

Motivating Contributions for Home Computer Security

by

Richard L. Wash

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in The University of Michigan
2009

Doctoral Committee:

Professor Jeffrey K. MacKie-Mason, Chair
Professor Mark Steven Ackerman
Professor Judith Spencer Olson
Associate Professor Brian D. Noble

© Richard L. Wash

All Rights Reserved

2009

to Emilee, Kumo, and Ursus, who brighten my life every day

Acknowledgments

A dissertation depends on the efforts and support of many people, and my dissertation is no exception. I have received an exceptional amount of help and support to get where I am today, and I thank everyone for their support. Several people deserve individual recognition for their investment and assistance in this process.

First and foremost, I thank my wife Emilee. She has continually been a source of support, understanding, assistance, and ideas, and she was even a coauthor on one chapter of this dissertation. But most of all I thank her for the hours and hours of debating research topics, which have made me an immeasurably better researcher and constantly remind me what I love so much about my chosen career.

Jeff MacKie-Mason, my advisor, coauthor, and colleague during my time at SI, probably had the single biggest impact on my career as a researcher. It was Jeff that convinced me to come to SI to study. He taught me so much about research, supported me for my entire time at Michigan, paid attention to both my successes and failures, made the time necessary to help me, pushed me to always be more rigorous, and worked hard to create an environment where young researchers like me can flourish.

My dissertation committee has helped me greatly in broadening my skills as a researcher. Judy Olson leads by example and shows both how to be scientifically rigorous yet still push boundaries. Mark Ackerman has pushed me to look beyond my quantitative and positivist training and see value in many different approaches. And Brian Noble has helped me to understand where I came from (Computer Science), where I'm going, and how the two relate.

I particularly want to thank George Furnas for making time for me in his busy schedule, despite health problems, and demonstrating how a truly great researcher thinks. George taught me to seek connections between everything I see, and to see great ideas in everyone I meet.

I want to thank my friends and colleagues at SI for all of their help. Finishing a PhD is a difficult task and cannot be done alone. The BlearyTheory and ICD lab groups have been great places for constructive feedback, and have accelerated and shaped my development

as a researcher. I will be lucky if I can find such high-quality feedback and support going into the future. Tiffany Veinot provided very helpful feedback about rigor and validity in qualitative research, and Chapter 3 is greatly improved thanks to her comments.

Finally, I would like to thank my family and friends for being there always, and reminding me that there is more to life than dissertations and the tenure track.

While many people helped and supported me in many ways, three people made much more concrete contributions to this dissertation. Rachel Lipson assisted me by conducting the second round of interviews for Chapter 3. Emilee Rader was an invaluable coauthor on Chapter 4. And Jeff MacKie-Mason was an invaluable coauthor on Chapter 5.

Finally, I could not be here without financial support from a number of places. The University of Michigan, School of Information, and the National Science Foundation all provided money and resources to support my work and study. This material is based upon work supported by the National Science Foundation under Grant No. CNS 0716196. This material is also based upon work supported by the National Science Foundation under the STIET (Socio-technical Infrastructure for Electronic Transactions) IGERT Grant No. 0114368.

Table of Contents

Dedication	ii
Acknowledgments	iii
List of Tables	viii
List of Figures	ix
List of Appendices	x
Abstract	xi
Chapter 1 Introduction	1
1.1 The Botnet Problem and Home Security	1
1.1.1 The Need for Individualized Security	3
1.2 Designing A Social Firewall	5
1.2.1 Technical Design	6
1.3 Incentive-Centered Design	8
1.3.1 Behavioral Design Challenges	9
Chapter 2 Theoretical Background	12
2.1 Botnets and Home Computers	12
2.1.1 Botnets Create Negative Externalities	13
2.1.2 Understanding Non-Technical Users	14
2.1.3 Existing Technical Solutions	16
2.2 Motivating Contribution: The Problem	19
2.2.1 Public Goods	19
2.2.2 Social Loafing	21
2.3 Motivating Contribution: Applying Theory	23
2.3.1 Public Goods Mechanisms	23
2.3.2 Learning from Charitable Giving	24
2.3.3 Applying Social Psychology to Design Software	26
2.3.4 The Problem with Information Cascades	27
2.4 Motivating Contribution: Existing Systems	30

2.5	My Contribution	33
Chapter 3	Folk Models of Home Computer Security	34
3.1	Introduction	34
3.1.1	Existing Literature on Security Understanding	36
3.1.2	Botnets and Home Computer Security	37
3.2	Methods	39
3.3	Folk Models of Security Threats	43
3.3.1	Models of Viruses and other Malware	46
3.3.2	Models of Hackers and Break-ins	52
3.4	Following Security Advice	57
3.4.1	Anti-Virus Use	58
3.4.2	Other Security Software	59
3.4.3	Email Security	60
3.4.4	Web Browsing	60
3.4.5	Computer Maintenance	61
3.5	Botnets and the Folk Models	62
3.6	Limitations and Moving Forward	65
Chapter 4	Designing for Side Effects: Tagging on Delicious	67
4.1	An Overview of del.icio.us	69
4.2	Existing Knowledge	70
4.3	Study 1: Interviews	72
4.3.1	Results: Producer and Consumer Incentives	74
4.3.2	Incentives on del.icio.us	80
4.3.3	Conclusion	84
4.4	Study 2: Large-scale Data	85
4.4.1	Background and Hypotheses	86
4.4.2	Methods	87
4.4.3	Logistic Regression Results	90
4.4.4	Summary of Study 2	94
4.5	Study 3: Computational Modeling	94
4.5.1	Measures	95
4.5.2	Modeling Tag Choices	96
4.5.3	Computer Model Results	98
4.5.4	Distributional Equivalence Measure	99
4.5.5	Interpretation	101
4.5.6	Summary of Study 3	101
4.6	Summary and Implications for Design	102
Chapter 5	Using a Minimum Threshold to Motivate Contributions	104
5.1	Introduction	104
5.1.1	Background on public goods	106
5.2	Behavioral Model	108
5.2.1	The voluntary equilibrium	109

5.2.2	Not enough information	110
5.3	Setting a Minimum Threshold	111
5.3.1	An exclusion equilibrium	112
5.3.2	Will it work?	114
5.3.3	Will it always work?	115
5.3.4	Adjusting the threshold	116
5.3.5	Who contributes?	116
5.3.6	Summary of behavioral analysis	117
5.4	Private Value	118
5.4.1	Blocking private use	120
5.5	Discussion	120
5.5.1	Quality	120
5.5.2	Measurement	121
5.5.3	Authentication	121
5.5.4	Bootstrapping	122
5.6	Using a Minimum Threshold Mechanism	122
Chapter 6 Improving the Design of the Social Firewall		125
6.1	Design for Information Sharing	126
6.2	Lessons about Home Computer Users	126
6.2.1	Design suggestions	128
6.3	Lessons from the Delicious Study	129
6.3.1	Design suggestions	129
6.4	Lessons from the Minimum Threshold Study	130
6.4.1	Design suggestions	131
6.5	Going Forward	132
Appendices		134
Bibliography		212

List of Tables

Table

3.1	A fragment of the data matrix from the initial analysis of Round 1. It includes a basic descriptions of each subject’s statements for each of the major questions in the interview.	42
3.2	Intermediate data matrix developed for analysis. This matrix includes a number of facets of mental models vertically matched with each of the 10 Round 2 participants. More details about the entries in this table can be found in Appendix B	44
3.3	A sample data matrix from near the end of the analysis. This matrix shows which folk model was held by the Participants in Round 2. A similar table was developed for the participants in Round 1.	45
3.4	Summary of folk models about viruses, organized by model features	47
3.5	Summary of folk models about hackers, organized by model features	52
3.6	Summary of Expert Security Advice. Each folk model responds to this advice differently.	59
3.7	How each folk model would probably react to the stylized facts about botnets	63
4.1	Description statistics about our respondents	72
4.2	Logistic Regression Results. Coefficients for tag_dummies and per user random effects are omitted here, but can be found in Appendix C.	91
4.3	Fitted Probabilities for the top 4 tags on <i>101 Cookbooks</i>	92
4.4	Measures of distributional equivalence and numerical identity.	99
B.1	A sample data matrix from near the end of the analysis. This matrix shows which folk model was held by the Participants in Round 1. The question marks indicate insufficient data to distinguish. A similar table for the participants in Round 2 is Table 3.3.	147

List of Figures

Figure

1.1	Screenshot of a ZoneAlarm personal firewall popup	6
2.1	The Collective Effort Model of Karau and Williams (1993)	22
4.1	The user interface of del.icio.us	70
4.2	This chart depicts the number of respondents who used these strategies for information discovery. The light bars are tag-related strategies, and the dark bars are username-related strategies	80
4.3	Tag frequency distributions for the real world data and strategies implemented in the computer model on a log-log scale.	98
4.4	Shows the frequency distribution shape of KS statistics for real world data and modeled tag choice strategies.	100
6.1	Diagram of the main interaction in a social firewall	127

List of Appendices

Appendix

A	Interview Guide for Folk Models Study	134
A.1	Round 1	134
A.2	Round 2	136
B	Intermediate Analysis of Folk Models of Home Computer Security	139
B.1	Interpreting the Facets in Table 3.2	139
	B.1.1 Viruses	139
	B.1.2 Hackers	142
	B.1.3 Relationships	145
B.2	Folk Models of the 33 Participants	146
C	Full Results of Logistic Regression	148
D	Proofs of Major Theorems in Chapter 5	205
D.1	Proof of Lemma 5.1	205
D.2	Proof of Proposition 5.1	206
D.3	Proof of Lemma 5.2	208
D.4	Proof of Proposition 5.2	209
D.5	Proof of Proposition 5.3	210

Abstract

Recently, malicious computer users have been compromising computers en masse and combining them to form coordinated botnets. The rise of botnets has brought the problem of home computers to the forefront of security. Home computer users commonly have insecure systems; these users do not have the knowledge, experience, and skills necessary to maintain a secure system. I take steps toward designing a socio-technical system that will hopefully help home computer users make better security decisions. Designing such a system requires additional knowledge before a successful system can be developed.

First, more information is needed about the knowledge and skills that home computer users currently possess. I conducted an interview study of home computer users and identified eight distinct mental models of security threats; four are models of “viruses,” and four are models of “hackers.” The respondents in this study use the models to decide which security precautions should be used and which can be ignored.

Second, to share information, users need an incentive to exert the time and effort required for sharing. I describe two mechanisms that can be used in social computing systems to encourage contribution. I illustrate the first mechanism, the side effect mechanism, by describing how it is used in a popular social bookmarking website. I also illustrate a design feature that is important when applying this mechanism: incentive alignment. The second mechanism that I describe is technically simple: set a minimum threshold and exclude users who don’t contribute enough. I develop a theory of how users are likely to respond to such a mechanism and use that theory to characterize when such a mechanism should be used.

Finally, I bring all of these findings together to suggest some preliminary design features for a socio-technical security system to help home computer users. While there are many unanswered questions, these design features can serve as a starting point for future work in the area.

Chapter 1

Introduction

Home computer users commonly have insecure systems. Recently, this insecurity has been exploited by criminals to build and control large networks of compromised ‘zombie’ computers that are then used for numerous Internet crimes. Home computer users typically lack the knowledge and experience of professional system administrators, and consequently do not make the same quality of security decisions as specialists.

I believe that a social computing system for sharing security-relevant information has the potential improve the security decisions of home users without requiring costly-to-obtain information from experts. To work, such a system will need to be designed to take advantage of the capabilities and motivations of home users.

My approach in this dissertation is to use rigorous, social science based understanding of home computer users to develop concrete, technical design ideas. I make three primary contributions with this research. First, I describe how home computer users understand information security and how they use that understanding to make security-relevant decisions. Second, I provide concrete design ideas for social computing systems that can be used to encourage users to contribute useful information. And finally, I illustrate how these results can be used by synthesizing them into design features for a social computing system intended help home computer users improve those security-relevant decisions.

1.1 The Botnet Problem and Home Security

Criminals and law enforcement have an adversarial relationship, in which “the law” is constantly trying to stop “the bad guys.” As a result, criminals tend to have a number of secondary objectives motivated by their desire to continue their work. To begin with, criminals value stealth. They often choose hideouts and targets where they are unlikely to be observed and caught by law enforcement. Criminals also value mobility. Being able

to pick up and move when one plan or hideout is discovered greatly aids in avoiding law enforcement. Closely related to this, criminals find redundancy to be useful. Redundancy allows one part of their plan to be halted by law enforcement, as another part will just take its place and allow the plan to be fulfilled.

For information criminals and hackers these secondary objectives are likewise valuable. One growing strategy among information criminals is the use of botnets. A botnet is a distributed army of computers not owned by, but under at least partial control of an attacker. The attacker breaks into each computer and leaves a remote control program running. Once the compromised, 'zombie' computer receives instructions from the attacker, it usually independently follows the instructions without further contact. Often attackers automate the process, allowing them to build extremely large botnets. Sizes in the tens of thousands are considered moderate, and networks of 100,000 or more are not uncommon (Sieberg, 2006).

Botnets fulfill all three secondary objectives. They are stealthy because they provide a layer of indirection between the attacker and his targets. If security professionals or law enforcement manage to track down a zombie computer, they still have not discovered the identity of the actual attacker. Botnets enhance mobility since they can accept commands from anywhere. They also provide great redundancy. Finding and shutting down one or two of the zombies has little impact on the botnet. The innovative structure of botnets has enabled criminals to evade law enforcement and commit a large amount of information crime (Sieberg, 2006; Stone, 2006).

Home computers, or computers whose primary use is for consumer or other home purposes, seem to be disproportionately represented in botnets. To develop strategies that reduce botnets, we need to understand why this is so. First, the targets are numerous: over 60 million home computers attached to the Internet in 2003 in the U.S. alone (Day et al., 2005). Second, home computers are typically not under the administrative control of security experts. Ordinary home users frequently are undereducated about how to detect and fix information security problems.

Another less obvious problem contributes to the ease of creating botnets from home computers: user motivations. Home computer users generally do not directly suffer the ill effects of having their computer under the control of an attacker. Botnets are often programmed to be active only while the host computer is otherwise idle so that any slowdown does not impact the host computer's user. Additionally, attackers will use one zombie to attack other computers on the Internet. For example, attackers routinely use botnets to send large amounts of spam email. The compromised home computer does not directly receive much of this email, but others do. While society would like to have this computer fixed, the actual home user isn't suffering much from her computer being a zombie. If she fixed her

computer, it wouldn't stop the flood of spam she receives. This is known as the problem of (negative) *externalities*. This is true for other types of attacks also. In general, attackers have a strong motivation to remain stealthy and only use botnets to attack other computers; this strategy reduces the chances that any zombie owner will find it worthwhile to fix her computer.

Despite my rhetoric of "home computer users," many of these problems extend beyond the home; most of my analysis and understanding in this dissertation is likely to generalize to a whole class of users who are unsophisticated in their security decisions. This includes many university computers, computers in small business that don't have much technical expertise, and personal computers used for business purposes.

1.1.1 The Need for Individualized Security

The HoneyNet project recently conducted a study of botnet activity obtained from their sensors. (Zhuge et al., 2007, with a good summary by Andrew Jaquith on his blog¹) Botnet activity is strongly diurnal, meaning that they are active during the day and inactive at night. This likely comes from infected machines being powered-off at night. There are large numbers of bots; Over 1.5 million bots were tracked during this study, though the average size of a botnet was only 800 bots. Over 50% of the methods that the botnets used to spread themselves were exploits of three well-known security holes: the ASN1², DCOM³, and LSASS⁴ vulnerabilities. Password guessing was only used 7% of the time. On average, a command-and-control server was alive for 54 days. This means that once a botnet is created, it is used for, on average, almost two months before it is either taken down or voluntarily removed from service.

Another finding relates to the malware that was found on these botnets. The authors ran all 90,000 unique samples of malware (average 250 per day) through nine popular anti-virus tools. Note that all of this malware was actively circulating on the Internet at the time of collection. Within 1 hour of finding the malware, the best antivirus tool detected only 92% of the malware. This means that around 7,200 instances of malware would not be detected by the best anti-virus software. The other tools decreased in effectiveness, with the worst tool only detecting around 50% of the malware. 30 days after discovering the malware, the best tool was up to 94% effectiveness.

¹http://www.securitymetrics.org/content/Wiki.jsp?page=Welcome_blogentry_041207_1

²<http://www.microsoft.com/technet/security/bulletin/MS04-007.msp>

³<http://www.microsoft.com/technet/security/bulletin/MS03-026.msp>

⁴<http://www.microsoft.com/technet/security/bulletin/MS04-011.msp>

From these results, I have some hypotheses about the botnet problem. First of all, it appears that botnets are primarily targeting unpatched home computers. The diurnal pattern of compromised hosts indicates that these machines are often taken offline at night, as many home computers are. The botherders (the people who run the botnets) still mainly spread using well-known vulnerabilities, most of which have existing patches. This indicates that most of the compromised machines have not been properly patched and kept up-to-date for some reason. Also, modern anti-virus tools are insufficient to detect most of the malware being used on these botnets. It is possible for a computer to be part of a botnet despite the fact that its anti-virus tool says everything is fine. With botnets becoming more stealthy over time, this is a serious problem.

Computer security experts know who is being compromised, and how, but still cannot stop it from happening. Particularly for home computers, part of the problem is that most security software is a one-size-fits-all solution. Everyone is instructed to use an anti-virus package, but most of these packages aren't customized for the user(s) of that computer. Anti-virus software has the difficult job of detecting all malware while letting through every possible legitimate use of the computer. As an example, consider the DCOM server in Windows, which I indicated above is one of the three most commonly exploited security holes. Most home computers do not need to be listening for external DCOM connections. However, the few users with a home network must have this enabled in order to share files. Therefore, it is important that anti-virus tools not flag this activity as dangerous so they don't interfere with home networking. In trying to find a lowest-common-denominator security policy, most security vendors have had to implement a policy that is too weak to stop real malicious activity.

There are a number of factors that lead to this approach. First, having a single common policy for everyone is a low-cost solution. Customizing every installation to the needs of each end user is expensive. Most security vendors find it easier to adopt one policy that can then be updated centrally and kept up-to-date easily. For example, new virus definitions are sent to everyone's home computer antivirus system whenever a new virus is found. Secondly, most home users do not have the technical skill to customize their security detection software. Home users are only aware of high-level uses of the computer (e.g. "I use email") and not the corresponding technical details that are important for customizing detection software ("outgoing port 25 must be allowed").

Individualized security could significantly help with the home computer security problem, and hence the botnet problem. If users that did not need DCOM could easily block it or turn it off, that would prevent many compromises. The problem is that there are MANY such decisions, most of which cannot be made cheaply by the average home computer

user. To make a sweeping generalization, large businesses have much better track records largely because they can afford to customize their security detection to their computer use. Businesses that use Lotus Notes for email get suspicious when Microsoft Outlook is run. They know which machines should be running web servers and IRC servers, and can customize their security detection software to notice when an errant machine is running unusual software. Home users don't have this luxury; since some users run web servers at home, a web server is not an automatic red flag. However, most home users know whether their own computer should be running a web server.

Customizing detection for specific computer uses is currently too costly for most home users to undertake. Individualized security has the potential to make a difference in the security of home computers, but only if it can be done without too much cost in terms of time and money, and without requiring too much sophistication on the part of the end users.

1.2 Designing A Social Firewall

Individualized security requires input from users. But for home computers, users are frequently under-educated about security and other technical issues and generally find it difficult to make appropriate technical choices. One option that some individualized security systems have tried is to provide expert-written guidance as “help” for each of the security decisions that a user has to make. For example, the ZoneAlarm personal firewall system gives basic expert-written help, and provides a link to the Internet for more expert-written help.⁵ Firefox, the popular web browser, provides expert-written advice in all of its security-related popups.

However, there are a very large number of security decisions that need to be made to properly secure a home computer. Common expert-written firewall rulesets often run into the hundreds or thousands of rules (Al-Shaer and Hamed, 2004). The number of possible rules of interest is even larger. It is very costly and time-consuming to have experts write useful advice for all the possible security decisions that a home computer use might need to make. Also, it is not clear if the home computer users understand the advice from experts; much of this advice currently found online uses technical jargon and is written assuming a base level of security knowledge that home computer users might not possess.

I propose that home computer users can fill this gap by providing security advice to each other. There are a large number of home computer users (over 60 million according to Day et al. (2005)), and these users can write using language that other home computer

⁵See Figure 1.1 for an example ZoneAlarm popup. The expert-written help link is the button labeled “More Info” next to the title “SmartDefense Advisor.”

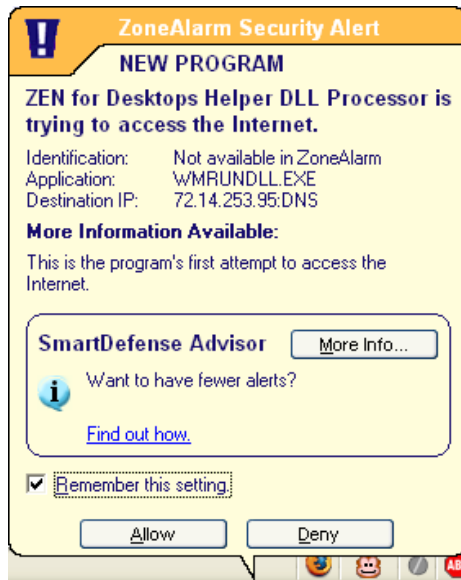


Figure 1.1 Screenshot of a ZoneAlarm personal firewall popup

users understand. While these users may not be familiar with computer security, security decisions involve more information than just security concerns. Security decisions need to balance security concerns with other factors such as usability and usefulness of the computer. Home computer users are in a unique position to know exactly what they want to use their computer for; for example, only the user knows whether allowing a given program to run is essential or irrelevant to her use of the computer. Home computer users also might be able to identify when a new security rule interferes with their use of the computer.

To accomplish this information sharing among home computer users, I propose a specially designed social computing system. This system would integrate social information sharing with a personal firewall system to provide technological support for users sharing information about personal firewall decisions.

However, the design of this *social firewall* is not straightforward; there are a number of design issues that must be addressed before such a system would work. I devote the majority of this dissertation to addressing some of these issues; but first I describe a basic technical design.

1.2.1 Technical Design

Firewalls are a form of *intrusion detection and prevention systems* (IDPS) (Scarfone and Mell, 2007). Their design goal is to prevent unauthorized users from taking unwanted actions on a computer. They monitor computer activities and can stop certain technical actions in order to implement a *policy* stating which activities are acceptable and which are

not. For example, a firewall might monitor network packets and can block those directed to particular TCP ports.

Firewalls have several technical design challenges: accurately identifying behaviors, offering a policy language flexible enough to map owner preferences over behaviors into actions, and preventing compromise of the firewall itself, among others. One especially difficult challenge for unsophisticated users is to configure the firewall policy appropriately.

Most firewalls focus on highly technical actions; many focus solely on creating network connections. However, there are many other user actions that could also benefit from a firewall. One can imagine a firewall policy for clicking on attachments in emails; executable attachments are disallowed, but images and Word documents are OK. Browsers have firewall-type policies for URLs. Web pages with invalid SSL certs, or that fall on a known blacklist can be denied, as well as forms that submit information across domains. Most systems also have an implicit policy for these actions, but they could be subject to user decisions as well.

A common approach to the customization problem is *interactive policy generation* (see, e.g., ZoneAlarm. A sample interactive policy query can be seen in Figure 1.1). A firewall policy may specify one of three actions for any given event (including a default for not otherwise specified events): allow, deny, or ask the user for a decision in real time. Interactivity offers flexibility and convenience for users who may want to postpone decisions until a event is encountered. Interactive policy generation allows detailed policies to be developed that take advantage of information only the user has, such as what applications are legitimately on the computer.

However, interactive policy generation is often the brunt of criticism: unsophisticated users may not know the risks or the benefits of permitting certain requests, and may face high costs of acquiring that knowledge, thus preventing effective policy decisions. Also, frequent policy popups can annoy users. A common belief (supported by my interviews in Chapter 3) is that many users always click 'allow' for every popup because they don't know how to make better decisions.

Firewalls targeted at home computers that use interactive policy generation are often called *personal firewalls*. There are a number of examples of personal firewall software on the market: ZoneAlarm from Checkpoint software, Sunbelt Personal Firewall, Microsoft Windows Firewall, McAfee Internet Security, Comodo Firewall Pro, and Norton 360, to name a few.

I propose adding a new feature to this class of software: user-contributed content. Users can contribute information to the system for each type of security decision they are faced with, and that information is then re-distributed in some form to other users of the system

when they face a similar decision. Technically, this is straightforward to implement. Each installation of the personal firewall software acts a client, collecting information and displaying the aggregated information from other users. There is a central server that stores all of the contributed information and then sends appropriate sets of information to client computers to help users make security decisions. This information sharing changes the personal firewall software into a social system.

However, there are many aspects of the design of the user-contributed content feature that are unspecified. What type of information should be shared? Should it be aggregated? How can the feature motivate contribution? I devote the remainder of this dissertation to better understanding design tradeoffs for this feature, and at the end I propose more concrete designs to accomplish this information sharing.

1.3 Incentive-Centered Design

I devote most of my dissertation to understanding how to design the information sharing aspect of a social firewall. To this end, I take an *incentive-centered design* approach.

Incentive-centered design is a paradigm for technology design that focuses on how technical features influence user behavior. All features of a software application influence user behavior. The What-You-See-Is-What-You-Get (WYSIWYG) editor in Microsoft Word encourages the user to continually focus on formatting, but the markup style of LaTeX encourages users to separately consider content and layout.

In applying incentive-centered design, I focus on these behavioral changes. I try to understand how and why specific features lead to given behavioral changes. In general, I look for *incentive mechanisms*: patterns in the design of computer systems that lead to specific, intended, and predictable user behavior. This allows me to ask first “how do I want the users of my system to behave?” and then design features that lead to the desired behavior.

Incentive-centered design becomes particularly interesting in multi-user systems. Human behavior in these systems depends not only on the technical design, but also on the behavior of other users. Strategic behavior, such as not contributing and free riding on the contributions of others, is commonly found in multi-user systems. I strive to understand not only how the technical design influences an individual’s behavior, but also how it provides incentives for multiple people and how it interacts with strategic incentives between people.

Because of the focus on understanding the causes of user behavior, incentive-centered design has its roots in the social sciences. I draw on theories of behavior from psychology and economics to try to predict how users will react to specific technologies.

1.3.1 Behavioral Design Challenges

There are a number of user behaviors that are critical to the success of the information sharing feature of a social firewall. These issues are common to many different social computing systems, and solutions that work for one might also generalize to help other systems.

First, I focus on providing incentives for users to contribute information to social computing systems. How can we design a social firewall to encourage users to contribute their information? *Contribution* is primarily an issue of quantity: are we getting enough information from users? I list this behavior first because without contributions, the other behavioral issues don't matter.

The second issue that depends on user behavior is *quality*. How can we make sure that what users contribute is actually worthwhile? This is a big issue in a social firewall; low quality information may do more harm than good. There are two classes of strategy for approaching quality. First, it is sometimes possible automatically to filter high-quality contributions from low-quality contributions. This filtering can be done programatically as long as there is some method of measuring quality. Alternatively, you can have users rate others' contributions (like Amazon.com's "Was this review helpful?") and use those ratings to identify high-quality contributions. Second, rather than trying to identify or measure the quality of contributions, it may be possible to provide an incentive that ensures most contributions are high quality. Wikipedia's "barn star" system⁶ publicly rewards high-quality contributions, thus encouraging all contributors to increase quality.

Another issue for social computing systems is *user retention*. How can these systems keep their users coming back regularly? And why do some users stop using these systems? Getting a user to contribute to a social firewall is good, but with security threats constantly evolving, it is important to keep users using the system. How can we get users to stick around and continue contributing? And, when is it better to let them leave?

Many social computing systems work best when users *collaborate* with each other. For example, Wikipedia is a much better encyclopedia when users collaborate and co-author articles and edit existing articles; its competition Google Knol has exclusively single-author articles. How can we motivate users to not just contribute, but to work together so that the sum is greater than the parts? Collaboration might play an important role in a social firewall to resolve conflicting mental models and to resolve disagreements about policy advice.

Since most social computing systems are open to pretty much anybody, one of the big issues is *maliciousness*: not all contributions support the goals of the system. A social fire-

⁶<http://en.wikipedia.org/wiki/Wikipedia:Barnstars>, retrieved July 10, 2009

wall is particularly likely to have attackers manipulate the system by contributing erroneous content that helps further their attacks. Is it possible to discourage people from this type of contribution? In some ways this is similar to the issue of quality, but focuses on the opposite side: reducing unwanted contributing rather than encouraging valuable contributions.

Social computing systems do not just pop into existence; they must grow over time. The first problem that any social media system faces is how to get the first few contributions from users. This is known as the *bootstrapping* problem. The basic problem is that each user gets value from the system that depends on how much useful information is currently in the system; but when the system is first created there isn't any information contained therein, and therefore no one will want to use it. Technically, this is an instance of a positive network externality; each user that joins the system makes the system more valuable to everyone else, but that user doesn't take that value into account when deciding if he or she should join. Bootstrapping problems are particularly difficult to deal with precisely because there isn't much to work with; since very few users are yet on the system there isn't much value that can be used to motivate people to join.

And finally, social computing systems can be improved when end-users find innovative new ways of using the system. For example, users on Twitter originally came up with "@replies" and "#hashtags."⁷ *End-user innovation* is extremely valuable because it allows users to customize the experience of using the system to make it more useful. Encouraging this type of innovation can greatly increase the value of a social firewall.

There are many behaviors that I would like to see from users of a social firewall. However, creating a design that induces all of them is a large and difficult task. In this dissertation I focus primarily on one important behavioral challenge: encouraging user contributions. I focus on this contributions because without contributions, the other behaviors are moot. Contributing information is the fundamental, defining behavior in user-contributed content systems. Some of my results also help with other behavioral issues (particularly quality of contributions), but those issues are not the focus of this dissertation. Additionally, I chose to study what users currently know in order to better understand what information they can contribute, and what information from others might be useful to have. I feel that a successful user-contributed content system should match the needs of the information users with the capabilities and knowledge of the contributors. This knowledge helps to design contribution mechanisms that induce users to contribute *useful* information.

The research literature has many results that might be helpful for this design problems.

⁷Twitter co-founder Biz Stone mentions this fact in his interview on the Freakonomics blog from the New York Times website: <http://freakonomics.blogs.nytimes.com/2009/06/12/biz-stone-answers-your-twitter-questions/>

Before I discuss my novel ideas I first describe some existing knowledge that can be applied to this design problem.

Chapter 2

Theoretical Background

No knowledge exists in a vacuum. There is much that is currently known that is relevant to designing a social firewall. In this chapter I discuss some of this knowledge that I build upon.

I begin by describing what is known about botnets and non-technical computer users, and a few of the technologies that non-technical users can use to improve their security. This knowledge helps to put the current state of home computer security in context, both in its successes and its weaknesses. Next, I describe a number of social science theories related to motivation. Specifically, these are theories from psychology and economics that relate to groups of people working together to create something. These two fields see this problem different, and propose different ways to think about the problem and to solve it. Finally I end by describing some existing socio-technical systems and the way they encourage contributions.

2.1 Botnets and Home Computers

Computer security has been an active area of research for a number of decades. And there has been a vibrant computer security industry for many years. This has lead to numerous innovations that help protect computers from attackers. Consequently, attackers have had to be creative for their attacks to succeed.

In botnets, modern attackers have discovered a very powerful method of attack that is very difficult for the computer security community to defend against. Botnets are an instance of a well-known and well-studied problem in economics, and many important properties are well described by economic theory. Largely, home computer users are not keeping their computers secure and have enabled botnets to flourish. Next I describe some economic

theory that helps explain why it may be entirely rational for home computer users to leave their computers with little security.

2.1.1 Botnets Create Negative Externalities

The botnet problem is not entirely unique. It is an instance of a *negative externality* problem from economics. An externality is any effect, either cost or benefit, that arises as a result of one person's actions and causes an effect on another person not mediated through a transaction. Parties in a transaction can take into account their costs and benefits when deciding whether and how to transact, but third parties that are affected by an externality have no say. Negative externalities are externalities that impose costs on third parties; pollution is the classic example. Negative externalities lead to more harmful actions than would be socially efficient. (Mas-Colell et al., 1995, esp. chapter 11)

Botnets create an externality (Camp and Wolfram, 2000). Normally, a hacker compromises a victim and this compromise can be seen as a transaction. The victim can spend time, money, and effort to better secure their computer from attack, but has chosen to bear the costs of attack rather than spend more to make their computer secure. Hackers obviously benefit from the compromise. All of the costs and benefits of the attack accrue to either the hacker or the victim. However, with botnets, the compromised computer becomes a tool that can be used to attack third parties. Botnets cause third parties to care whether the victim's computer is compromised, and the third party would prefer that the victim spent more time, effort, and money on security than he or she rationally would.

There are a number of standard solutions to negative externality problems (Mas-Colell et al., 1995, ch. 11). The two most common policy solutions are a Pigovian tax, and the assignment of property rights to encourage Coasian bargaining. The government can impose a Pigovian tax (named after the economist Arthur Pigou who advocated them) on transactions to reduce the number of transactions, and hence reduce the externality. Some attempts to control pollution are Pigovian taxes. This is unlikely to work for hacked computers; compromises are already against the law, so collecting such taxes would be difficult.

An alternative suggested by Ronald Coase is to assign property rights. When property rights are well-defined, the people involved in the externality have an incentive to bargain with each other and "internalize" the externality. However, Coasian bargaining breaks down when the transactions costs — the costs of doing the bargaining — are too high. For the botnet problem, these costs are very high; it is very costly for a botnet victim (such as a company who is being extorted) to negotiate with all of the home computer users in the country. Likewise, since hacking is a crime, it is difficult for the victims to negotiate

with hackers. When these negotiations are initiated by hackers, it is rightfully considered extortion. (Ratliff, 2006)

2.1.2 Understanding Non-Technical Users

Hackers have targeted home computers because they are low-hanging fruit. The vast majority of home computers are administered by people who have little security knowledge or training. While home users may be concerned with the security of their computers, they usually lack the training and expertise to effectively use security technologies. Ross Anderson's 1993 study of Automated Teller Machine (ATM) fraud found that the majority of the fraud committed using these machines was not due to technical flaws, but to errors in deployment and management failures. These problems in the banking industry illustrate the types of security problems found in home computers and the difficulty that even professionals face in producing effective security.

Existing research has investigated how non-expert users deal with security and network administration in a home environment. Dourish et al. (2004) conducted a related study, inquiring not into mental models but how corporate knowledge workers handled security issues. They found that people saw security technologies as a barrier (like a locked door) that keeps the bad guys out. Security technologies were expected to keep out all potential bad guys, and technologies that focused on one specific type of bad (like anti-spam technologies without anti-virus capabilities) were seen as partial or imperfect. Users also felt futility with security, referencing unknown others (like hackers) who will always be one step ahead. Most users, lacking the time and inclination to deal with security, attempted to delegate their security concerns. Dourish et al. found that users chose to trust in technology (like a firewall), delegate decisions to a person (like a knowledgeable colleague), trust in an organization (we have a good support group), or trust that an institution (like a bank) would protect them.

Gross and Rosson (2007) studied what security knowledge end users possess in the context of large organizations. They specifically studied the people in the organization who were not directly responsible for security (IT staff is usually where the direct responsibility lies), but who still had some security concerns because they had access to data their organization felt should be kept confidential. End users' security knowledge was neither comprehensive nor sufficient to maintain proper security, but common security actions such as locking the screen when away were better understood and practiced. All the participants were aware of some sensitive information they had access to, and knew to protect it and to be wary of being tricked into revealing it (social engineering). Gross and Rosson also noted that their

participants frequently conflated security and functionality failures.

Grinter et al. (2005) interviewed home network users and found that homes generally had a single person who assumed the role of system administrator. It was his or her job to maintain a network, troubleshoot and fix problems, and help others with network connectivity, and this person unanimously resented the amount of time spent in this role. They also found that both this system administrator and other members of the household felt a sense of uneasiness when the system administrator had to troubleshoot and fix computers belonging to other members of the household.

Combining the results from these papers, it appears that many users exert much effort to avoid security decisions. All three papers report that users often find ways to delegate the responsibility for security to some external entity; this entity could be technological (like a firewall), social (another person or IT staff), or institutional (like a bank). Users do this because they feel like they don't have the ability to maintain proper security, or to deal with problems when things go wrong. However, these papers do report that despite this delegation of responsibility, many users still make numerous security-related decisions on a regular basis. These papers do not explain how those decisions get made; rather, they focus mostly on the anxiety these decisions create.

Camp (2006) proposed using mental models as a framework for communicating complex security risks to the general populace. She did not study how people currently think about security, but proposed five possible models that may be active. These models take the form of analogies or metaphors with other similar situations: physical security, medical risks, crime, warfare, and markets. Asgharpour et al. (2007) built on this by conducting a card sorting experiment that matches these analogies with the mental models of users. They found that experts and non-experts show sharp differences in which analogy their mental model is closest to.

Another related area of literature concerns designing security software for non-technical users. There has been some work in the field of human-computer interaction that attempts to design security systems that non-specialists can easily understand and use. Cranor and Garfinkel (2005) edited a book containing a number of such research papers. All technology systems impose a usability cost on users: the cost of time, education, frustration, and skill required to use the system. The security and usability community has focused on designing technologies with a very low usability cost in the hope that user behavior will change and users' decisions will improve. This design approach is justified by the fundamental benefit-cost calculus of decision theory: if marginal cost is reduced for an activity with desirable outcomes, individuals will perform more of that activity.

2.1.3 Existing Technical Solutions

A number of technologies have been designed with the same basic goal in mind: to stop malicious people from taking unwanted actions on computer systems. There are three popular technologies that are particularly relevant to this dissertation: firewalls, anti-virus systems, and security updates.

Personal Firewalls A large and popular class of these security technology falls under the heading Intrusion Detection and Prevention Systems (Scarfone and Mell, 2007). In short, an Intrusion Detection System (IDS) is a piece of technology (hardware and/or software) that watches the behavior of other technologies for patterns that indicate unwanted activities. When such behavior is noticed, the typical response is to notify a human system administrator. However, some IDSs can also take automatic actions to prevent this behavior from continuing or causing harm. These systems are usually Intrusion Prevention Systems (IPS), since they actively attempt to prevent unwanted behavior, rather than simply detecting it.

Intrusion Detection and Prevention Systems (IDPS) are generally classified by the type of technological behavior they monitor. Network-based IDPSs monitor network traffic and network devices to look for patterns. Host-based IDPSs generally monitor a single computer for signs of unwanted behavior. Host-based systems are valuable because they are directly able to monitor many important behaviors of computers which are likely to change if that computer is compromised. However, they usually lack the sophistication to detect changes across multiple computers. Network-based systems have a view of a larger portion of the network so they can detect patterns a) before the network traffic gets to the end computers, and b) across multiple computers. However, they lack the direct access to monitor most of the software running on the end hosts, so can often miss intrusions. A number of modern IDPSs are taking a hybrid approach, combining data from both host-based sensors and network-based sensors to look for patterns of unwanted behavior.

A simple version of an IDPS is known as a firewall. Firewalls are devices that monitor computer behavior and have the ability to stop certain behaviors from occurring. (Cheswick et al., 2003) For example, a network-based firewall monitors network packets and has the ability to block packets to a certain destination, or for a specific application such as email. Firewalls implement a policy that states which behaviors are acceptable and which behaviors should be stopped.

While most firewalls are network-based, there is a growing class of host-based firewall systems. These systems are often called “personal firewalls,” particularly when the intent is for the end computer user to specify the firewall policy, rather than having an external system administrator set the policy. Many host-based firewalls take advantage of having

interactive access to the user to relax the requirement that a firewall policy be fully specified in advance. These firewalls have three possible actions for any behavior: allow the behavior, deny the behavior, or ask the user for a policy decision in real time. This capability for interactive policy generation has the potential to simplify firewall policy management, but requires the end user to be capable of understanding technical details in order to decide on an appropriate policy. Many users are unable to understand the functionality of the firewall in sufficient detail to make appropriate security decisions, as security policy decisions require the decision maker to trade off the benefits of access with the potential security risks.

Firewalls are proactive security systems. They try to prevent security problems before they start by restricting what a computer can do. Properly used firewalls do not need to be updated as hackers develop new techniques for attacks; however in reality sometimes new versions of firewalls are needed to combat particularly creative attacks.

Anti-Virus Systems Another popular type of Intrusion Detection and Prevention System is an anti-virus system. Anti-virus systems are largely marketed to home computer users and are designed to detect and remove computer viruses. They are very popular for personal computers, but are less common among corporate IT infrastructure where more sophisticated technologies can be used.

Most anti-virus systems work by looking for “signatures,” or recognizable patterns from known computer viruses. These systems are effective at detecting viruses that are well-known and have been around for a while. They fail, however, to detect novel viruses and viruses that continually change their own code in order to evade detection. Anti-virus systems are reactive security systems. They react only to known security threats and cannot anticipate or prevent previously unknown security problems.

Because anti-virus systems can only detect known viruses, it is important for these systems to continually update their database of known virus signatures. Many anti-virus systems come with a subscription service that keeps these databases current; but these subscriptions do not last forever and eventually the user must upgrade to the latest version and pay for an additional subscription. Because of this, some home computer users have out-of-date anti-virus systems that cannot detect newer viruses.

Security Updates Modern personal computers are general purpose computing machines. This means that in theory these computers can run any application that is written for them. However, most computers are intentionally limited to run only those applications approved by the computer’s user.

All software has bugs; bugs in software are mostly harmless and at most cause an

inconvenience to the user. However, a small subset of software bugs are particularly bad because they are *exploitable*: a properly crafted input can cause the application to execute arbitrary code. This class of bugs is particularly bad because it means hackers and viruses can exploit these bugs to get their programs running on victims' computers.

New exploitable bugs are being discovered all the time. The United States maintains a national database of known bugs vulnerable to exploitation in this way. Currently it averages 13 new vulnerabilities reported per day. Every year since 2000 has seen at least 1000 new vulnerabilities reported to this database.¹

It is not unusual for software applications to issue "patches" that fix various software bugs. Usually, these patches will be aggregated and formed into a "service pack," or simply just fixed for the next version of the application. However, for exploitable bugs, it is important that the patch for the bug be available quickly so that affected users can fix their software and therefore prevent hackers from exploiting the bug. Many software vendors work hard to release patches for exploitable bugs in a timely manor in order to protect their users. Microsoft has even gone so far to declare the second Tuesday of each month "patch Tuesday" where they release patches for all new exploitable bugs they know of.²

Many security experts advise all computer users to "stay up to date" on patches by frequently checking for and applying any new patches for exploitable bugs. Staying up to date on patches makes it more difficult for hackers and viruses to compromise a computer. A number of technologies are used to help users stay up to date; the most prevalent one is Microsoft Windows Update. This system automatically checks with Microsoft on a regular schedule, looking for newly available patches that have not yet been applied to the computer. It then automatically downloads and installs them for users unless they have configured the computer to require manual permission.

This automatic system has helped many users stay up to date, but doesn't work for everyone. Sometimes patches can accidentally break some other functionality, so many users don't install patches, or delay installation to wait for other users to detect problem patches. Also, many computers are not permanently connected to the Internet, and therefore cannot always check for and download new patches when they become available. For these reasons and others, many computers still are vulnerable to well-known exploitable bugs in common software. Above I mentioned that some botnets primarily spread through three well-known exploitable bugs: the ASN1³, DCOM⁴, and LSASS⁵ bugs. Patches have existed

¹<http://nvd.nist.gov/>

²<http://www.microsoft.com/technet/security/Bulletin/advance.msp>

³<http://www.microsoft.com/technet/security/bulletin/MS04-007.msp>

⁴<http://www.microsoft.com/technet/security/bulletin/MS03-026.msp>

⁵<http://www.microsoft.com/technet/security/bulletin/MS04-011.msp>

for these bugs for years, yet many computers are still vulnerable.

2.2 Motivating Contribution: The Problem

The problem of inducing contributions to a social firewall shares many attributes with a well-studied problem. Below I describe how social scientists characterize the basic problem of contribution. Also, since the problem is not unique, many others have tried to solve this problem in various different ways. I will describe both general approaches to solving the contribution problem and then specific technologies that address the contribution problem in interesting ways. Much work in the past has not distinguished the problem of inducing contribution and the problem of inducing quality. I find that separating these two problem can be quite beneficial, but most of the previous research has treated these as two aspects of the same problem.

Economists call the problem the “voluntary provision of public goods” problem, and psychologists refer to this problem as “social loafing.” But the basic problem is the same: human beings naturally contribute less effort when others are involved than they should. The fields of economics and psychology have different approaches to studying this problem, and that has led them to different types of solutions.

2.2.1 Public Goods

User contributions to a social computing system can be seen as contributions, in the form of information, to a single shared information pool. All users of the system have access to this pool. This shared information pool has the properties of a public good (Samuelson, 1954). In particular, the pool is *non-rivalrous* since using the information pool does not materially reduce the value of the pool to other people. To use a familiar example, once National Public Radio broadcasts a program, consumption by one listener does not crowd out consumption by other listeners. For information, nonrivalry is generally true because the incremental costs of (digital) reproduction and distribution are approximately zero, and thus multiple instances of the information can be “consumed” without “using it up”.

Shared information pools in social media are also commonly *non-exclusive*; the information in the system is available to anyone anytime. The information contained on Wikipedia, del.icio.us, and Twitter is available for free to anyone with a web browser and a network connection. However, non-exclusivity is a design choice; social media systems could technically exclude users from accessing the public information pool. This potential for exclusivity opens up new opportunities for creating incentive mechanisms. I explore this

opportunity in Chapter 5.

When public goods are created through voluntary contributions, they generally have the problem of *underprovision*: users prefer to “free ride” and use the public good without contributing, relying on other people to do the hard work of creating it (Samuelson, 1954). To see this, assume there are N people in the system, indexed by $i = 1, \dots, N$. Each person is given w_i dollars (or effort or some other common unit). Each person then has to choose how to allocate this w_i dollars: they contribute g_i it to the public good and x_i to all of the other private uses. The public good G will then be made up of everyone’s contributions: $G = \sum_{i=1}^N g_i$ Each person chooses an allocation that maximizes their utility subject to their budget constraint:

$$\begin{aligned} & \max_{x_i, g_i} u_i(x_i, G) \\ \text{s.t. } & x_i + g_i = w_i \\ & G = \sum_{i=1}^N g_i \\ & g_i \geq 0; x_i \geq 0 \end{aligned}$$

Let G_{-i} be the sum of everyone’s contribution to the public good except user i . If we add this to both sides of the budget constraint equation, the budget constraint for person i becomes

$$x_i + G = w_i + G_{-i}$$

In a Nash equilibrium, each person treats G_{-i} as a given. From this, we can see person i chooses his or her optimal level of G by “topping up” G_{-i} , the amount expected from everyone else. Person i treats G_{-i} as his or her “social income” that can only be spent on the public good. Under normal assumptions, it is straightforward to prove that a Nash equilibrium exists and is unique for this setting. (Andreoni, 2006, e.g.)

In equilibrium, each user will choose an allocation such that $\partial u_i / \partial G = \partial u_i / \partial x_i$. If this wasn’t true, then user i could benefit from shifting some funds from x_i into G or vice versa. Another way of saying this is that the marginal rate of substitution between goods for person i (MRS_i) is equal to 1. However, Samuelson (1954) proved that the efficient level of the good is achieved when the sum of the marginal rates of substitution ($\sum_{i=1}^n MRS_i$) is equal to one. When there is more than one person using a public good, this is not achieved, and everyone voluntarily chooses to contribute inefficiently little to the public good. That is, each person rationally chooses to free ride on the efforts of others and only contribute when it is worthwhile to “top off” the public good.

Of course, if everyone prefers to free ride, then the information pool tiny. We see the

free rider problem in social media: Adar and Huberman (2000) found that almost 70% of users of a popular peer-to-peer system contribute nothing at all. While Wikipedia has over 9 million registered users, only 166,066 (less than 2%) have contributed effort in the last 30 days⁶. Of those who contribute to Wikipedia, 50% do not return after their first day of contribution (Jian and MacKie-Mason, 2008).

This basic result is fairly extreme; for many public goods it predicts that no one will contribute any amount at all, and that the public good therefore won't exist. However, we observe these types of public goods all of the time. The basic problem is that in the above model, all contributions are equivalent. Therefore, each person would prefer that others contribute so that he can save his budget for personal use. However, we often observe that, all else being equal, many people prefer that the contributions come from themselves. An improved theory includes the idea that some people receive a *warm glow* from contributing to a public good. (Andreoni, 2006) These people gain utility both from having access to the public good G , and from a warm glow that comes from contributing g_i . When you include a warm glow in the model, other people's contributions are only imperfect substitutes for one's own.

Warm glow, as a motive for contribution, is slightly different than altruism. Andreoni (2006) uses a great metaphor to illustrate the distinction:

Just like hunger tells a person it is *time* to eat but taste buds tells the person what they *want* to eat, it is altruism that should tell you what to give, but warm-glow tells you how much to give.

2.2.2 Social Loafing

Psychology has a different perspective on this same problem. It identifies the problem as one of *social loafing*, which is the reduction in motivation when individuals work collectively (Karau and Williams, 1993). Specifically, psychologists see the problem when people work together on a group outcome, and not when individuals work co-actively on separate individual outcomes. This difference is one of perception; an individual only social loafs when he or she perceives a group outcome rather than his or her own individual outcome.

A meta-analysis (Karau and Williams, 1993) of the social loafing literature proposed a useful model that integrates the results of many experiments into a coherent theory. The basic logic of the theory states that "social loafing occurs because there is usually a stronger perceived contingency between individual effort and valued outcomes when working individually." Basically, people are less motivated to work toward group goals because they

⁶<http://en.wikipedia.org/wiki/Special:Statistics>, retrieved on March 16, 2009

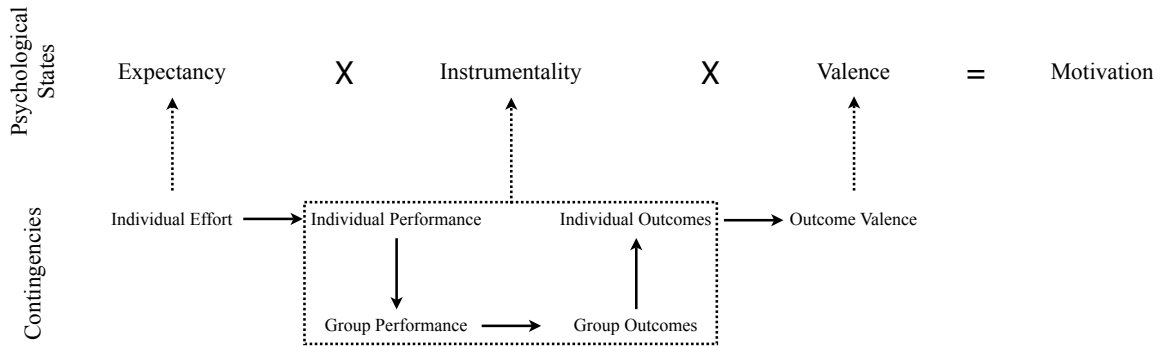


Figure 2.1 The Collective Effort Model of Karau and Williams (1993)

can't see as strong of a connection between their effort and the final outcome that they value as they can for individual work. Therefore, social loafing can be reduced by increasing the perceived value and importance of the individual contributions, and the salience of their connection to the final outcome.

This theory, called the Collective Effort Model, (CEM) is diagrammed in Figure 2.1. It puts forward three psychological factors that influence the strength of motivation: expectancy, instrumentality, and valence of outcomes. Individuals need to *expect* that high levels of effort will lead to high levels of performance. It is vital for motivation that high performance be seen as *instrumental* in obtaining the outcome. And finally, the outcome must be have high *valence* to the individual; that is, the individual must see the outcome as desirable. Lessening any of these factors will reduce the individual's motivation to put forth effort.

Karau and Williams then break down *instrumentality* into three component parts, all of which must be present for a person to perceive their contribution as instrumental. First, individuals must believe that their individual performance has a strong influence on group performance. For user-contributed information pools, this means that individuals must believe that their contributions will help others contribute. Second, individuals must believe that the group performance determines the final group outcome. This means that the individuals believe that the information pool is useful to the group as a whole. And third, individuals must believe that the group outcome is related to the outcome they as an individual will experience. For information pools, individuals believe that they will benefit from the group doing well. All three relationships need to be salient and strong in the individuals mind; any relationship that is weak consequently weakens the motivation to contribute and thus reduces contributions.

This model makes a number of useful predictions. For example, motivation decreases as group size increases. The CEM model gives a reason for this: with larger groups, individuals

will not believe that their contributions are as important in the final group outcome. It predicts that motivation will increase when individuals believe that their contributions are unique. This is important for user-contributed information pools because each person's contribution likely contains different information. The CEM model suggests that reminding users of this fact might lead to stronger motivation to contribute.

2.3 Motivating Contribution: Applying Theory

A number of researchers have worked to apply social science theory to design “mechanisms” that encourage people to contribute. In economics, this research falls in the subfield called *mechanism design*. This subfield has many tools that might be useful in designing software. These tools include methods of mathematical modeling that help researchers understand the incentives that people face and the consequences of their choices. Researchers have used mathematical modeling to identify a number of useful properties for these mechanisms. A mechanism for contribution is *Pareto efficient* if there exists no option for allocating contributions such that no one is worse off. A mechanism is *individually rational* if all participants are voluntarily willing to participate; no one ends up worse off than if they had not participated. A mechanism is *budget balanced* when there is neither excess, unused contributions nor a need for outside, third party contributions to make the mechanism work. Any given mechanism may exhibit several different desirable properties, but of course there are limits to how many desirable properties any mechanism can simultaneously exhibit. For example Myerson and Satterthwaite (1983) were able to prove that, under weak assumptions, it is impossible to design a mechanism for bilateral trade with all three of these properties.

Many interesting designs have been proposed and a number of general design principles have emerged. For example, one design principle states that systems can elicit honest information from people by ensuring that the greatest benefits accrue when the participant is honest; this can often be achieved by disconnecting a participant's information revelation from the received benefits. This is exemplified by a *second-price auction* (also known as a Vickrey auction): participants can submit any bid they want, but have to pay the value of the second highest bid. Thus, there is no incentive to lie because their bid does not determine the price they pay.

2.3.1 Public Goods Mechanisms

Economic theorists propose several approaches to raising the level of private contribution to a public good. For example, Groves and Ledyard (1977), Clarke (1971), Bagnoli and

Lipman (1989), and Varian (1994) all present mechanisms with different properties to address this problem. Most mechanisms from the economics literature face practical problems in any application, but in particular using these mechanisms to motivate contributions to a social firewall is not straightforward.

First, many of these mechanisms assume that contributions can be returned if unused. This works when contributions are money because money is easy to return. This doesn't work when contributions are in the form of information. The cost of contributing information is primarily the time and effort involved in some or all of data collection, analysis, drafting, formatting, editing, annotating, and organizing the information. It is simply impossible to refund time and effort.

Second, a number of these mechanisms have users propose how much should be contributed, and then the mechanism aggregates this information and decides how much each user should contribute. This too works when contributions are money because dividing money into any size contribution is easy. This is difficult to make work when contributions are information because dividing information into smaller chunks is problematic. There are some situations where something like this might work, but in general, telling users how much information to contribute is infeasible.

These mechanisms, however, are a good starting point for designing new mechanisms. They illustrate a way of thinking about incentive mechanisms that can be applied to new designs (and indeed, I use this way of thinking in Chapter 5). It may also be possible to make minor modifications to some of these designs to adjust them for the different nature of information as a contribution medium.

2.3.2 Learning from Charitable Giving

Andreoni (2006) provides a survey of literature on the topic of philanthropy, or the voluntary giving of resources to support a public good. Of particular interest is his discussion of mechanisms that are used by charities to encourage giving. Most of these mechanisms were originally observed in the wild, and then economic models were developed to explain how they work. He makes an interesting and useful distinction between capital campaigns, where the charity has a one-time need for large amounts of capital, and continuing campaigns, where the charity is raising funds for day-to-day operations. Charities tend to use very different forms of fund-raising for these different types of campaigns. For my purposes, I am mostly interested in continuing campaigns, as I don't want my system to rely on one-time, special contributions; however, mechanisms from capital campaigns might be important to get the system off the ground and solve the bootstrapping problem.

Capital campaigns almost universally begin by finding a small number of people who are willing to make very large donations – usually about 1/3 of the total desired amount. They then announce these large donations and who they are from, and collect the rest of what they need through numerous small donations. Economists have explained these facts in two ways. First of all, the initial donations can allow a minimum threshold to be reached, guaranteeing that some public good will be produced. This eliminates a potential no-contribution equilibrium for a threshold good (Andreoni, 1998). Secondly, such capital campaigns vary widely in quality; some are very worthwhile, others are not. The quality of the to-be-produced public good is private information to the charity. Receiving large donations from trusted individuals provides a credible *signal* to the numerous potential small donors that the cause is worthy, without requiring them to each go through a costly process to learn more about the proposed public good (Vesterlund, 2003). This suggests that, to get a user-contributed content system started, it might be valuable to find a high-profile individual, and get that person to make a lot of contributions to the system. Doing so signals that the system is worthwhile and likely to be around for a while, but also signals that the system is likely to overcome the network effects problems and grow to a self-sustaining size.

For continuing campaigns, charities usually use three different approaches. Charities directly ask for money, which is an extremely effective tool. They recognize donors, usually grouping them into categories by donation amount. And finally, they often use some form of charity auction or lottery. Economists have explained the power of the ask mainly with search costs – it is too costly for donors to figure out which charity to contribute to, and how to do the contribution. By soliciting donations, charities can eliminate these costs and make donating easier (Andreoni and Payne, 2003). Recognizing donors most likely works through social means – by publishing donor names, it encourages donors who seek to signal their wealth (Glazer and Konrad, 1996). Also, grouping them into categories encourages donors to contribute to the next higher category. Empirical evidence indicates that the majority of donations occur at the bottom end of a reporting bracket (Harbaugh, 1998). Charity lotteries effectively act as a subsidy on giving, even for risk-neutral donors (Morgan, 2000; Morgan and Sefton, 2000).

All three mechanisms here are potentially useful in software. User-contributed content systems should directly ask for contributions; however, I suspect that the “power of the ask” is greatly diminished when the target is not a charity. Recognizing contributors can be a powerful motivator in multi-user software systems. Many systems (such as Amazon.com’s book reviews) label contributions with the name of the contributor. Also, some systems have “leader boards” that recognize users who contribute the most.⁷ Finally, special lotteries have

⁷Amazon.com has a leader board that recognizes its top contributors: <http://www.amazon.com/>

potential to encourage contributions, though I am not aware if any user-contributed content system has ever tried to use one.

2.3.3 Applying Social Psychology to Design Software

Ling et al. (2005) took a different approach inspired by social psychology. They conducted a series of experiments in which they took theories from social psychology and attempted to apply them to the design of a social software system. In particular, they used these theories to attempt to influence the amount of contribution of movie ratings in MovieLens, an online movie recommendation system. Movie recommendations in MovieLens were chosen because they are an instance of the social loafing problem (Karau and Williams, 1993); users benefit from the ratings of others in the form of more and higher quality recommendations, but have little individual incentive to rate movies themselves.

In particular, Ling et al. looked to the Collective Action Model from Karau and Williams (1993) for concrete behavioral patterns that they could turn into design recommendations and test in a field experiment. The concrete behavioral patterns that they looked to are:

- People contribute more when they see their contribution as important to the group
- People contribute more when they are similar to others in the group
- People contribute more when they perceive their contribution as unique and important to the group
- People contribute more when the group benefits of contribution are made salient
- People contribute more when they are reminded of the intrinsic benefits of contribution
- People contribute more when they are reminded of the multiple benefits of contribution

They also looked to the goal-setting work of Locke and Latham (2002) to provide more behavioral patterns that can be turned into design recommendations:

- People contribute more when assigned challenging, numeric goals
- People contribute more when assigned individual goals rather than group goals
- People contribute more when not assigned overly-challenging goals

These behavioral predictions were turned into design recommendations by sending MovieLens users emails that specifically made salient one of the above points for the user.

review/top-reviewers. Amazon recently revised their algorithm for choosing the top reviewers because one woman, Harriet Klausner, was so much more productive than everyone else that no one was able to catch her. As of July 12, 2009, she had written over 19,000 reviews, averaging multiple reviews per day. The second most productive reviewer had written less than 7,000 reviews. Rankings using the old algorithm are available here: <http://www.amazon.com/review/top-reviewers>

The experimenters then measured changes in rating behavior among the users, looking for increases or decreases in contributions after these emails. However, many of these behavioral patterns did not make good design recommendations for this study. Only two of these behavioral patterns made good predictive design recommendations:

- Users contributed more when they believed their contributions were unique
- Users contributed more when assigned specific, challenging goals that are not overly difficult

Most of the remaining behavioral patterns that were turned into design recommendations did not have a statistically significant effect on contributions. The experiments were inconclusive on whether those recommendations would motivate contributions. However, a couple of the behavioral patterns actually caused users to reduce contribution at a statistically significant level, contrary to theoretical predictions:

- Users contributed less when their personal benefits were made salient
- Users contributed less when the benefits to the group were made salient
- Users contribute less when assigned individual goals than when assigned group goals

Social psychology can be a very useful source of concrete design recommendations for social software systems. However, this research shows that there are often complex interactions created by the various design decisions that can nullify the motivational effects. Social psychologists worked hard to isolate and identify the behavioral patterns described above, but in a real system the users must use the system as a whole and interactions between these patterns can cause behavior that is difficult to predict.

Rashid et al. (2006) continued this line of research. They found that individual contributions on MovieLens can be increased by displaying how valuable a potential contribution would be to other users.

2.3.4 The Problem with Information Cascades

A naive ‘social’ firewall system could simply aggregate users’ policy decisions and make those decisions available to others. For example, when presented with a firewall policy choice, the system could report information such as ‘10 of 30 (33%) of users chose Allow.’ Such a system would be simple to construct. And, a naive analysis would suggest that as long as the modal user makes a correct choice, such a system could improve security. However, since this system aggregates *decisions* and not the information that led to the decision, it would suffer from a problem known as *information cascades*. (Bikhchandani

et al., 1992) In short, users would rationally start to ignore their own opinions and follow the crowd, even if the crowd is wrong.

Imagine this situation: You need to decide between two restaurants for dinner tonight. You've never been to either one, and don't know anything about either one. But you know for sure that one of them will give you a good meal, and the other will be less than satisfactory, but you don't know which is which. Also you receive a signal that suggests you go to either the restaurant on the left (L), or the restaurant on the right (R), but the signal is imperfect. It is correct with probability $p > 0.5$. How do you choose?

One thing you can do is watch people go in ahead of you. If they are in the same position you are, with zero information except an imperfect signal, you can learn something from their choices. Say for example that person 1 chooses the restaurant on the right. You can infer that he received a R signal. Person 2 approaches and has to make a choice. She saw person 1 choose right, so believes that the probability that the right restaurant is the good one is p . She also received a signal. If her signal is R, then she will choose the right restaurant. If her signal is L, then she's seen an L signal and knows person 1 got an R signal. She's back at 50/50 probability, so will just flip a coin to choose. You get to choose third. You saw person 1 go right (and therefore got an R signal). If person 2 goes right, then chances are she also got an R signal. Even if you get an L signal, you should go right also and follow the crowd because you've observed more than one R signal and only one L signal. The person after you goes through this same logic, and knows that you should go R regardless of your signal, so they can't figure out your signal. However, from watching the first two people, they should also go right, and so should everyone after them. This is an information cascade – by observing behavior of people before you, you come to the point where you rationally ignore your own information and follow the crowd. And so should everyone else. But here's the interesting part: if the first two people got erroneous signals (or one erroneous signal and a unfortunate coin flip), then everyone ends up making a bad choice. This happens with non-trivial probability⁸.

This is the main point of the literature on information cascades (Bikhchandani et al., 1998). When people have a discrete set of possible actions and can observe the actions of others, they will end up in an information cascade in which everyone rationally ignores their own private information and follows the crowd. Furthermore, there is a non-trivial probability that the cascade will occur on a non-optimal outcome. The result seems to depend on the discrete-ness of the possible actions – if individuals can choose actions from a

⁸In this example, the probability of an erroneous cascade is $(1 - p)^2 + p(1 - p) * 0.5$. If $p = 0.75$ (an OK signal), then there is approximately a 15% chance of an erroneous cascade. If $p = 0.90$ (a good signal), then the chance of an erroneous cascade drops to about 5%.

continuous space (like stock prices), then they can still incorporate their private information, the cascade will not occur, and the group of people will eventually converge on the optimal choice. Cascades are rational because the group will cascade on the optimal choice slightly more often than if everyone chose solely based on their private signal. However, these cascades are fragile. A single person deviating from the cascade (in theory) is enough to dislodge the cascade and potentially change all following people to the other choice. The literature claims that cascades can explain much follow-the-leader type behavior, and particularly when this type of behavior is fragile and subject to sudden changes.

The result still holds if individuals can only observe summary statistics of past behavior (like the percentage of people who choose ‘right’) rather than the actual behavior. It also holds if individuals receive continuous signals rather than binary signals. If people have heterogeneous signal qualities, meaning that some signals are more reliable than others, then the people with the most reliable signals become fashion leaders and can derail an existing cascade and start a new one based on their signals. (Bikhchandani et al., 1992) Anderson and Holt (1997) have verified information cascades in a laboratory setting. They found that the cascade was imperfect (some people ignored the cascade and chose based on their own signal), and not quite as fragile as theory predicts.

Information cascades are a particularly large problem for a distributed firewall system.⁹ Consider a personal firewall system that records users’ allow/deny decisions and sends these decisions to a central server. This server then aggregates these decisions and reports this aggregate information to other users when they are faced with a similar decision. “10 of 30 (33%) of other users chose Allow.” Such a system meets the pre-conditions for information cascades presented above. It would be particularly prone to information cascades, with users ignoring their own personal feelings and going with the decision of the crowd. In a firewall system, this is extremely bad. It means that users have stopped contributing information to the system, because we cannot infer anything from their choice. But also, by figuring out which vulnerabilities have an incorrect cascade, an attacker can exploit large numbers of vulnerable people. Since cascades are highly sensitive to the early decisions, an attacker can influence cascades by being an early adopter and create large-scale vulnerabilities.

There are a number of methods to get around the information cascades problem. Information cascades are a problem because each user only observes the decision, but not the outcome of the decision – whether the decision was a good one or not. One well-known solution is to find a way to incorporate outcome information. For example, reporting the number of users who chose *Allow* and then caught a virus might get around the information cascades problem for some decisions. However, then the social firewall will need to find a

⁹Goecks et al. (2009) also discovered this problem for personal firewall systems.

way to induce users to provide this outcome information, with sufficient accuracy.

Another potential solution is to segment the population. Rather than reporting statistics for the whole population of users, the social firewall could instead figure out which users are ‘similar’ and report statistics for just that group. These similar users could be found in the same way that a standard recommender system works. Since each person will have their own idiosyncratic group of similar users, this might be enough to prevent the information cascades problem. To my knowledge, no one has looked at this idea as a solution to the information cascades problem. As future work, I am interested in developing a model of this situation to see if social clustering can diminish the information cascades problem. However, this is beyond the scope of my dissertation.

2.4 Motivating Contribution: Existing Systems

Software systems have been recognized as providing incentives for behavior for a long time. A number of interesting software designs have taken incentives seriously as an important component in the design of the system. Some of these have even received rigorous research attention. Below I summarize a few lines of research that are particularly relevant to my research.

Spam Spam can be characterized as unwanted content, usually in the form of advertising or malicious content, that is contributed to systems in which the barrier to bulk contribution is low. Email is the classic example, but spam also exists on wikis, in blog comments, on instant messaging, and on mobile phones. Spam is fundamentally an incentive problem – spammers naturally have a strong incentive to spam because they are sending commercially viable advertising for which the barrier to sending is very low – and many incentive-based solutions have been proposed. Loder et al. (2006) proposed requiring contributors to post a ‘bond’ that can be collected if the contribution is later deemed to be spam. This can dramatically increase the costs of contribution and provides a disincentive that is largely efficient if the proceeds from the bond goes to the party that exerts the cost of evaluation. von Ahn et al. (2003) designed a system that can effectively distinguish between humans and computer programs which has been used to fight spam. This works by eliminating the cheap automating capabilities of computers for abusing large-scale webmail system and requiring relatively expensive human labor to send spam in bulk.

Dwork and Naor (1993) proposed a method of increasing the cost of spam that requires contributors to prove that they have done a certain amount of costly computation. This is another disincentive based on increasing the costs of spam. Laurie and Clayton (2004)

produce some cost estimates and conclude that such proof of work systems will not work practically. However, Liu and Camp (2006) propose a modified proof-of-work system that sufficiently increases the costs of spammers such that it might work.

Peer to Peer Systems Peer-to-peer file sharing systems all must solve a basic problem in order to work: users of the system must choose to make their files available for download (share files) if there is going to be any content available on the network. In this way, content on a peer-to-peer network is a voluntarily provided public good that suffers from the usual underprovision problems. Bittorrent (Cohen, 2003), one of the more popular peer-to-peer systems, attempts to solve this problem by including a tit-for-tat mechanism that punishes users who don't contribute. Krishnan et al. (2004) propose that users naturally have an incentive to share content because by sharing with a third party I can reduce the download burden of the peers from whom I want to download. However, Jian and MacKie-Mason (2006) estimate this this would be a weak incentive in large file sharing networks. They then go on to propose a theory of generalized reciprocity. In this theory, users share content with the belief that what goes around comes around and that others will in turn share with them. Jian and MacKie-Mason characterize sharing network structures in which this belief holds and sharing is an equilibrium.

User-Contributed Content Systems User-contributed content systems are web-based applications that allow users to submit content to the system and then share that content publicly. While a good number of these systems exist, not all of them have succeeded in getting users to contribute valuable information. There have been a few studies of these systems that try to understand users motivations and incentives for contribution.

Wikipedia is one of the largest and most popular user contributed content system. I am not aware of much research that provides concrete results on why users contribute to Wikipedia. Cosley et al. (2007) were able to motivate increased contributions by providing existing users with customized suggestions of Wikipedia articles that the users might be interested in editing and improving. They conducted a field experiment that showed an increase in edits and contributions due to their automatically-provided recommendations. Bryant et al. (2005) describe the process where users progress from novices to expert 'Wikipedians,' increasing their contributions as they gain expertise. Burke and Kraut (2008) look at the challenge of building a consensus in an online community by studying how Wikipedians decide which users get promoted to administrator status. They found that the written criterion are important but not strictly enforced, and the community also developed some additional implicit criterion that was important for promotion decisions.

Facebook is a very large social networking website with over 250 million users¹⁰, meaning that it has a larger population than any country other than China, India, and the United States.¹¹ Ellison et al. (2007) found that people use Facebook to create and maintain ‘bridging social capital.’ (Putnam, 2000) This means that users are trying to maintain the ‘weak ties’ in their social network that Granovetter (1973) found can be very valuable. Burke et al. (2009) studied how new users to Facebook contribute photos, and found strong evidence of social learning; users learned from their ‘friends’ how to contribute and what kinds of contributions were appropriate.

Preece and Schneiderman (2009) study the spectrum of online social activity and present a theory of how individual users develop over time from lurking readers to basic contributors on up to social leaders in an online community. They believe that any vibrant online community supports all of these diverse types of participation. They don’t predict how users move up the scale of contribution and participation, but they do try to characterize the types of users and the types of contribution at each level of participation. This process is a more general version of the process described by Bryant et al. (2005).

Games Luis von Ahn has come up with a very interesting method of motivating users: give them a fun game that has a beneficial side-effect. His first, and probably best, example is the ESP Game, now better known as the Google Image Labeler (von Ahn and Dabbish, 2004). The ESP game randomly matches two players together and presents the pair of them with a series of images. For each image, the players must independently type words in that describe the image, and get points when both players agree on a word. These words that are agreed upon then are good labels for that image, since the image is all the users have in common. If you present the same image to many different pairs of players, then you end up with a good list of labels, and an estimate (frequency of agreement) for how well that word describes the image. Image recognition and labeling is a computationally hard (NP-Hard) problem in computer science, but humans are innately good at it.

Two other games have been published by von Ahn: Peekaboom (von Ahn et al., 2006b) is a game for locating objects in images, and Verbosity (von Ahn et al., 2006a) is a game for discovering common-sense facts. All three games here take advantage of skills that most humans have but computers find difficult or impossible. Labeling images, locating objects in images, and discovering common-sense facts about the world are all NP-Hard problems in computing, but are fairly easy for most people to do. The difficulty in creating these games is making them fun. Luis went through many prototype designs (over 30, private

¹⁰<http://www.facebook.com/press/info.php?statistics>, retrieved on August 9, 2009.

¹¹World Population statistics retrieved from Wikipedia at http://en.wikipedia.org/wiki/List_of_countries_by_population on August 9, 2009.

communication) in order to come up with these three games. It is not clear how to make such games fun, but it is still worthwhile to try. People will often spend many hours playing games, and harnessing that time and effort to do productive work can be very fruitful.

Recently, Microsoft began using a new game called ‘Page Hunt’ to motivate users to help them improve the search results from their new ‘Bing’ search engine (Ma et al., 2009). The game rewards users for helping them provide better metadata about query results, provide alternative queries, and identify ranking issues.

2.5 My Contribution

Overall, there is no strong consensus for how to motivate users of a user-contributed content system to contribute information. Both psychology and economics have studied this general problem and put forth solutions, but those solutions are imperfect for a number of reasons. In particular, none of the existing solutions are good candidates for how to motivate users of a social firewall system to contribute information about security decisions.

In this dissertation, I describe my research into how to motivate users to contribute information to a social firewall. I begin in Chapter 3 by describing the information that home computer users possess in the form of folk models. I use an interview study to understand how users differ in the way they think about security. With this knowledge, I can better design systems to take advantage of this heterogeneity in user beliefs to increase contributions through information sharing. By understanding how home computer users make security decisions, I can tailor the system to solicit information that would be the best influence on others’ decisions. Next, in Chapter 4, I describe a mixed methods case study of one current user-contributed content system: delicious.com. From this case study I discover one powerful method of encouraging users to contribute information: have users contribute for personal reasons and expose the information publicly as a side effect. I also describe an important constraint that is important for such designs to work: incentive alignment. In Chapter 5, I describe a simple variation on some public goods mechanisms from economics that works for users contributing information: setting a minimum threshold for contribution and denying access to anyone who doesn’t meet this threshold. Using a mathematical model, I predict how a users will react to such a design and provide advice for using this mechanism. Finally, in Chapter 6, I gather all of this knowledge and use it to propose a more concrete design for a social firewall. Together, I use a variety of research methods to propose a number of novel mechanisms for inducing users to contribute information to social computing systems, and then give examples of how those mechanisms can be applied to the design of a social firewall system.

Chapter 3

Folk Models of Home Computer Security

3.1 Introduction

Home users are installing paid and free home security software at a rapidly increasing rate.¹ These systems include anti-virus software, anti-spyware software, personal firewall software, personal intrusion detection / prevention systems, computer login / password / fingerprint systems, and intrusion recovery software. Nonetheless, security intrusions and the costs they impose on other network users are also increasing. One possibility is that home users are starting to become well-informed about security risks, and that soon enough of them will protect their systems that the problem will resolve itself. However, given the “arms race” history in most other areas of networked security (with intruders becoming increasingly sophisticated and numerous over time), it is likely that the lack of user sophistication and non-compliance with recommended security system usage policies will continue to limit home computer security effectiveness.

Recently, home computer security has been rising in importance because of the emergence of *botnets*. A botnet is a distributed army of computers not owned by, but under at least partial control of an attacker. The attacker breaks into each computer and leaves a remote control program running. Once the zombie receives instructions from the attacker, it usually independently follows the instructions without further contact. Often attackers automate the process, allowing them to build extremely large botnets. Sizes in the tens of thousands are considered moderate, and networks of 100,000 or more are not uncommon Sieberg (2006). Home computers, or computers whose primary use is for consumer or other

¹Despite a worldwide recession, the computer security industry grew 18.6% in 2008, totaling over \$13 billion according to a recent Gartner report. (Contu and Cheung, 2009)

home purposes, seem to be disproportionately represented in botnets.

To design security technologies (like a social firewall) that will induce changes in *user behavior*, it is first necessary to understand how users make security decisions, and thence to characterize the security problems that result from these decisions. To this end, I have conducted a qualitative study to understand users' *mental models* (Johnson-Laird, 1980; D'Andrade, 2005) of attackers and security technologies. Mental models describe how a user thinks about a problem; it is the model in the person's mind of how things work. People use these models to make decisions about the effects of various actions. (Johnson-Laird et al., 1998)

In particular, I investigate the potential existence of folk models for home computer users. Folk models are mental models that are not accurate in the real world, thus leading to erroneous decision making, but are shared among similar members of a culture. (D'Andrade, 2005) It is well-known that in technological contexts users often operate with incorrect folk models. (Adams and Sasse, 1999) To understand the rationale for home users' behavior, and in particular to design technology that induces improved behavior, it is important to understand the decision model that people use. If technology is designed on the assumption that users have correct mental models of security threats and security systems, it will not induce the desired behavior when they are in fact making choices according to a different model.

As an example of folk models, Kempton (1986) studied folk models of thermostat technology in an attempt to understand the wasted energy that stems from poor choices in home heating. He found that his respondents possessed one of two mental models for how a thermostat works. According to the 'valve' model a thermostat works like a faucet: turning it higher makes more heat come out. In the 'feedback' model, the thermostat turns the heater on if the temperature is too low and off if it is high enough. The 'feedback' model is closer to an expert's understanding of thermostats, but both models have flaws. Both models cause thermostat users to make poor decisions, but both models can lead to correct decisions that the other model gets wrong. For example, the 'valve' theory predicts that more fuel will be consumed at higher temperatures than at lower ones, and the 'feedback' theory would disagree. This prediction is correct, though the reasoning is wrong. The extra fuel comes not from a valve that is open farther to produce more heat, but from the fact that hotter houses lose heat to the outside environment faster than cool houses. Kempton concludes that "Technical experts will evaluate folk theory from this perspective [correctness] – not by asking whether it fulfills the needs of the folk. But it is the latter criterion [...] on which sound public policy must be based." The same argument holds for technology design: whether the folk models are correct or not, technology should be designed to work well with

the folk models actually employed by users.²

For home computer security, I examine two related problems: (1) how do home computer users conceptualize potential security threats? and (2) how do these users use their mental model to decide upon security responses? I found that home computer users conceptualize computer security threats in multiple ways; consequently, users make different decisions based on their conceptualization.

3.1.1 Existing Literature on Security Understanding

Hackers have targeted home computers because they are low-hanging fruit. The vast majority of home computers are administered by people who have little security knowledge or training. While home users may be concerned with the security of their computers, they usually lack the training and expertise to effectively use security technologies. Ross Anderson's 1993 study of Automated Teller Machine (ATM) fraud found that the majority of the fraud committed using these machines was not due to technical flaws, but to errors in deployment and management failures. These problems illustrate the difficulty that even professionals face in producing effective security.

Existing research has investigated how non-expert users deal with security and network administration in a home environment. Dourish et al. (2004) conducted a related study, inquiring not into mental models but how corporate knowledge workers handled security issues. Gross and Rosson (2007) also studied what security knowledge end users possess in the context of large organizations. And Grinter et al. (2005) interviewed home network users about their network administration practices.³ Combining the results from these papers, it appears that many users exert much effort to avoid security decisions. All three papers report that users often find ways to delegate the responsibility for security to some external entity; this entity could be technological (like a firewall), social (another person or IT staff), or institutional (like a bank). Users do this because they feel like they don't have the ability to maintain proper security, or to deal with problems when things go wrong. However, these papers do report that despite this delegation of responsibility, many users still make numerous security-related decisions on a regular basis. These papers do not explain how those decisions get made; rather, they focus mostly on the anxiety these decisions create.

I add structure to these observations by describing the folk models that home computer users use to make security decisions. I also focus on differences across people, and look at

²It may be that users can be re-educated to use more correct mental models, and that incentives could induce re-education, but generally it will be harder to embed incentives for society-wide re-education into technology than in social policy, so I focus on behavioral changes, not educational changes.

³More information about these papers can be found in Section 2.1.2.

different methods of dealing with security issues rather than trying to find general patterns. These mental models may explain differences observed between users in these studies, and provide a context and reasoning for many of the decisions that these researchers observed.

Camp (2006) proposed using mental models as a framework for communicating complex security risks to the general populace. Asgharpour et al. (2007) built on this by conducting a card sorting experiment that matches these analogies with the mental models of users. They found that experts and non-experts show sharp differences in which analogy their mental model is closest to.

Camp et al. began by assuming a small set of analogies that they believe function as mental models. Rather than pre-defining the range of possible models, I treat these mental models as a legitimate area for inductive investigation, and endeavor to uncover users' mental models in whatever form they take. This prior work confirms that the concept of mental models may be useful for home computer security, but made assumptions which may or may not be appropriate. I fill in the gap by inductively developing an understanding of just what mental models people actually possess. Also, given the vulnerability of home computers and this finding that experts and non-experts differ sharply (Asgharpour et al., 2007), I focus solely on non-expert home computer users.

Another related area of literature concerns designing security software for non-technical users. There has been some work in the field of human-computer interaction that attempts to design security systems that non-specialists can easily understand and use. (Cranor and Garfinkel, 2005) My work will help to inform such designs by helping designers better understand the capabilities and needs of home computer users when managing security.

3.1.2 Botnets and Home Computer Security

This research is motivated by a recent development in computer security. In the past, computers were targeted by hackers approximately in proportion to the amount of value stored on them or accessible from them. Computers that stored valuable information, such as bank computers with account numbers and access to transfer funds were a common target, while home computers used to view personal pictures were fairly innocuous. However, recently attackers have used a technique known as a 'botnet.' Briefly, the attacker hacks into a number of computers and installs special control software on those computers. These hacks can be direct or through a virus/worm. This control software then listens for commands from a master control computer. The hacker can give the master control computer a single command, and it will be carried out by all of the compromised computers (called zombies) it is connected to. (Bacher et al., 2005; Barford and Yegneswaran, 2006) This enables crimes

that require large numbers of computers, such as spam, click fraud, and distributed denial of service (Trend Micro, 2006). Observed botnets range in size from a couple hundred zombies to 50,000 or more zombies. Hacker techniques for developing and using botnets have been evolving rapidly. (Bacher et al., 2005)

Since any computer with an Internet connection will be an effective zombie, hackers have logically turned to attacking the most vulnerable population – home computers. Home computer users are usually untrained and have few technical security skills. While some software has improved the average level of security of this class of computers, home computers still represent the largest population of vulnerable computers with decent Internet connections. As these computers are compromised, they are often used to commit crimes against other people. The vulnerability of home computers is a security problem for many companies and individuals who are the victims of these crimes, even if their computers are secure.

The popular press is reporting that botnets are currently being used for a number of different crimes. 9 of out 10 email messages on the Internet are spam (Stone, 2006), and 80% of those messages are being sent through botnets (Markoff, 2007; Sieberg, 2006). Botnets are used to steal personal information, and to conduct various types of online fraud such as click fraud and trust fraud (Sieberg, 2006). Botnets are also used for extortion. Criminals will contact various websites and threaten to use a botnet to overwhelm the website, preventing legitimate users from accessing it. Unless, of course, the website pays up. Ratliff (2006) vividly describes the difficulties a number of victims have had in dealing with this crime. In January 2007 John Markoff reported in the *New York Times* that

[B]otnets are being blamed for the huge spike in spam that bedeviled the Internet in recent months, as well as fraud and data theft. Security researchers have been concerned about botnets for some time because they automate and amplify the effects of viruses and other malicious programs. What is new is the vastly escalating scale of the problem” (Markoff, 2007).

Some facts about the current severity of the problem (all reported in (Markoff, 2007):

- David Dagon of the Georgia Institute of Technology says there is scientific consensus that botnet programs are present on about 11% of the more than 650 million computers attached to the Internet.
- Rick Wesson, CEO of Security Intelligence, reports that sensor data identifies more than 250,000 new botnet infections daily.
- MessageLabs reports that more than 80% of all spam now originates from botnets.

- In December 2006 Trend Micro identified one 24-hour period in which computers from a single Internet service provider generated more than one billion spam messages.
- In a single data file retrieved from a botnet, 54,926 log-in credentials and 281 credit card numbers could be found. This data affected 1,239 companies.

Botnets are a major problem for the Internet today, and the threat they pose is growing. Compromised home computers form the majority of most botnets today. Home computer users appear to be unable to prevent their computers from being co-opted into botnets by Internet hackers. In this chapter, I use my description of the folk models to understand why home computer users cannot prevent their computers from being compromised, and what they are currently doing to protect their computers.

3.2 Methods

Though home computer users have little technical training, they do have some idea of the security threats they face and the potential countermeasures; indeed, the market for home security software is quite active. I conducted a qualitative inquiry into how home computer users understand and think about potential threats to their computer and information security.

I began by conducting a series of 23 semi-structured interviews. Respondents were chosen from a snowball sample (Kuzel, 1992) of home computer users evenly divided between two midwestern U.S. cities in two different states. I began with a few home computer users that I knew in these states. I then asked them to refer me to others in the area who might be information-rich informants. I screened these potential respondents to exclude people who had expertise or extensive training in computers or computer security. From those not excluded, I purposefully selected respondents for maximum variation (Kuzel, 1992); I chose respondents from a wide variety of backgrounds, ages, and socio-economic classes. Ages ranged from undergraduate (19 years old) up through retired (over 70). Socio-economic status was not explicitly measured, but ranged from recently graduated artist living in a small efficiency up to a successful executive who owns a large house overlooking the main river through town. Selecting for maximal variation allows me to document diverse variations in folk models and identify important common patterns. (Kuzel, 1992)

After interviewing the chosen respondents, I asked them to refer me to more people with home computers who might provide lots of useful information. In this way, I grew my potential interview pool by “snowballing” out from an initial pool of people. This snowballing through recommendations helped to ensure that the contacted respondents

would information-rich (Kuzel, 1992) and cooperative. These new potential respondents were also screened, selected, and interviewed.

The purpose of qualitative research is not to generalize to a population; rather, it is to explore phenomenon in depth. To avoid misleading readers, I do not report how many users possessed each folk model. Instead, I describe the full range of folk models I observed. To develop depth in my exploration of the folk models of security, I used an iterative methodology as is common in qualitative research. (Onwuegbuzie and Leech, 2007) I conducted multiple rounds of interviews punctuated with periods of analysis and tentative conclusions. The first round of 23 interviews was conducted in Summer 2007. Preliminary analysis proceeded throughout the academic year, and a second round of 10 interviews was conducted in Summer 2008, for a total of 33 respondents. This second round was more focused on models of viruses, hackers, and identity theft, and specifically searched for negative cases of earlier results. (Onwuegbuzie and Leech, 2007) Interviews averaged around 45 minutes each; they were audio recorded and transcribed for analysis.

I developed an (IRB approved) interview protocol that pushes subjects to describe and use their mental models, based on formal methods presented by D'Andrade (2005). I specifically probed for past instances where the respondents would have had to use their mental model to make decisions, such as past instances of security problems, or efforts undertaken to protect their computers. By asking about instances where the model was applied to make decisions, I enabled the respondents to uncover beliefs that they might not have been consciously aware of. It also ensured that the respondents believe the model enough to base choices on it.

My focus in the first round was broad and exploratory. I asked about any security-related problems the respondent had faced or was worried about; I also specifically asked about viruses, hackers, data loss, and data exposure (identity theft). I probed to discover what countermeasures the respondents used to mitigate these risks. Since this was a semi-structured interview, I followed up on many responses by probing for more information.

After some preliminary analysis of this data, I drew some tentative conclusions and had many more questions that needed clarification. It seemed that many respondents had very distinct mental models of 'hackers' and 'viruses' and that there were a couple of major types of each model. To better elucidate these models and to look for negative cases, I conducted 10 more second-round interviews in a third midwestern U.S. city in a third state. For the second round I developed a new (IRB approved) interview protocol. I focused more on three specific threats that subjects face: viruses, hackers, and identity theft. I challenged respondents to describe instances where these threats influenced their actions.

For this second round, I also used an additional interviewing technique: hypothetical sce-

narios based on knowledge gained in the previous interviews. This technique was developed to help focus the respondents and elicit additional information not present in the first round of interviews. I presented the respondents with three hypothetical scenarios and asked the subjects for their reaction. The three scenarios correspond to each of the three main themes for the second round: finding out you have a virus, finding out a hacker has compromised your computer, and being informed that you are a victim of identity theft. Respondents were asked to give initial reactions and also describe what they think might have happened based on the very little information provided in the scenario; basically, they were asked to describe what they thought would happen if all they knew was that they have a virus or were compromised by a hacker. The interview guide and the details of the scenarios is available in Appendix A.

For each scenario, after the initial description and respondent reaction, I would then add an additional piece of information that appeared to not be salient in the mental models that I discovered after the first round. For example, one preliminary finding from the first round was that people never talked about the creation of computer viruses; it was unclear how they would react to a computer virus that was created by people for a purpose. In the virus scenario, I informed the respondents that the virus in question was written by the Russian mafia. This scenario was taken out of recent news linking the Russian mafia to widespread viruses such as Netsky, Bagle, and Storm.⁴

Another preliminary finding from the first round was that respondents always talked about hackers as breaking in, looking around, and then leaving. No one mentioned the idea of the hacker running an application on the compromised computer. In the hacker scenario, I informed the respondents that the hacker has left a program running on their computer. I asked the subjects if they thought this was reasonable, and what they thought the program might do. Technically, hackers always have some program running on the compromised computer, so this scenario is fairly realistic, even if it is outside of the mental model of some respondents.

Finally, in the identity theft scenario, the first round of data collection could not accurately identify what was included in the model of identity theft; some people thought that using a credit card number without permission was identity theft and others thought that would not be identity theft. To this end, the identity theft scenario started by informing the respondent that they had been the victim of identity theft and asking them for their reaction and what they thought they meant. Then I offered two clarifications: first, that what actually happened was that someone had used their credit card number to make a large purchase; and second, that someone had taken out a large loan in their name. After each clarification, I

⁴<http://www.linuxinsider.com/story/33127.html?wlc=1244817301>

	<i>Alice</i>	<i>Bob</i>	<i>Carol</i>	<i>Deborah</i>	...
<i>Virus Experience</i>	Husband's laptop had one; caused it to freeze. Son's laptop; "ate" hard drive; got from download.	Grandmother got virus in email; rebooted into "safety" mode and displayed skull and crossbones	2 different viruses; had to format and resintall both times. ISP told her	No viruses	...
<i>Worries about Hackers</i>	Trust "computer companies" to deal with it	No problems, but "always in the back of my head"	"You have my music. Wahoo. If they really care that much, go ahead and look around"	I'm not important enough to be targetted. Still, doesn't put CC number in computer	...
<i>Sources of Information</i>	20/20 story about MySpace. Lots of stories from clients	Personal experience and stories from family members	ISP told her she had a virus. Learned some working collections for a large bank.	Her sons warn her about opening attachments. Feels confused. Got a crash course from job.	...
⋮	⋮	⋮	⋮	⋮	⋮

Table 3.1 A fragment of the data matrix from the initial analysis of Round 1. It includes a basic descriptions of each subject's statements for each of the major questions in the interview.

asked the respondent for their reaction and if that was indeed identity theft in their mind.

Once I had all of the data collected and transcribed, I conducted both inductive and deductive coding of the data to look both for predetermined and emergent themes. (Miles and Huberman, 1994) I began with a short list of major themes I expected to see from my pilot interviews, such as information about viruses, hackers, identity theft, countermeasures, and source of information. I identified and labeled (coded) instances when the respondents discussed these themes. I then expanded the list of codes as I noticed interesting themes and patterns emerging.

Once all of the data was coded, I summarized the data on each topic by building a data matrix in the style of Miles and Huberman (1994). A fragment of this matrix can be seen in Table 3.1. This data matrix helped me to identify basic patterns in the data across subjects, to check for representativeness, and to look for negative cases. (Onwuegbuzie and Leech, 2007) After building the initial summary matrices, I identified patterns in the way respondents

talked about each topic, paying specific attention to word choices, metaphors employed, and explicit content of statements. Specifically, I looked for themes in which users differ in their opinions (negative case analysis). These themes became the building blocks for the mental models: each theme became a feature of a model, and a model consists of a collection of features. I built a second matrix that matched subjects with these features of mental models. The part of this matrix for the 10 participants in Round 2 can be seen in Table 3.2. This second matrix allowed me to identify and characterize the various mental models that I encountered. Table 3.3 Shows which participants from Round 2 had each of the 8 models. A similar table was developed for the Round 1 participants, and is available in Appendix B. I then took the description of the model back to the data, verified when the model description accurately represented the respondents descriptions, and looked for contradictory evidence and negative cases. (Onwuegbuzie and Leech, 2007) This allowed me to update the models with new information or insights garnered following up on surprises and incorporating outliers. This was an iterative process; I continued updating model descriptions, looking for negative cases, and checking for representativeness until I felt that the model descriptions I had accurately represented the data. In this process, I developed further matrices as data visualizations, some of which appear in my descriptions below.

3.3 Folk Models of Security Threats

I identified a number of different folk models in the data. Each of these folk models were shared by multiple respondents in this study. I divide the folk models into broad categories based on a distinction that most subjects possessed: 1) models about viruses, spyware, adware, and other forms of malware which everyone referred to under the umbrella term ‘virus’; and 2) models about the attackers, referred to as ‘hackers,’ and the threat of ‘breaking in to’ a computer.

Each respondent had at least one model from each of the two categories. For example, Nicole⁵ believed that viruses were mischievous, and hackers are criminals who target big fish. These models are not necessarily mutually exclusive. For example, a few respondents talked about different types of hackers and would describe more than one folk model of hackers.

Note that by listing and describing these folk models, in no way do I intend to imply that these models are incorrect or bad in any way. They are all certainly incomplete, and do not exactly correspond to the way malicious software or malicious computer users behave. But, as Kempton (1986) learned in his study of home thermostats, what is important is not

⁵All subjects have been given pseudonyms for anonymity.

	<i>Christine</i>	<i>Dana</i>	<i>Erica</i>	<i>Floyd</i>	<i>Gail</i>	<i>Hayley</i>	<i>Irving</i>	<i>Jack</i>	<i>Kenneth</i>	<i>Lorna</i>
<i>Creator</i>	No Creator	Teenager	No Creator	Teenager	Teenager	Criminal	Teenager + Crime	Teenager + Crime	Teenager	Teenager
<i>Effects</i>	Errors	Annoying	Errors	Annoying	Annoying	Annoying + Spy	Spy	Annoying + Spy	Errors + Spy	Annoying
<i>Only Visible?</i>	Y	Y	Y	N	N	Y	N	N	N	Y
<i>How to catch</i>	Active	Active + Passive	Active	Active + Passive	Passive	Passive	Active	Happen + Active	Active	Active
<i>Sources</i>	Email	Web + Email + Downloads	Web + Downloads	Web + Downloads	Webpages	Web + Email + Downloads	Email	Downloads + Email	Email	Email + Web
<i>Identity</i>	Teenage	Teenager	Criminal	Anyone	Criminal	Teenager + Crime	Criminal	Teenager + Crime	Teenager	Teenager
<i>Behavior</i>	Secrets	Break Stuff	Big Fish	Big Fish	ID Theft	ID Theft + Secrets	Big Fish	ID Theft + Break Stuff	Big Fish + Databases	Databases
<i>How to Prevent</i>	No Info	Care on Internet	Care on Internet	Passwords + No Info	Care on Internet	Futility	Trust Software + Passwords	Care on Internet	Trust Software	Trust Institutions
Relationship	<i>Hacker & Virus</i>	Separate	Separate	Separate	Separate	As Tool	Separate	As Tool	As Tool	Separate

Table 3.2 Intermediate data matrix developed for analysis. This matrix includes a number of facets of mental models vertically matched with each of the 10 Round 2 participants. More details about the entries in this table can be found in Appendix B

		<i>Christine</i>	<i>Dana</i>	<i>Erica</i>	<i>Floyd</i>	<i>Gail</i>	<i>Hayley</i>	<i>Irving</i>	<i>Jack</i>	<i>Kenneth</i>	<i>Lorna</i>
Viruses	Viruses are Bad										
	Buggy Software	x		x							
	Mischief		x		x	x					x
	Support Crime						x	x	x	x	
Hackers	Graffiti	x	x			x		x			
	Burglar					x	x		x		
	Big Fish			x	x			x			
	Contractor									x	x

Table 3.3 A sample data matrix from near the end of the analysis. This matrix shows which folk model was held by the Participants in Round 2. A similar table was developed for the participants in Round 1.

how accurate the model is but how well it serves the needs of the home computer user in making security decisions.

Additionally, there is not “correct” model that can serve as a comparison. Even security experts will disagree as to the correct way to think about viruses or hackers. To show an extreme example, Medin et al. (2006) conducted a study of expert fishermen in the Northwoods of Wisconsin. They looked at the mental models of both Native American fishermen and of majority-culture fishermen. Despite both groups being experts, the two groups showed dramatic differences in the way fish were categorized and classified. Majority-culture fishermen grouped fish into standard taxonomic and goal-oriented groupings, while Native American fishermen groups fish mostly by ecological niche. This illustrates how even experts can have dramatically different mental models of the same phenomenon, and any single expert’s model is not necessarily correct. However, experts and novices do tend to have very different models; Asgharpour et al. (2007) found strong differences between expert and novice computer users in their mental models of security. In Section 3.4 I discuss how well each of the models work for my respondents.

Common Elements of Folk Models Most of the respondents made a distinction between ‘viruses’ and ‘hackers.’ To them, these are two separate threats that can both cause problems. Some people believed that viruses are created by hackers, but they still usually saw them as distinct threats. Some people realized this and tried to describe the difference; for example at one point in the interview Irving tries to explain the distinction by saying “The hacker is an individual hacking, while the virus is a program infecting.” After some thought, he clarifies

his idea of the difference a bit: “So it’s a difference between something automatic and more personal.” This description is characteristic of how many respondents think about the difference: viruses are usually more programatic and automatic, where hacking is more like manual labor, requiring the hacker to be sitting in front of a computer entering commands.

This distinction between hackers and viruses is not something that most of the respondents had thought about; it existed in their mental model but not at a conscious level. Upon prompting, Dana decides that “I guess if they hack into your system and get a virus on there, its gonna be the same thing.” She had never realized that they were distinct in her mind, but it makes sense to her that they might be related. She then goes on to ask the interviewer if she gets hacked, can she forward it on to other people?

This also illustrates another common feature of these interviews. When exposed to new information, most of the respondents would extrapolate and try to apply that information to slightly different settings. When Dana was prompted to think about the relationship between viruses and hackers, she decided that they were more similar than she had previously realized. Then she began to apply ideas from one model (viruses spreading) to the other model (can hackers spread also?) by extrapolating from her current models. This is a common technique in human learning and sensemaking. (Russell et al., 1993) I suspect that many details of the mental models were formed in this way. Extrapolation is also useful for analysis; how respondents extrapolate from new information reveals details about mental models that are not consciously salient during interviews. (Collins and Gentner, 1987; D’Andrade, 2005) During the interviews I used a number of prompts that were intended to challenge mental models and force users to extrapolate in order to help surface more elements of their mental models.

3.3.1 Models of Viruses and other Malware

All of the respondents had heard of computer viruses and possessed some mental model of their effects and transmission. The respondents focused their discussion primarily on the effects of viruses and the possible methods of transmission. In the second round of interviews, I prompted respondents to discuss how and why viruses are created by asking them to react to a number of hypothetical scenarios. These scenarios help me understand how the respondents apply these models to make security-relevant decisions.

All of the respondents used the term ‘virus’ as a catch-all term for malicious software. Everyone seemed to recognize that viruses are computer programs. Almost all of the respondents classify many different types of malicious software under this term: computer viruses, worms, trojans, adware, spyware, and keyloggers were all mentioned as ‘viruses.’ The

	<i>Bad</i>	<i>Buggy Software</i>	<i>Mischief</i>	<i>Support Crime</i>
<i>Creator</i>	Unspecified	Bad people	Mischievous hackers	Criminals
<i>Purpose of viruses</i>	Unspecified	No purpose	Cause mischief; cause annoying problems	Gather information for identity theft
<i>Effects of infection</i>	General notion of bad things happening	Same effects as buggy software, but more extreme	Annoying problems with computers	No direct harm to computer; stolen information
<i>Method of transmission</i>	“Catch” viruses; miscellaneous methods of catching them	Must be manually downloaded and executed	Passive “catching” by visiting shady websites or opening shady email	Spread automatically, or installed by hackers

Table 3.4 Summary of folk models about viruses, organized by model features

respondents don’t make the distinctions that most experts do; they just call any malicious computer program a ‘virus.’

However, a few subjects made a distinction one specific type of malware and rest of the malware, which are still called ‘viruses.’ For example, Jack distinguishes ‘trojans’ from other ‘viruses.’ To Jack, trojans are like their namesake; a trojan is “something that you’re fooled into taking into your computer.” But viruses are “kind of hidden” programs that “attach themselves to harmless files.” Viruses have an “automatic mechanism for reproducing” where trojans don’t. Note that all other types of malware are still referred to as ‘viruses.’

Thanks to the term ‘virus,’ all of the respondents used some sort of medical terminology to describe the actions of malware. Getting malware on your computer means you have ‘caught’ the virus, and your computer is ‘infected.’ Everyone who had a Mac seemed to believe that Macs are ‘immune’ to virus and hacking problems.

In addition to the mental model that each respondent possessed, they seemed to know one or two pieces of security advice that they believed came from security experts. Every respondent in this study believed the advice to not open attachments from strangers. Everyone believed that doing this will lead to getting a virus. However, different people fit this information into their mental model in different ways.

Overall, I found four distinct folk models of ‘viruses.’ These models differed in a number of ways. One of the major differences is how well-specified and detailed the model was, and therefore how useful the model was for making security-related decisions. One model was very under-specified, labeling viruses as simply ‘bad.’ Respondents with this model

had trouble using it to make any kind of security-related decisions because the model didn't contain enough information to provide guidance. Two other models (the *Mischief* and *Crime* models) were fairly well-described, including how viruses are created and why, and what the major effects of viruses are. Respondents with these models could use them to extrapolate many different situations and use them to make many security-related decisions on their computer. Table 3.4 summarizes the major differences between the four models.

3.3.1.1 Viruses are Generically 'Bad'

A few subjects had a very under-developed model of viruses. These subjects knew that viruses cause problems, but these subjects couldn't really describe the problems that viruses cause. They just knew that they were generically 'bad' to get and should be avoided.

Respondents with this model knew of a number of different ways that viruses are transmitted. These transmission methods seemed to be things that the subjects had heard about somewhere, but the respondents did not attempt to understand these or organize them into a more coherent mental model. Zoe believed that viruses can come from strange emails, or from "searching random things" on the Internet. She says she had heard that blocking popups helps with viruses too, and seemed to believe that without questioning. Peggy had heard that viruses can come from "blinky ads like you've won a million bucks."

Respondents with this model are uniformly unconcerned with getting viruses: "I guess just my lack of really doing much on the Internet makes me feel like I'm safer." (Zoe) A couple of people with this model use Macintosh computers, which they believe to be "immune" to computer viruses. Since they are immune, it seems that they have not bothered to form a more complete model of viruses.

Since these users are not concerned with viruses, they do not take any precautions against being infected. These users believe that their current behavior doesn't really make them vulnerable, so they don't need to go to any extra effort. Only one respondent with this model uses an anti-virus program, but that is because it came installed on the computer. These respondents seem to recognize that anti-virus software might help, but are not concerned enough to purchase or install it.

3.3.1.2 Viruses are Buggy Software

One group of respondents saw computer viruses as an exceptionally bug-ridden form of regular computer software. In many ways, these respondents believe that viruses behave much like most of the other software that home users experience. But to be a virus, it has to be 'bad' in some additional way. Primarily, viruses are 'bad' in that they are poorly written

software. They lead to a multitude of bugs and other errors in the computer. They bring out bugs in other pieces of software. They tend to have more bugs, and worse bugs, than most other pieces of software. But all of the effects they cause are the same types of effects you get from buggy software: viruses can cause computers to crash, or to “boot me out” (Erica) of applications that are running; viruses can accidentally delete or “wipe out” information (Christine and Erica); they can erase important system files. In general, the computer just “doesn’t function properly” (Erica) when it has a virus.

Just like normal software, viruses must be intentionally placed on the computer and executed. Viruses do not just appear on a computer. Rather than ‘catching’ a virus, computers are actively infected, though often this infection is accidental. Some viruses come in the form of email attachments. But they are not a threat unless you actually “click” on the attachment to run it. If you are careful about what you click on, then you won’t get the virus. Another example is that viruses can be downloaded from websites, much like many other applications. Erica believes that sometimes downloading games can end up causing you to download a virus. But still, intentional downloading and execution is necessary to be infected with a virus, much the same way that intentional downloading and execution is necessary to run programs from the Internet.

Respondents with this model did not feel that they needed to exert a lot of effort to protect themselves from viruses. Mostly, these users tried to not download and execute programs that they didn’t trust. Sarah intentionally “limits herself” by not downloading any programs from the Internet so she doesn’t get a virus. Since viruses must be actively executed, anti-virus program are not important. As long as no one downloads and runs programs from the Internet, no virus can get onto the computer. Therefore, anti-virus programs that detect and fix viruses aren’t needed. However, two respondents with this model run anti-virus software just in case a virus is accidentally put on the computer.

Overall, this is a somewhat underdeveloped mental model of viruses. Respondents who possessed this model had never really thought about how viruses are created, or why. When asked, they talk about how they haven’t thought about it, and then make guesses about how ‘bad people’ might be the ones who create them. These respondents haven’t put too much thought into their mental model of viruses; all of the effects they discuss are either effects they have seen or more extreme versions of bugs they have seen in other software. Christine says “I guess I would know [if I had a virus], wouldn’t I?” presuming that any effects the virus has would be evident in the behavior of the computer. No connection is made between hackers and viruses; they are distinct and separate entities in the respondent’s mind.

3.3.1.3 Viruses Cause Mischief

A good number of respondents believed that viruses are pieces of software that are intentionally annoying. Someone created the virus for the purpose of annoying computer users and causing mischief. Viruses sometimes have effects that are often much like extreme versions of annoying bugs: crashing your computer, deleting important files so your computer won't boot, etc. Often the effects of viruses are intentionally annoying such as displaying a skull and crossbones upon boot (Bob), displaying advertising popups (Floyd), or downloading lots of pornography (Dana).

While these respondents believe that viruses are created to be annoying, they rarely have a well-developed idea of who created them. They don't naturally mention a creator for the viruses, just a reason why they are created. When pushed, these respondents will talk about how they are probably created by "hackers" who fit the *Graffiti* hacker model below. But the identity of the creator doesn't play much of a role in making security decisions with this model.

Respondents with this model always believe that viruses can be "caught" by actively clicking on them and executing them. However, most respondents with this model also believe that viruses can be "caught" by simply visiting the wrong webpages. Infection here is very passive and can come from just from visiting the webpage. These webpages are often considered to be part of the 'bad' part of the Internet. Much like graffiti appears in the 'bad' parts of cities, mischievous viruses are most prevalent on the bad parts of the Internet.

While most everyone believes that care in clicking on attachments or downloads is important, these respondents also try to be careful about where they go on the Internet. One respondent (Floyd) tries to explain why: cookies are automatically put on your computer by websites, and therefore, viruses being automatically put on your computer could be related to this.

These 'bad' parts of the Internet where you can easily contract viruses are frequently described as morally ambiguous webpages. Pornography is always considered shady, but some respondents also included entertainment websites where you can play games, and websites that have been on the news like "MySpaceBook" (Gina). Some respondents believed that a "secured" website would not lead to a virus, but Gail acknowledged that at some sites "maybe the protection wasn't working at those sites and they went bad." (Note the passive tense; again, she has not thought about how site go bad or who causes them to go bad. She is just concerned with the outcome.)

3.3.1.4 Viruses Support Crime

Finally, some respondents believe that viruses are created to support criminal activities. Almost uniformly, these respondents believe that identity theft is the end goal of the criminals who create these viruses, and the viruses assist them by stealing personal and financial information from individual computers. For example, respondents with this model worry that viruses are looking for credit card numbers, bank account information, or other financial information stored on their computer.

Since the main purpose of these viruses is to collect information, the respondents who have this model believe that viruses often remain undetected on computers. These viruses do not explicitly cause harm to the computer, and they do not cause bugs, crashes, or other problems. All they do is send information to criminals. Therefore, it is important to run an anti-virus program on a regular basis because it is possible to have a virus on your computer without knowing it. Since viruses don't harm your computer, backups are not necessary.

People with this model believed that there are many different ways for these viruses to spread. Some viruses spread through downloads and attachments. Other viruses can spread "automatically," without requiring any actions by the user of the computer. Also, some people believe that hackers will install this type of virus onto the computer when they break in. Given this wide variety of transmission methods and the serious nature of identity theft, respondents with this model took many steps to try to stop these viruses. These users would work to keep their anti-virus up to date, purchasing new versions on a regular basis. Often, they would notice when the anti-virus would conduct a scan of their computer and check the results. Valerie would even turn her computer off when it is not in use to avoid potential problems with viruses.

3.3.1.5 Multiple Types of Viruses

A couple of respondents discussed multiple types of viruses on the Internet. These respondents believed that some viruses are mischievous and cause annoying problems, while other viruses support crime and are difficult to detect. All users that talked about more than one type of virus talked about both of the previous two virus folk models: the mischievous viruses and the criminal viruses. One respondent, Jack, also talked about a third type of virus that was created by anti-virus companies, but he seemed like he felt this was a conspiracy theory, and consequently didn't take that suggestion very seriously.

For the respondents with multiple models, they generally would take all of the precautions that either model would predict. For example, they would make regular backups in case they caught a mischievous virus that damaged their computer, but they also would regularly

	<i>Graffiti</i>	<i>Burglar</i>	<i>Big Fish</i>	<i>Contractor</i>
<i>Identity of hacker(s)</i>	Young technical geek	Some criminal	Professional criminal hackers	Young technical geek
<i>Level of organization</i>	Solo, or to impress friends	Unspecified	Part of a criminal organization	Solo, but a contractor for criminals
<i>Reason for break-ins</i>	Cause mischief	Look for financial and personal information	Look for financial and personal information	Look for financial and personal information
<i>Effects of break-ins</i>	Lots of computer problems; requires reinstall	Possible harm to computer; exposure of personal information	No harm to computer; exposure of personal information	Exposure of personal information
<i>Target(s)</i>	Anyone; doesn't matter	Opportunistic; could be me	Not me; only looking for rich or important people	Not me; looking for large databases of info
<i>Am I a target?</i>	Possibly	Possibly	No	No

Table 3.5 Summary of folk models about hackers, organized by model features

run their anti-virus program to detect the criminal viruses that don't have noticeable effects. This fact suggests that information sharing between users may be beneficial; when users believe in multiple types of viruses, they take appropriate steps to protect against all types.

3.3.2 Models of Hackers and Break-ins

The second major category of folk models describe the attackers, or the people who cause Internet security problems. These attackers are always given the name "hackers," and all of the respondents seemed to have some concept of who these people were and what they did. The term "hacker" was applied to describe anyone who does bad things on the Internet, no matter who they are or how they work.

All of the respondents describe the main threat that hackers pose as "breaking in" to their computer. They would disagree as to why a hacker would want to "break in" to a computer, and to which computers they would target for their break ins, but everyone agreed on the terminology for this basic action. To the respondents, breaking in to a computer meant that the hacker could then use the computer as if they were sitting in front of it, and could cause a number of different things to happen to the computer. Many respondents stated that they did not understand how this worked, but they still believed it was possible.

My respondents described four distinct folk models of hackers. These models differed mainly in who they believed these hackers were, what they believed motivated these people, and how they chose which computers to break in to. Table 3.5 summarizes the four folk models of hackers.

3.3.2.1 Hackers are Digital Graffiti Artists

One group of respondents believe that hackers are technically skilled people causing mischief. There is a collection of individuals, usually called “hackers,” that use computers to cause a technological version of mischief. Often these users are envisioned as “college-age computer types” (Kenneth). They see hacking computers as sort of digital graffiti; hackers break in to computers and intentionally cause problems so they can show off to their friends. Victim computers are just a canvas for their art.

When respondents with this model talked about hackers, they usually focused on two features: strong technical skills and the lack of proper moral restraint. Strong technical skills provide the motivation; hackers do it “for sheer sport” (Lorna) or to demonstrate technical prowess (Hayley). Some respondents envision a competition between hackers, where more sophisticated viruses or hacks “prove you’re a better hacker” (Kenneth); others see creating viruses and hacking as part of “learning about the Internet” (Jack). Lack of moral restraint is what makes them different than others with technical skills; hackers are sometimes described as people as maladjusted individuals who “want to hurt others for no reason.” (Dana) Respondents will describe hackers as “miserable” people. They feel that hackers do what they do for no good reason, or at least no reason they can understand. Hackers are believed to be lone individuals; while they may have hacker friends, they are not part of any organization.

Users with this model often focus on the identity of the hacker. This identity – a young computer geek with poor morals – is much more developed in their mind than the resulting behavior of the hacker. As such, people with this model can usually talk clearly and give examples of who hackers are, but seem less confident in information about the resulting break-ins that happen.

These hackers like to break stuff on the computer to create havoc. They will intentionally upload viruses to computers to cause mayhem. Many subjects believe that hackers intentionally cause computers harm; for example Dana believes that hackers will “fry your hard drive.” (Dana) Hackers might install software to let them control your computer; Jack talked about how a hacker would use his instant messenger to send strange messages to his friends.

These mischievous hackers were seen as not targetting specific individuals, but rather

choosing random strangers to target. This is much like graffiti; the hackers need a canvas and choose whatever computer they happen to come upon. Because of this, the respondents felt like they might become a victim of this type of hacking at any time.

Often, victims like this felt like there wasn't much they could do to protect themselves from this type of hacking. This was because respondents didn't understand how hackers were able to break into computers, so they didn't know what could be done to stop it. This would lead to a feeling of futility; "if they are going to get in, they're going to get in." (Hayley) This feeling of futility echoes similar statements discussed by Dourish et al. (2004).

One specific type of mischief that some respondents talked about is looking for secrets. Usually, respondents envisioned hackers as going after some sort of government secrets or high-profile corporate secrets, and the hackers did this for the thrill of it. Hackers interested in secrets weren't really looking to sell or profit from these secrets; rather, these secrets were an end unto themselves. Respondents who believed that hackers were looking for secrets felt that they were definitely not intentional targets of hackers because they didn't have any secrets stored on their computer. However, they felt that they might be hacked "by mistake." (Christine)

3.3.2.2 Hackers are Burglars Who Break Into Computers for Criminal Purposes

Another set of respondents believe that hackers are criminals that happen to use computers to commit their crimes. Other than the use of the computer, they share a lot in common with other professional criminals: they are motivated by financial gain, and they can do what they do because they lack common morals. They would "break into" computers to look for information much like a burglar will break into houses to look for valuables. The most salient part of this folk model is the behavior of the hacker; the respondents could talk in detail about what the hackers were looking for but spoke very little about the identity of the hacker.

Almost exclusively, this criminal activity is some form of identity theft. For example, respondents believe that if a hacker obtains their credit card number, for example, then that hacker can make fraudulent charges with it. But the respondents weren't always sure what kind of information the hacker was specifically looking for; they just described it as information the hacker could use to make money. Ivan talked about how hackers would look around the computer much like a thief might rummage around in an attic, looking for something useful. Erica used a different metaphor, saying that hackers would "take a digital photo of everything on my computer" and look in it for useful identity information. Usually, the respondents envision the hacker himself using this financial information (as opposed to

selling the information to others).

One consequence of this targeting of information is that computers are not harmed by the break-ins. Since hackers are simply looking for information, they generally do not cause harm to the computer. They simply rummage around, “take a digital photo,” possibly install monitoring software, and leave. The computer continues to work as it did before. The main concern of the respondents is how the hacker might use the information that they steal.

These hackers choose victims opportunistically; much like a mugger chooses his victims, these hackers will break into any computers they know about to look for valuable information. Or, more accurately, the respondents don’t have a good model of how hackers choose, and believe that there is a decent chance that they will be a victim someday. Gail talks about how hackers are opportunistic, saying “next time I go to their site they’ll nab me.” Hayley believes that they just choose computers to attack without knowing much about who owns them.

Respondents with this belief are willing to take steps to protect themselves from hackers to avoid becoming a victim. Gail tries to avoid going websites she’s not familiar with to prevent hackers from discovering her. Jack is careful to always sign out of accounts and websites when he is finished. Hayley shuts off her computer when she isn’t using it so hackers cannot break into it.

Also, a few respondents believe that this type of hacker will sometimes leave viruses on the computer to continue to collect information. For these people, running an anti-virus program is important to catch these viruses and stop these hackers.

3.3.2.3 Hackers are Criminals who Target Big Fish

Another group of respondents had a conceptually similar model. This group also believes that hackers are Internet criminals who are looking for information to conduct identity theft. However, this group has thought more about how these hackers can best accomplish this goal, and have come to some different conclusions. These respondents seemed willing to believe in “massive hacker groups” (Hayley) and other forms of organization and coordination among criminal hackers.

Most tellingly, this group believes that hackers only target the “big fish.” Hackers primarily break into computers of important and rich people in order to maximize their gains. Every respondent who holds this model believes that he or she is not likely to be a victim because he or she is not a big enough fish. They believe that hackers are unlikely to ever target them, and therefore they were safe from hacking. Irving believe that “I’m small potatoes and no one is going to bother me.” They often talk about how other people are more likely targets: “Maybe if I had a lot of money” (Floyd) or “like if I were a bank

executive” (Erica).

For these respondents, protecting against hackers isn’t a high priority. Mostly they find reasons to trust existing security precautions rather than taking extra steps to protect themselves. For example, Irving talked about how he trusts his pre-installed firewall program to protect him. Both Irving and Floyd trust their passwords to protect them. Basically, their actions indicate that they believe in the speed bump theory: by making it slightly hard for hackers using standard security technologies, hackers will decide it isn’t worthwhile to target them.

3.3.2.4 Hackers are Contractors Who Support Criminals

Finally, there is a sort of hybrid model of hackers. In this view, hackers the people are very similar to the mischievous graffiti-hackers from above: they are college-age, technically skilled individuals. However, their motivations are more intentional and criminal. These hackers are out to steal personal and financial information from people.

Users with this model show evidence of more effort in thinking through their mental model and integrating the various sources of information they have. This model can be seen as a hybrid of the mischievous graffiti-hacker model and the criminal hacker model, integrated into a coherent form by combining the most salient part of the mischievous model (the identity of the hacker) and the most salient part of the criminal model (the criminal activities). Also, everyone who had this model expressed a concern about how hacking works. Kenneth stated that he doesn’t understand how someone can break into a computer without sitting in front of it. Lorna wondered how you can start a program running; she feels you have to be in front of the computer to do that. This indicates that these respondents are actively trying to integrate the information they have about hackers into a coherent model of hacker behavior.

Since these hackers are first and foremost young technical people, the respondents believe that these hackers are not likely to be the identity thieves. They believe that the hackers are more likely to sell this identity information for others to use. Since the hackers just want to sell information, the respondents reason, they are more likely to target large databases of identity information such as banks or retailers like Amazon.com.

Respondents with this model believed that hackers weren’t really their problem. Since these hackers tended to target larger institutions like banks or e-commerce websites, their own personal computers weren’t in danger. Therefore, no effort was needed to secure their personal computers.

However, all respondents with this model expressed a strong concern for who they do business with online. These respondents would only make purchases or provide personal

information to institutions they trusted to get the security right and figure out how to be protected against hackers. These users were highly sensitive to third parties possessing their data.

3.3.2.5 Multiple Types of Hackers

Some respondents believed that there were multiple types of hackers. Most of the time, these respondents would believe that some hackers are the mischievous graffiti-hackers and that other hackers are criminal hackers (using either the burglar or big fish model, but not both). These respondents would then try to make the effort to protect themselves from both types of hacker threats as necessary.

It seems that there is some amount of cognitive dissonance that occurs when respondents hear about both mischievous hackers and criminal hackers. There are two ways that respondents resolve this: the simplest way to resolve this is to believe that some hackers are mischievous and other hackers are criminals, and consequently keep the models separate; a more complicated way is to try to integrate the two models into one coherent belief about hackers. This latter option involves a lot of effort making sense of the new folk model that is not as clear or as commonly shared as the mischievous and criminal models. The ‘contractor’ model of hackers is the result of this integration of the two types of hackers.

3.4 Following Security Advice

Computer security experts have been trying to provide security advice to home computer users for many years now. There are many websites devoted to doling out security advice, and many technical support forums where home computer users can ask security-related questions. There has been much effort to simplify the security advice so regular computer users can easily understand and follow this advice.

However, many home computer users do not follow this advice. This is evident from the large number of security problems that plague home computers. There is a disagreement among security experts as to why this advice isn’t followed. Some experts seem to believe that home users do not understand the security advice, and therefore more education is needed. Others seem to believe that home users are simply incapable of consistently making good security decisions (Cranor, 2008). However, none of these explanations explain which advice does get followed and which advice does not.

The folk models described above help to understand how home computer users choose which expert advice to follow and which advice to ignore. For example, respondents who

believe that viruses are buggy software believe that it is not important to run anti-virus software because if they simply refuse to run unknown software, then they will never get a virus. By better understanding why people choose to ignore certain pieces of advice, we can better craft that advice to have a greater effect.

In Table 3.6, I list 12 common pieces of security advice for home computer users. This advice was collected from the Microsoft Security at Home website⁶, the CERT Home Computer Security website⁷, and the US-CERT Cyber-Security Tips website⁸, and most of it appears on multiple of these websites. This advice represents the distilled wisdom on many computer security experts, and is the closest thing to an agreed-upon expert mental model. This table then summarizes, for each folk model, whether that advice is important to follow, helpful but not essential, or not necessary to follow.

To me, the most interesting entries indicate that users believe that a piece of security advice is not necessary to follow (labeled ‘xx’ in the table). These entries show how home computer users apply their folk models to determine for themselves whether following a given piece of advice is important. Also interesting are the entries labeled ‘??’; these entries indicate places where users believe that the advice will help with security, but do not see the advice as so important that it must always be followed. Often users will decide that following advice labeled with ‘??’ is too costly in terms of effort or money, and decide to ignore it. Advice labeled ‘!’ is extremely important, and the respondents feel that it should never be ignored, even if following it is inconvenient, costly, or difficult.

3.4.1 Anti-Virus Use

Advice 1–3 has to do with anti-virus technology: Advice #1. states that anti-virus software should be used; #2. states that the virus signatures need to be constantly updated to be able to detect current viruses; and #3. states that the anti-virus software should regularly scan a computer to detect viruses. All of these are best practices for using anti-virus software.

Respondents mostly use their folk models of viruses to make decisions about anti-virus use, for obvious reasons. Respondents who believe that viruses are just buggy software also believe it is not necessary to run anti-virus. This is because they think they can keep viruses off of their computer by controlling what gets installed on their computer; they believe viruses need to be executed manually to infect a computer, and if they never execute one then they don’t need anti-virus.

Respondents with the under-developed folk model of viruses, who refer to viruses as

⁶<http://www.microsoft.com/protect/default.aspx>, retrieved July 5, 2005

⁷<http://www.cert.org/homeusers/HomeComputerSecurity/>, retrieved July 5, 2005

⁸<http://www.us-cert.gov/cas/tips/>, retrieved July 5, 2005

	<i>Virus Models</i>				<i>Hacker Models</i>			
	<i>Viruses are Bad</i>	<i>Buggy Software</i>	<i>Mischief</i>	<i>Support Crime</i>	<i>Graffiti</i>	<i>Burglar</i>	<i>Big Fish</i>	<i>Contractor</i>
1. Use anti-virus	??	xx	??	!!		!!	xx	xx
2. Keep anti-virus updated	xx	xx	??	!!				xx
3. Regularly scan computer with anti-virus	xx	xx	??	!!				xx
4. Use security software (firewall, etc.)	xx		??		??	??	??	xx
5. Don't click on attachments	!!	!!	!!	!!	!!	!!		
6. Be careful downloading from websites	??	!!	??	!!	??	??	xx	xx
7. Be careful which websites you visit		xx	!!	??	!!	!!	??	!!
8. Disable scripting in web and email								xx
9. Use good passwords					??		??	xx
10. Make regular backups		??	!!	xx	!!	xx	xx	xx
11. Keep patches up to date		??	xx	!!	!!	!!	xx	xx
12. Turn off computer when not in use		xx	xx	!!	??	!!	xx	xx

- !! Important It is very important to follow this advice
- ?? Maybe Following this advice might help, but it isn't all that important to do
- xx Not Necessary It is not necessary to follow this advice
- Not Applicable This model does not have anything to say about this advice, or there is insufficient data from the interviews to determine an opinion

Table 3.6 Summary of Expert Security Advice. Each folk model responds to this advice differently.

generically 'bad,' also do not use anti-virus software. These people understand that viruses are harmful and that anti-virus software can stop them. However, they have never really thought about specific harms a virus might cause to them. Therefore, they don't feel that they have a strong reason to prevent those harms from occurring, and generally find it unnecessary to exert the effort to follow the best practices around anti-virus software.

Finally, one group of respondents believe that anti-virus software can help stop hackers. Users with the burglar model of hackers believe that regular anti-virus scans can be important because these burglar-hackers will sometimes install viruses to collect personal information. Regular anti-virus use can help detect these hackers.

3.4.2 Other Security Software

Advice #4 concerns other types of security software. This states that home computer users should run a firewall or other type of Internet security suite. I think that most of the respondents didn't understand what this security software did other than a general notion of

providing “security.” As such, no one included the security software as an important component of their mental model. Respondents who held the graffiti-hacker or burglar-hacker models believed that this software must help with hackers somehow, even though they don’t know how, and would suggest installing it. But since they don’t understand how it works, they wouldn’t consider it of vital importance. This highlights an opportunity for home user education; I suspect that if these respondents better understood how security software help protect against hackers, then they might be more interested in using it and maintaining it.

One interesting belief about this software comes from the respondents who believe hackers only go after big fish. For these respondents, security software can serve as a speedbump that discourages hackers from casually breaking into their computer. For these people, they don’t care exactly how it works as long as it does something.

3.4.3 Email Security

Advice #5 is the only piece of advice about email on my list. It states that you shouldn’t open attachments from people you don’t recognize. Everyone in my sample was familiar with this advice and had taken it to heart. Everyone believed that viruses can be transmitted through email attachments, and therefore not clicking on unknown attachments can help prevent viruses.

3.4.4 Web Browsing

Advice 6-9 all deal with security behaviors while browsing the web. Advice #6 states that users need to ensure that they only download and run programs from trustworthy sources. Many types of malware are spread through downloads. Advice #7 states that users should only browse webpages from trustworthy sources. There are many types of malicious websites such as phishing websites, and some websites can spread malware simply by visiting the site and executing the javascript on the website. Advice #8 states that users should disable scripting like Java and JavaScript in their web browsers. Often there are vulnerabilities in these scripts, and some malware uses these vulnerabilities to spread. And Advice #9 suggests using good passwords so attackers cannot guess their way into your accounts.

Overall, many respondents would agree with most of this advice. However, no one seemed to understand the advice about web scripts; indeed, no one seemed to even understand what a web script was. Advice #8 was largely ignored because it wasn’t understood.

Everyone seemed to think that they need to be careful in choosing what to download. Downloads were strongly associated with viruses in the respondents’ minds. However, only users with well-developed models of viruses (the *Mischief* and *Support Crime* models)

believed that viruses can be “caught” simply by browsing web pages. People who believed that viruses were buggy software didn’t see browsing as dangerous because they weren’t actively clicking on anything to run it.

While all of the respondents expressed some knowledge of the importance of passwords, few went to extra efforts to make good passwords. Everyone seemed to know that, in general, passwords are important, but they couldn’t really say why. Respondents with the *graffiti* hacker model would sometimes put extra effort into their passwords so that mischievous hackers couldn’t mess up their accounts. And respondents who believed that hackers only target big fish thought that passwords could be an effective speed bump to prevent hackers from casually targeting them.

Respondents who believed in hackers as contractors to criminals uniformly believed that they were not targets of hackers and were therefore safe. However, the one piece of advice they did strongly believe in was to be careful in choosing which websites to do business with. Since these hackers targeted web businesses with lots of personal or financial information, it is important to only do business with websites that are trusted to be secure enough to resist these hackers.

3.4.5 Computer Maintenance

Finally, Advice 10-12 is all about computer maintenance. Advice #10 suggests that users make regular backups in case some of their data is lost or corrupted. This is good advice for both security and non-security concerns. Advice #11 states that it is important to keep the system patched with the latest updates to protect against known vulnerabilities that hackers and viruses can exploit. Advice #12 echoes the old maxim that the most secure machine is one that is turned off.

Different models had dramatically different suggestions as to which types of maintenance are important and which don’t need to be done. For example, mischievous viruses and graffiti hackers can cause data loss, so users with those models feel that backups are very important. But users who believe in more criminal viruses and hackers don’t feel that backups are necessary; hackers and viruses are trying to steal information, not to delete it.

Patching is one of the more important pieces of advice; hackers and viruses need vulnerabilities to exploit. Most of the respondents only experience patches through the automatic updates feature in their operating system or in their software. Respondents mostly associated the patching advice with hackers; respondents who felt that they would be a target of hackers also felt that patching was an important tool to stop hackers. Respondents who believed that viruses are buggy software feel that viruses also bring out more bugs in other software on the

computer; patching the other software makes it more difficult for viruses to cause problems.

3.5 Botnets and the Folk Models

This study was inspired by the recent rise of botnets as a strategy for malicious attackers. Understanding the folk models that home computer users employ in making security decisions helps us to understand why botnets are so successful. Modern botnet software seems designed to take advantage of gaps and security weaknesses in multiple folk models.

I begin by listed a number of stylized facts about botnets. These facts are not true about all botnets and botnet software, but these facts are true about many of the recent and large botnets.

1. *Botnets attack third parties.* When a botnet viruses compromises a machine, that machine only serves as a worker. That machine is not the end goal of the botnet. The owner of the botnet intends to use that machine (and many others) to cause problems for third parties.
2. *Botnets only want the Internet connection* Most botnets compromise computers because they are connected to the Internet. The only real resources that are used is the Internet connection; botnet software rarely takes up much space on the hard drive, rarely looks at any data on the hard drive, rarely occupies much memory, and usually don't use much CPU.
3. *Botnets don't directly harm the host computer.* Most botnet software, once installed, does not directly cause harm to the machine it is running on. It consumes resources, but often botnet software is configured to only use the resources at times they are otherwise unused (like running in the middle of the night). They might indirectly cause harm to the host machine; for example, the botnet may send out spam that ends up cluttering the host machine's inbox. Some botnets even install patches and software updates so that other botnets cannot also use the computer.
4. *Botnets spread automatically through vulnerabilities.* Botnets often spread through automated compromises. The will automatically scan parts of the internet for vulnerable, compromise any vulnerable computers it finds, and install copies of the botnet software on the compromised computers. This does not require any human intervention; neither the botnet owner nor the zombie owner nor the vulnerable computer owner need to be using their computer at the time.

	<i>Virus Models</i>				<i>Hacker Models</i>			
	<i>Viruses are Bad</i>	<i>Buggy Software</i>	<i>Mischief</i>	<i>Support Crime</i>	<i>Graffiti</i>	<i>Burglar</i>	<i>Big Fish</i>	<i>Contractor</i>
Botnets attack third parties	?	-	-	-	-	-	-	-
Botnets only want the Internet connection	-	-	-	-	?	-	-	-
Botnets don't harm the host computer	-	-	-	+	-	+	+	+
Botnets spread automatically	?	-	-	-	-	-	-	-

- + Makes sense It makes sense that malicious software / attackers would do this
- ? Related This statement is odd, but viruses or hackers might do something similar
- Unusual Malicious software / attackers that do this would be unusual

Table 3.7 How each folk model would probably react to the stylized facts about botnets

These stylized facts about botnets are not true for all botnets, but for many of the current, large, well-known, and well-studied botnets around now. I believe that botnet software effectively takes advantage of the limited and incomplete nature of the folk models of home computer users. Table 3.7 illustrates how each model does or does not incorporate the possibility of each of the stylized facts about botnets.

Botnets attack third parties None of the hacker models would predict that compromises would be used to attack third parties. Respondents who held both the *Big Fish* mental model and the *Contractor* mental model believe that, since hackers don't want anything on the computer, they would target other computers and leave the unwanted computer alone. Respondents with the *Burglar* model believe that they might be a target, but only because the hacker wants something that might be on their computer. They would believe that once the hacker either finds what they were looking for, or cannot find anything interesting, then the hacker would leave. Respondents with the *Graffiti* model believe that hacking and vandalizing the computer is the end goal; it would never cross their mind to then use that computer to attack third parties.

None of the respondents used their virus models to discuss potential third parties either. A couple of respondents with the *Viruses are Bad* model mentioned that once they got a virus, it might try to "spread." However, they had no idea how this spreading might happen. Spreading is a form of harm to third parties; however, it is not the coordinated and intentional harm that botnets cause. Respondents who employed the other three virus models never mentioned the possibility of spreading beyond their computers. They were mostly focused on what the virus would do to them, and not to how it might affect others. Also, since they

had an idea of how viruses spread, those ideas only involved spreading through webpages and email. They don't run a webpage on their computer, and no one acknowledged that a virus could use their email to send copies out.

Botnets only want the Internet connection No one in this study could conceive of a hacker or virus that only wanted the Internet connection of their computer. The three crime-based hacker models (*Burglar*, *Big Fish*, and *Contractor*) all believe that hackers are actively looking for something stored on the computer. All the respondents with these three models believed that their computer had (or might have) some specific and unique information that hackers wanted. Respondents with the *Graffiti* model believed that computers are a sort of canvas for digital mischief. I would guess that they might believe that botnet owners would only want the Internet connection; they believe there is nothing unique about their computer that makes hackers want to do digital graffiti on their computer.

None of the virus models would have anything to say about this fact. Respondents with the *Viruses are Bad* model and the *Buggy Software* models didn't attribute any intentionality to viruses. Respondents with the *Mischief* and *Support Crime* models believed viruses were created for a reason, but didn't seem to think about how using the computer to spread.

Botnets don't harm the host computer This is the one stylized fact on this list that any respondents explicitly mentioned. Respondents with the *Supports Crime* model believe that viruses might try to hide on the computer and not display any outward signs of their presence. Respondents who employ one of the other three virus models would find this strange; to them, viruses always create visible effects. To users with the *Mischief* model, these visible effects are the main point of the virus!

Additionally, the three folk models of hackers that relate to crime all include the idea that a 'break in' by hackers might not harm the computer. To these respondents, since hackers are just looking for information, they don't necessarily want to harm the computer. Respondents who use the *Graffiti* model would find compromises that don't harm the computer to be strange, as the main purpose of 'breaking into' computers is to vandalize them.

Botnets spread automatically The idea that botnets spread without human intervention would be strange to most of the respondents. Almost all of the respondents believed that hackers had to be sitting in front of some computer somewhere when they were "breaking into" computers. Indeed, two of the respondents even asked the interviewer how it was possible to use a computer without being in front of it.

Most of the respondents believed that viruses generally also required some form of human

intervention in order to spread. Viruses could be ‘caught’ by visiting webpages, or by downloading software, or by clicking on emails. But all of those required someone to do something with the computer. Only one subject explicitly mentioned that viruses can “just happen” (Jack). Respondents with the *Viruses are Bad* model knew that viruses could spread, but didn’t know how they might spread. These respondents might not be surprised to learn that viruses can spread without human intervention, but probably haven’t thought about it enough for that fact to be salient.

Summary Botnets are extremely cleverly designed software. They take advantage of home computer users by operating in a very different manor from the one conceived of by the respondents in this study. The only stylized fact listed above that a decent number of my respondents would recognize as a property of attacks is that botnets don’t cause harm to the host computer. And not everyone in the study would believe this; some respondents had a mental model where not harming the computer wouldn’t make sense.

This analysis illustrates why eliminating botnets is so difficult. Many home computer users probably have similar folk models to the ones possessed by the respondents in this study. If so, then many home computer users never really thought about a number of the properties of the attacks that botnets use to compromise home computers. Some of the properties even seem counter to the way that home computer users believe that hackers and viruses operate.

3.6 Limitations and Moving Forward

By conducting a series of 33 interviews, I was able to discover 8 different folk models that describe how home computer users view security threats against their computers. These models are divided into two major classes of threats: viruses and hackers. Respondents viewed the threats against their computer in a number of different ways, and the folk models are common descriptions of these threats. The respondents would make decisions using some sort of mental model in their head; I have tried to understand these mental models, and classified them into these 8 folk models.

In this study, I am able to describe in detail a number of common folk models of threats to home computer systems. Previous literature (Dourish et al., 2004; Gross and Rosson, 2007) was able to describe some basic beliefs held by non-technical users; I provide structure to these theories by understanding how home computer users group these into semi-coherent mental models in their mind. However, the models I discovered are not exhaustive; other home computer users might have other models. Indeed, the snowball sampling method

increases the chances that I will interview users with similar folk models. Additionally, this method makes it impossible for me to estimate how prevalent each model is. Such information might be useful in understanding nationwide vulnerability to botnets, but I leave such estimates to future work.

Understanding these folk models helps to explain why users strictly follow some security advice from computer security experts and ignore other advice. Respondents in this study had different beliefs about possible threats, and consequently chose different countermeasures based on those beliefs. Some countermeasures are important since they directly address threats in a way that the respondents understand. Other countermeasures seem unnecessary either because the respondent doesn't understand how the countermeasure provides security, or because they don't believe that there is a threat.

Understanding these folk models also helps to understand why botnets work so well on the Internet today. Botnets have a number of properties that would seem unusual to many of the respondents. Since respondents are not looking for threats like botnets, they often do not take the appropriate countermeasures that are necessary to prevent being compromised and becoming part of a botnet.

There is some hope. A number of respondents believed multiple types of viruses, or multiple types of hackers. When a respondent believed that there are multiple types, they tended to take all of the countermeasures that were appropriate for both types. This suggests that educating users about the multiple types of threats they truly face might help them to make better security decisions. However, the *Contractor* model illustrates one possible problem with this approach. Respondents with the *Contractor* model seemed to have integrated two different models of hackers into a single, new idea of hackers. They then used this model to reason that they were not likely to be a victim of hackers, and consequently take few precautions to protect themselves.

Chapter 4

Designing for Side Effects: Tagging on Delicious

Users of social computing websites both produce and consume the information found on the site. This creates a novel problem for web-based software applications: how can website designers induce users to produce information that is useful for others? I study this question by looking at how people use the social bookmarking website del.icio.us. I find that users in our sample use metadata reflecting who bookmarked a webpage to support information seeking rather than free-form keyword metadata (tags). I explain this finding by describing how del.icio.us provides a private incentive to contribute both bookmarks and tags, and makes these publicly available as a side effect of contribution. However, differences in the private incentive to contribute bookmarks and the incentive to contribute tags make bookmarks more useful to consumer than tags.

Websites like flickr.com, YouTube.com, and del.icio.us belong to a growing category of Internet applications broadly referred to as social computing sites. The information presented by these sites is generated by the users themselves, rather than by an agency officially tasked with that responsibility, like a publishing company or news distributor. Users of social computing websites play two roles with respect to the information the sites contain: they can act as *producers* (contributors of information) or *consumers* (seekers of information). Consider the social bookmarking website del.icio.us (<http://del.icio.us>), which provides the capability for users to bookmark web pages and associate user-generated metadata, or tags, with them (Marlow et al., 2006). Information consumers are able to discover new web pages using del.icio.us only because the actions of information producers caused the web pages to be stored by del.icio.us in the first place, through the action of saving the web pages as bookmarks. This aspect of social computing — that the community of information consumers benefits from the contributions of information producers — presents an interesting research question. How is it that producers, possibly through actions intended

to serve purely personal goals, come to generate information that is beneficial for the larger community?

An incentive is something that influences a person to choose one course of action over the alternatives. Often, incentives are thought of in terms of money: a bonus for meeting performance goals, or a subject payment in an experiment. However, incentives can be anything that induces expectations of a future positive outcome or benefit. For example, the capability of del.icio.us to store one's bookmarks in such a way that they are available from any computer connected to the Internet provides an incentive to use del.icio.us rather than a web browser's built-in bookmark feature. Incentives are important in social computing because they motivate or induce certain types of actions and not others. By identifying the pattern of incentives, we better understand how user behaviors leads to both individual outcomes and a community of users.

In this chapter, I focus on del.icio.us as a case study of the incentives in social computing. Many researchers have studied del.icio.us as the canonical example of a collaborative tagging system for information management (Golder and Huberman, 2006). In particular, I am interested in how the technical design of del.icio.us provides incentives to producers that influence the information they contribute, and how those incentives then indirectly influence how consumers use the site. I discovered that even when users contribute for only individual, non-social reasons, a social information system can succeed in producing a broadly-useful pool of shared information.

In the first study, Emilee Rader and I conducted a series of semi-structured interviews with twelve regular users of del.icio.us. We discovered that metadata reflecting the identity of the user who saved a web page, which is automatically associated with each bookmark when it is created, was more useful for consumers' information seeking than the user-generated tags. We also discovered that producers created public bookmarks and tags for their own private, and not social, reasons.

This finding was surprising; existing literature has suggested that most tags are applied for social reasons. (Golder and Huberman, 2006; Halpin et al., 2007; Sen et al., 2006) To better understand and validate this finding, we conducted two subsequent studies that empirically examined tagging patterns on del.icio.us and that tested these competing hypotheses about how users choose tags. For Study 2, we conduct a large-scale empirical analysis that uses a logistic regression on tag choice data to look for evidence to test three different hypotheses about tag choices. In Study 3, we use a computer model that assumes different tag choice strategies for simulated users and compares the resulting aggregate tagging patterns to empirical patterns on del.icio.us. In both of these studies, our findings confirm the qualitative results from Study 1.

Finally, I will bring in some theory from economics to characterize the incentives that are important in del.icio.us. I refer to this pattern of incentives as the *Side Effect Mechanism* and describe the major difficulty in implementing this mechanism: *incentive alignment*. del.icio.us provides a great illustration of this mechanism because one feature on del.icio.us (bookmarks) implements this well, and another feature (tags) implements this poorly. I use del.icio.us as a case study to illustrate that even when users contribute for only individual, non-social reasons, a social information system can succeed in producing a broadly-useful pool of shared information. Applying this theory from economics allows me to generalize these findings beyond the single social computing system studied, and provide concrete design recommendations for future social computing systems.

4.1 An Overview of del.icio.us

I begin with a description of the interface and functionality of del.icio.us, and some important definitions. Users of del.icio.us willing to create an anonymous user account are able to save web pages as bookmarks. When bookmarks are created, it is possible to associate tags with them. In del.icio.us, tags are restricted to a single word, and plurals or different spellings of the same word are treated as different tags. When a user creates a new bookmark, the interface displays *recommended tags* selected automatically by the system, *your tags* which are all tags chosen in the past by that user, and *popular tags* for that particular web page. Each bookmark has the following metadata automatically associated with it when it is created: the username of the information producer, the tags selected, and the date and time the bookmark was created. Users may also associate a “note”, or text string, with each bookmark they create. By default, bookmarks and tags in del.icio.us are public information. See Figure 4.1(a)¹ for an example of the del.icio.us interface for creating, or posting, bookmarks.

Information consumers browsing del.icio.us view subsets of bookmarks delimited by metadata such as a particular username, tag, or user-tag combination. For example, clicking the tag *library* in the list of popular tags on del.icio.us displays all web pages bookmarked by any del.icio.us user having the tag *library* associated with them. Each user account is a collection of bookmarks delimited by the the username of the person who created the bookmarks, as shown in Figure 4.1(b)¹. For example, this particular user posted the ASIS&T call for papers to del.icio.us one day ago using the tags “asist, conference, 2007, paper.” The metadata for a given web page can also be displayed including the usernames of all the users

¹Yahoo! updated the user interface of del.icio.us in August 2008. Since all of our data is from before this interface change, I only discuss the old interface to the system.

url

description

notes

tags

recommended tags
[information](#) [technology](#)

your network
[for:joshua](#)

popular tags
[associations](#) [library](#) [information](#) [IA](#) [informationscience](#) [techn](#)

(a) The old interface for posting bookmarks

del.icio.us / bierdoctor /

[your bookmarks](#) | [your network](#) | [subscriptions](#) | [links for you](#) | [post](#)

All your items (143)

« earlier | later » page 1 of 3

[2007 ASIS&T Annual Meeting Call for Papers](#) [edit / delete](#)
 to assist conference 2007 paper ... [saved by 12 other people](#) ... 1 day ago

[A Journey to the Dark Side - New York Times](#) [edit / delete](#)
 to forrwash beer ... on oct 19

[WGN Weather Weblog](#) [edit / delete](#)
 to weather chicago blog skilling ... [saved by 15 other people](#) ... on oct 18

[Index - Labradoodle Discussion Forum](#) [edit / delete](#)
 to dog labradoodle information forum ... on oct 18

[view.jpg \(JPEG Image, 1280x720 pixels\) - Scaled \(85%\)](#) [edit / delete](#)
 to photos webcam banff ... [saved by 2 other people](#) ... on oct 18

[Flickr: Photos tagged with labradoodle](#) [edit / delete](#)
 to dog photos labradoodle ... on oct 18

(b) One user’s collection of bookmarks

Figure 4.1 The user interface of del.icio.us

who bookmarked it, and all the tags ever associated with it. That specific web page has been bookmarked by 12 other people. All metadata items are also links that filter collections of bookmarks. The Library of Congress home page has been bookmarked in del.icio.us by 3060 different users, and tagged “library” by 1337².

If a registered user wishes to follow the latest bookmarks saved by a certain person, or having a certain tag, it is possible to “subscribe” to users, or tags, or a combination. New bookmarks satisfying these metadata criteria will then appear in the user’s account. For example, if user A subscribes to user B, every time B posts a new bookmark it will appear when user A clicks the “your network” link in her account (see Figure 4.1(b)). In addition, registered users can subscribe to all bookmarks associated with a given tag, or all bookmarks posted by a given user with a given tag, which can be accessed by clicking on the “subscriptions” link. Finally, the “links for you” link returns bookmarks that have been saved by one user specifically for another user, using a special tag format supported by del.icio.us.

4.2 Existing Knowledge

A number of researchers have studied del.icio.us and the phenomenon of social bookmarking and tagging. User-contributed metadata, also known as *tags*, provides a means for users to associate personally salient keywords or labels with content items (Golder and Huberman, 2006; Sen et al., 2006), enabling them to find the content later via information they are predisposed to recognize or recall (Lansdale, 1988). Tagging helps users “package” information

²As of April 17, 2008

for future information seeking and reuse (Markus, 2001). Tagging has not only been applied to personal information management; many *collaborative tagging* systems have appeared in recent years. Collaborative tagging systems such as del.icio.us and citeulike.org publicly expose individual users' associations between content items and tags, thereby providing visibility into words others have used to tag similar items. Grudin (2006) suggests that collaborative tagging can be a low-effort solution for shared or group information management, because it does not require that users try to conform to a controlled vocabulary or organization scheme. However, in other shared information management contexts, the effort required to “package” information is necessary for effective information reuse (Markus, 2001).

In a collaborative tagging system, users interested in viewing content tagged a certain way by others can browse the system by clicking on tags. Tags provide the “information scent” (Pirolli, 2005) that connects users with information; they are the infrastructure upon which information organization and finding takes place, allowing users to navigate by recognition rather than recalling terms by which to search (Teevan et al., 2004). This has interesting consequences when one considers the potential utility of tags for information management, finding, and re-finding. If a given tag is applied in an inconsistent manner among many users, more variability exists in the content items displayed when a user browses to a particular tag. For example, users tend to assign high-level tags like “technology” and personal tags like “to read”, as well as words like “apple” that can refer to a computer or a fruit, and “photos” or “pictures” which are synonyms.

Golder and Huberman (2006) and Halpin et al. (2007) found (for del.icio.us) that the frequency distribution of tags used on a given site tends to stabilize over time, with a definite most appropriate tag and a power-law distribution of tags. A number of researchers have tested designs to improve tagging systems like the one in del.icio.us (Rivadeneira et al., 2007; Sen et al., 2006; Storey et al., 2006; Xu et al., 2006) Ames and Naaman (2007) qualitatively studied motivations for tagging in Flickr. They found tags were used both for organization and to communicate, and were used both for selfish and for social purposes. However, they could not identify to what extent tags were social or selfish (as we do in Studies 2 and 3). Also, they did not describe how tags were used by information consumers, and could not link motivations for creating tags to the eventual use of tags by consumers.

By default, bookmarks and tags in del.icio.us are public information, meaning that any user may browse any other user's bookmarks and tags without logging in to the system. This makes the del.icio.us corpus a public good, meaning that the information in the corpus can be accessed simultaneously by everyone without ever being used up (Mas-Colell et al., 1995). One of the distinguishing features of public goods is that most of the time individuals

N=12 (8 Men, 4 Women)	Mean	Std. Dev.
Total Bookmarks	950	1030.77
Total Unique Tags	400	356.08
Months Using del.icio.us	15.90	8.73
New Bookmarks / Week	16.15	25.35
New Bookmarks / Day	2.31	3.62

Table 4.1 Description statistics about our respondents

are insufficiently motivated to contribute to public goods relative to what would be best for the community or society as a whole. The standard solution is to have the government provide the good, as is the case for the provision of national defense (the army). Voluntary provision of public goods by individuals is an open research problem (Andreoni, 2006). Chapter 2 reviews existing work that looks at ways to get around this problem.

4.3 Study 1: Interviews

We conducted twelve 1.5 hour semi-structured interviews with users of del.icio.us during the summer of 2006. All had used del.icio.us for multiple months, and had posted bookmarks to del.icio.us about once a week on average, for a number of months before the study took place. Five of the twelve users were masters students or recent graduates at a local university, three were PhD students, one was an undergraduate, and three were information technology professionals. A number of del.icio.us users responded to fliers posted around campus and to Internet postings on del.icio.us. Our respondents were selected from that group to represent a wide variety of usage patterns. For example, the total number of bookmarks that had been saved by a single respondent varied from a low of 60 to a high of approximately 3000. Table 4.1 provides descriptive statistics on respondents' use of del.icio.us. Our selection of these particular respondents for participation in the study reflects our desire to collect data on a range of possible user behaviors.

Our sample consisted of highly educated, tech-savvy students and professionals, meaning that our respondents were likely to be more sophisticated and self-aware in their use of technology than the average home computer user. As a result, our respondents were probably more likely to have attempted to optimize their personal usage of del.icio.us, to expend more effort when taking advantage of the available functionality, and to tolerate usability problems. In addition, respondents self-selected for participation when they responded to our advertisements, which indicates that their use of del.icio.us was salient and important enough to them that they were willing to volunteer to participate. We believe that

by requiring that respondents be regular users of del.icio.us for several months before the study took place, such bias was unavoidable. Our choice to use this criterion was motivated by a desire to interview people who could recall many past instances when they had used del.icio.us. Our sample therefore consists of users likely to have explored different features and uses, biasing our results toward a greater variety of activities than might be seen in a sample obtained based on different criteria.

The interviews were comprised of three phases. In the first phase, the interviewer asked general questions about respondents' use of del.icio.us: how often do you use it?, what do you bookmark?, how do you choose tags?, and similar high-level questions. The second phase consisted of ten search tasks. Respondents were sequentially presented with ten printouts of web pages found in del.icio.us, five bookmarked by the respondent and five bookmarked by others, and were to find them using only del.icio.us. They were instructed to think aloud during this task. Results from the search tasks will not be presented here, though the search tasks did prompt the users to make a number of interesting statements about bookmarking and tagging that are presented below, and were followed up on later in the interview. Finally, the interviewer looked through the respondent's bookmark history and asked questions designed to trigger retrospective accounts of past actions, such as "Tell me about that bookmark. What were you doing when you posted it? Tell me about the tags you chose." The interviewer also asked detailed questions about respondents' use of their subscriptions.

We recorded and transcribed the interviews, and then coded them using Atlas.ti . The analysis was conducted in a similar fashion to Miles and Huberman (1994). We began informal coding with a list of classes of behavior upon which to focus, and the code list developed as we proceeded. We identified the stated motivations of the respondents, their actions undertaken using del.icio.us, and the outcomes of those actions, including inferences about benefits they received from using del.icio.us. We created summary matrix displays showing which users exhibited similar motivations, subsequently undertook which actions, and received which benefits. Coding for these three things (among others) enabled us to locate and analyze possible instances where incentives had influenced behavior. It is important to note that this is not an exact science; unlike monetary incentives that can be explicitly identified and measured, identifying factors that induce users to behave in certain ways is an interpretive process. Nevertheless, and despite the varied usage patterns of our respondents, patterns emerged that present intriguing evidence for the role incentives play in del.icio.us. I describe these patterns below.

4.3.1 Results: Producer and Consumer Incentives

In this section we present data describing three categories of activities users engage in with del.icio.us: bookmarking, tagging, and information seeking. We first describe the motivations that our respondents, in their role as information producers, cite as the reasons they contribute to del.icio.us. Following a distinction made by the respondents, we divide these results into the two distinct producer actions: contributing bookmarks and contributing tags. Next, we discuss how these respondents seek information on del.icio.us, using both information from bookmarks and information from tags. This allows us to see how the user-contributed information is or is not used by consumers on del.icio.us.

4.3.1.1 Why Bookmark?

Respondents reported three motivations for bookmarking web pages:

- To keep track of useful or interesting web pages
- To access bookmarks from multiple computers
- To achieve recognition from other users of del.icio.us

The most frequently mentioned motivation for bookmarking web pages, reported by all respondents, was a desire to have ready access to information they found useful or interesting. In other words, our respondents created bookmarks so that they could easily go back to useful or interesting webpages in the future. As one respondent, Fred³, said: “Any web page that I see, I basically ask myself this question: Would I ever have a need to find this again? And if I do I just bookmark it.” Seven respondents valued the ability to access their bookmarks from multiple computers. Zoe liked del.icio.us “because I’m working on so many computers and so many different places, it’s just made me so much more efficient. It’s a lifesaver.” The action of bookmarking leads to the benefit of future access to web pages that contain information important enough to save. The websites that respondents bookmarked can be divided into a few categories:

- Topics of specific interest to the respondent. Alice bookmarked PhD programs she was interested in, Bob bookmarked library-related links, Charlie liked web pages on sustainability, Oscar looked for programming skills, and Marvin was into community informatics. There is a different list of topics for every respondent in our study.

³Respondents have been given pseudonyms. While some of the pseudonyms are the same, none of the respondents from Chapter 3 participated in this study.

- Pages the respondent hasn't finished reading. Bob described this well: "Umm, I get to that situation where I have eight different tabs open in Firefox and I don't really have the time to read them all. [I'll] get up and do something, and so I'll bookmark a bunch of them so that I will go back and in theory read them later." Half of these respondents expressed discontent over rarely actually returning to these web pages.
- Reference information or Internet tools. Nine respondents reported this type of bookmark. Examples include new search engines (Zoe), manuals for the Perl programming language (Oscar and Eve), and collaborative text editors (Eve).
- Novelty or funny web pages. Fred described these well: "something funny. Like a video of a monkey sniffing itself or something. [...] Or if something is just, oh wow cool, a new story or an amusing rant or a blog post, those get added [...] as well." Another example was when Victor bookmarked a web page "because I thought the title was so ridiculous."

In addition, eight out of twelve respondents were motivated to bookmark in order to share web pages with other people. They used a variety of actions for sharing: using del.icio.us' built-in tag convention for sending a bookmark to another user (the "for:username" tag); using a previously agreed-upon tag; through knowing that a particular person subscribes to their bookmarks; or just by explicitly instructing the person to go look for it. As Trent explained, he does this "in lieu of [having] sent an e-mail with a link in it." Five of those eight, however, reported that this motivation only rarely influenced them to bookmark. Two respondents bookmarked webpages they had created in del.icio.us, hoping that other del.icio.us users would find them. Also, one respondent mentioned that he believes some search engines index del.icio.us, and therefore bookmarks pages to increase their Google PageRank (Brin and Page, 1998).

Social recognition only functions as a motivation when the respondent is aware of other users' behavior, and in our sample this awareness varied widely. One respondent seemed to have no knowledge of others looking at his bookmarks. Seven respondents had directly told other people to look at their bookmarks, either indicating a specific bookmark, as in: "I couldn't remember the Dog Judo link but I wanted him to check it out, so I sent him a thing that said Go to Dog Judo on my del.icio.us" (Eve), or directing them to a certain tag. Six respondents were aware of other people who subscribed to their bookmarks; often this awareness came from conversations with friends. Isaac was aware of his friends' subscription because, "Like every so often [a friend] will say I noticed you bookmarked that or [another friend] will say that."

Half of our respondents mentioned a general awareness that the bookmarks they post are public information, unless they specify that they should be private. However, they reported that this awareness rarely affected their actions. For example: “I do make a conscious decision of whether or not I want it to be available for everybody but 98% of the time I don’t care” (Charlie); and, “Even though [my use of del.icio.us] is oriented primarily towards myself, the awareness that it is public never goes away totally” (Trent).

Bookmarks on del.icio.us are primarily created to enable *re-finding* by the bookmark creator. This is a personal, not social, reason for bookmarking. Some bookmarks are created for more social reasons, but the majority of bookmarks created by our respondents are done so without explicit social purpose.

4.3.1.2 Why Tag?

The most frequently mentioned motivation respondents reported for tagging was to *organize* their bookmarks and make it easier to find them if the need arose in the future. Respondents chose specific tags that they believed would help them organize their bookmark collection, using one or more heuristics to help choose good tags:

- Reuse tags he or she has applied before
- Create and adhere to mental rules or definitions for specific tags
- Choose terms he or she expects to search on

These are only heuristics, do not apply in all cases, and were not necessarily applied consistently. In addition, these data are self-reports, and other unconscious factors could also be a factor in the choice of tags. We have attempted to verify the reports by manually looking through the respondents’ tagging history and by eliciting multiple instances of the heuristics during the interviews. Respondents’ observable bookmarking behaviors indicate a fair amount of compliance with these heuristics.

To reuse tags, respondents placed priority on choosing tags they had used in the past. “I will not add a new tag until I have a group of things that I think it goes with,” said Zoe. Reusing old tags made bookmarks easier to find by minimizing the length of respondents’ tag lists, which most respondents reported searching visually when they wanted to find a bookmark. Victor described the problem his tag reuse solves: “One of my friends, his tag section goes way down below the fold [...] I’m like ‘How on earth do you sort through all these?’ And he said, ‘I don’t.’”

Respondents often reported that they had created mental rules or definitions for a number of their tags. For example, Peggy described some of her rules about tags related to blogs:

“So ‘blogs’ are usually other people’s blogs. ‘Blogging’ would be something that’s about usually research about blogging. And then if it’s something like Blogger for instance or LiveJournal then that would be a ‘bloggingtool’.” The creation of such rules is also observed in the creation of folders in which to store documents (Whittaker and Sidner, 1996). One advantage of tags over folders is that a single web page can be associated with multiple tags, reducing the effort involved with selecting one and only one location for the information. However, the more tags one has, the more overhead is involved with remembering the mental rules necessary for distinguishing among tags. People in general have a hard time being consistent within themselves with the tags they use (Golder and Huberman, 2006), and our respondents were no different. Trent described how he handles this problem: “I’ve been sloppy in the past about ‘collaborative’ and ‘collaboration,’ so this one got tagged as both. Just to make sure that I got coverage.” Eve’s tags showed the same characteristics: “So, apparently I’m using funny and humor interchangeably. And not reliably. So I should remember that when I’m looking for something funny I also label it humor. [...]” Respondents also had problems with singular/plural tags and with misspelled tags, all of which create the situation where multiple tags have the same logical meaning. Respondents with this problem speak of their tags as “dirty,” and the occasional act of fixing this as “cleaning” their tags.

Seven respondents reported choosing tags by trying to guess what terms they might search on in the future to find the bookmark. Eve described her thought process:

Interviewer: So, on the librarian video, how did you choose the tags that you have?

Eve: [...] If I were looking for this again [...] I’d be like “What was that video about the girl in the library with that guy?” But girl and guy is not very helpful, so library and video won.

In addition to the general heuristics for tag choice, respondents used tags to represent personally meaningful categories. Five respondents used tags to represent projects. Whenever they bookmarked a web page related to the project, one of the tags they applied to that web page was the project name. Marvin said this was “so I can just type in [the project name] and the things related to that project should show up if I did it right.” Ten of the twelve respondents used tags for purely personal purposes. This is similar to the “functional purpose” tags of (Golder and Huberman, 2006), and the personal tags of (Sen et al., 2006b). Rather than placing icons in a specific location on the desktop to serve as reminders, as seen in some personal information management studies (Barreau and Nardi, 1995; Bruce et al., 2004), respondents were using special tags as reminders within del.icio.us. The “toread” tag is one example, used by at least four of the respondents. Fred had a “wishlist” for items he

would like to purchase. Both Alice and Oscar used the tag “research” to refer to web pages they wanted to remember to return to, because they might be useful for their respective research projects. The meaning of all of these tags is highly subjective and personal, and are of limited usefulness to others because they can only be correctly interpreted and understood by someone who knows the context.

Four out of twelve respondents mentioned consciously choosing tags for non-personal reasons. These respondents are each trying to build a collection of links on a specific topic that would benefit the larger community of del.icio.us users. Alice said, “I tag everything on [topic of interest] I can find. I was so frustrated when I started working with this stuff that I just couldn’t find information about it. [...] There aren’t many places for it so I have probably collected one of the larger lists out there.” For Alice, wanting to be known as an expert on the topic of the collection seemed to be an additional motivation beyond personal organization and community benefit.

4.3.1.3 Information Seeking

Information consumers are the beneficiaries of others’ bookmarking activities when they view other users’ bookmarks on del.icio.us. Our respondents primarily consumed information from del.icio.us by browsing to find new information. Browsing has been defined as, “a kind of searching in which the initial search criteria or goals are only partially defined or known in advance” (Chang and Rice, 1993). We found that respondents’ goals for discovering new information fell into three categories:

- Novelty information: “something entertaining” (Alice)
- Topical information: web pages relevant to specific topics
- Social information: updates on friends’ interests and activities via following their bookmarks

The most common action for novelty discovery, undertaken by seven respondents, was subscribing to someone they knew personally. Eve said: “I check out [my friend, he] always goes to really interesting places,” and, Bob reported that he subscribed to a friend because he, “like[s] to pick his brain for cool stuff.” Respondents’ topical discovery, or seeking web pages that contain information on specific topics, was either a one-time seeking behavior, or it was due to a continuing interest in a particular topic. For one-time seeking, the most common action was to click on the “Saved by X other people” link. Seven respondents reported doing this for topical discovery, and two for novelty discovery. Bob described his reasoning:

“If I’ve got something bookmarked myself and it says ‘Saved by X other people,’ then it’s more intriguing to me if there are very few people who have saved it. Because that means I belong to this elite group of people who actually find this stuff interesting. [...] And then maybe I’ll take a look at what else they’ve bookmarked because if they are interested in something that I’m interested in maybe they’ve got other stuff that I’d be interested in.”

For continuing topical discovery, respondents again reported subscribing to someone they knew in real life, or occasionally someone who is famous, or was found using “Saved by X other people.” As Zoe discovered, “certain people tend to tag the same things I’m interested in.” Five respondents reported looking on del.icio.us for other users with similar interests, and then subscribing to those users.

Finally, social discovery was used to keep tabs on friends. Peggy reported, “It’s just interesting to see what it is they’re up to. So like my friend Matt who’s not in the area anymore, I think I get a sense of what it is he’s doing.” Mostly, this occurred through subscriptions. Charlie gives a good example:

“One of my friends [...] just got, you know, just got a job in San Francisco. I believe she went out to interview and then all of a sudden there’s like 50 links to apartment search in San Francisco, and a few days later she tells me oh I got the job in San Francisco and I was like I know.”

Figure 4.2 illustrates the number of respondents who reported using each of the six information discovery strategies we found. Interestingly, most respondents reported taking advantage of user metadata for information seeking and discovery, but not tag metadata. Tags were rarely mentioned in the context of novelty and social discovery. Because assessments of how interesting or entertaining a web page might be is a subjective judgment, it is reasonable to expect that it would be difficult to find a tag which would capture this assessment accurately. Only one respondent reported browsing tags (like “funny”), or searching del.icio.us, for novelty purposes. In social discovery, it is the user and not the topic that is of interest. Since bookmarks are always automatically associated with the user who bookmarked them, tags are not needed. Finally, a number of respondents struggled with using tags for topical discovery; three explicitly mentioned trying to do so and failing find the information they wanted. Only one respondent (Trent) subscribed to any tags, and he was careful to block users (using del.icio.us’s built-in blocking mechanism) who post too many bookmarks for which “none of [them] fit my definition.” Only four respondents mentioned browsing del.icio.us for one-time topical discovery by looking at specific tags, and they reported doing it only rarely.

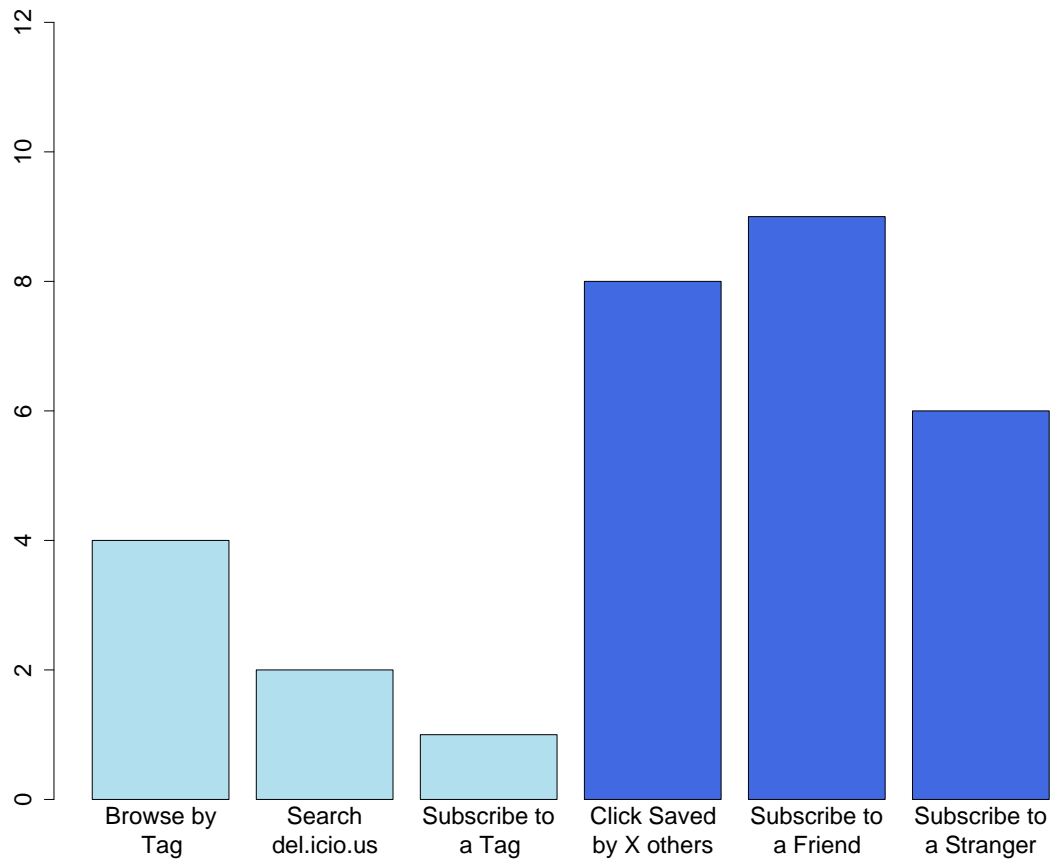


Figure 4.2 This chart depicts the number of respondents who used these strategies for information discovery. The light bars are tag-related strategies, and the dark bars are username-related strategies

4.3.2 Incentives on del.icio.us

One of the difficulties in understanding why people contribute to social computing systems is that each person is not acting alone. One person's decision is potentially affected by everyone else on the site, and everyone else's decision is in turn potentially affected by that one person's decision. For example, some respondents reported occasionally bookmarking webpages because they know others will see them and comment on them. This strategic interplay between contribution decisions can make understanding the reasons behind these decisions difficult.

Del.icio.us is a good system to study because many bookmarks on del.icio.us are created for non-strategic reasons. The reason that our respondents reported for creating the majority of their bookmarks is a personal reason; they choose to bookmark webpages because they think that they might go back and visit the website sometime in the future. And tags are cho-

sen mainly to support this potential re-finding activity; our respondents' tags are chosen to help themselves organize their own collection of bookmarks. Users create these bookmarks for personal reasons, and would continue to do so even if no one else ever saw them.

However, del.icio.us makes all bookmarks public by default. Anyone can go and view the bookmarks from anyone else. This public benefit is a *side effect* of the personal reason that users contribute bookmarks. The main purpose of creating bookmarks isn't to benefit others, but this benefit happens because of the default-to-public nature of the site.

This suggests a general mechanism, or design pattern, for inducing user contributions. Social computing systems can provide a strictly private benefit from contributions — potentially by emulating something that users already do on their computers like store bookmarks — and then make these contributions public by default. Making things publicly available as a side effect of normal, personal use can encourage contribution to a social computing system.

4.3.2.1 Clutter, Tags, and Language Use

However, del.icio.us also illustrates how simply making data public is not enough. As seen in Figure 4.2 and discussed above, most of our respondents found it very difficult to use tags for information discovery. There are a number of problems with tags related to language use that complicate their use for discovering new information.

First, in describing the 'vocabulary problem,' Furnas et al. (1987) state that random pairs of people choose the same label for an object on average about 20% of the time. This robust tendency results from humans' imprecise and flexible use of language in conversational settings, where meaning is determined by the surrounding context and complex communication processes. This suggests that if two random users having different knowledge and situational contexts create a tag for the same web page, there is an 80% chance on average that they will NOT choose the same tag. Similarly, if an information consumer attempts to imagine what tags might be applied to the information she is looking for, chances are low that she will end up using the same words to represent the same concepts in the same way as others have used them.

Second are synonymy (multiple words that can be used interchangeably in the same context) and plurals/tenses of the same word. Eight respondents noticed this problem just within their own bookmarks! Trent mentioned being "sloppy in the past about 'collaborative' and 'collaboration'" and Charlie "at one point had 'recipe' and 'recipes'." Del.icio.us uses exact word matches, so the difference here is important. Third is polysemy, or one word that has many meanings or senses. One example of this is the word "python" which can represent a snake, a programming language, or a comedy troupe from Great Britain. Expertise can also contribute to polysemy: people with different levels of expertise may end up using the

same word to represent different concepts — their internal rules for what that tag represents are different. Fred subscribed to the ‘security’ tag, but as a computer security expert he found many of the bookmarks with that tag too basic for him. However, a novice user would likely find the web pages he bookmarked under ‘security’ too advanced. Finally, users of del.icio.us tend to create tags that have personal or figurative meanings only they have the requisite contextual knowledge to understand, like the “research” or tag several respondents used to refer to their respective projects.

These characteristics of language use produce a great deal of variability in the collections of bookmarks delimited by specific tags. Many tags have a problem with *clutter*; each new bookmark associated with a given tag makes it more difficult to find things with that tag, in much the same way that each new book or paper on a desk makes it more difficult to find anything on that desk. This is particularly important for ongoing discovery through subscriptions, where the rate of incoming bookmarks depends on how prolific others are; too many bookmarks make it difficult to keep with the influx of web pages. Fred reported experiencing this when he subscribed to the tag “security”: “I’d go through it [the list of bookmarks], but I end up skipping a lot.” Ultimately, he stopped using that subscription altogether. We did not find evidence that our respondents attempted to match their internal rules for tag use with the ways in which others applied the same tags, which exacerbates the clutter problem with all the language use issues just discussed.

4.3.2.2 Incentive Alignment

The bookmarks associated with a given user don’t normally have a major clutter problem; most users have a reasonable number of bookmarks and can find things when needed. Above, we describe how a number of respondents look through other users’ collections of bookmarks to discover new information. Rarely were clutter problems mentioned. If we look back at how our respondents choose bookmarks, we get some insight as to why a user’s bookmarks have relatively little problem with clutter.

Bookmarks represent an implicit endorsement of a web page; when an information producer chooses to post a web page to del.icio.us, she is in effect making a statement that this particular web page is valuable enough to save for later. Since our respondents wanted to re-find webpages using del.icio.us, they reported that they were careful to only bookmark pages worth re-finding. This limits the clutter associated with each username; if they were to bookmark many more websites, re-finding would be more difficult. Our respondents had an incentive to limit the clutter associated with their usernames: they wanted to be able to easily re-find webpages in the future.

For bookmarks, the incentives faced by our respondents in their role as information

producer were *aligned* with the information discovery goals our respondents reported in their role as consumers. As producers, our respondents were motivated to create collections of bookmarks associated with their usernames with limited clutter. This limited clutter is an important property for other users who, acting as information consumers, look at these collections for discovery. Fred even indicated some level of awareness of this incentive alignment: “[I have] mostly just a general sense of the network, of people. If someone tracks the stuff I post enough, then I assume they care about the eigenvalues of the things that I like. And so I figure if I like it, they’ll like it.”

When contributing tags, on the other hand, our respondents did not report attempts to limit system-wide clutter associated with those tags. None of our respondents checked to see how other users on del.icio.us had used a tag. Our respondents indicated that rarely they would use a tag in the same way they had seen others use it on del.icio.us, meaning that the only time that other users’ tag choices did influence the choices of our respondents, it actually lead to increased clutter. Our respondents simply chose tags primarily for personal organization. There is little incentive to expend the effort required to combat the naturally occurring system-wide problems of clutter resulting from overuse of tags, the vocabulary problem, polysemy and synonymy. The incentives for contributing tags are not aligned with the goals of our respondents as information consumers because they do not induce users to limit the system-wide clutter associated with tags.

This discussion illustrates an important lesson for designing social computing systems. When users are motivated to contribute for personal reasons and those contributions are shared as a *side effect*, it is important to ensure that incentives for contribution are *aligned* with the goal of the users as information consumers. This was accomplished for bookmarks and usernames on del.icio.us by only allowing users to associate bookmarks with their own username. This ensured that the collection of bookmarks associated with that username is relatively clutter-free and useful, because making that collection useful is exactly why those bookmarks were contributed in the first place. del.icio.us failed to do this with tags because anyone can associate a bookmark with any tag. Without further incentives to limit clutter on those tags, users naturally overused tags and contributed to a number of problems associated with normal language use. This resulted in multiple orders of magnitude more bookmarks associated with many tags than are found associated with most users. For most tags in the system, the collection of bookmarks associated with those tags is very cluttered and difficult to use for information discovery; indeed none of our respondents use tags for this goal.

4.3.2.3 Locus of Control

One way to think about the difference between bookmarks and tags is that the collection of bookmarks created by a given user is under a single locus of control (the user), while no central authority oversees the way words are used as tags. When choosing tags for a new bookmark, none of our respondents expressed concern about increasing the number of bookmarks system-wide already associated with a tag. This difference is similar to the architectural difference between the Internet and Cable TV (MacKie-Mason et al., 1996). Content-aware architectures, like cable television, have the capacity for editorial control over the content they deliver. Cable TV providers have an incentive to make it easier for information consumers to find something interesting to watch: the larger the audience, the more money they make. They do so by limiting their offerings to those that appeal to a majority of their audience. Content-blind architectures like the Internet have no single locus of control, and therefore there is no agent for whom an incentive to limit content offerings might exist. Anybody can create a web page or a podcast and make it available on the Internet, but there is no guarantee that it will ever be noticed. Bookmarks in del.icio.us are content-aware: because an information producer has control over his collection of bookmarks, he also has an incentive to place limits on that collection so that it meets his needs. No such incentive exists for tags, which are content-blind.

4.3.2.4 Bookmarks are Endorsements

However, there is yet one more wrinkle to be considered. Bookmarks represent an implicit endorsement of a web page; when an information producer chooses to post a web page to del.icio.us, she is in effect making a statement that this particular web page is valuable enough to save for later. There is no similar assumption for tags. del.icio.us provides information on which tags and bookmarks are currently popular based on aggregate usage statistics; however, this method of creating a valuation for tags fails for topical discovery because popular tags are often the noisiest. Because tagging doesn't necessarily represent an endorsement, aggregating tagging decisions doesn't have the same effect as aggregating bookmarking decisions.

4.3.3 Conclusion

Del.icio.us users control their bookmark history, and receive benefits from using del.icio.us to store their bookmarks. This has effectively aligned the incentives for information producers with the needs of information consumers on del.icio.us. However, no one controls the uses of a tag, and the natural tendencies of language use preclude the applications of

tags in a way that is beneficial for information seeking. We have shown how the design and architecture of del.icio.us, a social computing website, can influence the choices that users make.

As social computing becomes more pervasive, it is important to understand how such systems can induce users to contribute. Future systems that rely on user-contributed content will need to provide users with an incentive to contribute, and might be able to learn from del.icio.us. Users can be induced to contribute by providing a private usefulness to contributions and having the social nature of the contributed information be an intentional *side effect*. However, systems that use this design pattern need to ensure that the incentives for information production are *aligned* with the needs and goals of users who consume the information. Misaligned incentives can lead to users abandoning certain features (like the use of tags for information discovery), and possibly to halting the use of the system altogether.

4.4 Study 2: Large-scale Data

In Study 1, we claim that the incentives to contribute bookmarks are aligned with the goals of del.icio.us users as they consume information, but the incentives for creating tags are not aligned. However, this claim was based on data from only 12 users. This is not enough data to make the claim that enough users of del.icio.us are making bookmarking and tagging decisions in this way to actually affect the available information on del.icio.us for consumption.

Additionally, our finding that users choose tags for personal reasons was very surprising. Much existing literature claims that users of del.icio.us are following a social process for choosing tags (Golder and Huberman, 2006; Halpin et al., 2007; Sen et al., 2007), and that this social process results in a large-scale bottom-up classification system. This claim is so widespread that it has spawned new terminology (“social tagging” for the process, “folksonomy” for the classification system) complete with a Wikipedia page⁴.

To address these two issues, we undertook two studies that examined a very large dataset of tagging choices on del.icio.us. Our objective in this research was to look for a pattern of evidence indicating that either a social process or a personal process is affecting tag choices. Golder and Huberman (2006) speculate that users might imitate each others’ tag choices; in other words, tag choices might be influenced by tags that had been previously applied to the same web page by other users. However, it is reasonable to assume that there might

⁴http://en.wikipedia.org/wiki/Social_tagging. The same page describes both social tagging and folksonomy.

be other sources of influence on users' tag choices having to do with personal information management goals. In Study 1, we found that users of del.icio.us chose tags for organizing and re-finding their own bookmarks according to mental rules and definitions they had established, striving for consistency within their own personal "controlled vocabulary". Finally, users might desire to expend as little effort as possible when choosing tags, and simply select tags suggested in the del.icio.us posting interface when they create a new bookmark.

We conducted a multiple-method investigation that teases apart these competing explanations. In Study 2, we performed a logistic regression analysis of a large sample from del.icio.us, in which we evaluated the influence of several predictors on users' tag choices. In Study 3 we developed a computer model in which we assume a number of different tag choice strategies one at a time, and compare aggregate patterns in model results against the same measures in the data from del.icio.us⁵. We found evidence that users' tag choices are not a result of imitation of others' tags; instead, they follow an individual, idiosyncratic pattern. This suggests that personal information management goals, rather than social processes, have a greater influence on tag choices in del.icio.us. This finding supports our claims in Study 1 and helps to give credence to the idea of incentive alignment (and mis-alignment) on del.icio.us.

4.4.1 Background and Hypotheses

Furnas et al. (1983) began the study of tagging with their paper on the vocabulary problem, in which they reported that when two random people create a label for the same document, they choose identical words less than 20% of the time. Tagging has been studied in a mobile context (Ames and Naaman, 2007) and for photos (Marlow et al., 2006). It has been applied to personal information management (Cutrell et al., 2006; Tang et al., 2008), and in a corporate environment (Millen et al., 2007, 2006). Researchers want to better understand tagging patterns (Halpin et al., 2007; Golder and Huberman, 2006) and make recommendations for how users might produce better tags (Farooq et al., 2007; Sen et al., 2007, 2006). We focus here in particular on the findings of Golder and Huberman (2006) and Sen et al. (2006), because they motivated and guided our investigation most directly.

Golder and Huberman (2006) argue that users' tag choices are not random; instead, consensus emerges for which tags best represent a given web page. They show that web pages bookmarked in del.icio.us demonstrate a stable frequency distribution following a power-law pattern in which the same few tags are chosen by many users, while most other

⁵Our database schema and code for our computer model and analyses may be downloaded from <http://bierdoctor.com/papers/cscw08>

tags are selected by only one or two people. Golder and Huberman hypothesize that when a user bookmarks a web page in del.icio.us, their tag choices are influenced by tags that had been previously applied to that web page by others (p. 206). They illustrate this imitation hypothesis through a mathematical construct: the stochastic urn of Polya (Page, 2006). For users to behave according to Polya's Urn, they must randomly select tags from the tag distribution for a given webpage. This means that if the tag "library" makes up 13.5% of all tags applied to the Library of Congress home page, users must somehow choose "library" 13.5 times out of 100. However, the del.icio.us interface does not provide users with sufficient information about the tag frequency distribution to behave according Polya's Urn. Rather, users are presented a nonrandom, biased sample in the posting interface: the *recommended* and *popular* tags (see Figure 4.1(a)). Golder and Huberman suggest that imitation occurs via these tags presented in the interface, but do not address the distinction between biased sampling methods and the unbiased random draws of Polya's Urn. In our computer model we implemented several different forms of sampling from the tag distribution, allowing us to clarify the difference.

Sen et al. (2006) manually assigned tags from MovieLens, a movie recommendation system, to one of three *classes*: factual, subjective, or personal. Through a field experiment manipulating the information displayed in the MovieLens tagging interface, they found that users imitated tag classes when tagging movies, and concluded that "community influence plays an important role in vocabulary" (p186). In our analysis we focus on a more fine-grained dependent variable, individual users' exact tag choices, rather than subjectively assigning tags to classes. This allows us to test competing hypotheses from the literature using a larger dataset containing tag choices made over a longer period of time, albeit without the experimental control afforded by the ability to make changes to the interface. The difference in the unit of analysis (tag classes versus exact tag choices) allows us to potentially reach different conclusions. As we will show below, our findings contradict those of Sen et al.; we found little support for the hypothesis that users imitate one another's exact tag choices.

4.4.2 Methods

Over two weeks in January 2007, we downloaded the entire bookmark and tag history for approximately 20,000 different web pages in del.icio.us. The web pages were chosen by periodically sampling the "recently posted" and "popular" del.icio.us pages. We randomly chose 30 web pages from our sample that had been bookmarked by at least 100 users. Then, in June 2007 we downloaded the complete public bookmark and tag histories for all of the

approximately 12,000 users who had ever bookmarked any of these 30 web pages. In other words, our dataset contains the complete tag histories for 30 web pages bookmarked in del.icio.us, as well as tag histories for all users who ever bookmarked any of those 30 web pages as of June 2007.

4.4.2.1 Model and Data Setup

We used a logistic mixed model regression Agresti (2007) to evaluate the influence of three hypotheses on users' tag choices:

1. *Imitation*: Users imitate tags that previous users have applied to a web page
2. *Organizing*: Users re-use tags that they have applied to other web pages
3. *Recommended*: Users choose tags that are suggested via the del.icio.us posting interface⁶

If the *imitation* hypothesis has a strong influence, tags previously associated with a given web page by other users will be correlated with tag choices. We can assume a social process is at work, and a socially constructed vocabulary is truly emerging. If tagging behavior is determined more by *Organizing* than by *Imitation*, then we expect tags a user has applied before to other web pages to be correlated with tag choices. Finally, if the *Recommended* hypothesis is true, users' tag choices are influenced by tags suggested in the del.icio.us posting interface.

We model the dependent variable — the choice of a single tag — as a yes/no choice. We lack evidence indicating what tags users have or have not viewed prior to choosing tags; therefore we cannot directly observe when a user makes a 'no' choice. We make a simplifying assumption that each user made a yes/no choice about all tags that had been applied to the particular web page at the time our data were collected. Basically, we assumed that each tag on our list of observed tags was at least implicitly considered by all subjects and a yes/no choice was made about it. We attempt to estimate the probability of saying "yes" to each tag as a function of three different factors included in the model as predictors. First, if *Imitation* is shown to have strong influence on a particular tag choice by a particular user, then the probability that a tag is chosen should be higher if the word has been used previously as a tag. This would be reflected in the model as a large, positive coefficient for the "used.onsite" predictor. Second, if *Organizing* is shown to have strong influence,

⁶It is difficult to concretely specify this hypothesis because del.icio.us does not reveal its method for selecting tags to suggest, and the method may have changed multiple times.

the probability that a word is chosen should be higher if the word has been previously used by that user as a tag for a different web page. This would be reflected by a large, positive coefficient for “used.byuser”. For the *Recommended* hypothesis, the algorithm for selecting tags to display in the posting interface is not publicly known; however, some experimentation with del.icio.us has led us to believe that a tag is much more likely to be recommended if it has both been applied previously to that web page and used previously by the user. Therefore, we approximated the *Recommended* hypothesis by including an interaction term that is 1 when both used.onsite = 1 and used.byuser = 1.⁷

The model also includes several controls for other factors that may influence the probability of choosing a tag. Some tags seem to “fit” the web page better than others (i.e., *library* for the Library of Congress home page), and are more frequently applied. Since the data include repeated measures for each tag, it is important to control for per-tag variability using fixed effects. This is represented in the model by “tag_dummys”.⁸ Also, some users tend to assign more tags to their bookmarks than others; we controlled for this within-user variability using random effects. Finally, we account for temporality in the used.onsite variable.⁹ This variable is 0 for early bookmarks and 1 for later bookmarks, switching after a tag is used. We believe used.onsite controls for any autocorrelation that might result from the previous use of certain tags, and therefore a time series model is not necessary. The model is set up as follows:

$$\text{tag_chosen} = f(\text{used.onsite}, \text{used.byuser}, \text{interaction}, \\ \text{tag_dummys}, \text{random_effect}(\text{user}))$$

⁷We do not claim that this interaction term fully identifies the recommended hypothesis. We only claim that *if* the Recommended hypothesis describes a large number of tag choices, *then* the coefficient on the interaction term will be positive and large. There are many other effects than can also cause a large positive coefficient here, including a non-linear interaction between the first two hypotheses. However, this is OK. The Recommended hypothesis predicts a large coefficient; therefore if we do not observe a large coefficient, that casts doubt on the Recommended hypothesis. Since we aren’t sure how del.icio.us chooses tags to recommend, this is at best a weak test and we hesitate to draw strong conclusions from it.

⁸Each row of the data table is associated with a user considering a single tag. The dummy variable for ‘library’ is 1 for the rows when the user is choosing whether to apply the tag ‘library’ and 0 when the user is considering other tags.

⁹This does not necessarily account for temporality in the used.onsitevariable; it is possible that users only took notice when a tag had been used many times before. We ran a number of alternative models with a ‘number of previous applications’ variable (and some polynomial combinations of this variable to account for non-linearity) and the results were substantially similar to those we report below. We chose to report the simple binary used.onsitevariable because it is easier to interpret and because there seemed to be little effect from additional applications of a tag.

4.4.3 Logistic Regression Results

We estimated the model using maximum likelihood estimation, separately for each of the 30 web pages in the study. This allowed us to compare web pages and determine whether an overall pattern exists.¹⁰ We summarize the estimates for the model coefficients in Table 4.2.

In logistic regression, the dependent variable is dichotomous, meaning it takes only two possible values. The model is used to estimate the probability of the dependent variable taking on the value 1, given a set of predictors. This probability is represented in the form of *odds*. For example, a probability of 50% can be represented as 1:1 odds, and 2:1 odds translates to a 66% probability. The coefficients for the predictors in a logistic regression model are the natural logarithm of odds *ratios*, or the ratio of the odds of one possible outcome divided by the odds of another outcome. In the model, our predictors are dummy variables that can be either 1 or 0. Therefore, the coefficient represents the natural logarithm of the ratio between the odds that a tag will be chosen when the value of the predictor is 1 to the odds when the predictor is 0. If the coefficient is positive, then the probability of a tag being chosen is greater when the value of the predictor is 1 (or true). If the coefficient is negative, the probability of a tag being chosen is greater when the predictor is 0 (or false).

To interpret the results in Table 4.2, first focus on the columns for used.onsite, used.byuser, and Interaction. The values in these columns are the coefficient estimates for predictors representing our three hypotheses. The size of the coefficient and whether it is positive or negative indicates whether that predictor increases or decreases the probability of a given tag being chosen, and how strong the effect is. From these coefficients, we can calculate the predicted probability of being chosen for each tag applied to a given web page. An example of fitted probabilities for “101 Cookbooks” is presented in Table 4.3. The remaining columns of Table 4.2 present the results of statistical tests to evaluate the validity of our model.

The three hypotheses are operationalized as follows:

1. *Imitation*: When bookmarking a given web page, users choose tags previously associated with that web page by other users ($\text{Used.onSite} > 0$)
2. *Organizing*: When bookmarking a given web page, users choose tags they had applied before to other web pages ($\text{Used.byUser} > 0$)
3. *Recommended*: When bookmarking a given web page, users choose tags suggested in the del.icio.us posting interface, operationalized in our model as tags that had

¹⁰Combining the data for all 30 web pages into one large dataset proved computationally infeasible.

¹¹These two web pages in our sample had more users bookmark them than listed (1427 and 1137 respectively) but we truncated the dataset for computational reasons.

Title	Users	Used.onSite	Used.byUser	Interaction	$G_m(df)$	R_L^2	P	λ_p
A List Apart: Alternative Style	395	-0.1665	3.764 ***	-0.5780 *	6019 33 ***	0.5218	0.9824	0.2322 ***
London Underground History	369	-0.3365 *	3.226 ***	-0.0501	6638 34 ***	0.4923	0.9778	0.1049 ***
Haiku	161	-0.7128 *	2.368 ***	0.8593 *	2100 20 ***	0.5081	0.9494	0.2086 ***
Spread Firefox	214	-1.0990 ***	2.825 ***	0.5331 *	2277 24 ***	0.4360	0.9799	0.1293 **
PayPalSucks.com	121	-0.4083	3.140 ***	-0.1475	1146 17 ***	0.4174	0.9557	0.0760
OS X Maintenance	282	-0.5596 **	3.106 ***	-0.1510	3686 28 ***	0.4648	0.9744	0.1935 ***
The Library of Congress	552	-0.4079 ***	3.740 ***	-0.3113 *	7882 39 ***	0.4455	0.9921	0.0986 ***
GDI+ FAQ main index	114	-0.1986	3.528 ***	-0.6113	1299 21 ***	0.4602	0.9485	0.1974 ***
MetaGer	174	-0.4910 *	4.776 ***	-1.3510 ***	1318 20 ***	0.3952	0.9736	0.1367 **
eHomeUpgrade	270	-0.1153	3.712 ***	-0.5184 *	3495 35 ***	0.4207	0.9797	0.0809 *
Getting started with SSH	938	-0.0064	3.289 ***	-0.3577 *	18337 43 ***	0.5645	0.9846	0.1625 ***
err.the_blog	456	0.4622 .	3.578 ***	-0.4908 .	7622 31 ***	0.5496	0.9755	0.2847 ***
Beer Advocate - Respect Beer.	489	0.0400	3.222 ***	-0.2351	6899 27 ***	0.5357	0.9846	0.2392 ***
Old Computers	258	-0.2279	4.055 ***	-0.6511 **	3770 28 ***	0.4777	0.9785	0.1637 ***
DotNetNuke	714	-0.1742	3.659 ***	-0.6486 ***	12376 55 ***	0.4761	0.9878	0.0950 ***
BibDesk	303	-0.4937 **	3.859 ***	-0.2865	5941 35 ***	0.5116	0.9800	0.2163 ***
Tiny Icon Factory	819	0.0009	2.916 ***	0.4367 **	15902 58 ***	0.5041	0.9865	0.1337 ***
Mint: A Fresh Look at Your Site	560	-0.1202	3.570 ***	-0.3691 *	10730 46 ***	0.4701	0.9869	0.0430 *
Telegraph newspaper online	447	-0.4688 **	4.350 ***	-0.7939 ***	5221 23 ***	0.5094	0.9890	0.2109 ***
Glimpses? The Uncanny Valley	166	0.1536	2.995 ***	0.0883	2300 36 ***	0.3701	0.9668	0.0364
DVDStyle	157	-0.7305 *	2.656 ***	0.4941	2469 20 ***	0.4974	0.9532	0.2153 ***
digg labs / swarm	499	-0.4685 ***	2.876 ***	0.5907 ***	9877 55 ***	0.4768	0.9867	0.1044 ***
Flickr: The HDR Pool	596	-0.3258 .	3.208 ***	-0.2578	9210 29 ***	0.5472	0.9833	0.2459 ***
Sxip Identity	496	-0.2473 .	3.958 ***	-0.8236 ***	8318 39 ***	0.4833	0.9870	0.1158 ***
Many Eyes	466	0.3220 *	3.032 ***	-0.0818	9276 54 ***	0.4729	0.9820	0.1421 ***
Obscure Sound - Indie Music Blog	116	-0.4727	2.899 ***	0.0480	1136 15 ***	0.5008	0.9488	0.3354 ***
JotSpot Wiki (dojomanual)	218	0.0855	3.82 ***	-1.1510 **	3150 29 ***	0.5009	0.9533	0.2314 ***
BasKet Note Pads	124	-0.7212 **	3.211 ***	-0.3224	2183 24 ***	0.4612	0.9584	0.1339 ***
101 Cookbooks ¹¹	1000	0.1086	4.297 ***	-1.1060 ***	18231 43 ***	0.6028	0.9932	0.2595 ***
Snipplr - Code 2.0 ¹¹	850	0.4304 ***	3.536 ***	-0.1440	20329 83 ***	0.4934	0.9888	0.1199 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 4.2 Logistic Regression Results. Coefficients for tag_dummies and per user random effects are omitted here, but can be found in Appendix C.

used.onsite	used.byuser	food	cooking	recipes	blog
no	no	0.2034	0.2285	0.2183	0.0340
yes	no	0.2216	0.2482	0.2374	0.0377
no	yes	0.9494	0.9561	0.9535	0.7209
yes	yes	0.8737	0.8892	0.8833	0.4879

Table 4.3 Fitted Probabilities for the top 4 tags on *101 Cookbooks*

previously been both applied the web page and used by the user on other web pages (Interaction > 0)

A Wald test¹² can be done on each parameter estimate, similar to the standard t-test used in Ordinary Least Squares (OLS) regression. It compares the Null hypothesis that the true value of the parameter is 0 with the alternative hypothesis that the parameter is not 0. The stars in Table 4.2 show the statistical significance of these Wald tests.¹³

The *Imitation* hypothesis was supported for one web page, ManyEyes, which has a positive (though small) parameter estimate significant at the 5% level. Although the Wald test for the used.onsite predictor is significant for 13 sites, 12 have a negative parameter estimate, which does not support the *Imitation* hypothesis. From this we reject Hypothesis 1. However, the *Organizing* hypothesis is supported for all 30 web pages at the 0.1% level. The parameter estimates are quite high, indicating a strong effect. From this pattern, we conclude that Hypothesis 2 is supported. Finally, the *Recommended* hypothesis is supported for 4 of the 30 web pages at the 5% level (the other 12 significant estimates are negative). However, because we are uncertain how well we have approximated the recommendation algorithm in del.icio.us, we hesitate to draw conclusions about this hypothesis.

To illustrate the pattern of our results, Table 4.3 shows the model-estimated probabilities of choosing the 4 most frequently used tags (as of June 2007) on the *101 Cookbooks* web page for an average user. When a tag has been used previously by a user, our analysis shows a much greater probability of it being chosen again than if the user had not used it before. This pattern is consistent across all 30 webpages.

Model Fit and Diagnostics We conducted two different types of goodness-of-fit tests to ensure that these results actually represent what is in the data. The first test is analogous to the standard goodness-of-fit test for OLS regression. In OLS regression, the *F* statistic

¹²The likelihood ratio test is more conservative, but requires more time to compute; this can be problematic for very large samples (like ours). Using the Wald statistic can increase the standard error when the estimated coefficient is large, leading to failure to reject the null hypothesis (Type II error) (Menard, 2002).

¹³Multi-collinearity can produce large standard errors, making it impossible to get statistically significant estimates. We frequently rejected the null, indicating that collinearity is not a problem (Judge et al., 1985).

is a statistical test that the model actually fits the data. Technically, it is a hypothesis test that the specified model fits the data better than the simplest possible model – the mean of the data. For logistic regression, the G_m statistic is analogous to the F statistic. It compares the specified model to the mode of the data, which is the simplest explanatory statistic for a binary variable. The G_m test is statistically significant at the 0.1% level for all 30 models. The OLS R^2 statistic represents how much of the variability in the data the model is able to explain. It is a substantive, rather than statistical test of significance. The R_L^2 statistic is the logistic equivalent of R^2 (Menard, 2002), and represents the percentage of the likelihood explained by the model.¹⁴ For our models, R_L^2 indicates that the models explain about 50% of the likelihood on average. This indicates that there is definite room for improvement in understanding why users choose certain tags, but that our predictors account for a nontrivial portion of the likelihood.

The second test we conducted concerns the predictive efficiency of the model. With a binary independent variable, we can use the model to “predict” our dependent variable. To do this, we calculate the estimated probability (as we did in Table 4.3) and predict that a user will choose that tag if this probability is greater than 50%. The P column in Table 4.2 shows the percentage of tag choices that our model predicts correctly. For every web page, our model is over 94% accurate. However, this number can be misleading, as always predicting that the user will choose no tags can achieve above 90% correct for many web pages. To measure how much better our model predicts tag choices, we calculated the λ_p statistic Menard (2002). This statistic represents the “proportional reduction in errors” — how many fewer errors does our model make than expected? This statistic ranges from 1 when all errors have been eliminated, to 0 when we make the same number of errors as a simple predict-the-mode model, to potentially negative if we make more errors than expected. In general, our model allows us to predict tag choices approximately 10 to 20 percent better than a simple predictor, and never worse. This improvement is statistically significant at the 0.1% level for all but 6 web pages (and 4 of those 6 are significant at the 5% level).

4.4.3.1 Interpretation

The results in Table 4.2 have a clear pattern; of our three explanatory variables, the strongest influence is users’ previous tag choices. The coefficients on `used.byuser` consistently indicate a much larger influence than that of `used.onsite` or the interaction term. While user variability and individual tag ‘fit’ (represented by control variables in the model) play an important role in the choice of tags, the data indicate that users’ previous tag choices are also

¹⁴Menard (2002) points out that the standard R^2 statistic can be calculated for logistic regressions, but is biased. For this reason, we do not report it here.

important. This analysis also casts doubt on the *Imitating* and *Recommended* hypothesis, as operationalized in our model. We were only able to detect influence of these predictors in 1 and 4 web pages, respectively, and in these instances the influence was small. If there is a social process at work promoting a socially constructed vocabulary, we doubt that it takes the form of direct imitation. We are less sure about the effect of recommended or popular tags because we do not have a compelling measure of this explanatory variable.

4.4.4 Summary of Study 2

We believe ours is the first quantitative study of how users of del.icio.us choose tags to compare competing hypotheses from the literature. Our logistic regression showed that users' past tag choices had a large influence on future tag choices, while the fact that a tag had been used before on a web page had very little influence. Using logistic regression allows us to control for sources of variability that the cosine similarity measures used by Sen et al. (2006) do not. Also, our emphasis on exact tag choice rather than tag class means we are able to consider how the processes shaping the tag vocabulary on del.icio.us might affect its utility as a tool for personal and shared information management. del.icio.us users do not navigate by tag classes; specific words and the multiple meanings associated with them are important for finding and re-finding. It might also be that tagging is just different on del.icio.us and MovieLens. del.icio.us has a strong information management component (storing and organizing bookmarks), while it is less clear for what purpose tags might be chosen or used on MovieLens.

4.5 Study 3: Computational Modeling

The logistic regression analysis described above allows us to detect patterns in tag choices a posteriori; as such we are only able to speculate about what processes may have caused those patterns to occur. To address this weakness, we developed a computer model to evaluate competing explanations for the aggregate pattern of tags that appears on del.icio.us. We call these competing explanations *tag choice strategies*. In addition, the analysis described in Study 2 lumped together several forms of what might be considered "imitation" strategies into one explanatory variable. Computer modeling allows us to specify different forms imitation might take, and control what strategy is used to choose tags. It would be nice to instruct collections of real people to use one or more of the strategies suggested by the literature; we could then determine whether those tag choices resulted in tagging patterns similar to those found on del.icio.us. However, this technique would be prohibitively costly.

Computer modeling allows us to explore the effects of different strategies, and compare them with the real-world data (Nan et al., 2005). Such models cannot tell us which strategy or strategies real users of del.icio.us used; they can only tell us which strategies result in patterns of tags that are different from those observed in our large sample downloaded from del.icio.us, henceforth called the *real world* data. In other words, this technique cannot confirm which strategy is prevalent on del.icio.us, but it can be used to rule out possible explanations.

4.5.1 Measures

Axtell et al. (1996) described two types of measures for validating computer models against each other or against a real world dataset. *Distributional equivalence* is achieved when the distributions of results being compared are statistically indistinguishable; *numerical identity* exists when samples from different sources are shown to produce results that are numerically equivalent. We selected two measures to compare the tag choice strategies in our computer model against the real world data, one to test for distributional equivalence, and the other to test for numerical identity.

Baseline for Distributional Equivalence To establish a baseline measure against which to evaluate the distributional equivalence of tag choice strategies implemented in our model, we identified the theoretical distribution that most closely matched the tag frequency distribution in our del.icio.us sample. We fit the data from each web page to seven different discrete probability distribution families (discrete powerlaw, negative binomial, binomial, discrete lognormal, discrete exponential, poisson, and geometric), estimating parameters with maximum likelihood estimation, to discover which distribution fit “best” (a statistical determination (Clauset et al., 2007)). We then used a non-nested Kolmogorov-Smirnov (KS) test to conduct pairwise comparisons between these distributions. The KS test is a common goodness-of-fit test to determine how well a set of data points fits a particular theoretical distribution. We are using it here to fit our data to distribution types other than normal.

The discrete powerlaw distribution fit the empirically observed (real world) tag distributions better than the other seven distributions we tested. The fitted distribution had an average exponent α of 1.92 ± 0.40 . This is a low exponent for a powerlaw distribution, and indicates that the “long tail” of tags is very long and heavy. This low exponent also has another important implication. Newman (2005) explains that powerlaw distributions with an exponent less than 2 have an infinite (or undefined) mean. Therefore, estimates of a “mean” or average tag are undefined, and any inferential statistics based on the mean of the

tag distribution cannot be used.

Baseline for Numerical Identity To measure the extent of the vocabulary problem (Furnas et al., 1983), we calculated the average inter-user agreement (IUA) for a sample of 200 users from each of the 30 web pages in our sample from del.icio.us described above; this measure became our baseline for establishing numerical identity. On average, users who bookmarked these web pages chose the same tag only $14\% \pm 5\%$ of the time. IUA as a measure is sufficiently different than using the goodness of fit to a powerlaw distribution of tags as a measure because it yields different results in this study, and is therefore a complementary measure for characterizing a set of tag choices.

4.5.2 Modeling Tag Choices

We modeled 120 web pages for each of five tag choice strategies we implemented, described in detail below. Each modeled web page was paired with one of 30 real web pages used in Study One, and the number of users for each web page modeled was chosen to match the real web page. In essence, we are simulating what would happen if the same set of users bookmarked the real web page, but chose their tags according to one of our five hypothesized strategies (and bookmarked it in a random order). To simulate a user choosing tags for a web page, two choices have to be made. First, the computer model chooses how many tags that user will apply to the web page. Second, the model chooses which specific tags will be applied. These parameters are selected by the model for each web page based on the distribution of parameters we found on del.icio.us.

The tags from the matched real web page are ordered from most-frequently used to least-frequently used, with ties broken randomly; each tag is then mapped onto a number according to its rank in the frequency distribution. When the random-number generator produces a 1, this is mapped to the most-frequently-used tag, 2 onto the second most frequently used tag, and so on. Any numbers larger than the number of tags on the matched web page are left as numbers. For each user, the specific tags they choose depends on which tag choice strategy is being modeled. The only difference between these strategies is in specific tag choice; all other decisions (number of users, number of tags per user, etc.) are identical. We implemented five different tag selection strategies in our computer model:

Zipf: Zipf's law states that word frequency in most written works follows a powerlaw distribution. Therefore, del.icio.us users might naturally choose their words from this distribution (Newman, 2005). This could potentially account for Golder and Huberman's observation that the stable pattern in the tag frequency distribution for

web pages bookmarked on del.icio.us is evident even for less common tags not popular enough to be recommended in the del.icio.us posting interface (p. 206) (Golder and Huberman, 2006). The model chooses random numbers from the base powerlaw distribution until it has the required number of unique numbers. These numbers are then mapped onto tags as described above.

Organizing: Users might favor tags that they had used previously. This strategy was described in Study 1. Simulated users have a 50% chance of choosing tags according to Zipf's law, and a 50% chance of choosing tags they had used before. When choosing tags they had used before, the model computes the overlap (set intersection) between tags the user had ever used and tags that were ever applied to the matched web page. It then randomly chooses among the tags in this overlap set. If that is not enough tags, then additional tags are chosen randomly from the base powerlaw distribution.

Imitation-Urn: Imitation of other users' tag choices might be achieved using a path-dependent process, as described by Golder and Huberman (2006) in the Polya's Urn example. For users to imitate previous users' tag choices, it is necessary for those previous users to exist; the first few users who bookmark a web page have no one to imitate. To handle this, the first 20 simulated users draw as described above for *Zipf* and serve as 'seeders.' All users after the first 20 choose a tag from the current empirical distribution of tags for the simulated web page. This means that if there are two tags, 'A' and 'B', and 'A' has been used twice previously and 'B' only once, then tag 'A' is chosen with probability $\frac{2}{3}$ and tag 'B' is chosen with probability $\frac{1}{3}$. However, to ensure growth of the vocabulary beyond that used by the initial 20 seeders, each tag choice has a 10% probability of choosing a new, previously unused tag. This probability was chosen to match the average empirically observed probability from the del.icio.us data. The average web page in our original sample from del.icio.us has a new tag probability of $10.5\% \pm 8.3\%$.

Imitation-Popular: Users might prefer to click on the tags that are suggested in the del.icio.us posting interface. This was also hypothesized by Golder and Huberman (2006) to be a plausible form of imitation, via biased sampling. Suggested tags in the del.icio.us posting interface come in two forms: *recommended* and *popular*. del.icio.us has not publicized their algorithm for choosing which tags to display in the interface; however, we implemented a simple approximation in our model. We proposed that the tagging system could simply recommend the N most popular tags for that web page. Then users could randomly choose among those N tags. The model first creates 20 'seeders' in the same way it did for the *Imitation-Urn* strategy. All of

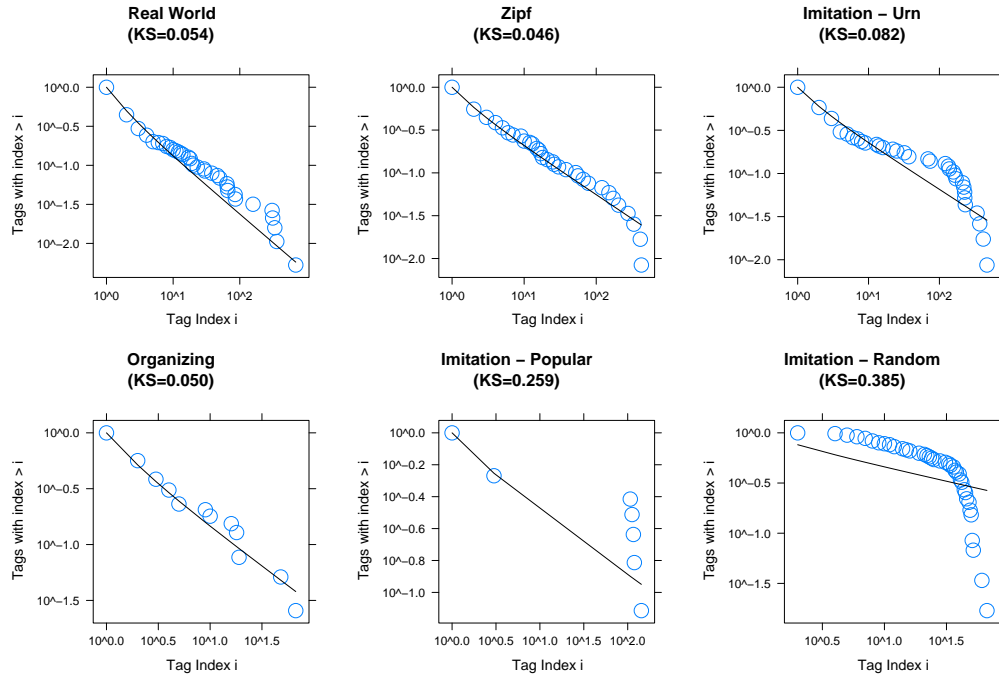


Figure 4.3 Tag frequency distributions for the real world data and strategies implemented in the computer model on a log-log scale.

the remaining users choose randomly among the $N = 5$ most popular tags at that point. If they need to apply more than 5 tags, then the remaining tags are chosen randomly from the base powerlaw distribution.

Imitation-Random: As a counterpoint to the flavors of imitation described above, we tested one final strategy. Rather than choosing randomly from the 5 most popular tags as in the *Imitation-Popular* strategy, users choose uniformly from among all tags previously used to that point (after the first 20 users have chosen tags according to Zipf’s law).

4.5.3 Computer Model Results

One of the benefits of computer modeling was that the development process forced us to be very explicit about what information users would need to follow a hypothesized strategy. Golder and Huberman (2006) suggested that the powerlaw distribution of tags for a given web page could arise from path-dependent choices. When trying to replicate these decisions for our simulation, we found that this only works if a user chooses tags from the empirical distribution *at the time of decision*. This is a very high information requirement for users; they must know the exact proportions of existing tags to choose appropriately. del.icio.us

Table 4.4 Measures of distributional equivalence and numerical identity.

	<i>Mean KS</i>	<i>St.dev. KS</i>	<i>Mean IUA</i>	<i>St.dev. IUA</i>
Real World	0.069	0.026	0.144	0.057
Zipf	0.080 *	0.011 ++	0.374 ***	0.074 +++
Organizing	0.084 ***	0.029	0.182 ***	0.052
Im-Urn	0.139 ***	0.067 +++	0.184 ***	0.056
Im-Popular	0.223 ***	0.149 +++	0.317 ***	0.088 +++
Im-Random	0.386 ***	0.063 +++	0.070 ***	0.042 +++

Wilcoxon test signif. codes: ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05

Levene test signif. codes: ‘+++’ 0.001 ‘++’ 0.01 ‘+’ 0.05

does not present this information in its interface; however we assume this knowledge for our simulations with the *Imitation-Urn* strategy so that it models a truly path-dependent process.

4.5.4 Distributional Equivalence Measure

For each simulated web page, we fit the tag distribution produced by the model to a discrete powerlaw distribution using maximum likelihood. We then conducted a Kolmogorov-Smirnov (KS) goodness-of-fit test to see how well the simulated distribution fit a powerlaw. A KS statistic ranges from 0 to 1; 0 means that the distribution is identical to a powerlaw, and higher numbers indicate greater deviation from a powerlaw (0.22 is a bad fit). The second and third columns (KS) of Table 4.4 show the mean and standard deviation of the KS statistic for each strategy. We used a Wilcoxon matched-pairs rank-sum test with Bonferroni correction to compare the set of KS statistics for each tag choice strategy (one for each simulated web page) against the set of KS statistics for the real world web pages, and found all comparisons to be statistically significantly different than the null (powerlaw) distribution. This is likely due to our large sample size for both the strategies modeled and our real-world sample. Therefore, it is more instructive in this case to consider practical, rather than statistical significance when interpreting the results of our analyses. In fact, in light of our large sample sizes we can interpret the significance of these results to mean that we can be confident the pattern of results we observed is unlikely to have occurred by chance. We can then focus on the actual differences observed, which for some strategies were large and for others were very small.

The mean KS statistic for the *Imitation-Popular* and *Imitation-Random* strategies indicate that data generated in this way do not fit a powerlaw very well. Figure 4.3 illustrates the tag distribution (on a log-log plot) for all five strategies on one simulated web page, along with the paired real world distribution. The straight line on each plot represents the

Density Plot of KS-test Statistic for Computer Model and Real World Power Law Fits

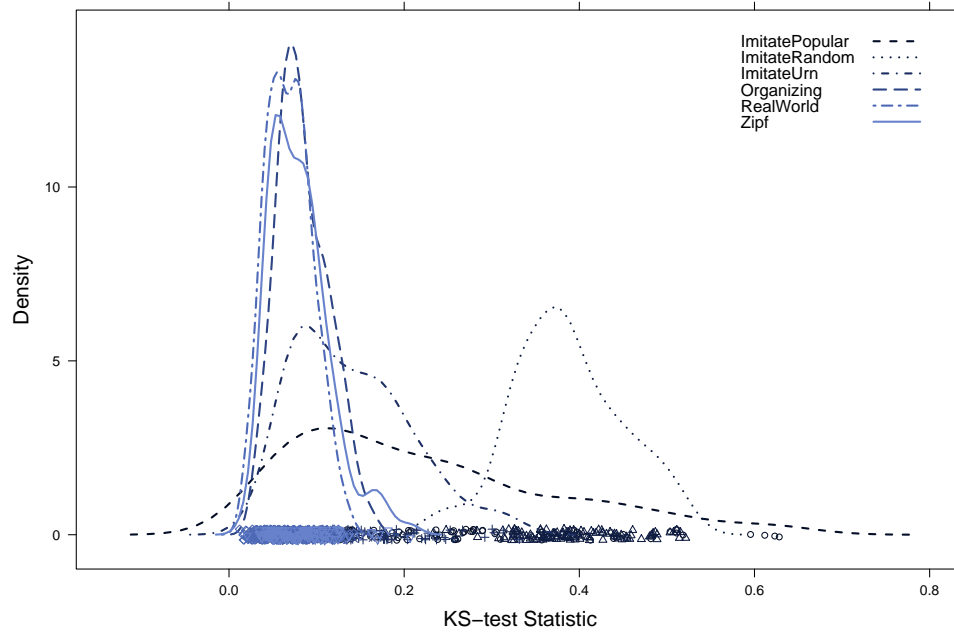


Figure 4.4 Shows the frequency distribution shape of KS statistics for real world data and modeled tag choice strategies.

theoretical powerlaw distribution that best fit the data. The non-powerlaw nature of the three *Imitation*-* strategies is noticeable compared to the nearly linear plots for the *Organizing* and *Zipf* strategies, as well as the *Real World* data. However, the distributions based on the *Zipf* and *Organizing* strategies fit as well as the real data from del.icio.us.

The mean KS statistic for the *Imitation-Urn* strategy is closer to that of the *Real World*, *Zipf*, and *Organizing* tag distributions than the other two *Imitation* strategies' distributions; however, the standard deviation of the *Imitation-Urn* strategy was significantly different from that of the *Real World* data using the Levene homogeneity of variance test. Figure 4.4 shows the density plot of KS statistics for each strategy; the narrow distributions clustered to the left are visibly different from the wide distributions produced by the *Imitation* strategies. We therefore conclude from our distributional equivalence measure that we can rule out the three *Imitation*-* strategies as plausible processes that might give rise to the distributional pattern we saw in our sample from del.icio.us.

Numerical Identity Measure We calculated the average inter-user agreement between simulated users of each modeled web page, for each strategy. Table 4.4 also provides the inter-user agreement means and standard deviations for each tag choice strategy, and for the real-world data from del.icio.us. IUA ranges from 0 to 1 and represents how often two

random users chose the same tag; higher numbers mean greater agreement. We again found that Wilcoxon matched-pairs rank-sum tests with Bonferroni correction were significant for pairwise comparisons between the *Real World* data and all tag choice strategies. Inter-user agreement was much higher for the *Zipf* and *Imitate-Popular* strategies than observed in the sample data from del.icio.us. The *Organizing* and *Imitation-Urn* strategies are negligibly different in terms of practical significance, as they are well within one standard deviation of the mean of the *Real World* data. They are also the only tag choice strategies for which the Levene test for homogeneity of variance was not significant when compared with the *Real World* data. From our numerical identity measure we can therefore rule out the following strategies as plausible processes that might produce inter-user agreement values we saw in our sample from del.icio.us: *Zipf*, *Imitation-Popular*, and *Imitation-Random*.

4.5.5 Interpretation

Based on the two measures described above, we can make the following determinations about the plausibility of each tag choice strategy producing data like that in our sample downloaded from del.icio.us:

1. *Zipf* — rule out based on numerical identity
2. *Organizing* — cannot rule out
3. *Imitation-Urn* — rule out based on distributional equivalence
4. *Imitation-Popular* — rule out based on both numerical identity and distributional equivalence
5. *Imitation-Random* — rule out based on both numerical identity and distributional equivalence

4.5.6 Summary of Study 3

We were able to rule out all tag choice strategies implemented in our computer model, except for the *Organizing* strategy. Our computer model allows us to go beyond the behavioral trace data and assume different strategies for choosing tags. We used this model to look at the large-scale tag choice patterns these strategies produce, rather than identifying patterns and speculating on what might have caused them, as Golder and Huberman (2006) reported in their paper. Our results corroborate and extend those from study 2, and indicate that the most plausible hypothesis among those we tested is that most tag selections in del.icio.us are governed by individual, idiosyncratic processes rather than a form of direct imitation.

4.5.6.1 Limitations and Future Work

It is important to note that through this research we are only able to **rule out** competing hypotheses. Data downloaded from del.icio.us are evidence which may be used to detect potential tagging strategies, but we cannot make assumptions about what information users may have seen and acted upon, or infer what a given user was thinking when making tag choices. Therefore, we are not able to say with absolute certainty that users choose tags according to their own personal organization scheme; nor can we determine whether popular tags are chosen more often due to imitation, because they are topically relevant, or for some as-yet unknown reason. The major weakness of the methods we have used that we cannot make any claims about users' perceptions, goals, or motivations that might shed more light on tagging strategies. In reality, the same tag choice strategy might not be used by all users, or even apply to all the tag choices of an individual user.

We also lack empirical evidence regarding the usefulness of tags for organizing, finding and re-finding personal and shared information. These limitations leave ample opportunity for future work in this area, including field studies of tagging behavior, measurement of the effectiveness of tags for information management, and experiments which will allow us to infer causal relationships between factors affecting tag choices and characteristics of the resulting tag distributions. The research described in this chapter leads us to focus our efforts on more rigorous investigations of the predictions of the *Organizing* hypothesis when tags are used for personal and shared information management.

4.6 Summary and Implications for Design

We report the results of three empirical studies of the social bookmarking website del.icio.us. In Study 1, we interviewed users of del.icio.us and found that they create bookmarks for personal organization reasons. This personal reason induces users to contribute information to del.icio.us, which then makes the information publicly available as a beneficial *side effect* of contribution. Other users of del.icio.us are then able to browse and discover this information. However, user metadata (information about who bookmarked a webpage) is much more useful to del.icio.us users than tag metadata because the collection of webpages associated with a user is fairly uncluttered and focused on a topic, but the collection of webpages associated with a tag has “too much noise” and suffers from many natural problems arising from human language use. We explain this difference by bringing in the concept of *incentive alignment* from economics; users have an incentive to produce uncluttered bookmarks, but there is no corresponding incentive for preventing clutter in tags.

In Studies 2 and 3, we focus on detecting patterns in tag choices on del.icio.us. We used two different methods, logistic regression performed on sample data collected from del.icio.us and computer modeling of tag choice strategies, to examine competing hypotheses describing processes that might produce observed tag choice patterns. Our logistic regression showed that users' past tag choices had a large influence on future tag choices, while the fact that a tag had been used before on a web page had very little influence. In addition, we were able to rule out all tag choice strategies implemented in our computer model, except for the *Organizing* strategy, by comparing simulated data with data from del.icio.us. In other words, our results indicate that the most plausible hypothesis among those we tested is that tag selection in del.icio.us is primarily governed by individual, idiosyncratic processes rather than a form of direct imitation. These results contradict both the hypotheses presented in Golder and Huberman (2006) and the results of Sen et al. (2006), and suggest that the potential for emergence of a socially-constructed vocabulary on del.icio.us due to tag imitation is unlikely.

In Study 1, we described an incentive for users to contribute bookmarks and tags to del.icio.us: personal organization of webpages that the user might want to return to. This incentive is personal, not social, despite the fact that these contributions are made public as a side effect. In Studies 2 and 3, we found evidence that this incentive does influence the large majority of contribution decisions on del.icio.us. We believe that most users contribute information to del.icio.us for personal reasons. Since most bookmarks and tags on del.icio.us are contributed for this reason, we can look at how users consume information on del.icio.us, and can connect their consumption behaviors to this incentive for production. In doing so, we find that this incentive leads to user metadata that meets the consumption needs and goals of del.icio.us users. But this incentive is insufficient to induce contributions of tags that are useful when consuming information on del.icio.us.

Chapter 5

Using a Minimum Threshold to Motivate Contributions

5.1 Introduction

Five of the 10 most visited websites are social computing systems¹, making Internet-scale social computing systems some of the fastest growing websites right now. Social computing systems collect, aggregate, and share user-contributed content, and therefore depend on the contributions of users to function properly. Not all social computing systems succeed in eliciting contributions; while Wikipedia has over 2.5 million articles and over 9 million registered users², its rival Citizendium (which runs the same MediaWiki software) only has around 20,000 articles and approximately 8,000 registered users³.

Human users are intelligent beings and cannot be programmed to behave; system designers need to provide incentives to encourage users to contribute. Users contribute primarily information to social computing systems, but contributing information is costly. Contributing requires making the effort to go to the website and either typing in information or clicking on information. Contributing also requires time to enter in the information, which has an *opportunity cost*: time that could have been spent on other things. To overcome this cost, social computing systems need to provide incentives that motivate users to voluntarily choose to spend their time and effort contributing.

There are many different ways to motivate users to contribute to social computing systems. For example, del.icio.us, a social bookmarking website, motivates contributions by providing easy online access to bookmarks and allowing users to organize their own

¹http://www.alexa.com/site/ds/top_sites, retrieved on March 14, 2009. The five social computing systems are YouTube(3), MySpace(6), Wikipedia(7), Facebook(8), and Blogger(9).

²<http://en.wikipedia.org/wiki/Special:Statistics>, retrieved on March 14, 2009

³<http://en.citizendium.org/wiki/Special:Statistics>, retrieved on March 14, 2009.

bookmarks (Wash and Rader, 2007). Facebook, the popular social networking system, motivates newcomer contributions of photographs by allowing other users to comment and by providing numerous examples of their friends contributions (Burke et al., 2009). Authors on Wikipedia are encouraged to contribute by having an automated robot suggest appropriate pages that need work (Cosley et al., 2007). Users on GlassDoor, a site for salary comparison data, must contribute information about their own salary to gain access to the aggregated salary data of others⁴.

In this paper we analyze a generalization of this last mechanism for encouraging contributions: a minimum threshold mechanism. This mechanism is technically simple: users must contribute a minimum amount of information to the system in order to receive access to the information from other users. While the mechanism is technically simple, how users will react is not. We identify circumstances under which setting a minimum contribution threshold for participation in a social information system will lead to an increase in breadth of contribution (by causing a larger proportion of users to make a non-trivial contribution), and cause the total quantity of contributions to increase.

In this chapter, Jeff MacKie-Mason and I develop a mathematical model of user behavior and use it to predict how users of a social computing system will alter their contributions when the minimum threshold mechanism is put in place. This type of model allows us to better understand strategic interactions between users; how much Alice is willing to contribute depends on what and how much is contributed by others, and their contributions in turn depend on hers. For example, Ephrati et al. (1994) used mathematical modeling to design a meeting scheduling system that users cannot manipulate. This method also allows us to explicitly generalize our results to a whole class of systems rather than studying only one specific system, and therefore provide constructive design suggestions for many similar systems.

Implementation and testing are also important, but we limit our current contribution to a theoretical analysis. Behavioral modeling provides a principled foundation for design that extends across different settings. The predicted user behavior, because it involves large numbers of users who have heterogeneous preferences and whose behavior depends on the strategic choices of others, is sufficiently complex that it warrants rigorously derived, testable predictions.

⁴<http://www.glassdoor.com/about/learn.htm>, retrieved on March 16, 2009

5.1.1 Background on public goods

User contributions to a social media system can be seen as contributions, in the form of information, to a single shared information pool. All users of the system have access to this pool. This shared information pool has the properties of a public good (Samuelson, 1954). In particular, the pool is *non-rivalrous* since using the information pool does not materially reduce the value of the pool to other people. To use a familiar example, once National Public Radio broadcasts a program, consumption by one listener does not crowd out consumption by other listeners. For information, nonrivalry is generally true because the incremental costs of (digital) reproduction and distribution are approximately zero, and thus multiple instances of the information can be “consumed” without “using it up”.

When public goods are created through voluntary contributions, they generally have the problem of *underprovision*: users prefer to “free ride” and use the public good without contributing, relying on other people to do the hard work of creating it (Samuelson, 1954). Of course, if everyone prefers to free ride, then the information pool will not get established in the first place. We see the free rider problem in social media: Adar and Huberman (2000) found that almost 70% of users of a popular peer-to-peer system contribute nothing at all. While Wikipedia has over 9 million registered users, only 166,066 (less than 2%) have contributed effort in the last 30 days⁵. Of those who contribute to Wikipedia, 50% do not return after their first day of contribution (Jian and MacKie-Mason, 2008).

Shared information pools in social media are also commonly *non-exclusive*; the information in the system is available to anyone anytime. The information contained on Wikipedia, del.icio.us, and Twitter is available for free to anyone with a web browser and a network connection. However, non-exclusivity is a design choice; social media systems could technically exclude users from accessing the public information pool. This potential for exclusivity opens up new opportunities for creating incentive mechanisms. The system can threaten to exclude users who do not meet specific criteria, and if crafted appropriately, this threat can encourage users to contribute more to the shared information pool.

Numerous researchers have looked at excludable public goods as a cost-sharing problem: a group of people who benefit from a public good need to find an agreeable method of dividing the cost of the public good (Moulin, 1994; Deb and Razzolini, 1999). In general, cost sharing mechanisms are designed for providing a known, fixed amount of shared resource, with cost shares allocated after the size is determined. This is not generally appropriate for information pools: rarely is it sensible to decide in advance how much information is the right amount, and then to require an individual to contribute his share. Cost sharing also

⁵<http://en.wikipedia.org/wiki/Special:Statistics>, retrieved on March 16, 2009

strongly depends on the fact that money is a perfect substitute for itself; my \$10 is the same as your \$10 when funding a bridge, but my information and your information might not be equivalent. Also, it is difficult to “refund” information that has been contributed, making it difficult to implement bidding-based mechanisms like that of Young (1998).

Bag and Winter (1999) propose one such mechanism. In it, all users submit a bid containing the amount of money they are willing to pay and the total size of the public good they want. The mechanism then chooses the set of users whose bids contain the amount of money necessary to give everyone in the set their desired total size of public good. Everyone outside of this set is excluded and their money returned to them. This mechanism might be adapted to use information rather than money, but we would need to be able to specify the size and composition of the pool without actually knowing the information already. This is unrealistic in many circumstances, but if it were practical in some settings, this mechanism has desirable properties: it is efficient, stable, and order-independent.

Feldman et al. (2006) propose a related mechanism that is promising for some applications: encourage contributions by degrading the service quality to users who contribute little. Degraded service is natural in their context (slower downloads in a peer-to-peer filesharing system). When service quality is measurable, controllable and uniformly valuable to users, degradation might serve as an effective motivation to contribute.

Mechanisms in this general family, including cost-sharing, degradation of service, and the threshold mechanism we analyze below, are fundamentally related to another familiar mechanism: pricing with exclusion. One way to create an encyclopedia is to pay authors to write it, and then provide access to the information only to those who buy it. A degradation or threshold mechanism requires users to “buy” access, but payment is measured in units of effort, or of information content, but not money. Thus, our mechanism can be seen as a contribution to an emerging literature on *non-monetary mechanisms* for social provision of shared information pools. Non-monetary mechanisms are especially appealing if social conventions rule out the use of pricing, or if the transaction costs of creating and enforcing a pricing system would be prohibitive.

There may also be useful non-monetary methods to encourage contributions that do not rely on excluding or degrading access. For example, Rashid et al. (2006) took a different approach inspired by social psychology. They found that individual contributions on MovieLens can be increased by displaying how valuable a potential contribution would be to other users. This builds on their previous work (Ling et al., 2005) that found that contributions increased when users are given information about the uniqueness of a potential contribution.

5.2 Behavioral Model

To better understand, and make testable predictions about how users will respond to a threshold exclusion mechanism, we developed a mathematical model of user behavior. We begin with a set of potential users of an information pool. For simplicity we number these users $1, 2, \dots, N$. Each user i is permitted to choose some amount of information to contribute; we call this amount x_i . We assume that there is a meaningful way for the system to measure the quantity and quality of relevant information along a single dimension.

Users receive some value from having access to the information pool and the information contributed by everyone else. It is neither obvious, nor trivial for our analysis, whether users benefit directly from contributing their own information to the pool. After all, they already have the information for their own use. For example, after collecting research on a topic in a personal notebook or file, why make the effort to write it carefully for others and transfer it to Wikipedia? On the other hand, when the information is in the pool, others may add value to it, to the benefit of the original contributor. For example, others may correct one's errors in Wikipedia. Or, others may add value to a personal photo collection in Flickr, by adding tags or comments.

To allow for either possibility, we model two types of information pools. First, *informative* pools are those in which information is collected, possibly automatically aggregated, and then redistributed. Both del.icio.us and GlassDoor are examples of informative pools. The important feature here is that the pool primarily functions as a way to aggregate and distribute information to its participants. For pools like this, adding my information to the pool doesn't increase the value I receive from the pool because I already know my own information. Let $x_{-i} = x_1 + \dots + x_{i-1} + x_{i+1} + \dots + x_N$ be the sum of everyone's contributions to the information pool except user i .⁶ We represent the value of an *informative* pool by the function $v_i(x_{-i})$. We assume that this function is increasing and concave; more information is better, but as the pool gets larger each new piece of information is worth less.

When a user benefits directly from adding her own information to the pool because others enhance its value, we call the pool *collaborative*. Wikipedia is a collaborative pool; open source software is another example. We represent the value from a collaborative pool with the function $v_i(X)$ where $X = x_1 + \dots + x_N$ is the sum of everyone's contributions.⁷ We assume this function is increasing and concave. We encompass both models by specifying value as $v_i(\alpha x_i + x_{-i})$, where $\alpha = 0$ for an informative pool, and $\alpha = 1$ for a collaborative

⁶To focus on our main point, we simplify by assuming that information can be measured in constant-quality units, so that it is meaningful to sum x_i and x_j .

⁷Complementarities and substitutions between different information contributions may have much richer structure, of course. We again simplify to focus attention on our main points.

pool. This seemingly small difference leads to qualitatively different predictions.

Contributing information is not costless. Depending on the information, and the target information pool, contribution requires time and effort for some or all of data collection, analysis, drafting, formatting, editing, annotating, and organizing. It is not material whether these costs are denominated in money: they are foregone resources. In particular, we are concerned with the *opportunity cost*: time used contributing to an information pool is not available for the user's most valuable alternative use of that time. This cost also depends on the amount of information contributed. We represent the cost of contributing with the function $c_i(x_i)$. We assume that this cost is increasing in the amount of information contributed, since contributing more information generally requires more time and effort. We also assume that this cost is convex, which means that it gets increasingly more costly to contribute information the more you contribute.

Combining the above sources of value and costs, we form a *utility function*, which is a description of each user's preferences. In this case, the user can choose x_i , the amount of information to contribute to the information pool, and the function describes how desirable the outcome is based on that choice (and the choices of everyone else in the system). Higher values of the function are more highly desired by the user, so a user will generally choose the contribution level that maximizes his or her utility function:

$$U_i(x_i) = v_i(\alpha x_i + x_{-i}) - c_i(x_i) \quad (5.1)$$

This specification is very general and can describe the preferences of a wide variety of users. By making only simple structural assumptions (e.g. more information is better), we can apply this model to a large number of users for many different types of social computing systems. This gives us the power to make general design recommendations that apply to all social computing systems.

5.2.1 The voluntary equilibrium

We begin by calculating how much information each of the N users of the system will voluntarily choose to contribute. In particular, we search for a *Nash equilibrium* of contributions, which is a level of contribution for each user such that no individual user will want to change his or her contribution once they learn what everyone else is contributing. Nash equilibria are a common tool for predicting behavior in game theory and decision theory because they are *stable*: if everyone is making choices that match a Nash equilibrium, then no one will want to change their choice and equilibrium will continue.

The Nash equilibrium for this system is different for the two types of information pools.

For informative pools like del.icio.us, each user’s value from the pool only depends on other users’ contributions and not on her own contribution. Therefore, whatever she chooses to contribute will not increase her value from the pool (since she already know her own information), but will increase her cost. Everyone will choose to contribute nothing: $x_i = 0$ for all i . In equilibrium, an informative pool will not contain any voluntary contributions. We know that many users may still choose to contribute for personal (non-strategic) reasons; we explore this further near the end of this paper.

For collaborative pools like Wikipedia, the Nash equilibrium is more complex. Individuals gain some value from contributing to the pool because, in a collaborative pool, the sum is greater than its parts. Fixing a typo in a Wikipedia article might be worthwhile because it improves the quality of the whole article, or, alternatively, content added to the pool may be more valuable to the contributor than keeping it to herself because the contributions of others (complementary material, comments, edits and corrections) add value to it.

Let \bar{x}_{-i} be the total contribution in equilibrium from everyone other than user i . A Nash equilibrium results when, given this amount contributed by other agents, no individual agent benefits from either increasing or decreasing her contribution a small amount. Mathematically, all agents will be simultaneously in a Nash equilibrium if $\partial v_i(x_i + \bar{x}_{-i})/\partial x_i = \partial c_i(x_i)/\partial x_i$ for all i with $x_i > 0$.

5.2.2 Not enough information

In 1954, Samuelson (1954) pointed out that for public goods like this, relying on voluntary contributions results in fewer contributions than we as a society would want. This occurs because each user’s contribution provides value to all of the other users of the information pool, but this value is not taken into account when that user is making his or her contribution decision.

To formalize this, imagine that there is a “system planner” who can force users to contribute any amount he wants. How much should he require each user to contribute? Suppose the system planner wishes to maximize the total utility for everyone in the system:

$$\max \sum_{i=1}^N U_i(x_i) = \max \sum_{i=1}^N v_i(\alpha x_i + x_{-i}) - c_i(x_i) \quad (5.2)$$

This maximization balances the value of the information pool to everyone who uses the system and the cost of contribution from each person. The system planner will cap contributions when the additional benefits to everyone are no longer worth the additional cost to the contributor. In a system with an informative information pool, the system planner

will choose an optimal size of the pool and then assign contributions to those users with the lowest total cost. For a collaborative pool the result is similar, but the system planner will assign contributions to everyone whose marginal net benefit is above a threshold.⁸

The amount the system planner would assign to a user is different from the amount that would be contributed voluntarily; each user will choose to cap his or her own contribution when the additional value to himself (or herself) is not worth the additional cost of contribution. Since the system planner is concerned with the benefits to everyone, and these benefits are by definition (weakly) greater than the benefits to any one individual, the system planner will choose higher levels of contribution than individuals will choose for themselves.⁹ Consequently, the system planner would prefer an information pool that is larger than the pool that is voluntarily provided; the voluntary pool is *underprovided*.

Evidence suggests that many social computing systems are underprovided; for example Adar and Huberman (2000) found that almost 70% of users of Gnutella contribute nothing at all.

5.3 Setting a Minimum Threshold

To combat this problem of underprovision in information pools, we explore a simple incentive to encourage users to contribute more information to the shared information pool: require users to contribute at least a minimum amount of information to the shared pool before they receive access to the rest of the information in the pool. This requirement is intended as an incentive to induce users to contribute more; however users are not robots and can make their own choices. Using this model, we describe users' reactions to this incentive mechanism.

We begin modeling this requirement by specifying a minimum threshold t . If a user contributes at least t information, then they are given access to the information pool. If they contribute less than t , then they are denied access and cannot benefit from the information in the pool. In a minimum threshold system, each user's utility function is now discontinuous:

$$U_i(x_i) = \begin{cases} v_i(\alpha x_i + x_{-i}) - c_i(x_i) & \text{if } x_i \geq t \\ -c_i(x_i) & \text{if } x_i < t \end{cases} \quad (5.3)$$

⁸These results are standard and follow directly from the maximization of (5.2).

⁹Mathematically, the system planner will choose each x_i such that $\sum_{j=1}^N v'_j(\cdot) = c'_i(x_i)$. Each user will choose x_i such that $v'_i(\cdot) = c'_i(x_i)$. $v_i(\cdot)$ is non-negative, increasing and concave, so the sum of $v'_i(\cdot)$ is always weakly greater than any individual $v'_i(\cdot)$. As $c_i(\cdot)$ is increasing and convex, the system planner will raise x_i to compensate. Samuelson (1954)

In order to better characterize differences among users, we assume that all of the users can be ordered by their marginal net benefit of contribution. Users with a low marginal net benefit of contribution will be given low indices, and users with a high marginal net benefit of contribution will be given high indices. Mathematically, for any given level of contribution x_i , we assume that $\frac{\partial}{\partial x_i} (v_i(\alpha x_i + y) - c_i(x_i)) = \alpha v'_i(\alpha x_i + y) - c'_i(x_i)$ is increasing in i for any constant y . This ordering must be the same for all values of x_i . In particular, this ordering must hold for $x_i = 1$, meaning that users are ordered by the benefit of contributing the first piece of information. The user who benefits the most from contributing one piece of information will have the highest index. Also, for informative pools ($\alpha = 0$) this is an ordering based solely on cost; the user with the highest cost of contributing one unit of information will have the lowest marginal net benefit and therefore the lowest index $i = 1$.

5.3.1 An exclusion equilibrium

When a system enforces a minimum threshold constraint, users will choose alter their behavior accordingly. Some users may choose to increase their contributions, and others may choose to decrease theirs. In this section we use our model to derive how users will react to this incentive mechanism.

To begin, we calculate user i 's best response given that everyone else contributes x_{-i} . Define $x_i^0(x_{-i})$ to be the level of contribution that user i would voluntarily choose to contribute if there were no threshold and everyone else contributed x_{-i} . This value is likely to be non-zero for collaborative pools like Wikipedia (for the reasons mentioned above) but will be zero for informative pools like del.icio.us since users receive no additional benefit from contributing to the pool.

Lemma 5.1 *Given the threshold t and everyone else's contribution of x_{-i} , user i would choose one of three options:*

$$x_i^* = \begin{cases} x_i^0(x_{-i}) & \text{if } x_i^0(x_{-i}) \geq t \\ t & \text{if } x_i^0(x_{-i}) < t \text{ and } v_i(\alpha t + x_{-i}) \geq c_i(t) \\ 0 & \text{if } x_i^0(x_{-i}) < t \text{ and } v_i(\alpha t + x_{-i}) < c_i(t) \end{cases}$$

Proof Sketch (complete proofs available in Appendix D): If the user would naturally contribute above the threshold, then she will continue to do so. If the user prefers to contribute less than the threshold, then she must decide whether the benefit of accessing the information pool is worth the higher cost of contributing enough information to meet the threshold. If so, she will contribute the threshold; if not then she will leave the system, not

receive access, and contribute nothing. ■

Lemma 5.1 describes each individual user’s best response once she knows every else’s decision. However, this is insufficient to predict what will happen in such a system, since when user i makes her choice, that changes the size of the pool, which then also might change all of the other user’s choices. Next, we describe a Nash equilibrium for this system: a stable point at which no one wants to change their decision once they see the final size of the pool. In this equilibrium, users naturally sort themselves into three groups based on their marginal benefits and costs. We ordered users such that users with high net benefits have a higher index i .

Proposition 5.1 *For a given threshold t , there exists a Nash equilibrium characterized by (i^0, i^*) such that users will choose:*

$$\begin{aligned} x_i^* &= 0 && \text{if } i \leq i^0 \\ x_i^* &= t && \text{if } i^* > i > i^0 \\ x_i^* &= x_i^0 && \text{if } i \geq i^* \end{aligned}$$

Proof Sketch: Users with high net benefit — users with index $i \geq i^*$ — want to contribute more than the threshold in any case, and thus will do so. Users with low net benefit — users with index $i \leq i^0$ — will find that increasing their contribution to the threshold level is not worth the increase in cost. They will stop using the system and contribute nothing. Finally, the users in the middle with a moderate marginal net benefits will choose to increase their contribution to the threshold level in order to continue receiving access to the information pool. ■

The exact values of i^* and i^0 will change as t changes. This equilibrium looks different for the different types of information pool. In particular, for informative pools like del.icio.us, no one will naturally choose to contribute above the threshold. In equilibrium $i^* = N$, and everyone either contributes the threshold or stops using the system.

Because of the threshold, moderate benefit users will increase their contribution and low benefit users will stop contributing and stop using the system. But in collaborative pools, users with high marginal benefits will voluntarily contribute above the threshold. These users are contributing in order to “top off” the pool; they make the pool slightly larger because they benefit from the interactions between their contributed information and the rest of the pool. As the pool gets larger, these users won’t need to contribute as much to get a desirable size of pool:

Lemma 5.2 *In a collaborative information pool, everyone who voluntarily contributes greater than t will alter their contribution in exactly the opposite direction as the overall change in the size of the information pool.*

Proof Sketch: By assumption, the value from the information pool is concave, which means that a user values each additional piece of information in the pool less and less as the pool gets larger. When one user observes others increasing their contributions, she will value contributions to the pool slightly less, and will correspondingly slightly decrease her contribution to lower her cost of contributing. ■

5.3.2 Will it work?

Some users increase their contribution and other users decrease their contribution, but it is not clear which group is larger. Does setting a minimum threshold actually lead to more contributions and a larger information pool? A very low threshold won't cause many new contributions but might drive people away. A very high threshold will drive many users away but the ones that remain will all be contributing lots of information.

Setting a threshold has a larger effect on systems with many users. Since everyone must contribute the threshold, more users means more people have increased their contributions, leading to a larger information pool. Therefore, using a minimum threshold makes the most sense on large-scale internet-based social computing systems.

Proposition 5.2 *If t is less than some maximum \bar{t} , then as long as the user population N is large enough there exists a Nash Equilibrium in which everyone contributes at least t information to the pool. Furthermore, if the pool is an informative pool, everyone is better off than without a threshold. If the pool is a collaborative pool, welfare improves as long as the voluntarily contributed pool is sufficiently small.*

Proof Sketch: Consider the situation in which everyone contributes t information to the pool. Everyone benefits from access to a pool of information (size Nt if it is an informative pool, and larger for a collaborative pool). However, there is a strong individual incentive to deviate; most users would rather contribute nothing (to reduce their costs) and free ride on the contributions of others. The threat of exclusion works here as long as N is large enough. Larger N means a larger and hence more valuable information pool, and being excluded from this pool is a more substantial loss. However, at some point, having more information in the pool doesn't help and the value from the pool is at a maximum. If the cost of contributing t is greater than this maximum, then further increasing the size of the pool

won't convince the user to contribute t . Therefore, this equilibrium only exists for small enough thresholds.

For an informative pool, without a threshold all users free-ride and no one contributes to the pool. The threshold t serves as a coordinating device, inducing all users to coordinate and contribute exactly t . As long as everyone contributes, everyone is better off when they spend the additional cost to gain access to the resulting large information pool.

For a collaborative pool, the voluntarily contributed pool has some value. A user might be willing to still receive access to a threshold pool but be worse off overall by being forced to increase their contribution (and hence, their costs). However, everyone benefits from these extra contributions. As long as the voluntarily contributed pool is small, the increases in value to everyone else makes these additional costs socially worthwhile. ■

5.3.3 Will it always work?

We showed in Proposition 5.2 that under fairly general conditions there is always a minimum threshold that increases total system value for an informative pool. But, for collaborative pools we guaranteed the existence of a welfare-improving threshold only when the total of freely contributed information is sufficiently small. Can we not show that there is always some threshold, perhaps small, that increases the value of a collaborative pool?

In a word, “no”. A minimum threshold is sometimes valuable, sometimes not. When the pool size with no threshold is large, each additional piece of information isn't worth as much. Further increasing the size of the pool doesn't add much value. Mathematically, this is a result of our assumption that value is concave.

To illustrate why a threshold might not help, suppose there are two types of users: a large group of ‘readers’ in which individuals contribute very little without a threshold, and a small group of ‘writers’ that contributes considerably more per person. We might call this the “Wikipedia case”. When a threshold is introduced, the readers will increase their contributions up to the threshold because they want to retain access to the pool. These contributions come at a cost; all of these users now have to spent more time and effort to make the contributions. However, the pool is already very large (because of the writers), so the additional contributions from the readers aren't worth very much. If there are enough readers who have to pay this additional cost, then overall system welfare may be lower even though the pool is larger.

We suspect this might be the case for Wikipedia, a resource to which contributions are high without any threshold. Imposing a threshold would inconvenience the vast majority of users, who currently contribute little or nothing, yet plausibly might not increase the value

of the pool much for others. Indeed, there is likely to be another source of loss: many users may simply stop using Wikipedia rather than make the threshold contribution.

This example illustrates design advice from our analysis that can be applied across many different settings: only consider using a minimum threshold to increase contributions when the information pool would be small otherwise.

5.3.4 Adjusting the threshold

Maximizing the size of the information pool is not equivalent to maximizing its value. Though users benefit from a larger pool, there is a cost incurred to create it: this tradeoff ensures that the optimal size is less than the maximal size.

However, user value and user cost are not observable by system designers. There is one formal link between pool size and pool value that may be helpful to system designers: an increase in pool size is *necessary* (albeit not sufficient) for a threshold to increase value. In fact, this relationship holds for any adjustments in threshold level, providing a pragmatic check:

Proposition 5.3 *If a system designer raises the threshold t and the total size of the information pool decreases, then aggregate welfare has decreased. If a system designer lowers the threshold t and the total size of the information pool increases, then aggregate welfare has increased.*

Proof Sketch: Raising the threshold causes everyone who is contributing the threshold to incur a greater cost of contributing. Also, since it causes the total size of the information pool to decrease, everyone receives less value from the pool. Finally, some users voluntarily chose to be excluded, which means they are no longer receiving the benefits of the information pool. All of these effects lead to a welfare decrease. The second statement is the converse; all of these effects are reversed leading to a welfare increase. ■

Proposition 5.3 provides dynamic guidance for setting a threshold. As the system designer changes the threshold, the effect of that change on the total size of the pool provides hints about the (unobservable) total system welfare.

5.3.5 Who contributes?

Introducing a minimum threshold changes the distribution of contributors, and of their contributions. Consider first an informative pool. From Proposition 5.1 we know that contribution *breadth* increases: more users will contribute than would in a strictly voluntary

equilibrium. This is also true in a collaborative pool; more users will contribute at least t information than would in a voluntary equilibrium.

We can also characterize the change in contribution *depth* in a collaborative pool. Two groups of users reduce the depth of their contributions: First, those with the lowest net benefits reduce their contribution to zero and leave the system. Second, those with the highest net benefit were already contributing more than the threshold so there is no direct pressure on them to increase their contribution. However, by Lemma 5.2, because they benefit from content others contribute, these users will slightly decrease their contributions.¹⁰ Those in the intermediate range of net benefits increase their contribution to exactly the threshold t .¹¹

One interesting implication of the analysis of contribution depth is that setting a minimum threshold decreases contribution *inequality*: low contributors increase, and high contributors decrease, their contributions.

5.3.6 Summary of behavioral analysis

We have found that under rather general circumstances, a well-chosen threshold can improve the social value of an informative pool. For collaborative pools, the desirability of a threshold depends on several factors, which complicates the designer's task, in ways that the formal analysis can characterize helpfully.

One critical finding is that for an increase in the threshold t to be an improvement, it is necessary (but not sufficient) that the total size of the pool be greater at the higher threshold. When an increase in welfare occurs, it is due solely to an increase in contributions from those who are contributing precisely the threshold amount (those "bound" by the mechanism). These users have intermediate net benefits of contributing. Those with high benefits are already contributing more than the threshold amount, and they in fact reduce their total contributions. Those with low benefits leave the system, thus reducing their (in any case, small) contributions. One qualitative implication of this result is that for a threshold to be beneficial, the number of people who increase their contributions to reach the threshold must be sufficiently large compared to the number who exit the system.

Not only must there be a large enough group who are induced to increase their contributions, but the size of the pool when there is no minimum threshold must not be too large or a threshold will not be beneficial. The system loses value from those participants who exit

¹⁰This only happens when the total size of the pool increases. However, by Proposition 5.3, no system would want to introduce a threshold if it leads to a smaller pool.

¹¹Formally, there will be some value of i , call it i'' , such that for all users $i > i''$ (high net benefits), contributions decrease, and in general this will include some users in this intermediate group who give the threshold amount t . These are users who would freely contribute more than t , but when a threshold increases the size of the pool reduce their contributions to t .

rather than meet the threshold. Those doing the extra contributing are bearing additional costs to create the larger pool. Therefore, the benefits to those who receive access must be reasonably large for the overall value of the system to increase. But if the no-threshold pool size is already large, then the gain will be modest, and will not offset the additional contribution costs and the value lost by users who exit.

A related implication is that a threshold mechanism is more likely to be beneficial if there aren't too many who opt out. We have assumed that the cost of letting users access the pool once it is created is approximately zero. There is always an increase in social value from letting all users access an existing pool. The threshold mechanism excludes some users to create an incentive to contribute content, but the value those users would have received from access is a pure social loss that offsets the value of increased content. In a social computing service in which much of the value comes from a large number of low-value users, a threshold mechanism may be ill-advised, because more value will be lost from these many excluded users than is gained by the remaining users.

From this last point we can see a deep connection to a fundamental result in the pricing of information goods. When an information good costs resources (e.g., effort) to create, but can be costlessly reproduced and distributed thereafter, charging a price to access or use the good is a common way to recover the costs of creating it. Yet, charging a price creates qualitatively the same tradeoff as the minimum threshold mechanism: it excludes those who get value from the good, but not enough value to be willing to pay the access price. Market pricing for information goods may increase social value (more goods are created in the first place), but at a cost of the value lost by excluding those not willing to pay.

These findings are very general and apply to many different systems, though they are limited by the assumptions we made in deriving them. One advantage of mathematical modeling is that the assumptions are explicit, so that one can check if they hold in any given system, and revise or extend the modeling to obtain testable predictions for those differing circumstances.

5.4 Private Value

Users can receive value from their information in two different ways. First, users receive value from having their information in the pool because, for example, contributions by others improve one's own information, and thus add value to it; we modeled this value above, and will now refer to it as the *social value*. Second, though we have focused on information *pools*, users might receive value directly from putting their information in the system, independent of any contributions by others. For example, users of del.icio.us value

it in part for its use as a stand-alone, web-accessible personal bookmarking tool (Chapter 4). We call this non-social benefit from contributing the *private value*, $p_i(x_i)$, and assume that it is weakly concave (and increasing) in the amount of information contributed. Note that this value might be zero: for example, if a user does not receive any independent benefit from including their information in the information system.

This private value includes all types of motivation not dependent on others' contribution, including many social motivations: personal benefits from social interactions, expected benefits from meeting new people, and warm glow from being altruistic. We do not make assumptions on how people derive value. Rather, we assume this value has a specific structure: value increases as you increase contributions, but each additional contribution is worth less (concavity). Any source of value consistent with this structure (social motivations, etc.) is covered by the model.

Recognizing this value in the utility function, optimizing users will choose x_i to maximize:

$$U_i(x_i) = p_i(x_i) + v_i(\alpha x_i + x_{-i}) - c_i(x_i) \quad (5.4)$$

Even if there is no information pooling, or if there are no other contributors, a user will naturally choose to contribute some amount of information for its private value.¹² We call this level of contribution \hat{x}_i^* , the private contribution.

For an informative pool, there are two possible ways this private value affects a minimum threshold equilibrium. First, it is possible that $\hat{x}_i^* > t$; user i might want to voluntarily contribute above the threshold for purely private reasons. If that is the case, then user i will contribute exactly \hat{x}_i^* .

The other possibility is that $\hat{x}_i^* \leq t$. In this case, the personal benefits serve to mitigate some of the costs associated with contribution. This can be seen by defining a new cost function $\hat{c}_i(x_i) = c_i(x_i) - p_i(x_i)$. With this, the utility function then becomes $\hat{U}_i(x_i) = v_i(\alpha x_i + x_{-i}) - \hat{c}_i(x_i)$, which has the same form as the utility function we used earlier. For all levels of contribution $x_i \geq \hat{x}_i^*$, this new cost function is increasing and convex but strictly smaller than the original cost function $c_i(\cdot)$. When $\hat{x}_i^* \leq t$, user i can retain pool access by contributing t , but now would do so at a lower absolute and marginal cost. Therefore, with private benefits, more users will contribute the threshold t and fewer users will choose to leave the system.

For collaborative pools, there is one additional effect. Users who already contribute above the threshold will contribute even more due to the private value. Users who would

¹²To find the private contribution, delete the social value $v(\cdot)$ and maximize expression 5.1. The contribution will be the value of x_i for which $\frac{\partial p_i}{\partial x_i} = \frac{\partial c_i}{\partial x_i}$. It is unique because of the concavity/convexity assumptions on these functions; c.f. Mas-Colell et al. (1995).

have contributed the threshold without private value may want to increase their contributions above the threshold.

5.4.1 Blocking private use

When a user is excluded, should he also be excluded from accessing his own private information? For example, if I am excluded from accessing del.icio.us, should I still be able to see my own bookmarks? Mathematically, in our model, this is the difference between an excluded user receiving $p_i(\cdot) - c_i(\cdot)$ and the user only having cost $-c_i(\cdot)$.

Unfortunately, the answer is “it depends.” If the system gives all users access to their private contributions even if they are excluded from the rest of the pool, some users who otherwise would have stopped using the system will instead use the system privately. These users will not have access to the full information pool, but their contributions can still be added to the pool for the benefit of everyone else. These benefits to every one can make it worthwhile for a system to allow users to access a private version of the system that does not include access to the full pool.

However, when the cost of contributing the threshold is very large, some users who would normally contribute the threshold will instead choose to use the private version of the system. These users end up contributing less information to the information pool, potentially leading to a small information pool and lower total system welfare. When deciding if a private version of the system should be offered, system designers need to assess which of these effects is larger: are the increased contributions from private users enough to offset the lost contributions from users who would otherwise contribute the threshold?

5.5 Discussion

There are several practical considerations for applying this mechanism in a social computing system. Here we discuss a few of them:

5.5.1 Quality

This mechanism focuses on the quantity of contributions to an information pool, but often contribution quality is as important or even more important. A threshold changes the distribution of contributions and thus might change the distribution of quality. In addition, if users can choose the quality of their contribution, a threshold might also (perhaps perversely) affect the quality choice contributors make.

First consider the change in the distribution of contributions: users with low to moderate net benefits contribute more information than they would voluntarily. Suppose, for example, that the cost of contribution is positively correlated with quality. This might hold because experts have a higher opportunity cost (they have better things to do with their time). If this assumption holds, then using a minimum threshold will induce more high-quality contributions and might reduce the low-quality contributions.

However, if users can choose the quality of their contributions, then they are likely to choose lower cost (easier) contributions. For example, if del.icio.us required a minimum number of bookmarks, users bound by this requirement might simply bookmark the first t websites they find, regardless of their quality.

Depending on the specifics of the information system, the designer might employ any of several quality control mechanisms. One is to include some measure of quality in the threshold measurement. For example, Wikipedia could set a threshold of t edits to articles that are not reverted within 2 weeks. This ensures a minimum quality level for contributions. Another method of ensuring quality is to use a secondary mechanism that induces higher quality contributions. For example, Amazon.com asks its users to rate with up to 5 stars the quality of each of its user-contributed reviews. It then provides public recognition for users whose reviews are rated highly.

5.5.2 Measurement

To model this mechanism we assumed that there exists a meaningful way to measure the quality-adjusted quantity of information contributions along a single continuous dimension, x . In order to implement this mechanism in an actual social media system, we do not need a continuous measure of quantity. A binary measure of whether the contribution is “enough” — if it meets or exceeds the minimum threshold — is sufficient. For example, the GlassDoor service doesn’t measure how much salary information a user contributed: if she contributes at all she is granted access to the site. One implication of our analysis, since we predict that many users will contribute exactly the minimum necessary, is that it is important to set the threshold such that the including the minimum contribution in the information pool is actually useful to other users.

5.5.3 Authentication

Excluding users who do not contribute enough depends on being able to identify them. This usually is done by requiring that users create accounts before accessing the information pool.

This is an additional cost of contribution, and might reduce use of the system (Gazzale and MacKie-Mason, 2008).

5.5.4 Bootstrapping

Another practical problem is the bootstrapping problem: how does the system react to new users before they have had an opportunity to contribute? This is important for two reasons. First, social computing systems are often an experience good; users need to experience the information pool to know how valuable it is to them so they can make an informed choice. Second, users often learn how to contribute by mimicking the contributions of others. Without being able to see others' contributions, new users will not know the appropriate social norms and conventions for the system. (Burke et al., 2009)

For these two reasons, it may be beneficial to implement an "introductory" period during which users can see and interact with the system without meeting the threshold constraint. At the end of the period the system can enforce the threshold. With such a practice, authentication is not sufficient: the system designer must now address the problem of "cheap pseudonyms" (Friedman and Resnick, 2001): users who create new accounts with a new "introductory" periods to avoid the threshold requirements.

5.6 Using a Minimum Threshold Mechanism

Most social computing analysis is at an individual level: e.g. lurkers are valuable (Nonnecke et al., 2006). In this study, we look at aggregate patterns, where it makes sense to ask about design tradeoffs across individuals: when is it worthwhile to lose the lurkers in order to gain more contributions from the remaining users?

An example will help illustrate this tradeoff. Facebook uses a minimum threshold. To access Facebook, you must 1) create an account, and 2) contribute a list of friends (your social network). Facebook doesn't have true lurkers; everyone contributes something. Facebook presumably made the decision that the increased contributions (knowing everyone's social network) allowed them to create enough extra value for their remaining users that it was worth losing the lurkers. Twitter made the opposite decision and does not have a minimum threshold.

We characterize this design tradeoff in concrete terms, and provide advice, summarized below, on when this tradeoff is worthwhile. We explicitly model the value from lurkers and consider it a loss when they can no longer access the system. We then ask *when* that loss is worthwhile. By looking at system-level properties, we are able to understand potentially

valuable tradeoffs that individual-level analyses miss.

The Facebook example also illustrates another useful point. Everyone who remains in the system will have contributed at least the minimum. If the threshold is chosen carefully, the system can take advantage of the knowledge that everyone contributes this information. On Facebook, everyone has to contribute social network information. Facebook is then designed to take advantage of the fact that all Facebook users have provided their social network information to the system; many features such as the privacy controls assume the existence of contributed social network information. This well-chosen threshold enables many useful features.

Because there is an unavoidable tradeoff, not all social computing systems will benefit from using a minimum threshold. Our analysis above characterizes how users will react to a system which uses this mechanism. We now use this information to provide concrete design guidance for using the minimum threshold mechanism. A social computing system should consider using the minimum threshold mechanism when:

- There are a large number of users in the system.
- Without an explicit mechanism, users contribute very little.
- Having more users contributing is more important than greater contributions from each user.

When using the minimum threshold mechanism,

- Users with high costs of contributing and low benefits of access will stop using the system.
- Setting a minimum threshold increases the breadth of contribution — more users contribute — but potentially sacrifices depth of contribution.
- Systems with an informative pool will see a greater increase in contributions than systems with a collaborative pool.

Finally, to use the minimum threshold mechanism,

- Watch the size of the pool as you change the threshold to know if the change helped.
- Set the threshold so that the minimum contribution has value to others, since most users will contribute exactly the minimum.

- Think carefully about whether to allow users to use the system privately (with access only to their own private contributions) when they are excluded from the rest of the system.

Even though it is costless to let everyone access an information pool and benefit from its contents, we are frequently better off to use an excludable public goods rule that imposes a minimum contribution. The reason is simple: without this incentive, participants will undercontribute, and fully or partially free-ride on the contributions of others. Exclusion is a knob the designer can turn to adjust the tradeoff between the benefits of inducing more contributions and the costs of withholding the value of information from some potential users.

Chapter 6

Improving the Design of the Social Firewall

In Chapter 1, I introduced an idea for a technology I called a *social firewall*: a socio-technical system that allows users to share information with each other about security-related firewall decisions. Firewall decisions are technically simple; the user chooses whether specific technical actions (such as allowing an application to access the network) should be Allowed or Denied. Designing such a system is difficult, at least partially because it depends critically on how its users behave. I discussed a number of different behaviors that are important to the success of such a system: contributing information, ensuring quality, preventing maliciousness, enabling collaboration, bootstrapping the initial pool of users, and encouraging end-user innovation. In this dissertation I have focused on the first, and most important, of these behaviors: encouraging users to contribute information to the system.

Encouraging contributions is not simple; contributing information requires time and effort. This time and effort is required for some or all of: data collection, analysis and understanding, drafting, editing, annotating, and organizing. It also requires the contributor to possess some information that he or she feels would be useful to the community created by the system, and be confident enough in that information to be willing to contribute it. As a result of these challenges, many users would prefer to use the system as a consumer, but free ride on the contributions of others and not spend their energy making new contributions. Unfortunately, if most of the users of the system feel this way, little information will be contributed and the system will not be very useful.

In this dissertation, I have sought to understand some ways that the technical design of socio-technical systems influences the resulting user behavior. In this chapter I summarize what I learned and apply it to improve the design of a social firewall.

6.1 Design for Information Sharing

One fundamental design question for a social firewall is “What information should be shared?” A very simple social firewall would take a standard personal firewall system and augment it by sharing the Allow/Deny decisions that are made by users. Such a system could simply display aggregate information about how other users have made the same decisions; for example, it could display something like “80% of the users have chosen Allow.” However, as I discussed in Section 2.3.4, this has a major problem: information cascades. Once a few people choose one way, it is rational for everyone that follows to follow the crowd, even if they privately believe the crowd is incorrect, and even if the crowd has made a bad choice. Goecks et al. (2009) independently discovered this problem, but did not present a solution.

Fortunately, the theory about information cascades provides some guidance for avoiding them. Cascades happen because each individual only observes the decisions of previous people, but not the outcomes; i.e. they never find out if the decisions were good decisions. One way to avoid cascades is to share not just decisions but outcomes (Bikhchandani et al., 1998): did this decision lead to a security incident, and did the decision lead to the user not being able to use the computer as they wanted to? Unfortunately, outcome information about security incidents is hard to come by; it is difficult to prove that no security incident happened, and even when it does, it is difficult to pinpoint the one security decision that enabled it. So sharing outcome information is infeasible.

The other feature leading to information cascades is the fact that there are a small number of discrete options. Each user has to choose between Allow and Deny. I can avoid information cascades by using a form of contribution that permits arbitrary information to be conveyed: textual comments. I propose that a standard personal firewall should be augmented with the ability to contribute free-text comments on each of the security decision popups. Those comments will then be displayed for other users in the future who receive the same popup on their computers, as illustrated in Figure 6.1. To achieve this, there should be a central server that collects contributions from everyone, and that can be queried for relevant comments for each popup by the personal firewall system.

6.2 Lessons about Home Computer Users

Free-text comments seem like a good solutions to the information cascades problem; but will they work? Are home computer users capable of contributing useful comments that help other users? The results of my study of home computer users in Chapter 3 indicate that

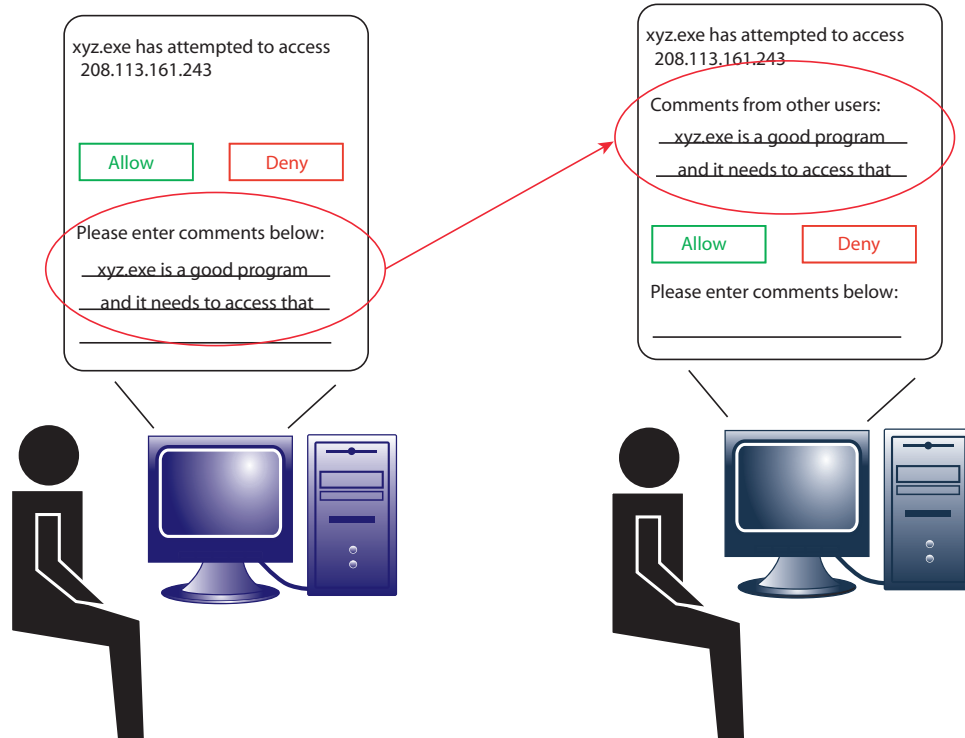


Figure 6.1 Diagram of the main interaction in a social firewall

users are likely to be able to contribute at least some useful information for each other, and that those contributions might be useful to others.

Home computer users conceptualize computer security threats in multiple ways; consequently, users make different decisions based on their conceptualization. In my interviews, I found four distinct ways of thinking about malicious software as a security threat: the ‘viruses are bad,’ ‘buggy software,’ ‘viruses cause mischief,’ and ‘viruses support crime’ models. I also found four more distinct ways of thinking about malicious computer users as a threat: thinking of malicious others as ‘graffiti artists,’ ‘burglars,’ ‘internet criminals who target big fish,’ and ‘contractors to organized crime.’

Each different mental model induces user to focus on different types of defenses; for example the ‘viruses cause mischief’ model is the only that induces users to make regular backups of data because these users believe that a virus might randomly delete something important. If other users were to be exposed to this conceptualization of viruses, then they too would hopefully see the value in regular backups and begin following that advice. Because the folk models differ in their security proscriptions, exposure to and understanding of multiple models of threats may induce users to undertake more precautions. In this small study, the few users who possessed multiple models of threats undertook all of the

precautions that would help protect against any threat in their models. In other words, they used an OR model of security: if security advice protected against Threat 1 OR Threat 2, then they followed it.

While the home computer user study is rigorous, I did not use a generalizable sampling method. I am able to describe a number of different folk models, but I cannot say how prevalent each model is in the population. I also cannot say if my list of folk models is exhaustive; there may be more models than I describe. It is possible that some of the models I describe are extremely common in the general population, and information sharing would not help among people with these common models. It is also possible that some of the models are extremely rare; information sharing should be encouraged for holders of these models when they are found.

6.2.1 Design suggestions

One goal of this research was to discover information that could be applied to designing better security technologies. To this end, I provide an example of how this information can provide some potentially useful design suggestions for a social firewall.

From this study, I better understand the value of textual comments in a social firewall. My respondents described making security-related decisions by applying their current understanding (their mental model) to better understand the risks involved. In addition to the decision, textual comments allow the user to convey reasons for making that decision, including relevant parts of their mental model(s). This is important because my respondents indicated that they intentionally ignored security advice that seemed irrelevant to their understanding of the risks; users would likely also ignore advice from others that did not include an understandable explanation. Users can also use textual comments to tell the stories that inspired certain security decisions. For example, they can tell a story about how a specific Deny decisions might have caused some application to stop working.

Additionally, I have learned from this study what type of information might be influential. Users are likely to get the most value out of comments that explain folk models. Such comments can help users understand why a particular decision is important by helping them to understand the security threat that the decision addresses. For example, users who think that all hackers are *Graffiti*-type could benefit from being exposed to the idea that hackers might steal information for identity theft. This could cause them to take additional precautions, like using anti-virus software. The system could specifically prompt users to contribute information about “what threat does this decision address?”

I suggest that the contribution interface (the security popup) have two separate text

boxes: 1) what threat the security decision affects, and 2) how that decision helps protect against that threat. By prompting users for these two pieces of information, the system will encourage users to contribute information from their mental model of threats. This information can then be used by other users to make Allow / Deny decisions.

6.3 Lessons from the Delicious Study

From the social bookmarking website del.icio.us, I learned in Chapter 4 that even when users contribute for only individual, non-social reasons, a social information system can succeed in producing a broadly-useful pool of shared information. Users of del.icio.us contribute bookmarks and tags for personal organization. The system then shares these contributions by default by making them publicly accessible as a side effect. The design of del.icio.us ensures that the contributed bookmarks are useful to others through *incentive alignment*: users are encouraged to limit and focus their bookmark choices to maintain their personal organization, and consequently, these limited, focused collections of bookmarks are useful to others.

Del.icio.us is only a single case study; the *side effect mechanism* it uses will not necessarily work for other social computing systems. However, it does illustrate the feasibility of this mechanism: it has worked for one specific social computing system. Most of the concrete findings come from the small sample qualitative study, but my description of the mechanism relies on the assumption that the trends found in that study are widespread on the site. I did verify that tags are commonly applied for personal organization reasons, but I did not verify that bookmarks are regularly chosen for personal reasons. If some of findings are highly idiosyncratic to my sample, then the description of the side effect mechanism might not hold true for del.icio.us.

6.3.1 Design suggestions

Private, individual reasons to contribute can be powerful motivators. Engineering a system to provide private benefits from contribution, and then making those contributions public as a *side effect*, can be an effective method of inducing users to share information electronically. To apply this mechanism to design a social firewall, we must provide users with a private, personal reason to contribute information that is aligned with the needs of other users when they consume that information.

I propose creating a supplemental interface where users can return and revise their security decisions easily. This interface allows users to better manage their own security.

A number of the respondents in the home security study who had used personal firewall systems expressed concern with choosing Deny; they felt that the decision was irreversible (or at least difficult to reverse) and didn't know what would break. Providing an interface where they can sort through previous security decisions and make changes might help with this problem.

This interface can also display the information the computer user has contributed about *why* he or she made that particular decision; both information about the threat and information about how that decision helps with the threat. By having this information available, users can better revise security decisions because they know more about what they were thinking when the original decision was made. This is a private reason to contribute information; the user can help his future self revise security decisions.

The contributed information can also be sent to a central server and made available to other users who are faced with similar decisions. This is the side effect mechanism; users contribute information for private reasons but the information is made public as a side effect. This design also properly aligns the contribution incentive with the needs of consumers. Users contribute the semi-structured information that will help them most revisit their decisions. This information will also be able to help others because the purpose is the same: understanding *why* the decision was (or should be) made in a certain way.

6.4 Lessons from the Minimum Threshold Study

Setting a minimum threshold has the potential increase the amount of contribution from a number of users of a social computing system. But, my analysis of this mechanism in Chapter 5 demonstrates that this mechanism has a tradeoff: some number of low marginal value users refuse to contribute and leave the system altogether, no longer able to receive any benefit from the system. Understanding this tradeoff is the core of understanding when to use a minimum threshold.

There are identifiable circumstances under which setting a minimum contribution threshold for participation in a social information system will lead to an increase in breadth of contribution by causing a larger proportion of users to make a non-trivial contribution, and cause the total quantity of contributions to increase. Minimum thresholds are best used when breadth of contribution (number of users contributing) is more important than depth of contribution (how much each user contributes). This mechanism also works best when there are many users in the system, since large numbers of users all contributing create more value from accessing the system, and thus provide a stronger incentive to contribute the minimum in order to gain access. Additionally, a minimum threshold system will have a

stronger effect in an informative pool than a collaborative pool.

The largest issue with my analysis of minimum thresholds is that it is a theoretical model; I have no empirical data to validate my claims. The model is predicated on the strong assumption that users will behave rationally. The Nash equilibrium solution concept assumes a static equilibrium, but in practice users choose their contributions over time. I believe that the Nash equilibrium is a good first approximation of human behavior in strategic settings; however, it is very important to validate Nash equilibrium predictions with real world data since users often deviate from fully rational behavior.

6.4.1 Design suggestions

A social firewall, as discussed so far, forms an informative information pool. Users do not collaborate with each other; having one's own information in the pool doesn't necessarily encourage others to build on it. This is a favorable condition for using a minimum threshold.

Previously, I suggested that it might be beneficial to expose home computer users to different types of folk models of security threats; this exposure might encourage them to recognize more security threats, follow more security advice, and exert more effort securing their machines. Since a social firewall has contributions from home computer users rather than experts, I suspect that it would be more beneficial if a wide variety of users contribute rather than having a small number of users contribute. My analysis Chapter 5 suggests that a minimum threshold system would increase the breadth of contribution by encouraging more users to contribute. This is one way to achieve this variety.

The Facebook example illustrates one good way of implementing a minimum threshold: have a mandatory information contribution that happens at signup. For a social firewall, one strategy that might work is to have each user choose one security decision from a list of common security decisions, and contribute answers to the two questions in every security prompt (listed above): 1) what threat does this security decision address, and 2) how does this security decision address the threat. Basically, users get to make a contribution to one popup of their choice. This is the minimum contribution required to gain access to the system.

Providing a list of possible security decisions ensures that users comment on decisions which could benefit from additional comments. But allowing users to choose from a list allows users to choose to contribute information that they feel strongly about, or feel knowledgeable about. This will hopefully help increase the quality of the contributions. However, as discussed in Chapter 5, some additional quality check should be done to ensure that users aren't just typing gibberish. If the final design of the social firewall includes some sort of

comment rating system (like Amazon's thumbs-up/thumbs down), then maybe access can be revoked if a user's contributions receive too many negative ratings.

Since the mandatory-contribution-at-signup system is an instance of a minimum threshold, my analysis in Chapter 5 states that this will increase the breadth of contribution by getting many different users to make contributions about a number of different security decisions. And since users are allowed to choose a decision to comment on, the comments are likely to be well-informed and well-explained (or at least better informed than other comments from that same user). By requiring them to comment on both of the points listed above, users are forced to explain their mental model of the threats they face. Since we are getting a wide variety of these explanations, subsequent users who see all of these comments will be exposed to many different ideas about the threats they face; this should help encourage them to make more secure decisions.

6.5 Going Forward

In this dissertation I focus on the primary behavioral concern in designing a social firewall: inducing users to contribute information that home computer users will be able to use to make better security decisions. The findings in Chapters 3-5 all apply beyond designing a social firewall. Chapter 3 describes the folk models that home computer users apply to making security decisions. This knowledge can be used to design better user education and better security systems. Chapters 4 and 5 describe general mechanisms that may apply in many different types of social computing systems. Both the side effect mechanism and the minimum threshold mechanism can be used on almost any social computing system; my analysis helps to understand how to use them and when they should be used.

There are many more mechanisms for inducing contribution than the ones I discussed in this dissertation. These two mechanisms do not work in every situation for every social computing system; much work is still needed to identify more complementary mechanisms for inducing contributions. However, I have identified two mechanisms that are general enough to work in most social computing systems, practically simple enough to be included in the current system, and powerful enough to make a difference in contributions.

There are still many important behavioral considerations that need to be addressed for a social firewall to be successful. If these mechanisms are successful in inducing contributions, then the next important challenge will be to address quality. Contribution and quality cannot completely be separated; these design suggestions above also focus on how to induce the type of contributions that will be most helpful. This is one important aspect of quality.

Quality is important because it is usually impossible to display all of the relevant contri-

contributions at one time. Some subset of contributions must be chosen for display. There are a number of possible ways to deal with this: Amazon.com uses a “thumbs up” system for allowing users to indicate which reviews are most valuable; Slashdot uses a meta-moderation system where random users are given points which they can use to vote a few contributions as helpful (Lampe and Resnick, 2004). Designing mechanisms to deal with quality is an important issue in designing social computing systems like a social firewall.

In addition to quality, mechanisms need to be designed for all of the behavioral challenges listed in Section 1.3.1: user retention, collaboration, preventing maliciousness, bootstrapping, and end-user innovation.

In this dissertation I also illustrate an incentive centered design approach to the design of social computing systems. I focused on identifying the behaviors that would be most beneficial to solving a specific problem (botnets), and then I worked on designing technology that would elicit those behaviors. I believe that this approach is a valuable way to design complex socio-technical systems to solve practical, real-world problems.

Appendix A

Interview Guide for Folk Models Study

The study was divided into two rounds. The first round was more exploratory. In this round, I asked a wide variety of questions about how users manage their security. In the second round, I focused in on three specific aspects of security: viruses, hackers, and identity theft.

A.1 Round 1

Below is a copy of the basic interview questions from round 1. I took a printout of this sheet with me on each interview, and used it to make notes about interesting observations, followup questions, and any other notes to myself.

This list of questions is very broad, and mostly intended to prompt users to talk about the security. I used it primarily to find out what kinds of threats users noticed, and how they dealt with those threats.

Folk Models of Security -- Specific Questions

Background and Demographics

- What do you do for a living?
- Do you own a computer?
 - What type? What does it run?
 - What do you use it for?
 - Email? Surfing web? Games? Work?
- Do you use a computer at work?
 - What kind?
 - What do you use that for?
- When you have problems with your computer, what do you do? Call someone? Who?

General Problems

- What different types of security problems exist in computers? (Write them down)
- For each type of problem:
 - Can you elaborate?
 - Who does it?
 - What kind of person is behind it?
 - Do you have any idea why they do it?
 - What are some ways you have of stopping it?
 - How does that work?
 - What do you have to do to make it work?
 - Is there any way around it?
- Are there any other information security problems you can think of?

Specific Threats

- Viruses + Worms
 - Do you know what a computer virus is?
 - How do you get a computer virus?
 - Can others get a computer virus from you?
 - What happens when you get one?
 - How do you get rid of a computer virus?
 - Have you ever had one?
 - How are viruses created? Why?
- Targeted Attacks
 - Do hackers ever try to break into your computer?
 - Why would they do this? What would they gain?
 - How do they break in?
 - Why do they break in? Are they looking for something?

- What do they do once inside?
- How can you stop them?
- Have you ever had this problem?
- Trojans / Downloaded programs (including spyware and adware)
 - Can programs you download from the Internet ever do something bad?
 - What do you mean by bad?
 - What kinds of programs will do this?
 - What exactly do they do?
 - Why do they do this?
 - How do you stop them?
 - Have you ever had one?

Defenses

- Passwords
 - Do you have a password for your computer?
 - Why?
 - Do you ever change it? Why?
 - Why are passwords necessary?
 - Do you have other passwords?
 - Any problems with so many?
 - Is it OK to write them down? Give them to others? Why or why not?
- Updates
 - Do you ever update your operating system?
 - Are the updates automatic?
 - Do you ever update other applications / programs on your computer?
 - Do you just buy new versions, or are you aware of smaller updates like patches?
- Do you have any other security software installed?
 - Firewall?
 - Do you ever interact with it? How?
 - How do you decide if its OK?
 - Anti-virus?
 - Has it ever caught a virus?
 - What do you do when it does?
 - Anti-spyware?
 - Has it ever found anything?
 - What do you do when it does?
- If not, have you considered security software? Why didn't you?

A.2 Round 2

In the second round of interviews, I used a revised list of questions based on what worked and didn't work in the first round. I augmented this list with three scenarios. These scenarios were intended to 1) provide a hypothetical instance of a security problem to better understand how users would react, and 2) challenge the users mental models and see how they resolve the challenges. Below I include both the revised list of questions and the scenarios.

Home Security Interview Study: Questions

Scenario 1:

A friend you trust calls you to tell you that a hacker has compromised your computer.

- What would be going through your head as they tell you this?
- Would you believe them?
- Would this surprise you?
- What do you think would happen to your computer?
- What is your best guess as to why the hacker would do this?
 - (followup this -- what information is he looking for? would he find it?)
- What is your best guess who it would be?
 - what kind of person?
- What is your best guess as to how they were able to access your computer?
- What would you do about it?

Your trusted friend tells you that the hacker left a program running on your computer.

- Would you believe him/her?
- Would this surprise you?
- What is your best guess about what this program does?

Scenario 2:

After briefly using your computer to check his or her email, a friend you trust tells you that you have a virus on your computer that was written by the Russian mafia.

- What would be going through your head as they tell you this?
- What do you think this virus would do?
 - Why?
- What is your best guess as to how you got this virus on your computer?
- Is this different from other viruses you have had or heard about?
 - How so?

Scenario 3:

You receive a call from the police. They tell you that they have evidence that you have been the victim of identity theft.

- What would be going through your head as they tell you this?
- What do you think they mean by "identity theft"?
- What would you do about it?
- What would be your best guess as to who this criminal would be?
- What would be your best guess as to how the criminal got the information needed?

The police inform you that someone has been using your main credit card.

- Do you think that this is "identity theft"?
- What would you do about it?
- What would be your best guess as to who this criminal would be?
- What would be your best guess as to how the criminal got the information needed?

The police also inform you that this person has applied for and received a large loan in your name.

- Do you think that this is "identity theft"?
- What would you do about it?
- What would be your best guess as to who this criminal would be?
- What would be your best guess as to how the criminal got the information needed?

Background Questions

- What do you do for a living?
- Do you use a computer for that?
 - What kinds of things do you use it for?
- You have a computer at home?
 - What type of computer is it? (Mac/PC)
 - What kinds of things do you use it for?
 - Does anyone else use it?
- Can you think of any problems you have had recently with your computer?

Viruses

- Have you ever had a virus on your computer?
 - Can you describe what happened?
 - When?
 - What did it do?
 - How did you know it was a virus?
 - How did you fix it?
 - Do you know where it came from / how you got it?
 - Do you know how the virus was created?
 - What do you think the virus was trying to do?
- Do you know anyone who has had a virus?
 - *Same questions as above*
- Are you worried about viruses on your computer?
 - What are you worried viruses will do?
 - What else can viruses do?
 - What do you do about it?
 - Is your A/V up to date?
 - Does it ever pop up and say anything?
 - What makes places seem unhygienic?
- If you got a virus, how would you know?
- Where do you think the virus came from?
 - Why do you think it was made? (Does this question even make sense to the user?)

Hackers

- Have you ever had a problem with hackers?
 - If so, can you describe it?
 - What happened?
 - Who did it? Why?
 - How was it detected? fixed?
- Do you know anyone who has had a problem with hackers?

- Are you worried about hackers?
 - Why?
 - What do you think they will do?
 - What else can they do?
 - What can you do about it? Are you doing about it?
 - Do you think you will have a problem with hackers?
 - Why or why not?
 - What do you do to protect yourself?

Identity Theft

- Do you shop online?
 - Can you remember the last time you shopped online? How did you pay? What information did you provide?
 - Do you provide your CC# in general? Address?
 - Are you worried about that?
 - What would make you not want to give out that information?
- Do you bank online? taxes online?
- Do you do financial stuff on your computer at all?
- Are you worried about Identity Theft?
 - Who are you worried about getting your info?
 - Who is safe to give your info to? How can you tell?
 - How can they get your personal info?

Integration Questions (last)

- Are all hackers looking for identity information, or do some of them look for other things?

Appendix B

Intermediate Analysis of Folk Models of Home Computer Security

B.1 Interpreting the Facets in Table 3.2

B.1.1 Viruses

B.1.1.1 Creator

There are a couple different understandings of how viruses are created. There are some interesting overlaps between people's ideas of virus creators and of hackers.

No creator Viruses are like biological viruses in that there is no explicit creator. They just exist, or are created accidentally by computer programmers. Respondents will acknowledge that they must have been created by someone, but never thought about who that is. Often these types of viruses do not have a purpose, they just have effects on computers. Much like a cold virus doesn't have a purpose, but causes various symptoms that are annoying.

Mischievous Individual Viruses are created by people who are being mischievous. These are usually technically talented, misguided teenagers. The motivation of these individuals for writing viruses is "sheer sport" (Lorna) or "who want to hurt people for no reason." (Dana) A slightly different but related motivation is writing viruses as part of "learning about the internet" (Jack) or "as a technical challenge" (Floyd).

Professional Criminal Viruses are created by one or more people who are basically professional criminals. People talk about "massive hacker groups" (Hayley) Often respondents talked about viruses being created to steal personal info (presumably for ID theft reasons).

Others believe that they were created for a purpose, but don't necessarily know enough to know what that purpose is. (Hayley) Usually, "making money" is the believed goal of the created, though occasionally they presume a goal of "taking down the government".

B.1.1.2 Effects they have

Most users didn't distinguish between different types of computer viruses. All viruses they got had a chance of having effects from a specific list. There are a couple different ways of thinking about the effects of viruses that reveal a lot about the underlying mental model of viruses.

Errors, Bugs, and Crashes Viruses are basically poorly-written software. They might not have been created intentionally, but they are badly written and their effects are exaggerated versions of normal software bugs. These effects occur mostly randomly. Sometimes the computer will get slow. Sometimes the computer will crash, or a program will crash ("boot me out" of applications (Erica)). It might accidentally delete or "wipe out" information (Christine and Erica) or delete system files (Jack). Overall, the computer "doesn't function properly." (Erica)

Mischiefous Problems Viruses are programs that cause mischief. The effects of viruses are mostly there to be really annoying. Displaying a skull and crossbones (1st round). Downloading pornography. (Lorna) Viruses can cause a bunch of popups. Deleting files and programs, or causing the computer not to boot are also in this category, but are more intentional effects rather than accidents. (Floyd) Despite being intentional, the respondent still thinks they are done "for no reason." (Floyd)

Facilitating Crime Viruses are intentionally placed on computers to facilitate some sort of criminal activity. Most of the time respondents talked about viruses that steal personal or financial information. Some users talk about viruses being "automatic" ways of collecting this information. (Irving) Some users would use the term "Spyware" here, but most did not. This also covers when people talk about popups as advertising? (Floyd?)

Focus on Visible Outcomes without Thinking About How and Why A number of users focused their comments primarily on effects that would effect the usefulness of the computer to them. They would only talk about effects that caused major interruptions, such as crashing, data loss, slowness, or popups. If it didn't effect their direct use of the computer, then they didn't think about it. Some users seemed to be cognizant of this; Lorna says there

might be “hidden ones” and Gail says “If it wasn’t an annoying virus I wouldn’t know.” Others seemed to presume either that the only effects were user-visible (Christine says “I guess I would know, wouldn’t I?”) or that any non-user-visible effects didn’t really matter.

B.1.1.3 Transmission and Prevention

The final major category of differences regards how viruses are transmitted, “caught,” and prevented. First, everyone assumes that no one gets viruses intentionally; it must be a mistake to get a virus.

Active Mistakes For some people, the main way you get viruses is by actively clicking on the virus or downloading the virus. You might not know that it is a virus, but you have to actively put it on your computer. The most common way this is stated is that viruses come from email attachments; you have to open the attachment (aka run the virus) to be infected. Often these emails come from people you don’t know. (Irving) Erica talks about how if you download games you can get a virus. This might be similar. Lorna talks about clicking on bad links in websites and believes this is more dangerous because she knows how to recognize bad emails but not bad links.

Passive Mistakes If you are hanging out in the bad parts of the Internet, than you can accidentally pick up a virus. For these people, there is no specific action that gets you a virus; you are just more susceptible to them if you go to shady websites. Downloading a game or going to “the FaceBook” or “the MySpace” (Dana) is also included. Floyd believes it is related to cookies, which are another thing that are automatically put on your computer by websites.

They Just Happen No one mentioned that viruses just happen. Despite the fact that many real viruses right now spread automatically without human intervention, exploiting bugs in listening software; all of the major viruses in the news recently have been of this type. Despite this, many people seemed to know that keeping all your software up to date is important. I don’t think they could have told me exactly why, but they know it is important for stopping viruses.

B.1.1.4 Sources

There are various places from which you can get viruses:

Emails Viruses come from emails. These are usually suspect emails from people you don't recognize. Some users believe you need to open the email, or open the attachment on the email, to be infected by the virus. Users who see email as a source of viruses often repeat the security lesson "don't open attachments from people you don't recognize" or "don't open attachments you weren't expecting."

Web Pages Viruses come from visiting web pages. One person things frequently places like "the FaceBook" and "the MySpace" will likely lead to getting a virus. People who feel this way think that just going to a web page will lead to contracting the virus. Most people feel that business webpages are OK, and more morally ambiguous webpages (like pornography or other entertainment webpages) are more likely to lead to infection. Most respondents also believed that a "secured" website would not lead to a virus. However, one respondent (Gail) acknowledged that at some sites "maybe the protection wasn't working at those sites and they went bad." Infection here is very passive and comes from visiting the website.

Downloads / Files Some viruses must be downloaded in order to get onto your computer. Therefore, by avoiding downloading files or program, a person can avoid being infected by those viruses. Often people talk about download games as a particularly risky activity because it might come with a virus.

B.1.2 Hackers

The word 'hacker' is a generic term that most respondents use to describe a computer criminal.

B.1.2.1 Identity

Different respondents had very different ideas of *who* hackers were. These are usually stereotypes of the type of person they believe is most likely to be a hacker.

Mischievous Technical Individual Hackers are individuals with strong technical skills but "a little unethical." (Christine) Sometimes they are envisioned as college-age "computer science types." (Kenneth) People with this model focus on two features: strong technical skills and the lack of proper moral restraint. Strong technical skills provide the motivation; hackers do it "for sheer sport" (Lorna) or to demonstrate technical prowess (Hayley). Lack of moral restraint is what makes them different than others with technical skills; hackers

are sometimes described as people as maladjusted individuals who “want to hurt others for no reason.” (Dana) Respondents will describe hackers as “miserable” people. They feel that hackers do what they do for no good reason, or at least no reason they can understand. Hackers are considered lone individuals.

Professional Criminal Hackers are criminals that happen to use a computer to commit crimes. Other than the computer as a method of operation, they are similar to most other criminals: They are motivated by some sort of financial gain; they can do what they do because they lack morals. Frequently this is associated with some form of ID theft. Sometimes but not always hackers can be part of some criminal group.

Anyone Hackers could be anyone. Often the respondent hasn’t thought a lot about who hackers are, but doesn’t have pre-conceived notions. They could be any kind of person. Sometimes they are described as having some amount of computer skill, but that is all they describe.

B.1.2.2 Behavior: What Do Hackers Do?

Hackers undertake many different activities as part of their hacking. Different respondents had very different opinions about what it was that hackers did.

Looking for Secrets Hackers break into computers to look for secrets. For example, hackers may be looking for information for industrial or governmental espionage purposes. (Christine) These are secrets that the hackers are not supposed to have. It is unclear why hackers would want this information; respondents don’t really think about how this information benefits the hackers (i.e. sold? used directly?). Subjects with this model feel that hackers would not target them because they do not have any secret information worth looking for.

Targeting Big Fish Hackers break into computers to look for information that can be directly used for financial gain. Looking for bank information like account numbers and passwords is common. Looking for credit card information is also common. Hackers specifically target the computers of people who have lots of money (the ‘big fish’), and the respondent feels that they don’t have enough money to be a target. “Maybe if I had a lot of money” (Floyd) they would be a target. Another thing that they might be looking for is usernames / passwords to places like Amazon which they can use to spend your money. This doesn’t cause harm to the computer.

Targeting Databases Hackers are looking for information that can be directly used for financial gain like credit card numbers or bank account information. However, the best place to go to get this is large databases of this information. Therefore, hackers go after places like Amazon.com or hacking into banks. Since each respondent has only a single person's (or single family's) worth of financial information on their computer, it is not worth breaking into their computer. This doesn't cause harm to the computer.

Random Financial Theft Hackers are looking for financial information. However, they either don't target specific individuals or they do so but poorly. One respondent talks about how he might be a victim "by accident." People with this model are concerned about their computer being broken into by a hacker looking for financial information, and take steps to prevent it. This doesn't cause harm to the computer.

Break stuff Hackers are mischievous individuals who like to cause havoc. They intentionally break things on your computer and put viruses on your computer. They might use your computer to send malicious emails or to use your IM client to send taunting messages to your friends. This frequently causes harm to the computer. There is no targeting here; hackers just break into whatever computer they can find and spread havoc. Because the hacking is random and causes damage, it is worth trying to prevent the attacks that cause lots of damage. (One subject talked about using his IM, and it wasn't bothering him enough to stop it.)

B.1.2.3 Prevention and Response

There are many way to deal with the hacker threat, and different users felt different needs to deal with it.

Trust in the Software For some people, an "anti-virus" program is really a catch-all security solution. it protects against hackers. People who believe this often put their trust in this software to protect them from attacks, or at least notify them of attacks. Others call this software a "firewall," but the effect is the same.

Passwords Passwords are an essential security tool to help protect against hackers. Passwords can protect access to important websites like bank accounts and Amazon.com. Regularly changing passwords is important so the hackers can't guess them. If you have a problem, you should change your passwords for protection.

Care on the Internet Hackers can find out about you from various websites on the Internet, so it is important to be careful. "All they need is an email address" to break into your computer. Don't distribute your information over the internet, whether that is email address or personal info (name, postal address, etc.). Don't go to useless websites. Some people take this so far as to always log out of websites they visit and to delete cookies regularly.

Hackers are like Viruses Hackers get into your computer much like viruses do. Therefore, don't download applications if you don't trust the source, and don't open suspicious emails.

Keep Targeted Information Off Computer Hackers break into computers to look for specific information like financial information. Therefore, you reduce the temptation for hackers to break in if you don't ever put that kind of sensitive information on your computer. Don't put your credit card number or social security number into the computer at all, and the hacker won't want to break in to get it.

Futility Hackers can always get in if they really want to; there is no way to stop them. "If they are going to get in, they're going to get in." (Hayley) There isn't anything you can do to stop a dedicated hacker, except maybe turning off the computer.

Trust in Institutions Trust large institutions to get security right. For example, you can give your information to your bank because your bank is a large institution that has a strong reason to be secure. Respondents with this belief don't really know how the security works, and don't care. They just know that the institutions will get it right.

B.1.3 Relationships

This section looks at the relationships among the major concepts.

B.1.3.1 Viruses and Hackers

This section looks at the relationships between the respondents' idea of "hacker" and their idea of "virus".

Viruses are Tools of Hackers Viruses are intentionally created by hackers, and serve to further the goals of the hacker. For the most part, there isn't much distinction here; hackers and viruses both can break into computers. Hackers make viruses, and viruses serve the

purposes of hackers. Often hackers will install viruses onto people's computers. Though not all software that a hacker installs is necessarily a virus.

Completely Separate Hackers and viruses are completely separate entities. Viruses may be created by people, or may just happen, but being hacked and getting a virus are very different. Hackers break in and do stuff on your computer, and then leave. (Lorna even thinks that hackers would need to be in front of a computer to run things, so she is confused how hackers work.) Viruses "infect" your computer and get into things. They stick around longer and can cause more problems. Often when faced with a virus with a known creator, the respondent really interprets it as a hacker tool rather than a virus. Also, Irving believes that viruses have different agency, and don't necessarily work at the behest of a hacker. "The hacker is an individual hacking, while the virus is a program infecting. So it's a difference between something automatic and more personal." Hackers intentionally do some actions on a computer to further their own goals. Where viruses are programs that automatically do actions.

Haven't Thought About It A number of respondents indicate that they had never thought about the relationship between viruses and hackers before. When prompted about how or why viruses are created, they might respond "Haven't the faintest idea" (Kenneth). They start out thinking they are completely separate. Usually, though, this changes throughout the interview; respondents realize that hackers are an obvious potential creator of viruses, and then state so. They then go on to describe a slightly different model of the relationship that is less independent. This realization is often described as a guess; for example Dana says "I guess if they hack into your system and get a virus on there, its gonna be the same thing."

B.2 Folk Models of the 33 Participants

Table 3.3 shows which models each of the 10 participants in Round 2 use. Below, I provide a table that shows the models that each of the 23 participants from Round 1.

	Alice	Bob	Carol	Deborah	Eve	Fred	Gina	Heather	Ivan	Justin	Karen	Laura	Nicole	Olivia	Peggy	Quinn	Robert	Sarah	Trudy	Valerie	William	Zoe	Mallory
Viruses																							
Viruses are Bad																							
Buggy Software	x	x	x	x	x	x	?	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Mischief		x					?	x															
Support Crime									x											x			
Hackers																							
Graffiti		x				?	?	?	x	?										x			
Burglar	x	x	x	x	x	?	x	?	x	?	x	x	x	x	x	x	x	x	x	x	x	x	x
Big Fish			x			?	?	?					x	x	x	x	x	x					
Contractor																							

Table B.1 A sample data matrix from near the end of the analysis. This matrix shows which folk model was held by the Participants in Round 1. The question marks indicate insufficient data to distinguish. A similar table for the participants in Round 2 is Table 3.3.

Appendix C

Full Results of Logistic Regression

```
[[1]]
A List Apart: Articles: Alternative Style: Working With Alternate Style Sheets
http://www.alistapart.com/stories/alternate/
0ce30dff2f6a9e9c6c753d4946f538a9
Total Number of Users of Site: 395
Number of Users in Fit: 395
Type of Data: bysite
```

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.46973	0.15626	-41.4039	< 2e-16 ***
used.onSiteTRUE	-0.16655	0.18286	-0.9108	0.36239
used.byUserTRUE	3.76397	0.20516	18.3461	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.57796	0.22498	-2.5689	0.01020 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classcss	6.14489	0.18985	32.3673	< 2.2e-16 ***
tag_classwebdesign	4.11294	0.17730	23.1979	< 2.2e-16 ***
tag_classsystem:unfiled	3.78741	0.24457	15.4861	< 2.2e-16 ***
tag_classjavascript	3.76703	0.17579	21.4294	< 2.2e-16 ***
tag_classwebdev	2.98115	0.22628	13.1747	< 2.2e-16 ***
tag_classdesign	2.73236	0.18983	14.3939	< 2.2e-16 ***
tag_classstylesheet	2.67296	0.34518	7.7437	9.659e-15 ***
tag_classswitcher	2.56177	0.36140	7.0886	1.355e-12 ***
tag_classhtml	2.48451	0.21542	11.5334	< 2.2e-16 ***
tag_classalternate	2.44257	0.39081	6.2501	4.102e-10 ***
tag_classweb	2.34598	0.20563	11.4089	< 2.2e-16 ***
tag_classstyle	2.32592	0.34946	6.6557	2.820e-11 ***
tag_classstylesheets	2.29659	0.39859	5.7618	8.325e-09 ***
tag_classstyleswitcher	2.09305	0.44563	4.6968	2.642e-06 ***
tag_classtutorial	1.76898	0.23866	7.4120	1.244e-13 ***
tag_classshowto	1.54644	0.26882	5.7528	8.777e-09 ***
tag_classutorials	1.44059	0.30985	4.6493	3.330e-06 ***
tag_classaccessibility	1.41654	0.36668	3.8631	0.0001119 ***
tag_classxhtml	1.25694	0.35172	3.5737	0.0003519 ***

tag_classprogramming	1.08749	0.28725	3.7858	0.0001532	***
tag_classdevelopment	1.03605	0.34107	3.0377	0.0023842	**
tag_classtech	0.93826	0.43632	2.1504	0.0315259	*
tag_classreference	0.90484	0.27729	3.2631	0.0011020	**
tag_classcode	0.86912	0.42409	2.0494	0.0404240	*
tag_classarticle	0.74539	0.45798	1.6276	0.1036131	
tag_classweb2.0	0.70087	0.33780	2.0748	0.0380043	*
tag_classblogs	0.49731	0.42830	1.1611	0.2455835	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.54896 +- 0.74091

Goodness of Fit Tests:

Residual Deviance: 5517 on 5.29e+04 degrees of freedom (p-value: 1)
AIC: 5581
Gm: 6019 on 33 degrees of freedom (p:value 0)
R^2_L: 0.5218
R^2: 0.377

Predictive Power Tests

Lambda_p: 0.2322 (d= 8.172 ; p-value: 3.03e-16) (for prediction models)
Tau_p: 0.6071 (d= 30.21 ; p-value: 1.966e-200) (for classification models)
Phi_p: 0.5066 (d= 22.38 ; p-value: 5.827e-111) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	51508	212
TRUE	717	493

Fitted Probabilities:

	css	webdesign	system:unfiled	javascript	webdev
Never Used	0.4194954	0.08652695	0.06402465	0.06281392	0.02963893
Used Only On Site	0.3795651	0.07423755	0.05473973	0.05369451	0.02520637
Used Only By User	0.9689050	0.80331876	0.74680502	0.74293127	0.56841463
Recommended Tag	0.9367082	0.65985826	0.58349555	0.57853334	0.38482484
	design	stylesheet	switcher	html	alternate
Never Used	0.02326253	0.02195032	0.01968605	0.01824908	0.01751264
Used Only On Site	0.01976415	0.01864551	0.01671632	0.01549270	0.01486580
Used Only By User	0.50664739	0.49179735	0.46406276	0.44491017	0.43457733
Recommended Tag	0.32785262	0.31489818	0.29141909	0.27572601	0.26742938
	web	style	stylesheets	styleswitcher	tutorial
Never Used	0.01592598	0.01561460	0.01517011	0.01241100	0.00900657
Used Only On Site	0.01351565	0.01325076	0.01287268	0.01052696	0.00763534
Used Only By User	0.41101174	0.40616404	0.39910950	0.35143937	0.28154953
Recommended Tag	0.24893712	0.24520518	0.23981729	0.20469203	0.15692434
	howto	tutorials	accessibility	xhtml	programming
Never Used	0.007222615	0.006501849	0.006348313	0.00541695	0.004576437
Used Only On Site	0.006121311	0.005509837	0.005379599	0.00458970	0.003877046
Used Only By User	0.238790117	0.220083415	0.215982746	0.19018238	0.165441882
Recommended Tag	0.129675755	0.118189813	0.115706002	0.10035105	0.086054746
	development	tech	reference	code	article
Never Used	0.004347976	0.003944520	0.003815361	0.003681995	0.003254879
Used Only On Site	0.003683371	0.003341377	0.003231904	0.003118868	0.002756895
Used Only By User	0.158461230	0.145852181	0.141737647	0.137448379	0.123426507

Recommended Tag	0.082094253	0.075020151	0.072733657	0.070361396	0.062686057
	web2.0	blogs	Other		
Never Used	0.003113565	0.002541588	0.001547240		
Used Only On Site	0.002637144	0.002152499	0.001310175		
Used Only By User	0.118689059	0.098993698	0.062633799		
Recommended Tag	0.060120095	0.049596770	0.030760703		

[[2]]

London Underground History - Disused Stations on London's Underground

<http://underground-history.co.uk/front.php>

1b33100ce08e52df31cbf3ad4c7ed801

Total Number of Users of Site: 369

Number of Users in Fit: 369

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-6.295967	0.136206	-46.2238	< 2e-16	***
used.onSiteTRUE	-0.336526	0.167991	-2.0032	0.04515	*
used.byUserTRUE	3.225885	0.179328	17.9887	< 2e-16	***
used.onSiteTRUE:used.byUserTRUE	-0.050057	0.203441	-0.2461	0.80564	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)	
tag_classlondon	6.33395	0.16855	37.5793	< 2.2e-16	***
tag_classunderground	6.20902	0.16603	37.3972	< 2.2e-16	***
tag_classtube	5.23681	0.17420	30.0618	< 2.2e-16	***
tag_classabandoned	5.12837	0.17970	28.5379	< 2.2e-16	***
tag_classhistory	4.71467	0.17685	26.6590	< 2.2e-16	***
tag_classsystem:unfiled	4.08762	0.22483	18.1808	< 2.2e-16	***
tag_classuk	3.93275	0.20095	19.5708	< 2.2e-16	***
tag_classtravel	3.35706	0.18880	17.7807	< 2.2e-16	***
tag_classsubway	3.13183	0.26311	11.9029	< 2.2e-16	***
tag_classurban	2.75051	0.24560	11.1992	< 2.2e-16	***
tag_classarchitecture	2.69069	0.21681	12.4102	< 2.2e-16	***
tag_classstations	2.50327	0.38223	6.5492	5.785e-11	***
tag_classtrain	2.30739	0.35536	6.4932	8.405e-11	***
tag_classcities	2.22978	0.34183	6.5231	6.888e-11	***
tag_classdisused	2.19067	0.44572	4.9149	8.884e-07	***
tag_classphotography	2.17302	0.21243	10.2295	< 2.2e-16	***
tag_classtransportation	2.15974	0.36868	5.8580	4.685e-09	***
tag_classphotos	2.12492	0.22885	9.2852	< 2.2e-16	***
tag_classengland	2.10894	0.37569	5.6135	1.983e-08	***
tag_classexploration	2.07942	0.40487	5.1361	2.806e-07	***
tag_classtrains	2.07387	0.37684	5.5033	3.727e-08	***
tag_classtransit	1.89631	0.45402	4.1767	2.958e-05	***
tag_classcool	1.77426	0.26414	6.7171	1.854e-11	***
tag_classreference	1.43518	0.23029	6.2321	4.603e-10	***
tag_classtransport	1.42647	0.46084	3.0953	0.0019659	**
tag_classfun	1.10763	0.28603	3.8724	0.0001078	***
tag_classinteresting	0.44898	0.45050	0.9966	0.3189497	
tag_classnetwork	0.34922	0.44939	0.7771	0.4370954	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.54614 +- 0.73902

Goodness of Fit Tests:

Residual Deviance: 6846 on 5.79e+04 degrees of freedom (p-value: 1)

AIC: 6912

Gm: 6638 on 34 degrees of freedom (p:value 0)

R²_L: 0.4923

R²: 0.3105

Predictive Power Tests

Lambda_p: 0.1049 (d= 4.03 ; p-value: 5.587e-05) (for prediction models)

Tau_p: 0.541 (d= 29.39 ; p-value: 6.857e-190) (for classification models)

Phi_p: 0.2951 (d= 12.82 ; p-value: 1.203e-37) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	56364	129
TRUE	1160	280

Fitted Probabilities:

	london	underground	tube	abandoned	history
Never Used	0.5094940	0.4782770	0.2574703	0.2372905	0.1706115
Used Only On Site	0.4259131	0.3956860	0.1985018	0.1818121	0.1281042
Used Only By User	0.9631682	0.9584706	0.8972215	0.8867829	0.8381579
Recommended Tag	0.9467119	0.9400462	0.8557145	0.8418036	0.7786819

	system:unfiled	uk	travel	subway	urban
Never Used	0.0990032	0.08602065	0.05026330	0.04053768	0.02804611
Used Only On Site	0.0727715	0.06298830	0.03642357	0.02929325	0.02019369
Used Only By User	0.7344922	0.70321701	0.57125550	0.51543106	0.42077969
Recommended Tag	0.6527053	0.61682200	0.47511855	0.41949675	0.33044856

	architecture	stations	train	cities	disused
Never Used	0.02646067	0.02203818	0.01818907	0.01685378	0.01621778
Used Only On Site	0.01904346	0.01584046	0.01305937	0.01209602	0.01163743
Used Only By User	0.40627317	0.36197339	0.31806191	0.30147204	0.29329974
Recommended Tag	0.31734999	0.27820302	0.24062129	0.22672772	0.21994361

	photography	transportation	photos	england	exploration
Never Used	0.01593863	0.01573154	0.01520146	0.01496408	0.01453516
Used Only On Site	0.01143621	0.01128694	0.01090497	0.01073395	0.01042500
Used Only By User	0.28965560	0.28692897	0.27985912	0.27664990	0.27078218
Recommended Tag	0.21693112	0.21468218	0.20887103	0.20624271	0.20145243

	trains	transit	cool	reference	transport
Never Used	0.01445578	0.012132590	0.010753549	0.007684838	0.007618711
Used Only On Site	0.01036783	0.008695817	0.007704367	0.005500954	0.005453516
Used Only By User	0.26968634	0.236174529	0.214868893	0.163159327	0.161973749
Recommended Tag	0.20056000	0.173596181	0.156777117	0.116965011	0.116068543

	fun	interesting	network	Other
Never Used	0.005550303	0.002880274	0.002607519	0.001840333
Used Only On Site	0.003970586	0.002058923	0.001863802	0.001315144
Used Only By User	0.123201930	0.067792724	0.061753617	0.044358359
Recommended Tag	0.087142468	0.047079950	0.042801260	0.030570706

[[3]]

Haiku | Desktop Operating System

http://haiku-os.org/
 2c4bd41cac0fb47fd81417063e2601aa
 Total Number of Users of Site: 161
 Number of Users in Fit: 161
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.27809	0.22554	-23.4023	< 2.2e-16 ***
used.onSiteTRUE	-0.71276	0.28417	-2.5082	0.01213 *
used.byUserTRUE	2.36807	0.31817	7.4428	9.855e-14 ***
used.onSiteTRUE:used.byUserTRUE	0.85931	0.36375	2.3623	0.01816 *

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classbeos	6.11850	0.28254	21.6552	< 2.2e-16 ***
tag_classos	5.96133	0.29315	20.3352	< 2.2e-16 ***
tag_classhaiku	5.35156	0.28567	18.7332	< 2.2e-16 ***
tag_classopensource	4.00121	0.29886	13.3881	< 2.2e-16 ***
tag_classopen-source	3.35136	0.35751	9.3740	< 2.2e-16 ***
tag_classdesktop	2.94076	0.33064	8.8941	< 2.2e-16 ***
tag_classsoftware	2.49620	0.30115	8.2890	< 2.2e-16 ***
tag_classsystem:unfiled	2.25063	0.47923	4.6963	2.649e-06 ***
tag_classgeek	1.44592	0.40107	3.6052	0.0003119 ***
tag_classfree	1.34736	0.36034	3.7391	0.0001847 ***
tag_classsystem	1.33802	0.51131	2.6169	0.0088745 **
tag_classstech	1.10165	0.42988	2.5627	0.0103855 *
tag_classunix	0.78757	0.45223	1.7415	0.0815879 .
tag_classfreeware	0.45829	0.47834	0.9581	0.3380234

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.73365 +- 0.85653

Goodness of Fit Tests:

Residual Deviance: 2033 on 8674 degrees of freedom (p-value: 1)
 AIC: 2071
 Gm: 2100 on 20 degrees of freedom (p-value 0)
 R²_L: 0.5081
 R²: 0.4123

Predictive Power Tests

Lambda_p: 0.2086 (d= 5.085 ; p-value: 3.681e-07) (for prediction models)
 Tau_p: 0.5773 (d= 19.85 ; p-value: 1.082e-87) (for classification models)
 Phi_p: 0.4922 (d= 15.27 ; p-value: 1.233e-52) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	8018	120
TRUE	320	236

Fitted Probabilities:

beos os haiku opensource open-source

Never Used	0.6985516	0.6644615	0.5183598	0.2180813	0.12711277
Used Only On Site	0.5318693	0.4926206	0.3454076	0.1202947	0.06663965
Used Only By User	0.9611521	0.9548390	0.9199407	0.7486051	0.60857771
Recommended Tag	0.9662691	0.9607535	0.9300911	0.7751693	0.64287968
	desktop	software	system:unfiled	geek	free
Never Used	0.08807861	0.05831058	0.04620088	0.02120333	0.01925154
Used Only On Site	0.04521385	0.02946479	0.02319809	0.01050934	0.00953236
Used Only By User	0.50768571	0.39799590	0.34087781	0.18784159	0.17326592
Recommended Tag	0.54420739	0.43357612	0.37452856	0.21122633	0.19527240
	system	tech	unix	freeware	Other
Never Used	0.019075880	0.015120997	0.011090464	0.008003836	0.005076276
Used Only On Site	0.009444528	0.007471244	0.005468447	0.003940269	0.002495299
Used Only By User	0.171931311	0.140835831	0.106934072	0.079312202	0.051660510
Recommended Tag	0.193808019	0.159518118	0.121756508	0.090694717	0.059330316

[[4]]

affiliates homepage | Spread Firefox
<http://www.spreadfirefox.com/?q=affiliates/homepage>
 34d1d5e34c5a7665d8be3f10ba08e82b
 Total Number of Users of Site: 214
 Number of Users in Fit: 214
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.58732	0.16035	-34.8441	< 2.2e-16 ***
used.onSiteTRUE	-1.09860	0.21409	-5.1314	2.876e-07 ***
used.byUserTRUE	2.82514	0.21869	12.9184	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	0.53306	0.26544	2.0082	0.04462 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classfirefox	6.26484	0.23792	26.3317	< 2.2e-16 ***
tag_classbuttons	4.97129	0.24542	20.2561	< 2.2e-16 ***
tag_classsystem:unfiled	4.43079	0.28025	15.8100	< 2.2e-16 ***
tag_classmozilla	3.96183	0.26909	14.7230	< 2.2e-16 ***
tag_classbrowser	3.59097	0.28083	12.7871	< 2.2e-16 ***
tag_classaffiliates	2.82902	0.43328	6.5292	6.611e-11 ***
tag_classresources	2.75852	0.34783	7.9306	2.182e-15 ***
tag_classbutton	2.65387	0.39749	6.6765	2.447e-11 ***
tag_classbanner	2.58742	0.43615	5.9324	2.985e-09 ***
tag_classmarketing	2.57353	0.36982	6.9588	3.431e-12 ***
tag_classinternet	2.22449	0.32144	6.9203	4.507e-12 ***
tag_classthunderbird	2.22344	0.40739	5.4578	4.820e-08 ***
tag_classlogo	1.90900	0.43237	4.4152	1.009e-05 ***
tag_classweb	1.89287	0.31025	6.1010	1.054e-09 ***
tag_classwebdev	1.47979	0.48438	3.0550	0.002250 **
tag_classicons	1.38447	0.44806	3.0899	0.002002 **
tag_classcommunity	1.07780	0.48536	2.2206	0.026378 *
tag_classopensource	0.66679	0.43613	1.5289	0.126294

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.72185 +- 0.84962

Goodness of Fit Tests:

Residual Deviance: 2945 on 2.373e+04 degrees of freedom (p-value: 1)
AIC: 2991
Gm: 2277 on 24 degrees of freedom (p:value 0)
R^2_L: 0.436
R^2: 0.279

Predictive Power Tests

Lambda_p: 0.1293 (d= 3.066 ; p-value: 0.002171) (for prediction models)
Tau_p: 0.5544 (d= 18.58 ; p-value: 4.642e-77) (for classification models)
Phi_p: 0.3341 (d= 9.089 ; p-value: 9.953e-20) (for selection models)

Actual vs. Predicted Values:

Table with 2 columns: FALSE, TRUE. Rows: FALSE (23152, 53), TRUE (425, 124)

Fitted Probabilities:

Large table of fitted probabilities for various categories like firefox, mozilla, browser, affiliates, resources, button, banner, marketing, internet, thunderbird, logo, web, webdev, icons, community, opensource, Other.

[[5]]

PayPalSucks.com is where you will learn about the PayPal Class Action Lawsuit, Abuse, Fraud & evil behind the PayPal system!

http://www.paypalsucks.com/
36a351c9102416c3e36c99ed564ceaa5
Total Number of Users of Site: 122
Number of Users in Fit: 121
Type of Data: bysite

Main Effects:

Table with 5 columns: Estimate, Std. Error, z value, Pr(>|z|). Rows: (Intercept), used.onSiteTRUE, used.byUserTRUE.

used.onSiteTRUE:used.byUserTRUE -0.14750 0.36970 -0.3990 0.6899

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classpaypal	5.43211	0.27399	19.8263	< 2.2e-16 ***
tag_classmoney	3.86593	0.30218	12.7934	< 2.2e-16 ***
tag_classsystem:unfiled	3.75990	0.31770	11.8349	< 2.2e-16 ***
tag_classsecurity	3.33207	0.29379	11.3417	< 2.2e-16 ***
tag_classebay	3.05844	0.31796	9.6191	< 2.2e-16 ***
tag_classprivacy	3.00494	0.31556	9.5226	< 2.2e-16 ***
tag_classfinance	2.89634	0.33115	8.7464	< 2.2e-16 ***
tag_classonline	2.74391	0.34964	7.8477	4.237e-15 ***
tag_classinternet	1.20634	0.39303	3.0693	0.002145 **
tag_classshopping	0.80203	0.46129	1.7387	0.082095 .
tag_classweb	0.52571	0.42621	1.2334	0.217408

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.84753 +- 0.92061

Goodness of Fit Tests:

Residual Deviance: 1600 on 7122 degrees of freedom (p-value: 1)

AIC: 1632

Gm: 1146 on 17 degrees of freedom (p:value 5.004e-233)

R^2_L: 0.4174

R^2: 0.2699

Predictive Power Tests

Lambda_p: 0.07602 (d= 1.441 ; p-value: 0.1496) (for prediction models)

Tau_p: 0.5148 (d= 13.78 ; p-value: 3.363e-43) (for classification models)

Phi_p: 0.3649 (d= 8.437 ; p-value: 3.246e-17) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	6724	73
TRUE	243	99

Fitted Probabilities:

	paypal	money	system:unfiled	security	ebay
Never Used	0.5628436	0.2119072	0.1947404	0.13618646	0.10707651
Used Only On Site	0.4611837	0.1516451	0.1385022	0.09486573	0.07383318
Used Only By User	0.9674646	0.8613052	0.8481473	0.78453767	0.73471600
Recommended Tag	0.9446199	0.7808087	0.7621232	0.67623448	0.61370071
	privacy	finance	online	internet	shopping
Never Used	0.10206703	0.09253549	0.08050590	0.01846829	0.012402607
Used Only On Site	0.07025659	0.06348564	0.05500362	0.01235395	0.008279512
Used Only By User	0.72415667	0.70194526	0.66910663	0.30292264	0.224831709
Recommended Tag	0.60094078	0.57463565	0.53702189	0.19953437	0.142642174
	web	Other			
Never Used	0.009436491	0.005599839			
Used Only On Site	0.006293161	0.003729700			
Used Only By User	0.180339450	0.115091034			
Recommended Tag	0.112063204	0.069425396			

[[6]]

OS X Maintenance And Troubleshooting

http://www.macattorney.com/ts.html

3969c4fca4fla80d89a3c024609731c3

Total Number of Users of Site: 282

Number of Users in Fit: 282

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.88907	0.16840	-34.9699	< 2.2e-16 ***
used.onSiteTRUE	-0.55956	0.19733	-2.8357	0.004573 **
used.byUserTRUE	3.10638	0.23238	13.3678	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.15095	0.25453	-0.5931	0.553137

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classmaintenance	5.50074	0.19121	28.7680	< 2.2e-16 ***
tag_classosx	5.02462	0.19533	25.7244	< 2.2e-16 ***
tag_classmac	4.68803	0.19535	23.9977	< 2.2e-16 ***
tag_classtroubleshooting	4.58733	0.21253	21.5846	< 2.2e-16 ***
tag_classmacosx	4.12309	0.22970	17.9501	< 2.2e-16 ***
tag_classsystem:unfiled	3.39128	0.30449	11.1376	< 2.2e-16 ***
tag_classapple	3.13999	0.21914	14.3286	< 2.2e-16 ***
tag_classhelp	2.59666	0.32946	7.8817	3.230e-15 ***
tag_classreference	2.51407	0.21440	11.7262	< 2.2e-16 ***
tag_classmacintosh	2.26490	0.35983	6.2943	3.087e-10 ***
tag_classsafari_export	2.12356	0.42486	4.9983	5.784e-07 ***
tag_classadmin	2.05456	0.43415	4.7323	2.220e-06 ***
tag_classshowto	2.02279	0.26291	7.6939	1.427e-14 ***
tag_classrepair	1.90270	0.40825	4.6606	3.153e-06 ***
tag_classsysadmin	1.67504	0.43140	3.8828	0.0001032 ***
tag_classstools	1.66413	0.26826	6.2034	5.525e-10 ***
tag_classsoftware	1.29758	0.27505	4.7175	2.387e-06 ***
tag_classsupport	1.18868	0.47128	2.5222	0.0116609 *
tag_classtutorial	1.15284	0.30899	3.7310	0.0001907 ***
tag_classstips	1.04985	0.33915	3.0955	0.0019647 **
tag_classutilities	1.01392	0.45710	2.2181	0.0265447 *
tag_classstech	0.72929	0.43298	1.6843	0.0921158 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.54262 +- 0.73663

Goodness of Fit Tests:

Residual Deviance: 4244 on 2.817e+04 degrees of freedom (p-value: 1)

AIC: 4298

Gm: 3686 on 28 degrees of freedom (p:value 0)

R²_L: 0.4648

R²: 0.3192

Predictive Power Tests

Lambda_p: 0.1935 (d= 5.88 ; p-value: 4.104e-09) (for prediction models)
 Tau_p: 0.5836 (d= 25.06 ; p-value: 1.27e-138) (for classification models)
 Phi_p: 0.4227 (d= 15.28 ; p-value: 1.014e-52) (for selection models)
 Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	27203	103
TRUE	618	276

Fitted Probabilities:

	maintenance	osx	mac troubleshooting	macosx
Never Used	0.4041204	0.2964111	0.2312908	0.2138726
Used Only On Site	0.2793103	0.1940342	0.1467156	0.1345518
Used Only By User	0.9380837	0.9039525	0.8704952	0.8587131
Recommended Tag	0.8815869	0.8222144	0.7676043	0.7491589

	system:unfiled	apple	help reference	macintosh
Never Used	0.07601331	0.06013902	0.03583265	0.03308595
Used Only On Site	0.04490126	0.03527622	0.02079629	0.01917925
Used Only By User	0.64761944	0.58838855	0.45362752	0.43324585
Recommended Tag	0.47454218	0.41260586	0.28976299	0.27306421

	safari_export	admin	howto	repair	sysadmin
Never Used	0.02263178	0.02115477	0.02050680	0.01822853	0.014571272
Used Only On Site	0.01305983	0.01219972	0.01182273	0.01049890	0.008379231
Used Only By User	0.34093558	0.32560579	0.31866845	0.29317988	0.248310357
Recommended Tag	0.20267824	0.19175630	0.18688049	0.16931346	0.139655401

	tools	software	support	tutorial	tips
Never Used	0.014415421	0.010036002	0.009009858	0.008695364	0.007851134
Used Only On Site	0.008289051	0.005759953	0.005168736	0.004987644	0.004501760
Used Only By User	0.246279288	0.184656979	0.168821230	0.163850758	0.150225387
Recommended Tag	0.138349505	0.100144514	0.090749681	0.087834905	0.079926557

	utilities	tech	Other
Never Used	0.007576162	0.005710198	0.002761909
Used Only On Site	0.004343580	0.003271158	0.001580193
Used Only By User	0.145696258	0.113709856	0.058266966
Recommended Tag	0.077324005	0.059306123	0.029506405

[[7]]

The Library of Congress
<http://www.loc.gov/>
 3b3cb60120b337fa373871347e15418c
 Total Number of Users of Site: 552
 Number of Users in Fit: 552
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.607801	0.092068	-71.7707	< 2.2e-16 ***
used.onSiteTRUE	-0.407889	0.119043	-3.4264	0.0006117 ***
used.byUserTRUE	3.739711	0.126823	29.4877	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.311265	0.148882	-2.0907	0.0365561 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
--	----------	------------	---------	----------

tag_classlibrary	5.386373	0.138544	38.8784	< 2.2e-16	***
tag_classreference	4.759235	0.142109	33.4900	< 2.2e-16	***
tag_classgovernment	4.656189	0.155975	29.8521	< 2.2e-16	***
tag_classlibraries	4.539032	0.164330	27.6215	< 2.2e-16	***
tag_classbooks	3.953035	0.155860	25.3627	< 2.2e-16	***
tag_classsystem:unfiled	3.897335	0.213840	18.2255	< 2.2e-16	***
tag_classcongress	3.400048	0.244616	13.8995	< 2.2e-16	***
tag_classresearch	3.351144	0.182111	18.4016	< 2.2e-16	***
tag_classhistory	2.824518	0.190835	14.8008	< 2.2e-16	***
tag_classimported	2.520151	0.260905	9.6592	< 2.2e-16	***
tag_classloc	2.417560	0.380332	6.3565	2.065e-10	***
tag_classbibliotecas	2.373027	0.385721	6.1522	7.642e-10	***
tag_classsafari_export	2.356934	0.314418	7.4962	6.571e-14	***
tag_classinformation	2.214834	0.293254	7.5526	4.266e-14	***
tag_classus	2.025083	0.437862	4.6249	3.747e-06	***
tag_classresource	1.962854	0.363481	5.4002	6.658e-08	***
tag_classcatalog	1.893764	0.423137	4.4755	7.622e-06	***
tag_classresources	1.838724	0.321878	5.7125	1.113e-08	***
tag_classimages	1.824687	0.298556	6.1117	9.857e-10	***
tag_classusa	1.821653	0.421370	4.3232	1.538e-05	***
tag_classbook	1.668408	0.378327	4.4100	1.034e-05	***
tag_classinfo	1.540372	0.424366	3.6298	0.0002836	***
tag_classliterature	1.472426	0.297691	4.9462	7.569e-07	***
tag_classsearch	1.448244	0.249212	5.8113	6.199e-09	***
tag_classarchive	1.332586	0.396459	3.3612	0.0007760	***
tag_classeducation	1.290172	0.307190	4.1999	2.670e-05	***
tag_classlaw	1.288687	0.396700	3.2485	0.0011601	**
tag_classpolitics	0.737252	0.352485	2.0916	0.0364756	*
tag_classfree	0.650329	0.414642	1.5684	0.1167853	
tag_classstools	0.593553	0.347736	1.7069	0.0878390	.
tag_classmaps	0.409538	0.410787	0.9970	0.3187845	
tag_classphotography	0.350238	0.410876	0.8524	0.3939828	
tag_classart	-0.059074	0.409448	-0.1443	0.8852820	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.56286 +- 0.75024

Goodness of Fit Tests:

Residual Deviance: 9810 on 1.766e+05 degrees of freedom (p-value: 1)

AIC: 9886

Gm: 7882 on 39 degrees of freedom (p:value 0)

R²_L: 0.4455

R²: 0.2369

Predictive Power Tests

Lambda_p: 0.09857 (d= 3.888 ; p-value: 0.0001012) (for prediction models)

Tau_p: 0.5453 (d= 30.42 ; p-value: 3.436e-203) (for classification models)

Phi_p: 0.3728 (d= 17.66 ; p-value: 8.8e-70) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	174831	267
TRUE	1123	419

Fitted Probabilities:

	library	reference	government	libraries	books
Never Used	0.2276854	0.13604134	0.12437774	0.11216955	0.06569587
Used Only On Site	0.1639240	0.09479412	0.08631358	0.07751082	0.04467426
Used Only By User	0.9254137	0.86888598	0.85669407	0.84170131	0.74742861
Recommended Tag	0.8580429	0.76350464	0.74439627	0.72147452	0.59044151
	system:unfiled	congress	research	history	imported
Never Used	0.06235861	0.03887502	0.03708843	0.02224193	0.01650174
Used Only On Site	0.04235646	0.02619503	0.02497603	0.01490309	0.01103552
Used Only By User	0.73676950	0.62993972	0.61846884	0.48910870	0.41388216
Recommended Tag	0.57690748	0.45333724	0.44124774	0.31805470	0.25595620
	loc bibliotecas	safari_export	information	us	
Never Used	0.014916763	0.014276319	0.014051619	0.012212991	0.010123525
Used Only On Site	0.009970249	0.009540125	0.009389259	0.008155646	0.006755585
Used Only By User	0.389234806	0.378701546	0.374922652	0.342256157	0.300901745
Recommended Tag	0.236912143	0.228955727	0.226127235	0.202230738	0.173336704
	resource	catalog	resources	images	usa
Never Used	0.009518563	0.00888878	0.008416768	0.008300422	0.008275486
Used Only On Site	0.006350593	0.00592916	0.005613420	0.005535608	0.005518932
Used Only By User	0.287975581	0.27401903	0.263206980	0.260493888	0.259909881
Recommended Tag	0.164600255	0.15531870	0.148233931	0.146470365	0.146091490
	book	info	literature	search	archive
Never Used	0.007108058	0.006259173	0.005850415	0.005711438	0.005090807
Used Only On Site	0.004738515	0.004171425	0.003898474	0.003805688	0.003391438
Used Only By User	0.231531794	0.209537140	0.198505025	0.194685733	0.177189751
Recommended Tag	0.127991426	0.114368865	0.107664884	0.105363619	0.094948397
	education	law	politics	free	tools
Never Used	0.004880433	0.004873226	0.002813387	0.002579771	0.002437729
Used Only On Site	0.003251060	0.003246251	0.001872816	0.001717167	0.001622543
Used Only By User	0.171090512	0.170880017	0.106135457	0.098166846	0.093253876
Recommended Tag	0.091365728	0.091242522	0.054681728	0.050358603	0.047711714
	maps	photography	art	Other	
Never Used	0.002028835	0.001912245	0.0012707522	0.0013479778	
Used Only On Site	0.001350200	0.001272559	0.0008454771	0.0008968813	
Used Only By User	0.078815401	0.074616096	0.0508269695	0.0537537130	
Recommended Tag	0.040013358	0.037796610	0.0254237746	0.0269292398	

[[8]]

GDI+ FAQ main index
<http://www.bobpowell.net/faqmain.htm>
 3bda811161161074f8c0f7a003dba847
 Total Number of Users of Site: 114
 Number of Users in Fit: 114
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.15428	0.36548	-3.1582	0.001587 **
used.onSiteTRUE	-0.19863	0.28311	-0.7016	0.482919
used.byUserTRUE	3.52840	0.35602	9.9107	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.61131	0.38833	-1.5742	0.115439

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)	
tag_classgdi+	1.14270	0.32166	3.5525	0.0003816	***
tag_classgraphics	-0.17186	0.33898	-0.5070	0.6121552	
tag_classgdi	-0.66461	0.37519	-1.7714	0.0764944	.
tag_classprogramming	-0.84539	0.33972	-2.4885	0.0128287	*
tag_classc#	-1.02034	0.35375	-2.8844	0.0039217	**
tag_classfaq	-1.75308	0.45286	-3.8711	0.0001083	***
tag_classsystem:unfiled	-1.84245	0.51161	-3.6013	0.0003166	***
tag_classdevelopment	-2.11846	0.39639	-5.3444	9.070e-08	***
tag_classdotnet	-2.45021	0.47126	-5.1992	2.001e-07	***
tag_classcsharp	-2.47653	0.54196	-4.5696	4.886e-06	***
tag_classwinforms	-2.59124	0.52990	-4.8900	1.008e-06	***
tag_classreference	-3.20541	0.44072	-7.2731	3.513e-13	***
tag_classasp.net	-3.30208	0.47687	-6.9244	4.377e-12	***
tag_classwindows	-3.93512	0.52431	-7.5054	6.125e-14	***
tag_classOther	-4.02836	0.31756	-12.6853	< 2.2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.54605 +- 0.73895

Goodness of Fit Tests:

Residual Deviance: 1524 on 5907 degrees of freedom (p-value: 1)
AIC: 1564
Gm: 1299 on 21 degrees of freedom (p:value 3.695e-262)
R^2_L: 0.4602
R^2: 0.3606

Predictive Power Tests

Lambda_p: 0.1974 (d= 3.977 ; p-value: 6.979e-05) (for prediction models)
Tau_p: 0.5712 (d= 16.24 ; p-value: 2.668e-59) (for classification models)
Phi_p: 0.4524 (d= 11.22 ; p-value: 3.377e-29) (for selection models)
Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	5484	64
TRUE	241	139

Fitted Probabilities:

	gdi+	graphics	gdi programming	c#	
Never Used	0.4971055	0.2097977	0.1395675	0.11923749	
Used Only On Site	0.4476397	0.1787589	0.1173757	0.09990285	
Used Only By User	0.9711626	0.9004518	0.8467731	0.82182023	
Recommended Tag	0.9374312	0.8009607	0.7108608	0.67233920	
	faq	system:unfiled	development	dotnet	csharp
Never Used	0.05179096	0.04757388	0.03651835	0.02648103	0.02581084
Used Only On Site	0.04286073	0.03934062	0.03013785	0.02181455	0.02125988
Used Only By User	0.65045510	0.62987273	0.56356938	0.48098689	0.47441953
Recommended Tag	0.45291384	0.43087704	0.36487115	0.29192899	0.28651798
	winforms	reference	asp.net	windows	Other
Never Used	0.02307812	0.01262102	0.011471341	0.006123948	0.005581822
Used Only On Site	0.01899960	0.01037092	0.009424257	0.005026265	0.004580864
Used Only By User	0.44593172	0.30337219	0.283337494	0.173502345	0.160536313
Recommended Tag	0.26365318	0.16229701	0.149578480	0.085414849	0.078407201

Never Used 0.005581822
 Used Only On Site 0.004580864
 Used Only By User 0.160536313
 Recommended Tag 0.078407201

[[9]]

MetaGer - die MetaSuche ber deutschsprachige Suchmaschinen
 http://www.metager.de/
 42c34e4f93236add471037122e00a521
 Total Number of Users of Site: 174
 Number of Users in Fit: 174
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.20370	0.19785	-26.3009	< 2.2e-16 ***
used.onSiteTRUE	-0.49103	0.23038	-2.1313	0.0330615 *
used.byUserTRUE	4.77568	0.31859	14.9900	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-1.35148	0.37043	-3.6484	0.0002638 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classsuchmaschinen	4.63613	0.24474	18.9434	< 2.2e-16 ***
tag_classmeta	3.79541	0.27878	13.6142	< 2.2e-16 ***
tag_classsuche	3.64052	0.28776	12.6511	< 2.2e-16 ***
tag_classsearch	3.56368	0.28058	12.7009	< 2.2e-16 ***
tag_classsearchengine	3.45284	0.28446	12.1382	< 2.2e-16 ***
tag_classsuchmaschine	3.31375	0.30151	10.9906	< 2.2e-16 ***
tag_classsystem:unfiled	2.93242	0.35179	8.3358	< 2.2e-16 ***
tag_classsuchen	2.48689	0.36440	6.8247	8.813e-12 ***
tag_classdeutsch	2.39514	0.36268	6.6040	4.003e-11 ***
tag_classimported	1.51267	0.47133	3.2094	0.001330 **
tag_classweb	1.05000	0.41107	2.5543	0.010640 *
tag_classgerman	0.82394	0.45902	1.7950	0.072656 .
tag_classinternet	0.34008	0.49958	0.6807	0.496039
tag_classstools	0.19423	0.47006	0.4132	0.679456

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.29043 +- 0.53892

Goodness of Fit Tests:

Residual Deviance: 2017 on 1.216e+04 degrees of freedom (p-value: 1)
 AIC: 2055
 Gm: 1318 on 20 degrees of freedom (p-value 4.325e-267)
 R²_L: 0.3952
 R²: 0.2689

Predictive Power Tests

Lambda_p: 0.1367 (d= 2.682 ; p-value: 0.007317) (for prediction models)
 Tau_p: 0.5547 (d= 15.38 ; p-value: 2.192e-53) (for classification models)
 Phi_p: 0.3483 (d= 7.904 ; p-value: 2.694e-15) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	11768	39
TRUE	283	90

Fitted Probabilities:

	suchmaschinen	meta	suche	search	searchengine
Never Used	0.3617970	0.1965032	0.1731908	0.1624625	0.14793819
Used Only On Site	0.2575769	0.1301852	0.1136279	0.1061157	0.09605114
Used Only By User	0.9853435	0.9666695	0.9613019	0.9583399	0.95368260
Recommended Tag	0.9141662	0.8212556	0.7973789	0.7846805	0.76536251

	suchmaschine	system:unfiled	suchen	deutsch	imported
Never Used	0.13124962	0.09352898	0.06198852	0.05686341	0.02433896
Used Only On Site	0.08463443	0.05939489	0.03887173	0.03558538	0.01503739
Used Only By User	0.94713614	0.92444900	0.88684029	0.87730130	0.74737125
Recommended Tag	0.73947035	0.65968362	0.55387982	0.53111310	0.31911012

	web	german	internet	tools	Other
Never Used	0.015463275	0.012373246	0.007663300	0.006630182	0.005466128
Used Only On Site	0.009520603	0.007608921	0.004703905	0.004068118	0.003352365
Used Only By User	0.650667461	0.597704454	0.478028640	0.441816503	0.394597823
Recommended Tag	0.227843027	0.190526006	0.126701066	0.111421919	0.093593134

[[10]]

eHomeUpgrade | Connected Home & Digital Lifestyle News [Digital Home, Digital Living, Digital Media]
<http://www.ehomeupgrade.com/>
573188a35ec17e0876456fecf6981741
Total Number of Users of Site: 270
Number of Users in Fit: 270
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.03054	0.15082	-39.985	< 2e-16 ***
used.onSiteTRUE	-0.11528	0.17872	-0.645	0.51891
used.byUserTRUE	3.71170	0.19469	19.064	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.51837	0.21881	-2.369	0.01783 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classgadgets	4.43104	0.19369	22.8771	< 2.2e-16 ***
tag_classhome	4.01819	0.20399	19.6977	< 2.2e-16 ***
tag_classsystem:unfiled	3.96025	0.23011	17.2099	< 2.2e-16 ***
tag_classtechnology	3.74716	0.20098	18.6441	< 2.2e-16 ***
tag_classtech	3.43598	0.21043	16.3287	< 2.2e-16 ***
tag_classblog	2.77073	0.20348	13.6166	< 2.2e-16 ***
tag_classhttpc	2.43448	0.29289	8.3118	< 2.2e-16 ***
tag_classnews	2.30387	0.20748	11.1043	< 2.2e-16 ***
tag_classmce	2.27713	0.36972	6.1591	7.318e-10 ***
tag_classmedia	2.26215	0.25114	9.0075	< 2.2e-16 ***
tag_classautomation	2.06736	0.37398	5.5280	3.239e-08 ***
tag_classhomeautomation	2.04530	0.43330	4.7203	2.355e-06 ***
tag_classhometheater	1.86547	0.40079	4.6545	3.247e-06 ***

tag_classgadget	1.83066	0.37452	4.8880	1.018e-06	***
tag_classmediacenter	1.75848	0.43822	4.0127	6.002e-05	***
tag_classpvr	1.52235	0.37531	4.0563	4.986e-05	***
tag_classelectronics	1.47928	0.37508	3.9439	8.017e-05	***
tag_classdigital	1.42464	0.39862	3.5739	0.0003516	***
tag_classhardware	1.26129	0.31072	4.0592	4.924e-05	***
tag_classblogs	1.25817	0.34173	3.6818	0.0002316	***
tag_classdiy	1.20048	0.36879	3.2552	0.0011331	**
tag_classentertainment	0.98864	0.46659	2.1189	0.0341020	*
tag_classtv	0.97039	0.35030	2.7702	0.0056022	**
tag_classgeek	0.88242	0.40389	2.1848	0.0289023	*
tag_classdaily	0.85394	0.40711	2.0976	0.0359447	*
tag_classreviews	0.83831	0.40349	2.0777	0.0377404	*
tag_classmagazine	0.55576	0.45963	1.2092	0.2266043	
tag_classvideo	0.50515	0.35879	1.4079	0.1591596	
tag_classaudio	0.44507	0.42266	1.0530	0.2923227	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.36981 +- 0.60812

Goodness of Fit Tests:

Residual Deviance: 4812 on 3.912e+04 degrees of freedom (p-value: 1)
AIC: 4880
Gm: 3495 on 35 degrees of freedom (p:value 0)
R^2_L: 0.4207
R^2: 0.2396

Predictive Power Tests

Lambda_p: 0.08092 (d= 2.407 ; p-value: 0.01609) (for prediction models)
Tau_p: 0.5301 (d= 22.29 ; p-value: 4.647e-110) (for classification models)
Phi_p: 0.3063 (d= 10.52 ; p-value: 6.707e-26) (for selection models)
Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	38173	112
TRUE	683	182

Fitted Probabilities:

	gadgets	home	system:unfiled	technology	tech
Never Used	0.1680522	0.1179122	0.1120185	0.09250877	0.06948922
Used Only On Site	0.1525456	0.1064403	0.1010542	0.08327504	0.06239520
Used Only By User	0.8920838	0.8454497	0.8377271	0.80663937	0.75345749
Recommended Tag	0.8143539	0.7437777	0.7325817	0.68883266	0.61857070
	blog	htpc	news	mce	media
Never Used	0.03697603	0.02669933	0.02350708	0.02290089	0.02256807
Used Only On Site	0.03308321	0.02386163	0.02100134	0.02045841	0.02016035
Used Only By User	0.61108890	0.52887922	0.49625863	0.48957349	0.48583062
Recommended Tag	0.45468457	0.37331848	0.34330105	0.33729674	0.33395636
	automation	homeautomation	hometheater	gadget	mediacenter
Never Used	0.01864824	0.01824881	0.01529113	0.01477578	0.01376096
Used Only On Site	0.01665160	0.01629422	0.01364892	0.01318819	0.01228104
Used Only By User	0.43745952	0.43203875	0.38855904	0.38032251	0.36346341
Recommended Tag	0.29211562	0.28757518	0.25217882	0.24567167	0.23254154
	pvr	electronics	digital	hardware	blogs

Never Used	0.01089834	0.010443688	0.009893800	0.008415366	0.008389341
Used Only On Site	0.00972325	0.009317154	0.008826053	0.007505963	0.007482729
Used Only By User	0.31077767	0.301627712	0.290243082	0.257778912	0.257181738
Recommended Tag	0.19307713	0.186455052	0.178307606	0.155618482	0.155208484
	diy	entertainment	tv	geek	daily
Never Used	0.007922747	0.006419984	0.006304639	0.005776783	0.005615490
Used Only On Site	0.007066199	0.005724966	0.005622037	0.005151036	0.005007126
Used Only By User	0.246315080	0.209126466	0.206124733	0.192101163	0.187719779
Recommended Tag	0.147793223	0.123050877	0.121095470	0.112040384	0.109238103
	reviews	magazine	video	audio	Other
Never Used	0.005528866	0.004173654	0.003968498	0.003737974	0.002398431
Used Only On Site	0.004929840	0.003720910	0.003537929	0.003332333	0.002137843
Used Only By User	0.185347645	0.146404902	0.140192614	0.133106525	0.089574675
Recommended Tag	0.107726181	0.083422158	0.079633048	0.075339747	0.049618892

[[11]]

Getting started with SSH - Kimmo Suominen
<http://kimmo.suominen.com/docs/ssh/>
589f84ca5ce34697974199d47c1cfe2e
Total Number of Users of Site: 938
Number of Users in Fit: 938
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-7.5993403	0.1357615	-55.9757	< 2e-16 ***
used.onSiteTRUE	-0.0064323	0.1611449	-0.0399	0.96816
used.byUserTRUE	3.2889931	0.1688023	19.4843	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.3577204	0.1788841	-1.9997	0.04553 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

Tag	Estimate	Std. Error	z value	Pr(> z)
tag_classssh	8.153240	0.132930	61.3349	< 2.2e-16 ***
tag_classsecurity	5.073887	0.127821	39.6951	< 2.2e-16 ***
tag_classtutorial	5.048455	0.129716	38.9192	< 2.2e-16 ***
tag_classshowto	4.975413	0.129573	38.3985	< 2.2e-16 ***
tag_classlinux	4.759314	0.126333	37.6729	< 2.2e-16 ***
tag_classadmin	4.484838	0.163412	27.4449	< 2.2e-16 ***
tag_classunix	4.238420	0.140814	30.0995	< 2.2e-16 ***
tag_classsystem:unfiled	4.052004	0.203668	19.8952	< 2.2e-16 ***
tag_classsysadmin	3.409188	0.178405	19.1092	< 2.2e-16 ***
tag_classscp	3.368176	0.245830	13.7013	< 2.2e-16 ***
tag_classshell	2.833875	0.193685	14.6314	< 2.2e-16 ***
tag_classnetwork	2.833692	0.172470	16.4301	< 2.2e-16 ***
tag_class tutorials	2.809390	0.189182	14.8502	< 2.2e-16 ***
tag_class tips	2.758361	0.173238	15.9224	< 2.2e-16 ***
tag_classreference	2.704907	0.156545	17.2787	< 2.2e-16 ***
tag_class cryptography	2.494989	0.283006	8.8160	< 2.2e-16 ***
tag_class computer	2.224255	0.229118	9.7079	< 2.2e-16 ***
tag_class software	1.957742	0.185539	10.5516	< 2.2e-16 ***
tag_class commandline	1.899745	0.414961	4.5781	4.691e-06 ***
tag_class geek	1.859139	0.275055	6.7592	1.388e-11 ***
tag_class vpn	1.764431	0.318252	5.5441	2.954e-08 ***

tag_classencryption	1.700403	0.334053	5.0902	3.576e-07	***
tag_classtoread	1.616547	0.324144	4.9871	6.128e-07	***
tag_classnetworking	1.486998	0.272912	5.4486	5.076e-08	***
tag_classubuntu	1.436628	0.305759	4.6986	2.620e-06	***
tag_classmac	1.431957	0.273284	5.2398	1.607e-07	***
tag_classtech	1.423481	0.305854	4.6541	3.254e-06	***
tag_classprogramming	1.394459	0.223929	6.2272	4.748e-10	***
tag_classapps	1.360999	0.416777	3.2655	0.0010926	**
tag_classwork	1.339767	0.370584	3.6153	0.0003000	***
tag_classguide	1.051644	0.366112	2.8725	0.0040729	**
tag_classweb	1.025253	0.260390	3.9374	8.238e-05	***
tag_classwindows	1.003318	0.265864	3.7738	0.0001608	***
tag_classtechnology	0.317040	0.410045	0.7732	0.4394144	
tag_classtools	0.255412	0.346926	0.7362	0.4616009	
tag_classopensource	0.156767	0.385072	0.4071	0.6839266	
tag_classserver	0.058128	0.443073	0.1312	0.8956232	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.83246 +- 0.9124

Goodness of Fit Tests:

Residual Deviance: 1.414e+04 on 1.772e+05 degrees of freedom (p-value: 1)

AIC: 1.423e+04

Gm: 1.834e+04 on 43 degrees of freedom (p:value 0)

R²_L: 0.5645

R²: 0.3761

Predictive Power Tests

Lambda_p: 0.1625 (d= 9.357 ; p-value: 8.2e-21) (for prediction models)

Tau_p: 0.5734 (d= 46.69 ; p-value: 0) (for classification models)

Phi_p: 0.5605 (d= 44.94 ; p-value: 0) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	172760	1266
TRUE	1461	1795

Fitted Probabilities:

	ssh	security	tutorial	howto	linux
Never Used	0.6350399	0.07409293	0.07236703	0.06761427	0.05519914
Used Only On Site	0.6335478	0.07365286	0.07193642	0.06720989	0.05486464
Used Only By User	0.9790182	0.68212169	0.67658195	0.66039738	0.61039346
Recommended Tag	0.9700768	0.59854030	0.59241433	0.57466566	0.52119072
	admin	unix	system:unfiled	sysadmin	scp
Never Used	0.04251299	0.03353938	0.02799497	0.01491806	0.01432721
Used Only On Site	0.04225193	0.03333151	0.02782047	0.01482383	0.01423666
Used Only By User	0.54351233	0.48202598	0.43577106	0.28881242	0.28046204
Recommended Tag	0.45272612	0.39267547	0.34921403	0.22006089	0.21310268
	shell	network	tutorials	tips	reference
Never Used	0.008446968	0.008445429	0.008244336	0.007837405	0.007432495
Used Only On Site	0.008393263	0.008391734	0.008191909	0.007787546	0.007385192
Used Only By User	0.185960923	0.185933108	0.182282798	0.174799604	0.167222634
Recommended Tag	0.136977447	0.136955726	0.134108555	0.128292761	0.122432609
	cryptography	computer	software	commandline	geek

Never Used	0.006033652	0.004609179	0.003534657	0.003336154	0.003203824
Used Only On Site	0.005995198	0.004579762	0.003512074	0.003314834	0.003183347
Used Only By User	0.139991839	0.110455925	0.086858943	0.082367815	0.079350244
Recommended Tag	0.101605604	0.079420620	0.061991738	0.058703734	0.056499725
	vpn	encryption	toread	networking	ubuntu
Never Used	0.002915166	0.002734858	0.002515429	0.002210458	0.002102104
Used Only On Site	0.002896529	0.002717370	0.002499341	0.002196317	0.002088654
Used Only By User	0.072701327	0.068501168	0.063340180	0.056075383	0.053468126
Recommended Tag	0.051658065	0.048609905	0.044875359	0.039638758	0.037765156
	mac	tech	programming	apps	work
Never Used	0.002092327	0.002074705	0.002015478	0.001949285	0.001908413
Used Only On Site	0.002078940	0.002061430	0.002002581	0.001936811	0.001896200
Used Only By User	0.053232187	0.052806642	0.051373705	0.049767325	0.048772804
Recommended Tag	0.037595757	0.037290289	0.036262277	0.035110920	0.034398682
	guide	web	windows	technology	tools
Never Used	0.001431362	0.001394133	0.001363928	0.0006871293	0.0006460881
Used Only On Site	0.001422197	0.001385207	0.001355194	0.0006827266	0.0006419483
Used Only By User	0.037015391	0.036086097	0.035330847	0.0181047977	0.0170411669
Recommended Tag	0.026011611	0.025351301	0.024814937	0.0126488389	0.0119018526
	opensource	server	Other		
Never Used	0.0005854329	0.0005304722	0.0005005311		
Used Only On Site	0.0005816815	0.0005270727	0.0004973234		
Used Only By User	0.0154651486	0.0140328869	0.0132509415		
Recommended Tag	0.0107959133	0.0097917791	0.0092439425		

[[12]]

err.the_blog.find_by_title('Sessions N Such')

<http://errtheblog.com/post/24>

5b2c1c410192d0f355498d56a46b683e

Total Number of Users of Site: 457

Number of Users in Fit: 456

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-7.29075	0.24939	-29.2340	< 2e-16 ***
used.onSiteTRUE	0.46218	0.27898	1.6567	0.09758 .
used.byUserTRUE	3.57777	0.27667	12.9314	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.49082	0.29726	-1.6511	0.09871 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

Tag	Estimate	Std. Error	z value	Pr(> z)
tag_classsessions	7.07930	0.17970	39.3957	< 2.2e-16 ***
tag_classrails	5.81840	0.18964	30.6817	< 2.2e-16 ***
tag_classrubyonrails	4.44865	0.17954	24.7783	< 2.2e-16 ***
tag_classsession	3.51822	0.22814	15.4213	< 2.2e-16 ***
tag_classruby	3.22908	0.17367	18.5931	< 2.2e-16 ***
tag_classshowto	2.92281	0.19026	15.3620	< 2.2e-16 ***
tag_classperformance	2.83987	0.19902	14.2691	< 2.2e-16 ***
tag_classprogramming	2.46534	0.19033	12.9529	< 2.2e-16 ***
tag_classrror	1.78256	0.43342	4.1128	3.909e-05 ***
tag_classwebdev	1.65674	0.27557	6.0120	1.833e-09 ***
tag_classstoread	1.58466	0.30199	5.2474	1.543e-07 ***

tag_classtips	1.41860	0.24705	5.7421	9.352e-09	***
tag_classmemcached	1.29931	0.40209	3.2314	0.0012318	**
tag_classdevelopment	1.24584	0.25474	4.8905	1.006e-06	***
tag_classarticle	1.23786	0.33363	3.7103	0.0002070	***
tag_classdev	1.11862	0.45783	2.4433	0.0145539	*
tag_classtutorial	1.03799	0.25240	4.1125	3.913e-05	***
tag_classcache	0.82829	0.42431	1.9521	0.0509298	.
tag_classdatabase	0.75979	0.29615	2.5656	0.0102997	*
tag_classreference	0.67228	0.29406	2.2862	0.0222434	*
tag_classguide	0.59880	0.42689	1.4027	0.1607103	
tag_classweb	0.54103	0.29471	1.8358	0.0663900	.
tag_classmanagement	0.40252	0.38269	1.0518	0.2928904	
tag_classblog	0.33640	0.32165	1.0459	0.2956296	
tag_classdesign	-0.30497	0.40055	-0.7614	0.4464320	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.59188 +- 0.76934

Goodness of Fit Tests:

Residual Deviance: 6245 on 4.648e+04 degrees of freedom (p-value: 1)

AIC: 6305

Gm: 7622 on 31 degrees of freedom (p:value 0)

R²_L: 0.5496

R²: 0.4235

Predictive Power Tests

Lambda_p: 0.2847 (d= 11.56 ; p-value: 6.856e-31) (for prediction models)

Tau_p: 0.6297 (d= 36.12 ; p-value: 1.039e-285) (for classification models)

Phi_p: 0.5582 (d= 29.15 ; p-value: 9.21e-187) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	44619	302
TRUE	836	755

Fitted Probabilities:

	sessions	rails	rubyonrails	session	ruby
Never Used	0.4473333	0.1865864	0.05509121	0.02247706	0.01692871
Used Only On Site	0.5623563	0.2669478	0.08471701	0.03521799	0.02661022
Used Only By User	0.9666351	0.8914292	0.67604836	0.45146381	0.38133115
Recommended Tag	0.9656991	0.8886264	0.66974529	0.44438232	0.37459848
	howto	performance	programming	ror	webdev
Never Used	0.01251869	0.01153369	0.00795945	0.004037069	0.003561479
Used Only On Site	0.01972873	0.01818687	0.01257708	0.006393816	0.005642160
Used Only By User	0.31213296	0.29460675	0.22310972	0.126703833	0.113423084
Recommended Tag	0.30601772	0.28869077	0.21818538	0.123568864	0.110575107
	toread	tips	memcached	development	article
Never Used	0.003314644	0.002808907	0.002493829	0.002364285	0.002345541
Used Only On Site	0.005251880	0.004451887	0.003953244	0.003748175	0.003718500
Used Only By User	0.106374946	0.091589629	0.082136167	0.078193868	0.077620709
Recommended Tag	0.103683246	0.089234696	0.080002893	0.076154489	0.075595053
	dev	tutorial	cache	database	reference
Never Used	0.002082441	0.001921443	0.001558515	0.001455493	0.001333694
Used Only On Site	0.003301904	0.003046915	0.002471931	0.002308670	0.002115625

```

Used Only By User 0.069502199 0.064465557 0.052915436 0.049586183 0.045620696
Recommended Tag 0.067672866 0.062759850 0.051498513 0.048253866 0.044389950
                guide          web management          blog          design
Never Used      0.001239322 0.001169838 0.001018679 0.0009535658 0.0005023477
Used Only On Site 0.001966033 0.001855881 0.001616219 0.0015129698 0.0007972585
Used Only By User 0.042526023 0.040235007 0.035214060 0.0330354859 0.0176719471
Recommended Tag 0.041375145 0.039143595 0.034253985 0.0321328299 0.0171816227
                Other
Never Used      0.0006813535
Used Only On Site 0.0010812385
Used Only By User 0.0238233089
Recommended Tag 0.0231663344

```

[[13]]

```

Beer Advocate - Respect Beer.
http://beeradvocate.com/
5caf7547955925da56256a4a2f918d7b
Total Number of Users of Site: 489
Number of Users in Fit: 489
Type of Data: bysite

```

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.758360	0.149956	-45.0690	<2e-16 ***
used.onSiteTRUE	0.040025	0.179085	0.2235	0.8231
used.byUserTRUE	3.222382	0.197538	16.3127	<2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.235085	0.217872	-1.0790	0.2806

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classbeer	7.617432	0.173039	44.0215	< 2.2e-16 ***
tag_classreviews	4.775614	0.168864	28.2809	< 2.2e-16 ***
tag_classalcohol	4.756587	0.174626	27.2387	< 2.2e-16 ***
tag_classfood	3.722617	0.176009	21.1502	< 2.2e-16 ***
tag_classcommunity	3.610846	0.197011	18.3281	< 2.2e-16 ***
tag_classsystem:unfiled	3.462606	0.249887	13.8567	< 2.2e-16 ***
tag_classforums	3.356673	0.227316	14.7666	< 2.2e-16 ***
tag_classreference	3.190341	0.177215	18.0027	< 2.2e-16 ***
tag_classbrewing	2.688758	0.275070	9.7748	< 2.2e-16 ***
tag_classhomebrew	2.409450	0.303207	7.9466	1.918e-15 ***
tag_classdrink	2.379199	0.317377	7.4965	6.557e-14 ***
tag_classdrinks	2.157781	0.339412	6.3574	2.052e-10 ***
tag_classreview	2.067096	0.300228	6.8851	5.775e-12 ***
tag_classdrinking	1.929970	0.382675	5.0434	4.574e-07 ***
tag_classforum	1.896941	0.280962	6.7516	1.462e-11 ***
tag_classinformation	1.649026	0.333672	4.9421	7.730e-07 ***
tag_classblog	1.303412	0.246859	5.2800	1.292e-07 ***
tag_classdatabase	1.117846	0.382733	2.9207	0.003493 **
tag_classmagazine	0.369855	0.450683	0.8207	0.411843
tag_classculture	0.084205	0.449972	0.1871	0.851556
tag_classnews	-0.433333	0.450672	-0.9615	0.336287

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.74977 +- 0.8659

Goodness of Fit Tests:

Residual Deviance: 5980 on 6.501e+04 degrees of freedom (p-value: 1)
AIC: 6032
Gm: 6899 on 27 degrees of freedom (p-value 0)
R²_L: 0.5357
R²: 0.3775

Predictive Power Tests

Lambda_p: 0.2392 (d= 8.769 ; p-value: 1.799e-18) (for prediction models)
Tau_p: 0.6117 (d= 31.71 ; p-value: 1.082e-220) (for classification models)
Phi_p: 0.4895 (d= 22.02 ; p-value: 1.827e-107) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	63541	179
TRUE	823	494

Fitted Probabilities:

	beer	reviews	alcohol	food	community
Never Used	0.7024667	0.1210265	0.1190169	0.04583702	0.04118936
Used Only On Site	0.7107639	0.1253493	0.1232780	0.04761973	0.04279942
Used Only By User	0.9833974	0.7755008	0.7721708	0.54652502	0.51870846
Recommended Tag	0.9798934	0.7397322	0.7360522	0.49789507	0.46998838

	system:unfiled	forums	reference	brewing	homebrew
Never Used	0.03571715	0.03224280	0.02743763	0.01679723	0.01275607
Used Only On Site	0.03712159	0.03351537	0.02852614	0.01747120	0.01327008
Used Only By User	0.48166543	0.45529369	0.41444103	0.30001660	0.24480249
Recommended Tag	0.43329234	0.40748698	0.36802571	0.26071058	0.21055432

	drink	drinks	review	drinking	forum
Never Used	0.01238067	0.009946105	0.009091669	0.007935907	0.007680053
Used Only On Site	0.01287975	0.010348072	0.009459434	0.008257310	0.007991177
Used Only By User	0.23925318	0.201298903	0.187112744	0.167143636	0.162596179
Recommended Tag	0.20557004	0.171752769	0.159233696	0.141721180	0.137750990

	information	blog	database	magazine	culture
Never Used	0.006003841	0.004256903	0.003538477	0.001677947	0.001261546
Used Only On Site	0.006247488	0.004429972	0.003682446	0.001746350	0.001312996
Used Only By User	0.131592451	0.096864024	0.081800478	0.040460686	0.030716048
Recommended Tag	0.110857568	0.081090286	0.068294256	0.033530889	0.025411040

	news	Other
Never Used	0.0007522471	0.001159785
Used Only On Site	0.0007829426	0.001207091
Used Only By User	0.0185363606	0.028305721
Recommended Tag	0.0153017134	0.023406950

[[14]]

Old Computers - rare, vintage and obsolete computers
<http://oldcomputers.net/>
78649812ecefcdcl3a6bd58582db0b438
Total Number of Users of Site: 258
Number of Users in Fit: 258
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.02358	0.15242	-39.5190	< 2.2e-16 ***
used.onSiteTRUE	-0.22792	0.18923	-1.2045	0.228400
used.byUserTRUE	4.05548	0.20816	19.4828	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.65107	0.23386	-2.7841	0.005368 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classhistory	4.808523	0.210836	22.8069	< 2.2e-16 ***
tag_classcomputers	4.551049	0.215083	21.1595	< 2.2e-16 ***
tag_classretro	4.266903	0.218690	19.5111	< 2.2e-16 ***
tag_classcomputer	4.266560	0.218635	19.5145	< 2.2e-16 ***
tag_classsystem:unfiled	3.558011	0.275134	12.9319	< 2.2e-16 ***
tag_classhardware	3.411172	0.226172	15.0822	< 2.2e-16 ***
tag_classnostalgia	3.342932	0.291179	11.4807	< 2.2e-16 ***
tag_classstechnology	2.849169	0.234816	12.1336	< 2.2e-16 ***
tag_classgeek	2.737063	0.255930	10.6946	< 2.2e-16 ***
tag_classold	2.615249	0.340127	7.6890	1.483e-14 ***
tag_classobsolete	2.404661	0.396553	6.0639	1.329e-09 ***
tag_classvintage	2.296474	0.340530	6.7438	1.543e-11 ***
tag_classstech	1.830535	0.277011	6.6082	3.891e-11 ***
tag_classretrocomputing	1.811925	0.467428	3.8764	0.0001060 ***
tag_classapple	1.403370	0.314851	4.4573	8.302e-06 ***
tag_classfun	1.332045	0.295618	4.5060	6.607e-06 ***
tag_classmuseum	1.158257	0.450680	2.5700	0.0101693 *
tag_classimages	1.058347	0.340330	3.1098	0.0018724 **
tag_classcool	1.019019	0.362981	2.8074	0.0049949 **
tag_classreference	0.897997	0.282861	3.1747	0.0014999 **
tag_classscience	0.071794	0.381429	0.1882	0.8507017
tag_classprogramming	-0.209106	0.427117	-0.4896	0.6244339

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.45364 +- 0.67353

Goodness of Fit Tests:

Residual Deviance: 4122 on 3.3e+04 degrees of freedom (p-value: 1)

AIC: 4176

Gm: 3770 on 28 degrees of freedom (p:value 0)

R^2_L: 0.4777

R^2: 0.3197

Predictive Power Tests

Lambda_p: 0.1637 (d= 4.833 ; p-value: 1.345e-06) (for prediction models)

Tau_p: 0.5708 (d= 23.82 ; p-value: 1.976e-125) (for classification models)

Phi_p: 0.4439 (d= 16.18 ; p-value: 7.157e-59) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	32019	156
TRUE	554	295

Fitted Probabilities:

	history	computers	retro	computer	system:unfiled
Never Used	0.2288077	0.1865584	0.1472071	0.1471642	0.07830757
Used Only On Site	0.1910849	0.1544064	0.1208297	0.1207933	0.06335873
Used Only By User	0.9448216	0.9297561	0.9087779	0.9087495	0.83060377
Recommended Tag	0.8766876	0.8460507	0.8053086	0.8052549	0.67060407
	hardware	nostalgia	technology	geek	old
Never Used	0.06834419	0.06412503	0.04014016	0.03603668	0.03203616
Used Only On Site	0.05518347	0.05173166	0.03222272	0.02890425	0.02567444
Used Only By User	0.80893003	0.79815977	0.70704384	0.68329669	0.65636799
Recommended Tag	0.63739579	0.62148087	0.50051896	0.47252018	0.44229733
	obsolete	vintage	tech	retrocomputing	apple
Never Used	0.02611159	0.02349700	0.01487563	0.01460536	0.009754645
Used Only On Site	0.02090095	0.01879805	0.01187981	0.01166332	0.007781999
Used Only By User	0.60743952	0.58136385	0.46566311	0.46103564	0.362453835
Recommended Tag	0.39116165	0.36572076	0.26569841	0.26208341	0.190969414
	fun	museum	images	cool	reference
Never Used	0.00908924	0.007650367	0.006927999	0.006662607	0.005907650
Used Only On Site	0.00725017	0.006100640	0.005523787	0.005311898	0.004709267
Used Only By User	0.34613915	0.307924110	0.287050485	0.279069879	0.255383592
Recommended Tag	0.18019195	0.155928961	0.143226490	0.138467890	0.124651923
	science	programming	Other		
Never Used	0.002594445	0.001960321	0.002415143		
Used Only On Site	0.002066758	0.001561407	0.001923854		
Used Only By User	0.130527210	0.101816215	0.122593202		
Recommended Tag	0.058673648	0.044950589	0.054831776		

[[15]]

DotNetNuke - Web Application Framework > Home (DNN 3.2.2)
<http://www.dotnetnuke.com/>
 83f6625b93df099752919f24e1feb4e7
 Total Number of Users of Site: 714
 Number of Users in Fit: 714
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.45415	0.16066	-9.0512	< 2.2e-16 ***
used.onSiteTRUE	-0.17418	0.12840	-1.3566	0.1749
used.byUserTRUE	3.65935	0.14100	25.9523	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.64858	0.15262	-4.2495	2.142e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classdotnetnuke	0.57278	0.13131	4.3619	1.289e-05 ***
tag_classcms	0.25829	0.13369	1.9320	0.05336 .
tag_classasp.net	-0.26347	0.14018	-1.8795	0.06017 .
tag_classopensource	-0.59982	0.14105	-4.2525	2.114e-05 ***
tag_classdotnet	-0.99252	0.15808	-6.2786	3.416e-10 ***
tag_classsystem:unfiled	-1.31936	0.18902	-6.9799	2.954e-12 ***
tag_classprogramming	-1.78132	0.15172	-11.7407	< 2.2e-16 ***
tag_classdnn	-2.04299	0.23736	-8.6071	< 2.2e-16 ***
tag_classframework	-2.39351	0.20274	-11.8059	< 2.2e-16 ***

tag_classportal	-2.46453	0.21245	-11.6004	< 2.2e-16	***
tag_classasp	-2.54583	0.22815	-11.1585	< 2.2e-16	***
tag_classdevelopment	-2.65892	0.18412	-14.4410	< 2.2e-16	***
tag_classweb	-2.87913	0.18048	-15.9524	< 2.2e-16	***
tag_classwebdev	-3.01813	0.25341	-11.9100	< 2.2e-16	***
tag_classcontent	-3.08704	0.30535	-10.1099	< 2.2e-16	***
tag_classapplication	-3.21582	0.30294	-10.6152	< 2.2e-16	***
tag_classnuke	-3.34686	0.40930	-8.1770	2.909e-16	***
tag_classsoftware	-3.43863	0.19781	-17.3831	< 2.2e-16	***
tag_classdev	-3.45819	0.35845	-9.6477	< 2.2e-16	***
tag_classopen-source	-3.51068	0.39561	-8.8740	< 2.2e-16	***
tag_classstools	-3.52395	0.21113	-16.6905	< 2.2e-16	***
tag_classwebdesign	-3.56160	0.23218	-15.3398	< 2.2e-16	***
tag_classopen	-3.57456	0.37314	-9.5798	< 2.2e-16	***
tag_classsource	-3.57867	0.34466	-10.3831	< 2.2e-16	***
tag_classvb.net	-3.67999	0.41452	-8.8776	< 2.2e-16	***
tag_classnet	-3.78785	0.44713	-8.4715	< 2.2e-16	***
tag_classfreeware	-3.79465	0.28339	-13.3902	< 2.2e-16	***
tag_classc#	-3.81288	0.30139	-12.6510	< 2.2e-16	***
tag_classcommunity	-3.88962	0.31154	-12.4852	< 2.2e-16	***
tag_classcode	-3.91924	0.29892	-13.1114	< 2.2e-16	***
tag_classimported	-3.92077	0.43821	-8.9473	< 2.2e-16	***
tag_classstechnology	-3.93240	0.27487	-14.3062	< 2.2e-16	***
tag_classwebsite	-3.94607	0.41635	-9.4778	< 2.2e-16	***
tag_classmicrosoft	-3.95922	0.26398	-14.9981	< 2.2e-16	***
tag_classfree	-3.98334	0.27452	-14.5104	< 2.2e-16	***
tag_classcollaboration	-4.07367	0.36045	-11.3015	< 2.2e-16	***
tag_classinternet	-4.39063	0.32286	-13.5991	< 2.2e-16	***
tag_classblog	-4.41070	0.27597	-15.9826	< 2.2e-16	***
tag_classmanagement	-4.44234	0.37779	-11.7589	< 2.2e-16	***
tag_classdownload	-4.54933	0.37989	-11.9754	< 2.2e-16	***
tag_classresources	-4.55780	0.46155	-9.8749	< 2.2e-16	***
tag_classstool	-4.71980	0.42746	-11.0415	< 2.2e-16	***
tag_classphp	-4.92900	0.37404	-13.1779	< 2.2e-16	***
tag_classcomputer	-4.97685	0.45710	-10.8878	< 2.2e-16	***
tag_classwindows	-5.15544	0.39278	-13.1255	< 2.2e-16	***
tag_classwiki	-5.21916	0.41938	-12.4451	< 2.2e-16	***
tag_classweb2.0	-5.38025	0.41932	-12.8310	< 2.2e-16	***
tag_classdesign	-5.45344	0.37344	-14.6034	< 2.2e-16	***
tag_classOther	-5.49829	0.12912	-42.5817	< 2.2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.55419 +- 0.74444

Goodness of Fit Tests:

Residual Deviance: 1.362e+04 on 1.82e+05 degrees of freedom (p-value: 1)

AIC: 1.373e+04

Gm: 1.238e+04 on 55 degrees of freedom (p-value 0)

R²_L: 0.4761

R²: 0.2454

Predictive Power Tests

Lambda_p: 0.09502 (d= 4.737 ; p-value: 2.165e-06) (for prediction models)

Tau_p: 0.5413 (d= 38.16 ; p-value: 1.147e-318) (for classification models)

Phi_p: 0.3259 (d= 18.87 ; p-value: 2.052e-79) (for selection models)
 Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	179303	315
TRUE	1904	548

Fitted Probabilities:

	dotnetnuke	cms	asp.net	opensource	dotnet
Never Used	0.2928937	0.2322125	0.1521778	0.11365117	0.07968222
Used Only On Site	0.2581596	0.2026126	0.1310386	0.09724992	0.06780808
Used Only By User	0.9414740	0.9215422	0.8745416	0.83276766	0.77077187
Recommended Tag	0.8760137	0.8376332	0.7537959	0.68624141	0.59626098
	system:unfiled	programming	dnn	framework	portal
Never Used	0.05877231	0.03785242	0.02939377	0.02088408	0.01948029
Used Only On Site	0.04984537	0.03199500	0.02481151	0.01760436	0.01641732
Used Only By User	0.70802949	0.60440998	0.54046357	0.45305975	0.43552782
Recommended Tag	0.51576178	0.40157891	0.34061474	0.26676844	0.25310931
	asp development	web	webdev	content	
Never Used	0.01798652	0.01609425	0.01295436	0.011292225	0.010548217
Used Only On Site	0.01515480	0.01355632	0.01090608	0.009504224	0.008876965
Used Only By User	0.41565518	0.38847645	0.33761597	0.307264964	0.292795204
Recommended Tag	0.23805000	0.21814900	0.18291801	0.163050729	0.153863891
	application	nuke	software	dev	open-source
Never Used	0.009285526	0.008154355	0.007444708	0.007301571	0.006930774
Used Only On Site	0.007812753	0.006859755	0.006262062	0.006141522	0.005829290
Used Only By User	0.266858086	0.242014238	0.225581313	0.222182970	0.213243907
Recommended Tag	0.137835250	0.122988096	0.113427711	0.111475733	0.106381561
	tools	webdesign	open	source	vb.net
Never Used	0.006840020	0.006588939	0.006504649	0.006478138	0.005857617
Used Only On Site	0.005752876	0.005541478	0.005470514	0.005448195	0.004925840
Used Only By User	0.211025706	0.204825210	0.202722454	0.202058865	0.186215602
Recommended Tag	0.105126429	0.101636694	0.100459448	0.100088582	0.091325635
	net	freeware	c#	community	code
Never Used	0.005261816	0.005226326	0.005132441	0.004755117	0.004616973
Used Only On Site	0.004424392	0.004394526	0.004315517	0.003998011	0.003881776
Used Only By User	0.170419634	0.169459968	0.166910814	0.156511049	0.152640318
Recommended Tag	0.082760243	0.082245265	0.080880303	0.075356007	0.073317891
	imported	technology	website	microsoft	free
Never Used	0.004609910	0.004556887	0.004495260	0.004436801	0.004331554
Used Only On Site	0.003875833	0.003831221	0.003779370	0.003730187	0.003641640
Used Only By User	0.152441469	0.150945955	0.149201295	0.147539903	0.144532868
Recommended Tag	0.073213449	0.072428774	0.071515190	0.070647020	0.069080155
	collaboration	internet	blog	management	download
Never Used	0.003958900	0.002886622	0.002829423	0.002741541	0.002464040
Used Only On Site	0.003328143	0.002426291	0.002378191	0.002304292	0.002070958
Used Only By User	0.133718281	0.101066503	0.099257509	0.096464300	0.087532795
Recommended Tag	0.063492252	0.047057008	0.046165087	0.044791657	0.040430373
	resources	tool	php	computer	windows
Never Used	0.002443328	0.002078667	0.001686939	0.001608253	0.001345571
Used Only On Site	0.002053543	0.001746955	0.001417650	0.001351508	0.001130713
Used Only By User	0.086859272	0.074840952	0.061583261	0.058875529	0.049725025
Recommended Tag	0.040103352	0.034311346	0.028015852	0.026741974	0.022466483
	wiki	web2.0	design	Other	
Never Used	0.001262609	0.0010749570	0.0009991658	0.0009553857	

```

Used Only On Site 0.001060984 0.0009032713 0.0008395748 0.0008027819
Used Only By User 0.046798973 0.0401154376 0.0373900772 0.0358089243
Recommended Tag 0.021108820 0.0180248127 0.0167740176 0.0160501412
      Other
Never Used 0.0009553857
Used Only On Site 0.0008027819
Used Only By User 0.0358089243
Recommended Tag 0.0160501412

```

```

[[16]]
BibDesk | Home
http://bibdesk.sourceforge.net/
85ab919107e4cc79b345e996b3c0b097
Total Number of Users of Site: 303
Number of Users in Fit: 303
Type of Data: bysite

```

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.15078	0.15189	-40.4953	< 2.2e-16 ***
used.onSiteTRUE	-0.49366	0.18235	-2.7072	0.006786 **
used.byUserTRUE	3.85871	0.18529	20.8258	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.28654	0.21055	-1.3609	0.173531

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classbibtex	5.638640	0.179628	31.3906	< 2.2e-16 ***
tag_classbibliography	5.280009	0.181733	29.0537	< 2.2e-16 ***
tag_classlatex	5.178528	0.186233	27.8067	< 2.2e-16 ***
tag_classosx	4.086542	0.190762	21.4222	< 2.2e-16 ***
tag_classsoftware	4.048486	0.180458	22.4346	< 2.2e-16 ***
tag_classstex	3.967272	0.220097	18.0251	< 2.2e-16 ***
tag_classmac	3.824762	0.185886	20.5759	< 2.2e-16 ***
tag_classmacosx	2.430452	0.304465	7.9827	1.432e-15 ***
tag_classcitation	2.305393	0.371040	6.2133	5.187e-10 ***
tag_classresearch	2.056909	0.225721	9.1126	< 2.2e-16 ***
tag_classapps	2.048390	0.376757	5.4369	5.421e-08 ***
tag_classacademic	1.751879	0.346027	5.0628	4.130e-07 ***
tag_classwriting	1.712766	0.256836	6.6687	2.580e-11 ***
tag_classapplications	1.647059	0.462190	3.5636	0.0003658 ***
tag_classapplication	1.571421	0.462449	3.3980	0.0006787 ***
tag_classapple	1.389684	0.261524	5.3138	1.074e-07 ***
tag_classstool	1.241879	0.367318	3.3809	0.0007224 ***
tag_classopensource	1.111780	0.312997	3.5520	0.0003822 ***
tag_classpapers	1.104088	0.456603	2.4180	0.0156039 *
tag_classpublishing	1.054162	0.455933	2.3121	0.0207723 *
tag_classfree	1.004812	0.348104	2.8865	0.0038952 **
tag_classproductivity	0.996946	0.333548	2.9889	0.0027997 **
tag_classstools	0.991654	0.311207	3.1865	0.0014402 **
tag_classreference	0.958039	0.264673	3.6197	0.0002949 ***
tag_classcocoa	0.939856	0.424045	2.2164	0.0266635 *
tag_classeducation	0.520489	0.391780	1.3285	0.1840050
tag_classdatabase	0.360572	0.418455	0.8617	0.3888669

```

tag_classlibrary      0.070016   0.415981  0.1683 0.8663359
tag_classscience     -0.535984   0.441351 -1.2144 0.2245886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Standard Deviation on normal distribution of user values: 0.45009 +- 0.67088

Goodness of Fit Tests:

```

Residual Deviance: 5673 on 4.875e+04 degrees of freedom (p-value: 1)
AIC: 5741
Gm: 5941 on 35 degrees of freedom (p:value 0)
R^2_L: 0.5116
R^2: 0.3518

```

Predictive Power Tests

```

Lambda_p: 0.2163 (d= 7.743 ; p-value: 9.744e-15 ) (for prediction models)
Tau_p: 0.5979 (d= 30.25 ; p-value: 5.294e-201 ) (for classification models)
Phi_p: 0.4951 (d= 22.23 ; p-value: 1.641e-109 ) (for selection models)

```

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	47307	228
TRUE	750	498

Fitted Probabilities:

	bibtex	bibliography	latex	osx	software
Never Used	0.3746931	0.2950949	0.2744328	0.1126221	0.10887457
Used Only On Site	0.2678032	0.2035216	0.1875653	0.0718979	0.06939952
Used Only By User	0.9659925	0.9520265	0.9471732	0.8574753	0.85276089
Recommended Tag	0.9286655	0.9009422	0.8915099	0.7338552	0.72635667
	tex	mac	macosx	citation	research
Never Used	0.10124176	0.08899133	0.02365312	0.02093077	0.01640117
Used Only On Site	0.06433437	0.05627020	0.01457186	0.01288092	0.01007545
Used Only By User	0.84226884	0.82240051	0.53454168	0.50333193	0.44148048
Recommended Tag	0.70992123	0.67972121	0.34483558	0.31715490	0.26593192
	apps	academic	writing applications	application	
Never Used	0.016264297	0.012141672	0.011681380	0.01094664	0.010157292
Used Only On Site	0.009990832	0.007446349	0.007162767	0.00671031	0.006224516
Used Only By User	0.439380916	0.368144348	0.359093890	0.34411582	0.327251131
Recommended Tag	0.264272194	0.210753162	0.204320806	0.19384587	0.182298798
	apple	tool	opensource	papers	publishing
Never Used	0.008483676	0.007326555	0.006438533	0.006389512	0.006080237
Used Only On Site	0.005195502	0.004484839	0.003939883	0.003909811	0.003720113
Used Only By User	0.288561331	0.259189314	0.235000908	0.233620856	0.224801225
Recommended Tag	0.156753182	0.138191578	0.123413928	0.122584169	0.117314711
	free productivity	tools	reference	cocoa	
Never Used	0.005789156	0.005744055	0.005713914	0.005526074	0.005427045
Used Only On Site	0.003541617	0.003513963	0.003495483	0.003380324	0.003319619
Used Only By User	0.216318172	0.214987570	0.214095907	0.208494143	0.205509506
Recommended Tag	0.112300156	0.111518334	0.110995134	0.107721224	0.105986004
	education	database	library	science	Other
Never Used	0.003574724	0.003048042	0.002281224	0.0012457653	0.002127294
Used Only On Site	0.002185010	0.001862699	0.001393669	0.0007607687	0.001299550
Used Only By User	0.145346421	0.126585406	0.097787837	0.0558271599	0.091782256
Recommended Tag	0.072306869	0.062286557	0.047324096	0.0263841232	0.044265658

[[17]]

Tiny Icon Factory
http://tiny.media.mit.edu/
967c02486be93cc670a10c0c65203bea
Total Number of Users of Site: 819
Number of Users in Fit: 819
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-7.3233194	0.1250488	-58.5637	< 2.2e-16 ***
used.onSiteTRUE	0.0008949	0.1386778	0.0065	0.994851
used.byUserTRUE	2.9164998	0.1507284	19.3494	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	0.4366632	0.1608335	2.7150	0.006628 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classicons	6.265759	0.116006	54.0126	< 2.2e-16 ***
tag_classicon	5.688782	0.119439	47.6291	< 2.2e-16 ***
tag_classdesign	4.419483	0.112900	39.1451	< 2.2e-16 ***
tag_classgraphics	4.349501	0.120189	36.1889	< 2.2e-16 ***
tag_classsystem:unfiled	3.846498	0.198691	19.3592	< 2.2e-16 ***
tag_classstools	3.437055	0.120459	28.5330	< 2.2e-16 ***
tag_classart	3.418147	0.122188	27.9746	< 2.2e-16 ***
tag_classstiny	3.260455	0.228890	14.2446	< 2.2e-16 ***
tag_classmit	2.659850	0.244129	10.8953	< 2.2e-16 ***
tag_classinternet	2.625658	0.149964	17.5086	< 2.2e-16 ***
tag_classwebdesign	2.485551	0.142120	17.4891	< 2.2e-16 ***
tag_classfactory	2.355019	0.344235	6.8413	7.847e-12 ***
tag_classpixelart	2.252936	0.296062	7.6097	2.748e-14 ***
tag_classpixel	2.233686	0.238481	9.3663	< 2.2e-16 ***
tag_classfree	2.223560	0.152419	14.5885	< 2.2e-16 ***
tag_classgenerator	2.146741	0.191588	11.2050	< 2.2e-16 ***
tag_classcool	1.998115	0.188153	10.6196	< 2.2e-16 ***
tag_classgraphic	1.984908	0.280061	7.0874	1.366e-12 ***
tag_classbitmap	1.820360	0.411655	4.4221	9.777e-06 ***
tag_classcollection	1.790271	0.351462	5.0938	3.510e-07 ***
tag_classstool	1.760413	0.230752	7.6290	2.365e-14 ***
tag_classweb	1.733007	0.165024	10.5015	< 2.2e-16 ***
tag_classresources	1.647779	0.237329	6.9430	3.838e-12 ***
tag_classonline	1.531717	0.212689	7.2017	5.948e-13 ***
tag_classfun	1.475808	0.197602	7.4686	8.106e-14 ***
tag_classeditor	1.459206	0.254473	5.7342	9.796e-09 ***
tag_classdiy	1.408927	0.242106	5.8195	5.904e-09 ***
tag_classsmall	1.361061	0.442487	3.0759	0.0020984 **
tag_classfavicon	1.345137	0.382260	3.5189	0.0004333 ***
tag_classcommunity	1.326202	0.230406	5.7559	8.617e-09 ***
tag_classweb2.0	1.199472	0.204357	5.8695	4.371e-09 ***
tag_classuseful	1.137650	0.367298	3.0973	0.0019526 **
tag_classimages	1.079463	0.246470	4.3797	1.188e-05 ***
tag_classgallery	1.048047	0.294325	3.5608	0.0003697 ***
tag_classimage	0.963452	0.316450	3.0446	0.0023302 **

tag_classajax	0.957092	0.249962	3.8289	0.0001287	***
tag_classinteractive	0.930643	0.437934	2.1251	0.0335803	*
tag_classwebsite	0.917056	0.386299	2.3740	0.0175987	*
tag_classmedia	0.810046	0.271381	2.9849	0.0028367	**
tag_classcomputer	0.737366	0.328999	2.2412	0.0250105	*
tag_classillustration	0.679978	0.342616	1.9847	0.0471821	*
tag_classdownload	0.670182	0.314111	2.1336	0.0328766	*
tag_classdrawing	0.664827	0.380547	1.7470	0.0806322	.
tag_classfreeware	0.553447	0.314145	1.7618	0.0781104	.
tag_classsoftware	0.322181	0.274226	1.1749	0.2400441	
tag_classinteresting	0.304253	0.434623	0.7000	0.4839034	
tag_classjavascript	0.250759	0.314035	0.7985	0.4245760	
tag_classfunny	0.236745	0.342838	0.6905	0.4898505	
tag_classcode	0.098574	0.432971	0.2277	0.8199034	
tag_classsocial	0.073646	0.377508	0.1951	0.8453277	
tag_classtechnology	-0.180294	0.402885	-0.4475	0.6545090	
tag_classblog	-0.311065	0.357179	-0.8709	0.3838113	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.77761 +- 0.88182

Goodness of Fit Tests:

Residual Deviance: 1.564e+04 on 1.965e+05 degrees of freedom (p-value: 1)

AIC: 1.576e+04

Gm: 1.59e+04 on 58 degrees of freedom (p-value 0)

R²_L: 0.5041

R²: 0.2974

Predictive Power Tests

Lambda_p: 0.1337 (d= 7.452 ; p-value: 9.197e-14) (for prediction models)

Tau_p: 0.56 (d= 44.15 ; p-value: 0) (for classification models)

Phi_p: 0.412 (d= 27.99 ; p-value: 2.398e-172) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	192957	543
TRUE	2108	952

Fitted Probabilities:

	icons	icon	design	graphics	system:unfiled
Never Used	0.2577760	0.1632097	0.05196424	0.04862278	0.02997898
Used Only On Site	0.2579472	0.1633319	0.05200834	0.04866419	0.03000502
Used Only By User	0.8651733	0.7827836	0.50316579	0.48567426	0.36347310
Recommended Tag	0.9085866	0.8480670	0.61069188	0.59393085	0.46934767
	tools	art	tiny	mit	internet
Never Used	0.02010919	0.01973998	0.01690886	0.009345516	0.009034211
Used Only On Site	0.02012683	0.01975730	0.01692374	0.009353805	0.009042226
Used Only By User	0.27492745	0.27117442	0.24115374	0.148429863	0.144159763
Recommended Tag	0.37000243	0.36560597	0.32986263	0.212585386	0.206918095
	webdesign	factory	pixelart	pixel	free
Never Used	0.007862414	0.006906924	0.006240820	0.006122562	0.006061249
Used Only On Site	0.007869398	0.006913065	0.006246373	0.006128010	0.006066643
Used Only By User	0.127720198	0.113870611	0.103968885	0.102189186	0.101263861
Recommended Tag	0.184867666	0.166000493	0.152345086	0.149875813	0.148590150

	generator	cool	graphic	bitmap	collection
Never Used	0.005615584	0.004843780	0.004780531	0.004058161	0.003938344
Used Only On Site	0.005620584	0.004848095	0.004784791	0.004061779	0.003941857
Used Only By User	0.094483662	0.082511333	0.081516997	0.070014983	0.068080926
Recommended Tag	0.139131734	0.122265798	0.120855492	0.104433956	0.101653051
	tool	web	resources	online	fun
Never Used	0.003822934	0.003719971	0.003417099	0.003043794	0.002878763
Used Only On Site	0.003826344	0.003723289	0.003420148	0.003046510	0.002881333
Used Only By User	0.066210813	0.064536405	0.059578107	0.053398153	0.050641649
Recommended Tag	0.098958673	0.096541724	0.089359364	0.080354195	0.076318364
	editor	diy	small	favicon	community
Never Used	0.002831498	0.002693027	0.002567483	0.002527025	0.002479744
Used Only On Site	0.002834026	0.002695431	0.002569776	0.002529282	0.002481959
Used Only By User	0.049849396	0.047521147	0.045400939	0.044715786	0.043913895
Recommended Tag	0.075156216	0.071735232	0.068612510	0.067601873	0.066418099
	web2.0	useful	images	gallery	image
Never Used	0.002185228	0.002054493	0.001938585	0.001878742	0.001726610
Used Only On Site	0.002187180	0.002056328	0.001940317	0.001880421	0.001728153
Used Only By User	0.038890157	0.036644119	0.034644521	0.033609073	0.030967278
Recommended Tag	0.058978700	0.055639644	0.052660159	0.051114774	0.047164123
	ajax	interactive	website	media	computer
Never Used	0.001715682	0.001670973	0.001648461	0.001481418	0.001377710
Used Only On Site	0.001717216	0.001672466	0.001649935	0.001482742	0.001378942
Used Only By User	0.030776991	0.029997719	0.029604903	0.026680659	0.024856775
Recommended Tag	0.046879125	0.045711376	0.045122364	0.040729706	0.037982909
	illustration	download	drawing	freeware	software
Never Used	0.001300972	0.001288307	0.001281436	0.001146524	0.000910016
Used Only On Site	0.001302135	0.001289459	0.001282581	0.001147550	0.000910830
Used Only By User	0.023503037	0.023279267	0.023157829	0.020767637	0.016550692
Recommended Tag	0.035940651	0.035602780	0.035419388	0.031804847	0.025404899
	interesting	javascript	funny	code	
Never Used	0.0008938606	0.0008473406	0.0008355586	0.0007278080	
Used Only On Site	0.0008946602	0.0008480986	0.0008363061	0.0007284592	
Used Only By User	0.0162613904	0.0154274340	0.0152160072	0.0132784515	
Recommended Tag	0.0249647572	0.0236952045	0.0233731610	0.0204184370	
	social	technology	blog	Other	
Never Used	0.0007099017	0.0005507860	0.0004833022	0.0006595326	
Used Only On Site	0.0007105368	0.0005512788	0.0004837347	0.0006601227	
Used Only By User	0.0129557643	0.0100795743	0.0088549426	0.0120469979	
Recommended Tag	0.0199257408	0.0155265507	0.0136492560	0.0185372557	

[[18]]

Mint: A Fresh Look at Your Site
<http://haveamint.com/>
b4b81afe38922cfb189a39e7b26a5719
Total Number of Users of Site: 560
Number of Users in Fit: 560
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.91743	0.11126	-62.1757	< 2e-16 ***
used.onSiteTRUE	-0.12023	0.13349	-0.9007	0.36777
used.byUserTRUE	3.57049	0.13683	26.0952	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.36910	0.15439	-2.3907	0.01682 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classtats	6.27999	0.13000	48.3089	< 2.2e-16 ***
tag_classweb	4.32167	0.13657	31.6433	< 2.2e-16 ***
tag_classtools	4.26301	0.13761	30.9781	< 2.2e-16 ***
tag_classsystem:unfiled	3.77604	0.22226	16.9896	< 2.2e-16 ***
tag_classajax	3.50420	0.14313	24.4821	< 2.2e-16 ***
tag_classmint	3.48977	0.23310	14.9713	< 2.2e-16 ***
tag_classsoftware	3.48931	0.14130	24.6935	< 2.2e-16 ***
tag_classanalytics	3.25207	0.22602	14.3883	< 2.2e-16 ***
tag_classphp	3.21898	0.15584	20.6555	< 2.2e-16 ***
tag_classtatistics	3.20986	0.19519	16.4445	< 2.2e-16 ***
tag_classwebdesign	2.87388	0.15925	18.0461	< 2.2e-16 ***
tag_classdesign	2.83831	0.14549	19.5089	< 2.2e-16 ***
tag_classwebdev	2.76147	0.20358	13.5648	< 2.2e-16 ***
tag_classweb2.0	2.33318	0.17978	12.9778	< 2.2e-16 ***
tag_classwebstats	2.27357	0.37137	6.1221	9.235e-10 ***
tag_classwebsite	2.15901	0.29893	7.2224	5.107e-13 ***
tag_classtool	2.10178	0.26200	8.0220	1.040e-15 ***
tag_classtraffic	2.08066	0.30507	6.8203	9.084e-12 ***
tag_classtracking	2.06452	0.32566	6.3396	2.304e-10 ***
tag_classutilities	1.95580	0.27480	7.1173	1.101e-12 ***
tag_classanalysis	1.94148	0.35248	5.5080	3.629e-08 ***
tag_classinspiration	1.89574	0.24275	7.8095	5.743e-15 ***
tag_classseo	1.78649	0.26612	6.7132	1.904e-11 ***
tag_classclean	1.78577	0.42221	4.2296	2.341e-05 ***
tag_classsite	1.75346	0.41407	4.2347	2.288e-05 ***
tag_classblog	1.61784	0.20031	8.0767	6.655e-16 ***
tag_classcool	1.56212	0.26977	5.7907	7.011e-09 ***
tag_classwebapps	1.54714	0.44469	3.4791	0.0005031 ***
tag_classdevelopment	1.54328	0.25188	6.1271	8.948e-10 ***
tag_classcss	1.39750	0.19859	7.0370	1.965e-12 ***
tag_classjavascript	1.30544	0.22978	5.6814	1.336e-08 ***
tag_classmysql	1.26681	0.30994	4.0872	4.365e-05 ***
tag_classblogging	1.20179	0.32237	3.7280	0.0001930 ***
tag_classgui	1.09087	0.44030	2.4775	0.0132288 *
tag_classui	1.06944	0.41393	2.5836	0.0097775 **
tag_classapache	0.93940	0.36635	2.5642	0.0103412 *
tag_classblogs	0.37542	0.36736	1.0219	0.3068208
tag_classtech	0.25066	0.41163	0.6089	0.5425591
tag_classsearch	-0.11555	0.40467	-0.2855	0.7752368
tag_classbusiness	-0.12533	0.38173	-0.3283	0.7426601

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.66832 +- 0.81751

Goodness of Fit Tests:

Residual Deviance: 1.21e+04 on 1.573e+05 degrees of freedom (p-value: 1)

AIC: 1.219e+04

Gm: 1.073e+04 on 46 degrees of freedom (p-value 0)

R^2_L: 0.4701

R^2: 0.2372

Predictive Power Tests

Lambda_p: 0.04304 (d= 2.014 ; p-value: 0.04396) (for prediction models)

Tau_p: 0.5149 (d= 34.08 ; p-value: 1.504e-254) (for classification models)

Phi_p: 0.273 (d= 14.7 ; p-value: 6.861e-49) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	154894	305
TRUE	1763	398

Fitted Probabilities:

	stats	web	tools	system:unfiled	ajax
Never Used	0.3458268	0.06941229	0.06571746	0.04143217	0.03188481
Used Only On Site	0.3191542	0.06203697	0.05870996	0.03691192	0.02837535
Used Only By User	0.9494566	0.72606306	0.71424212	0.60566081	0.53923669
Recommended Tag	0.9201020	0.61902562	0.60509798	0.48495061	0.41774087
	mint	software	analytics	php	statistics
Never Used	0.03144221	0.03142816	0.02495643	0.02416367	0.02394943
Used Only On Site	0.02798005	0.02796751	0.02219207	0.02148519	0.02129417
Used Only By User	0.53564800	0.53553327	0.47630292	0.46805529	0.46578391
Recommended Tag	0.41423383	0.41412191	0.35797048	0.35040089	0.34832659
	webdesign	design	webdev	web2.0	webstats
Never Used	0.01723310	0.01664085	0.01542911	0.010108267	0.009528891
Used Only On Site	0.01531080	0.01478362	0.01370523	0.008973464	0.008458575
Used Only By User	0.38389487	0.37551645	0.35767749	0.266246498	0.254764438
Recommended Tag	0.27640328	0.26934523	0.25449451	0.181967350	0.173261646
	website	tool	traffic	tracking	utilities
Never Used	0.008506200	0.008036877	0.007870275	0.007745188	0.006952838
Used Only On Site	0.007549881	0.007132943	0.006984947	0.006873834	0.006170071
Used Only By User	0.233630077	0.223540143	0.219896518	0.217139120	0.199226324
Recommended Tag	0.157459907	0.150015648	0.147343030	0.145325934	0.132335901
	analysis	inspiration	seo	clean	site
Never Used	0.006854706	0.006550194	0.005876285	0.005872083	0.005686445
Used Only On Site	0.006082919	0.005812492	0.005214082	0.005210352	0.005045527
Used Only By User	0.196952683	0.189817337	0.173582731	0.173479538	0.168895424
Recommended Tag	0.130701039	0.125590531	0.114075635	0.114002938	0.110779815
	blog	cool	webapps	development	css
Never Used	0.004968842	0.004700818	0.004631253	0.004613501	0.003990168
Used Only On Site	0.004408446	0.004170523	0.004108774	0.004093016	0.003539757
Used Only By User	0.150703271	0.143709534	0.141876088	0.141406995	0.124615269
Recommended Tag	0.098108500	0.093287449	0.092028152	0.091706260	0.080264645
	javascript	mysql	blogging	gui	ui
Never Used	0.003640517	0.003503022	0.003283244	0.002939550	0.002877409
Used Only On Site	0.003229446	0.003107428	0.002912397	0.002607422	0.002552284
Used Only By User	0.114914995	0.111043262	0.104785952	0.094827726	0.093004304
Recommended Tag	0.073725943	0.071130430	0.066952823	0.060347657	0.059143923
	apache	blogs	tech	search	business
Never Used	0.002527413	0.001439514	0.001270891	0.0008815286	0.0008729502
Used Only On Site	0.002241746	0.001276652	0.001127085	0.0007817459	0.0007741377
Used Only By User	0.082599750	0.048729318	0.043261218	0.0303988574	0.0301116933
Recommended Tag	0.052308924	0.030447162	0.026972403	0.0188575523	0.0186773135
	Other				
Never Used	0.0009893969				

Used Only On Site 0.0008774150
Used Only By User 0.0339957155
Recommended Tag 0.0211185619

[[19]]
Telegraph newspaper online
http://www.telegraph.co.uk/
baaed76bf6c8be2facbd247b7c2d987c
Total Number of Users of Site: 447
Number of Users in Fit: 447
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.41293	0.13666	-46.9274	< 2.2e-16 ***
used.onSiteTRUE	-0.46879	0.17047	-2.7501	0.0059586 **
used.byUserTRUE	4.35048	0.18819	23.1181	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.79388	0.21586	-3.6777	0.0002353 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classnews	5.328874	0.172431	30.9043	< 2.2e-16 ***
tag_classnewspaper	5.162911	0.172551	29.9211	< 2.2e-16 ***
tag_classuk	4.864596	0.179767	27.0606	< 2.2e-16 ***
tag_classsystem:unfiled	4.771571	0.187075	25.5062	< 2.2e-16 ***
tag_classnewspapers	4.179773	0.203165	20.5733	< 2.2e-16 ***
tag_classstelegraph	4.002661	0.231650	17.2789	< 2.2e-16 ***
tag_classmedia	3.520044	0.211485	16.6444	< 2.2e-16 ***
tag_classnews(international)	2.455547	0.419189	5.8579	4.689e-09 ***
tag_classdaily	2.291876	0.266843	8.5889	< 2.2e-16 ***
tag_classinternational	2.240188	0.362550	6.1790	6.452e-10 ***
tag_classnoticias	2.183071	0.395246	5.5233	3.327e-08 ***
tag_classimported	2.044132	0.378003	5.4077	6.384e-08 ***
tag_classenglish	1.349521	0.435880	3.0961	0.001961 **
tag_classworld	1.316812	0.469257	2.8062	0.005014 **
tag_classinternet	0.549719	0.459587	1.1961	0.231651
tag_classpolitics	0.286737	0.425932	0.6732	0.500820
tag_classmagazine	0.026858	0.454509	0.0591	0.952878

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.48632 +- 0.69736

Goodness of Fit Tests:

Residual Deviance: 5029 on 7.016e+04 degrees of freedom (p-value: 1)
AIC: 5073
Gm: 5221 on 23 degrees of freedom (p-value 0)
R²_L: 0.5094
R²: 0.3336

Predictive Power Tests

Lambda_p: 0.2109 (d= 6.621 ; p-value: 3.558e-11) (for prediction models)
Tau_p: 0.5999 (d= 26.63 ; p-value: 2.813e-156) (for classification models)

Phi_p: 0.4806 (d= 18.66 ; p-value: 9.671e-78) (for selection models)
 Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	69050	157
TRUE	610	362

Fitted Probabilities:

	news newspaper	uk system:unfiled newspapers			
Never Used	0.2527386 0.2226963 0.1753266	0.1622799 0.09681197			
Used Only On Site	0.1746754 0.1520245 0.1174166	0.1081142 0.06285847			
Used Only By User	0.9632588 0.9569118 0.9427917	0.9375627 0.89257549			
Recommended Tag	0.8811907 0.8626874 0.8233887	0.8094518 0.70154248			
	telegraph	media news(international)	daily		
Never Used	0.08239275 0.05250622	0.01875456 0.01596823			
Used Only On Site	0.05319844 0.03351485	0.01181880 0.01005235			
Used Only By User	0.87437531 0.81116444	0.59702805 0.55710621			
Recommended Tag	0.66318987 0.54857761	0.29534341 0.26245586			
	international	noticias	imported	english	world
Never Used	0.015176045 0.01434561 0.012507980 0.006284203 0.006083207				
Used Only On Site	0.009550804 0.00902536 0.007863803 0.003941672 0.003815313				
Used Only By User	0.544317924 0.53011870 0.495420457 0.328951866 0.321772352				
Recommended Tag	0.252574048 0.24194453 0.217382161 0.121788840 0.118333434				
	internet	politics	magazine	Other	
Never Used	0.002834041 0.002180113 0.001682023 0.001637521				
Used Only On Site	0.001775312 0.001365341 0.001053205 0.001025323				
Used Only By User	0.180534452 0.144833276 0.115516360 0.112800391				
Recommended Tag	0.058668101 0.045721710 0.035630904 0.034719438				

[[20]]

Glimpses The Uncanny Valley
<http://www.arclight.net/~pdb/nonfiction/uncanny-valley.html>
 bcf280a637e27240df2658b8be0cd8c4
 Total Number of Users of Site: 166
 Number of Users in Fit: 166
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-4.615325	0.407631	-11.3223	<2e-16 ***
used.onSiteTRUE	0.153552	0.190561	0.8058	0.4204
used.byUserTRUE	2.995092	0.220440	13.5869	<2e-16 ***
used.onSiteTRUE:used.byUserTRUE	0.088308	0.242934	0.3635	0.7162

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classpsychology	3.57462	0.41653	8.5820	< 2.2e-16 ***
tag_classuncannyvalley	3.16174	0.42464	7.4457	9.643e-14 ***
tag_classrobots	2.65675	0.42886	6.1949	5.831e-10 ***
tag_classarticle	2.04723	0.43688	4.6860	2.786e-06 ***
tag_classrobotics	1.67272	0.45665	3.6630	0.0002493 ***
tag_classscience	1.62484	0.41827	3.8846	0.0001025 ***
tag_classuncanny	1.52581	0.49504	3.0822	0.0020546 **

tag_classarticles	1.40906	0.45934	3.0676	0.0021579	**
tag_classaesthetics	1.18562	0.50447	2.3502	0.0187612	*
tag_classanthropomorphism	1.09696	0.54332	2.0190	0.0434872	*
tag_classsystem:unfiled	0.93816	0.56243	1.6680	0.0953063	.
tag_classrealism	0.83783	0.56116	1.4930	0.1354270	
tag_classrobot	0.80973	0.52687	1.5368	0.1243301	
tag_classwriting	0.61702	0.44690	1.3807	0.1673776	
tag_classstoread	0.54458	0.49815	1.0932	0.2742981	
tag_classstheory	0.39859	0.50198	0.7940	0.4271750	
tag_classanimation	-0.18933	0.51575	-0.3671	0.7135526	
tag_classgraphics	-0.20321	0.48490	-0.4191	0.6751629	
tag_classbrain	-0.34234	0.59907	-0.5715	0.5676905	
tag_classinteresting	-0.36309	0.56364	-0.6442	0.5194519	
tag_classmovies	-0.39285	0.50875	-0.7722	0.4400069	
tag_classgames	-0.42126	0.47492	-0.8870	0.3750807	
tag_classresearch	-0.50199	0.53718	-0.9345	0.3500505	
tag_classfilm	-0.67957	0.54410	-1.2490	0.2116726	
tag_classliterature	-0.71632	0.58058	-1.2338	0.2172740	
tag_classdesign	-0.85956	0.48169	-1.7845	0.0743482	.
tag_classreference	-0.87973	0.49045	-1.7937	0.0728547	.
tag_classstech	-0.93509	0.53709	-1.7410	0.0816760	.
tag_classOther	-1.11246	0.37825	-2.9411	0.0032708	**
tag_classart	-1.11870	0.50628	-2.2096	0.0271310	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.80014 +- 0.8945

Goodness of Fit Tests:

Residual Deviance: 3915 on 2.071e+04 degrees of freedom (p-value: 1)

AIC: 3985

Gm: 2300 on 36 degrees of freedom (p:value 0)

R²_L: 0.3701

R²: 0.1926

Predictive Power Tests

Lambda_p: 0.03641 (d= 0.9902 ; p-value: 0.3221) (for prediction models)

Tau_p: 0.501 (d= 19.26 ; p-value: 1.261e-82) (for classification models)

Phi_p: 0.2776 (d= 8.771 ; p-value: 1.777e-18) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	19921	115
TRUE	573	141

Fitted Probabilities:

	psychology	uncannyvalley	robots	article	robotics
Never Used	0.2610132	0.1894506	0.1236218	0.07122028	0.05008716
Used Only On Site	0.2916969	0.2141596	0.1412412	0.08207050	0.05791858
Used Only By User	0.8759237	0.8236838	0.7381781	0.60515653	0.51311854
Recommended Tag	0.8999116	0.8561122	0.7821740	0.66124732	0.57305980
	science	uncanny	articles	aesthetics	anthropomorphism
Never Used	0.04785751	0.04354197	0.03893063	0.03137994	0.02879410
Used Only On Site	0.05536063	0.05040431	0.04510048	0.03639846	0.03341333
Used Only By User	0.50115128	0.47641256	0.44740192	0.39302563	0.37208646

Recommended Tag	0.56130623	0.53679339	0.50767098	0.45196068	0.43010658
	system:unfiled	realism	robot	writing	toread
Never Used	0.02467054	0.02236818	0.02176178	0.01801619	0.01677842
Used Only On Site	0.02864777	0.02598412	0.02528223	0.02094368	0.01950878
Used Only By User	0.33579858	0.31380240	0.30778260	0.26831031	0.25433022
Recommended Tag	0.39168997	0.36806145	0.36154921	0.31835272	0.30284441
	theory	animation	graphics	brain	interesting
Never Used	0.01453235	0.008124987	0.008013897	0.006980238	0.006837869
Used Only On Site	0.01690349	0.009460720	0.009331539	0.008129320	0.007963702
Used Only By User	0.22764662	0.140691157	0.139021583	0.123188560	0.120964727
Recommended Tag	0.27293390	0.172544326	0.170571785	0.151778978	0.149126805
	movies	games	research	film	literature
Never Used	0.006638721	0.006453996	0.005956402	0.004992073	0.004812823
Used Only On Site	0.007732020	0.007517104	0.006938118	0.005815781	0.005607121
Used Only By User	0.117836078	0.114915188	0.106955583	0.091138886	0.088140381
Recommended Tag	0.145390242	0.141896208	0.132347211	0.113251758	0.109613474
	design	reference	tech	Other	art
Never Used	0.004173203	0.004090220	0.003870787	0.003243715	0.003223620
Used Only On Site	0.004862455	0.004765832	0.004510318	0.003780034	0.003756629
Used Only By User	0.077286614	0.075860544	0.072069408	0.061071420	0.060714909
Recommended Tag	0.096395001	0.094652516	0.090013723	0.076503147	0.076063850
	Other				
Never Used	0.003243715				
Used Only On Site	0.003780034				
Used Only By User	0.061071420				
Recommended Tag	0.076503147				

[[21]]

DVDStyler - Home
<http://www.dvdstyler.de/>
c22affe0aa61f58edleabb98983fd0d1
Total Number of Users of Site: 157
Number of Users in Fit: 157
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.48113	0.24489	-22.3823	< 2e-16 ***
used.onSiteTRUE	-0.73047	0.29607	-2.4672	0.01362 *
used.byUserTRUE	2.65604	0.29481	9.0095	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	0.49408	0.33803	1.4617	0.14384

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classdvd	5.99914	0.28165	21.2996	< 2.2e-16 ***
tag_classauthoring	5.31423	0.27136	19.5836	< 2.2e-16 ***
tag_classsoftware	3.93993	0.25333	15.5523	< 2.2e-16 ***
tag_classvideo	3.48101	0.24807	14.0325	< 2.2e-16 ***
tag_classsystem:unfiled	3.32708	0.35910	9.2650	< 2.2e-16 ***
tag_classlinux	2.71712	0.25560	10.6302	< 2.2e-16 ***
tag_classfree	2.61797	0.26355	9.9334	< 2.2e-16 ***
tag_classwindows	2.26271	0.26432	8.5604	< 2.2e-16 ***
tag_classfreeware	2.16148	0.27656	7.8156	5.472e-15 ***

tag_classmultimedia	1.54910	0.41722	3.7129	0.0002049	***
tag_classopensource	1.40454	0.30862	4.5510	5.340e-06	***
tag_classstools	1.36404	0.31206	4.3711	1.236e-05	***
tag_classmedia	1.05422	0.36167	2.9149	0.0035583	**
tag_classvideos	0.85252	0.47488	1.7952	0.0726145	.

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.62496 +- 0.79054

Goodness of Fit Tests:

Residual Deviance: 2495 on 1.097e+04 degrees of freedom (p-value: 1)

AIC: 2533

Gm: 2469 on 20 degrees of freedom (p:value 0)

R²_L: 0.4974

R²: 0.3897

Predictive Power Tests

Lambda_p: 0.2153 (d= 5.681 ; p-value: 1.337e-08) (for prediction models)

Tau_p: 0.5828 (d= 21.71 ; p-value: 1.755e-104) (for classification models)

Phi_p: 0.46 (d= 14.85 ; p-value: 7.061e-50) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	10237	98
TRUE	416	239

Fitted Probabilities:

	dvd	authoring	software	video	system:unfiled
Never Used	0.6266811	0.4583705	0.17636106	0.11919010	0.10395375
Used Only On Site	0.4470828	0.2895905	0.09349672	0.06119217	0.05292435
Used Only By User	0.9598458	0.9233768	0.75303087	0.65834318	0.62292823
Recommended Tag	0.9496771	0.9048877	0.70650249	0.60337181	0.56601497
	linux	free	windows	freeware	multimedia
Never Used	0.05930049	0.05400508	0.03847856	0.03490319	0.019226834
Used Only On Site	0.02946983	0.02676245	0.01891164	0.01712202	0.009354458
Used Only By User	0.47303508	0.44840496	0.36299836	0.33992949	0.218233154
Recommended Tag	0.41475327	0.39090667	0.31029101	0.28905178	0.180586668
	opensource	tools	media	videos	Other
Never Used	0.016682151	0.016030608	0.01181016	0.009673838	0.004147347
Used Only On Site	0.008105591	0.007786365	0.00572379	0.004683204	0.002002009
Used Only By User	0.194575166	0.188306304	0.14543383	0.122113384	0.055983388
Recommended Tag	0.160173649	0.154800119	0.11844298	0.098949407	0.044724683

[[22]]

digg labs / swarm

<http://labs.digg.com/swarm/>

c2c2a5ab753a62bceal07elaelf841c8

Total Number of Users of Site: 501

Number of Users in Fit: 499

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.53312	0.10685	-61.1431	< 2.2e-16 ***

```

used.onSiteTRUE          -0.46853    0.13343   -3.5113 0.0004460 ***
used.byUserTRUE          2.87632    0.14188   20.2726 < 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE 0.59071    0.15772    3.7452 0.0001802 ***

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)	
tag_classdigg	6.3125109	0.1399020	45.1209	< 2.2e-16	***
tag_classswarm	5.8068838	0.1426805	40.6985	< 2.2e-16	***
tag_classvisualization	5.5862149	0.1447177	38.6008	< 2.2e-16	***
tag_classsystem:unfiled	4.4169076	0.1874902	23.5581	< 2.2e-16	***
tag_classflash	4.0674681	0.1499995	27.1165	< 2.2e-16	***
tag_classcool	3.9678010	0.1648299	24.0721	< 2.2e-16	***
tag_classweb2.0	3.7769007	0.1478322	25.5486	< 2.2e-16	***
tag_classnews	2.9276091	0.1583819	18.4845	< 2.2e-16	***
tag_classinterface	2.2541834	0.2323671	9.7010	< 2.2e-16	***
tag_classvisualisation	2.2390115	0.3710100	6.0349	1.591e-09	***
tag_classui	2.1387955	0.2673794	7.9991	1.253e-15	***
tag_classtraffic	1.9110653	0.3165526	6.0371	1.569e-09	***
tag_classtagging	1.7984802	0.2680947	6.7084	1.968e-11	***
tag_classsocial	1.7670326	0.2173801	8.1288	4.337e-16	***
tag_classawesome	1.7492734	0.4470905	3.9126	9.132e-05	***
tag_classlabs	1.7429629	0.4144498	4.2055	2.605e-05	***
tag_classdesign	1.6983548	0.1873628	9.0645	< 2.2e-16	***
tag_classexperimental	1.5734239	0.4190706	3.7546	0.0001736	***
tag_classcommunity	1.4939412	0.2470362	6.0475	1.471e-09	***
tag_classnetwork	1.4241785	0.2834172	5.0250	5.034e-07	***
tag_classvisual	1.4210834	0.4512092	3.1495	0.0016355	**
tag_classsocialsoftware	1.3170261	0.3925479	3.3551	0.0007934	***
tag_classfun	1.2781244	0.2492443	5.1280	2.928e-07	***
tag_classlive	1.2736978	0.4255342	2.9932	0.0027609	**
tag_classtag	1.2422381	0.4163238	2.9838	0.0028467	**
tag_classinteractive	1.1871142	0.3667840	3.2365	0.0012098	**
tag_classinformation	1.1289599	0.3502273	3.2235	0.0012663	**
tag_classtrends	1.0356668	0.3890792	2.6618	0.0077715	**
tag_classdata	1.0348539	0.3881288	2.6663	0.0076699	**
tag_classtech	0.9686161	0.3291522	2.9428	0.0032530	**
tag_classblog	0.9258900	0.2365397	3.9143	9.066e-05	***
tag_classdaily	0.8745296	0.3933596	2.2232	0.0262002	*
tag_classgeek	0.8351513	0.3938173	2.1207	0.0339507	*
tag_classgraphics	0.6572854	0.3358190	1.9573	0.0503168	.
tag_classinspiration	0.6561382	0.3667356	1.7891	0.0735937	.
tag_classanimation	0.6226773	0.3676853	1.6935	0.0903592	.
tag_classtags	0.5948162	0.3883870	1.5315	0.1256449	.
tag_classreference	0.5720605	0.2924418	1.9562	0.0504473	.
tag_classnetworking	0.5148498	0.4403035	1.1693	0.2422801	.
tag_classresearch	0.5062996	0.3500308	1.4464	0.1480530	.
tag_classmap	0.4677309	0.4105146	1.1394	0.2545460	.
tag_classtechnology	0.4030676	0.3333892	1.2090	0.2266627	.
tag_classtools	0.3756883	0.2905903	1.2928	0.1960646	.
tag_classajax	0.3004470	0.3462949	0.8676	0.3856109	.
tag_classinternet	0.2766444	0.3643666	0.7592	0.4477046	.
tag_classwebdesign	0.2162126	0.3636826	0.5945	0.5521717	.
tag_classsearch	0.0025052	0.3625578	0.0069	0.9944869	.

tag_classweb -0.0088744 0.3468290 -0.0256 0.9795865
tag_classmaps -0.1011101 0.4384185 -0.2306 0.8176065

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.73724 +- 0.85863

Goodness of Fit Tests:

Residual Deviance: 1.084e+04 on 1.337e+05 degrees of freedom (p-value: 1)
AIC: 1.094e+04
Gm: 9877 on 55 degrees of freedom (p:value 0)
R^2_L: 0.4768
R^2: 0.2728

Predictive Power Tests

Lambda_p: 0.1044 (d= 4.695 ; p-value: 2.66e-06) (for prediction models)
Tau_p: 0.5454 (d= 34.68 ; p-value: 1.431e-263) (for classification models)
Phi_p: 0.3832 (d= 20.84 ; p-value: 2.07e-96) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	131380	360
TRUE	1424	568

Fitted Probabilities:

	digg	swarm	visualization	system:unfiled	flash
Never Used	0.4450711	0.3260219	0.2795083	0.10753133	0.07830170
Used Only On Site	0.3342259	0.2324086	0.1953791	0.07012722	0.05048974
Used Only By User	0.9343623	0.8956769	0.8731850	0.68137787	0.60124893
Recommended Tag	0.9414697	0.9065543	0.8861093	0.70729729	0.63014855
	cool	web2.0	news	interface	visualisation
Never Used	0.07140428	0.05973655	0.02645478	0.013668032	0.01346500
Used Only On Site	0.04592015	0.03824512	0.01672416	0.008599104	0.00847072
Used Only By User	0.57713046	0.52999005	0.32537310	0.197401793	0.19500907
Recommended Tag	0.60663446	0.56027712	0.35274274	0.217476940	0.21490604
	ui	traffic	tagging	social	awesome
Never Used	0.012196664	0.009736866	0.008709127	0.008441783	0.008294420
Used Only On Site	0.007669165	0.006116810	0.005469062	0.005300647	0.005207828
Used Only By User	0.179756091	0.148586440	0.134899439	0.131271406	0.129259386
Recommended Tag	0.198480504	0.164716048	0.149804692	0.145843373	0.143644936
	labs	design	experimental	community	network
Never Used	0.008242673	0.007885899	0.006966214	0.006437379	0.006006203
Used Only On Site	0.005175237	0.004950571	0.004371707	0.004039031	0.003767887
Used Only By User	0.128550790	0.123635804	0.110723463	0.103136046	0.096859391
Recommended Tag	0.142870418	0.137494320	0.123338316	0.114998243	0.108086680
	visual	socialsoftware	fun	live	
Never Used	0.005987753	0.005399200	0.005194265	0.005171442	
Used Only On Site	0.003756287	0.003386324	0.003257541	0.003243199	
Used Only By User	0.096588978	0.087882327	0.084813589	0.084470627	
Recommended Tag	0.107788663	0.098182176	0.094791157	0.094412012	
	tag	interactive	information	trends	data
Never Used	0.005012085	0.004744552	0.004477705	0.004080488	0.004077185
Used Only On Site	0.003143073	0.002975005	0.002807402	0.002557976	0.002555903
Used Only By User	0.082069272	0.078011091	0.073929635	0.067790875	0.067739519
Recommended Tag	0.091756364	0.087264712	0.082742565	0.075932023	0.075875001

	tech	blog	daily	geek	graphics
Never Used	0.003816869	0.003657808	0.003475321	0.003341576	0.002798606
Used Only On Site	0.002392483	0.002292644	0.002178115	0.002094188	0.001753548
Used Only By User	0.063674426	0.061174085	0.058289999	0.056165661	0.047447971
Recommended Tag	0.071359063	0.068579066	0.065370177	0.063005071	0.053285742
	inspiration	animation	tags	reference	networking
Never Used	0.002795406	0.002703666	0.002629574	0.002570564	0.002427975
Used Only On Site	0.001751541	0.001694000	0.001647532	0.001610524	0.001521107
Used Only By User	0.047396145	0.045908070	0.044703064	0.043741296	0.041409781
Recommended Tag	0.053227896	0.051566635	0.050220920	0.049146545	0.046540940
	research	map	technology	tools	ajax
Never Used	0.002407353	0.002316483	0.002171747	0.002113217	0.001960350
Used Only On Site	0.001508176	0.001451198	0.001360452	0.001323758	0.001227929
Used Only By User	0.041071709	0.039579286	0.037193131	0.036225010	0.033687858
Recommended Tag	0.046162995	0.044494153	0.041824595	0.040741020	0.037900011
	internet	webdesign	search	web	maps
Never Used	0.001914328	0.001802270	0.001455991	0.001439539	0.0013128689
Used Only On Site	0.001199081	0.001128844	0.000911835	0.0009015269	0.0008221589
Used Only By User	0.032921556	0.031050914	0.025226964	0.0249486403	0.0228005420
Recommended Tag	0.037041567	0.034945280	0.028412317	0.0280998625	0.0256875894
	Other				
Never Used	0.0014523532				
Used Only On Site	0.0009095557				
Used Only By User	0.0251654340				
Recommended Tag	0.0283432431				

[[23]]

Flickr: The HDR Pool

<http://www.flickr.com/groups/hdr/pool/c41483478648f6b82e9178c4b0fd7417>

Total Number of Users of Site: 596

Number of Users in Fit: 596

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.76936	0.16795	-40.3051	< 2e-16 ***
used.onSiteTRUE	-0.32584	0.19459	-1.6745	0.09403 .
used.byUserTRUE	3.20806	0.20252	15.8409	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.25780	0.21833	-1.1808	0.23768

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classhdr	6.75149	0.14531	46.4618	< 2.2e-16 ***
tag_classphotography	5.80543	0.14740	39.3858	< 2.2e-16 ***
tag_classflickr	5.77136	0.14677	39.3216	< 2.2e-16 ***
tag_classsystem:unfiled	4.42149	0.19649	22.5018	< 2.2e-16 ***
tag_classphoto	4.17761	0.15976	26.1501	< 2.2e-16 ***
tag_classphotos	3.72899	0.16685	22.3490	< 2.2e-16 ***
tag_classart	3.53063	0.15663	22.5415	< 2.2e-16 ***
tag_classphotoshop	3.21363	0.17391	18.4790	< 2.2e-16 ***
tag_classimages	2.68917	0.20641	13.0286	< 2.2e-16 ***
tag_classhdri	2.30813	0.39268	5.8779	4.156e-09 ***

tag_classfoto	2.28812	0.39087	5.8539	4.801e-09	***
tag_classpictures	2.23156	0.30807	7.2437	4.367e-13	***
tag_classcool	1.84203	0.26392	6.9795	2.963e-12	***
tag_classgallery	1.72548	0.30342	5.6868	1.294e-08	***
tag_classimage	1.39521	0.36146	3.8599	0.0001134	***
tag_classinspiration	1.29085	0.35741	3.6117	0.0003042	***
tag_classcamera	1.28948	0.35724	3.6095	0.0003068	***
tag_classgraphics	1.05255	0.31526	3.3387	0.0008416	***
tag_classdigital	1.04007	0.41330	2.5165	0.0118524	*
tag_classcolor	0.65841	0.45378	1.4509	0.1467951	
tag_classdesign	0.33561	0.33537	1.0007	0.3169646	
tag_classshowto	0.13371	0.38733	0.3452	0.7299351	
tag_classstools	-0.18483	0.44311	-0.4171	0.6765825	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.51333 +- 0.71647

Goodness of Fit Tests:

Residual Deviance: 7620 on 7.924e+04 degrees of freedom (p-value: 1)

AIC: 7676

Gm: 9210 on 29 degrees of freedom (p:value 0)

R²_L: 0.5472

R²: 0.4001

Predictive Power Tests

Lambda_p: 0.2459 (d= 10.41 ; p-value: 2.237e-25) (for prediction models)

Tau_p: 0.6144 (d= 36.78 ; p-value: 3.93e-296) (for classification models)

Phi_p: 0.4826 (d= 24.79 ; p-value: 1.032e-135) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	77311	204
TRUE	1118	635

Fitted Probabilities:

	hdr	photography	flickr	system:unfiled	photo
Never Used	0.4955321	0.2760911	0.2693337	0.08723511	0.06967139
Used Only On Site	0.4149079	0.2158909	0.2101791	0.06454250	0.05129092
Used Only By User	0.9604635	0.9041429	0.9011490	0.70270064	0.64937994
Recommended Tag	0.9312817	0.8403029	0.8356776	0.56869939	0.50816678
	photos	art	photoshop	images	hdri
Never Used	0.04563480	0.03773375	0.02776743	0.01662320	0.011416307
Used Only On Site	0.03336814	0.02752963	0.02020187	0.01205636	0.008267913
Used Only By User	0.54182377	0.49233221	0.41394793	0.29481120	0.222152270
Recommended Tag	0.39748441	0.35107476	0.28265838	0.18911435	0.137428712
	foto	pictures	cool	gallery	image
Never Used	0.011192670	0.010583680	0.007193662	0.006407349	0.004613448
Used Only On Site	0.008105445	0.007663128	0.005203661	0.004633853	0.003334820
Used Only By User	0.218713780	0.209202507	0.151964900	0.137546540	0.102836950
Recommended Tag	0.135073862	0.128601161	0.090881649	0.081700491	0.060101469
	inspiration	camera	graphics	digital	color
Never Used	0.004158170	0.004152525	0.003279410	0.003238851	0.002213529
Used Only On Site	0.003005343	0.003001258	0.002369634	0.002340301	0.001598976
Used Only By User	0.093599986	0.093484309	0.075247449	0.074383239	0.052010952

Recommended Tag	0.054470015	0.054399795	0.043422299	0.042906639	0.029697836
	design	howto	tools	Other	
Never Used	0.001603833	0.001311002	0.0009537118	0.0011471098	
Used Only On Site	0.001158357	0.000946784	0.0006886867	0.0008283862	
Used Only By User	0.038210299	0.031444271	0.0230643631	0.0276175121	
Recommended Tag	0.021682358	0.017788859	0.0129992799	0.0155971978	

[[24]]

Sxip Identity

<http://www.sxip.com/>

c668255d3069ca020886a65c1ca2709c

Total Number of Users of Site: 496

Number of Users in Fit: 496

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-4.81222	0.37455	-12.8480	< 2.2e-16 ***
used.onSiteTRUE	-0.24729	0.14379	-1.7198	0.08547 .
used.byUserTRUE	3.95829	0.15415	25.6773	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.82356	0.17257	-4.7724	1.821e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classidentity	4.882652	0.371350	13.1484	< 2.2e-16 ***
tag_classauthentication	3.461847	0.373017	9.2807	< 2.2e-16 ***
tag_classssxip	3.167495	0.378505	8.3684	< 2.2e-16 ***
tag_classweb2.0	2.651762	0.371879	7.1307	9.985e-13 ***
tag_classsecurity	2.179750	0.373361	5.8382	5.277e-09 ***
tag_classsystem:unfiled	1.979156	0.408217	4.8483	1.245e-06 ***
tag_classidentity2.0	1.835189	0.405148	4.5297	5.907e-06 ***
tag_classssso	1.173056	0.430635	2.7240	0.0064494 **
tag_classsocial	0.845434	0.386614	2.1868	0.0287596 *
tag_classweb	0.609714	0.381822	1.5969	0.1102985
tag_classcompany	0.385437	0.465198	0.8285	0.4073625
tag_classvancouver	0.337707	0.519577	0.6500	0.5157147
tag_classidentity-management	0.076157	0.563347	0.1352	0.8924636
tag_classlogin	-0.098019	0.567081	-0.1728	0.8627710
tag_classinternet	-0.181688	0.423639	-0.4289	0.6680136
tag_classstrust	-0.282727	0.562647	-0.5025	0.6153198
tag_classprivacy	-0.326609	0.464744	-0.7028	0.4821972
tag_classmanagement	-0.328424	0.444413	-0.7390	0.4599031
tag_classstoread	-0.458888	0.526653	-0.8713	0.3835744
tag_classfoaf	-0.464450	0.561291	-0.8275	0.4079716
tag_classservice	-0.509132	0.506369	-1.0055	0.3146768
tag_classopenid	-0.519529	0.567851	-0.9149	0.3602427
tag_classstechnology	-0.609586	0.426569	-1.4290	0.1529913
tag_classstartup	-0.813678	0.518341	-1.5698	0.1164676
tag_classbusiness	-0.842691	0.437915	-1.9243	0.0543136 .
tag_classonline	-1.093213	0.535911	-2.0399	0.0413586 *
tag_classdevelopment	-1.320047	0.501210	-2.6337	0.0084455 **
tag_classstools	-1.435759	0.463381	-3.0984	0.0019454 **
tag_classajax	-1.653603	0.471702	-3.5056	0.0004556 ***

```

tag_classprogramming      -1.759301   0.515402  -3.4135  0.0006415 ***
tag_classOther            -1.916894   0.359304  -5.3350  9.554e-08 ***
tag_classopensource      -2.540697   0.562867  -4.5138  6.366e-06 ***
tag_classblog            -2.850812   0.560672  -5.0846  3.683e-07 ***

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.63427 +- 0.79641

Goodness of Fit Tests:

Residual Deviance: 8893 on 1.126e+05 degrees of freedom (p-value: 1)

AIC: 8969

Gm: 8318 on 39 degrees of freedom (p:value 0)

R²_L: 0.4833

R²: 0.2921

Predictive Power Tests

Lambda_p: 0.1158 (d= 4.737 ; p-value: 2.17e-06) (for prediction models)

Tau_p: 0.5513 (d= 31.9 ; p-value: 2.58e-223) (for classification models)

Phi_p: 0.3847 (d= 18.93 ; p-value: 6.011e-80) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	110666	276
TRUE	1183	467

Fitted Probabilities:

	identity authentication	sxip	web2.0	security
Never Used	0.5176004	0.2058092	0.1618229	0.10335787
Used Only On Site	0.4558990	0.1683077	0.1310142	0.08258336
Used Only By User	0.9825141	0.9313693	0.9099941	0.85788461
Recommended Tag	0.9506342	0.8230378	0.7760360	0.67414240
	system:unfiled identity2.0	sso	social	web
Never Used	0.05556335	0.04847435	0.02560161	0.01858232
Used Only On Site	0.04392474	0.03826048	0.02010535	0.01457046
Used Only By User	0.75495658	0.72735770	0.57911094	0.49787560
Recommended Tag	0.51359052	0.47761706	0.32044565	0.25362961
	company vancouver	identity-management	login	
Never Used	0.011811685	0.011267355	0.008696813	0.007316789
Used Only On Site	0.009247806	0.008820575	0.006804402	0.005722934
Used Only By User	0.384972675	0.373735440	0.314799808	0.278492742
Recommended Tag	0.176630704	0.169795952	0.136034597	0.116829798
	internet	trust	privacy management	toread
Never Used	0.006733462	0.006090304	0.005830350	0.005819842
Used Only On Site	0.005266003	0.004762339	0.004558808	0.004550580
Used Only By User	0.261996007	0.242934340	0.234954917	0.234628895
Recommended Tag	0.108469626	0.099078280	0.095229595	0.095073361
	foaf	service	openid technology	startup
Never Used	0.005083436	0.004862378	0.004812330	0.004399712
Used Only On Site	0.003974137	0.003801134	0.003761967	0.003439098
Used Only By User	0.211087630	0.203742825	0.202061344	0.187929964
Recommended Tag	0.083997563	0.080622946	0.079855665	0.073483799
	business	online development	tools	ajax
Never Used	0.003488071	0.002717194	0.002166945	0.001930623
Used Only On Site	0.002725956	0.002123151	0.001692995	0.001508283

Used Only By User 0.154906887 0.124865000 0.102111717 0.091980421 0.075331632
 Recommended Tag 0.059107451 0.046619537 0.037513199 0.033551724 0.027162377
 programming Other opensource blog Other
 Never Used 0.001397709 0.001194163 0.0006403096 0.0004696595 0.001194163
 Used Only On Site 0.001091821 0.000932779 0.0005000949 0.0003668000 0.000932779
 Used Only By User 0.068291641 0.058921234 0.0324637548 0.0240155965 0.058921234
 Recommended Tag 0.024504666 0.021006879 0.0113684696 0.0083625640 0.021006879

[[25]]

Many Eyes

<http://services.alphaworks.ibm.com/manyeyes/app>
 d934945421022e0bdda78ad71f1ef9b6

Total Number of Users of Site: 466

Number of Users in Fit: 466

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.849080	0.143052	-47.8781	< 2e-16 ***
used.onSiteTRUE	0.322004	0.163950	1.9640	0.04953 *
used.byUserTRUE	3.031681	0.167694	18.0786	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.081758	0.180679	-0.4525	0.65091

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classvisualization	6.324734	0.150637	41.9865	< 2.2e-16 ***
tag_classdata	5.318619	0.143525	37.0571	< 2.2e-16 ***
tag_classgraph	4.693085	0.149639	31.3627	< 2.2e-16 ***
tag_classchart	4.571233	0.156175	29.2699	< 2.2e-16 ***
tag_classibm	4.430531	0.157875	28.0636	< 2.2e-16 ***
tag_classcollaboration	4.003623	0.152814	26.1993	< 2.2e-16 ***
tag_classinformation	3.521132	0.163352	21.5555	< 2.2e-16 ***
tag_classsystem:unfiled	2.858407	0.286920	9.9624	< 2.2e-16 ***
tag_classcharts	2.496509	0.230988	10.8079	< 2.2e-16 ***
tag_classvisualisation	2.399992	0.303686	7.9029	2.725e-15 ***
tag_classweb2.0	2.083261	0.173383	12.0154	< 2.2e-16 ***
tag_classgraphing	1.865672	0.385218	4.8432	1.278e-06 ***
tag_classdatamining	1.861136	0.324658	5.7326	9.890e-09 ***
tag_classstools	1.852645	0.173558	10.6745	< 2.2e-16 ***
tag_classsocialsoftware	1.831637	0.277874	6.5916	4.351e-11 ***
tag_classcool	1.798221	0.222669	8.0758	6.705e-16 ***
tag_classeyes	1.795477	0.418292	4.2924	1.768e-05 ***
tag_classinfovis	1.759509	0.410322	4.2881	1.802e-05 ***
tag_classanalysis	1.671074	0.291854	5.7257	1.030e-08 ***
tag_classanalytics	1.638527	0.332490	4.9281	8.305e-07 ***
tag_classjava	1.526825	0.221091	6.9059	4.990e-12 ***
tag_classresearch	1.485140	0.218417	6.7996	1.049e-11 ***
tag_classgraphics	1.448012	0.221431	6.5394	6.179e-11 ***
tag_classstool	1.315140	0.273841	4.8026	1.566e-06 ***
tag_classstatistics	1.305405	0.237476	5.4970	3.863e-08 ***
tag_classgraphs	1.267121	0.354086	3.5786	0.0003455 ***
tag_classdatabase	1.095088	0.258757	4.2321	2.315e-05 ***
tag_classsocial	1.006150	0.251633	3.9985	6.375e-05 ***

tag_classinteractive	0.994587	0.387857	2.5643	0.0103380	*
tag_classvisual	0.970483	0.442907	2.1912	0.0284397	*
tag_classonline	0.931599	0.298619	3.1197	0.0018104	**
tag_classsocialnetworking	0.883005	0.443582	1.9906	0.0465223	*
tag_classmapping	0.866677	0.385576	2.2477	0.0245924	*
tag_classdesign	0.834267	0.229077	3.6419	0.0002707	***
tag_classknowledge	0.704055	0.444008	1.5857	0.1128116	
tag_classgenerator	0.662179	0.381944	1.7337	0.0829697	.
tag_classcommunity	0.621244	0.297940	2.0851	0.0370573	*
tag_classillustration	0.606318	0.437742	1.3851	0.1660201	
tag_classreference	0.520121	0.265631	1.9581	0.0502235	.
tag_classseconomics	0.390650	0.407325	0.9591	0.3375273	
tag_classgallery	0.367392	0.437581	0.8396	0.4011340	
tag_classpresentation	0.343570	0.405642	0.8470	0.3970064	
tag_classscience	0.128040	0.360378	0.3553	0.7223687	
tag_classfreeware	0.097137	0.435613	0.2230	0.8235446	
tag_classweb	0.072435	0.315810	0.2294	0.8185876	
tag_classeducation	0.060558	0.381335	0.1588	0.8738225	
tag_classstechnology	-0.299876	0.435148	-0.6891	0.4907377	
tag_classsoftware	-0.486394	0.402334	-1.2089	0.2266899	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.60113 +- 0.77533

Goodness of Fit Tests:

Residual Deviance: 1.034e+04 on 9.641e+04 degrees of freedom (p-value: 1)
AIC: 1.045e+04
Gm: 9276 on 54 degrees of freedom (p:value 0)
R^2_L: 0.4729
R^2: 0.2943

Predictive Power Tests

Lambda_p: 0.1421 (d= 6.454 ; p-value: 1.092e-10) (for prediction models)
Tau_p: 0.5619 (d= 36.08 ; p-value: 3.995e-285) (for classification models)
Phi_p: 0.3863 (d= 20.84 ; p-value: 2.073e-96) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	94166	276
TRUE	1457	563

Fitted Probabilities:

	visualization	data	graph	chart	ibm
Never Used	0.3718366	0.1779262	0.1037723	0.09297438	0.08176915
Used Only On Site	0.4495863	0.2299741	0.1377635	0.12391764	0.10943307
Used Only By User	0.9246545	0.8177564	0.7059275	0.68001367	0.64865503
Recommended Tag	0.9397766	0.8508732	0.7532334	0.72989313	0.70127534
	collaboration	information	system:unfiled	charts	
Never Used	0.05491663	0.03462477	0.01815170	0.01271005	
Used Only On Site	0.07423030	0.04715807	0.02487581	0.01745419	
Used Only By User	0.54642203	0.42647048	0.27708016	0.21067033	
Recommended Tag	0.60503048	0.48599854	0.32766925	0.25338421	
	visualisation	web2.0	graphing	datamining	tools
Never Used	0.01155416	0.008444006	0.006804063	0.006773477	0.006716592

Used Only On Site	0.01587380	0.011614542	0.009364653	0.009322664	0.009244569
Used Only By User	0.19506845	0.150059150	0.124365206	0.123872067	0.122953494
Recommended Tag	0.23556306	0.183338366	0.152971740	0.152384910	0.151291411
	socialsoftware	cool	eyes	infovis	
Never Used	0.006577881	0.006363083	0.006345758	0.006122942	
Used Only On Site	0.009054126	0.008759180	0.008735389	0.008429379	
Used Only By User	0.120705974	0.117204058	0.116920460	0.113257625	
Recommended Tag	0.148613664	0.144435088	0.144096354	0.139716861	
	analysis	analytics	java	research	graphics
Never Used	0.005607616	0.005429018	0.004858019	0.004660598	0.004491496
Used Only On Site	0.007721444	0.007476028	0.006691183	0.006419746	0.006187214
Used Only By User	0.104675148	0.101663906	0.091906687	0.088486319	0.085537123
Recommended Tag	0.129421987	0.125798867	0.114019288	0.109875557	0.106296604
	tool	statistics	graphs	database	social
Never Used	0.003934848	0.003896877	0.003751060	0.003160082	0.002891945
Used Only On Site	0.005421553	0.005369313	0.005168685	0.004355337	0.003986187
Used Only By User	0.075699968	0.075021627	0.072407861	0.061669614	0.056719353
Recommended Tag	0.094318257	0.093489953	0.090295561	0.077125095	0.071028122
	interactive	visual	online	socialnetworking	
Never Used	0.002858792	0.002790899	0.002684745	0.002557728	
Used Only On Site	0.003940539	0.003847055	0.003700878	0.003525958	
Used Only By User	0.056103846	0.054840989	0.052860004	0.050479327	
Recommended Tag	0.070268907	0.068710415	0.066263557	0.063319576	
	mapping	design	knowledge	generator	community
Never Used	0.002516408	0.002436354	0.002139536	0.002051970	0.001969831
Used Only On Site	0.003469050	0.003358792	0.002949926	0.002829287	0.002716118
Used Only By User	0.049702404	0.048193760	0.042560206	0.040886106	0.039310703
Recommended Tag	0.062358009	0.060489674	0.053499593	0.051418303	0.049458005
	illustration	reference	economics	gallery	presentation
Never Used	0.001940705	0.001780713	0.001564803	0.001528883	0.001492946
Used Only On Site	0.002675986	0.002455526	0.002157972	0.002108466	0.002058934
Used Only By User	0.038750881	0.035664689	0.031469893	0.030768675	0.030066136
Recommended Tag	0.048761010	0.044914843	0.039676831	0.038800070	0.037921325
	science	freeware	web	education	technology
Never Used	0.001203834	0.001167242	0.001138795	0.001125365	0.000785067
Used Only On Site	0.001660399	0.001609952	0.001570732	0.001552216	0.001082983
Used Only By User	0.024378854	0.023654527	0.023090705	0.022824301	0.016027771
Recommended Tag	0.030795340	0.029886211	0.029178291	0.028843727	0.020291978
	software	Other			
Never Used	0.0006515703	0.001059308			
Used Only On Site	0.0008988728	0.001461140			
Used Only By User	0.0133369240	0.021511980			
Recommended Tag	0.0168975218	0.027194941			

[[26]]

Obscure Sound - Indie Music Blog
<http://obscuresound.com/>
e2325a57fd50fbfa9ef65252e2269492
Total Number of Users of Site: 116
Number of Users in Fit: 116
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.176688	0.332822	-15.5539	< 2.2e-16 ***

```

used.onSiteTRUE          -0.472729   0.364469  -1.2970   0.1946
used.byUserTRUE          2.899205   0.431712   6.7156  1.873e-11 ***
used.onSiteTRUE:used.byUserTRUE  0.047964   0.465324   0.1031   0.9179

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classmusic	4.56071	0.34289	13.3007	< 2.2e-16 ***
tag_classindie	3.80723	0.33525	11.3564	< 2.2e-16 ***
tag_classmp3	3.78032	0.33049	11.4385	< 2.2e-16 ***
tag_classmp3blog	3.65311	0.34677	10.5346	< 2.2e-16 ***
tag_classblog	3.43983	0.33898	10.1475	< 2.2e-16 ***
tag_classsystem:unfiled	3.07146	0.40470	7.5894	3.213e-14 ***
tag_classblogs	2.38274	0.37608	6.3357	2.362e-10 ***
tag_classdownload	2.01316	0.44480	4.5260	6.011e-06 ***
tag_classaudio	1.03329	0.51854	1.9927	0.04629 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.61925 +- 0.78693

Goodness of Fit Tests:

Residual Deviance: 1133 on 4161 degrees of freedom (p-value: 1)

AIC: 1161

Gm: 1136 on 15 degrees of freedom (p:value 8.558e-233)

R²_L: 0.5008

R²: 0.4297

Predictive Power Tests

Lambda_p: 0.3354 (d= 6.265 ; p-value: 3.729e-10) (for prediction models)

Tau_p: 0.6399 (d= 16.85 ; p-value: 1.124e-63) (for classification models)

Phi_p: 0.6014 (d= 14.93 ; p-value: 2.243e-50) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	3781	73
TRUE	141	181

Fitted Probabilities:

	music	indie	mp3	mp3blog	blog
Never Used	0.3506959	0.2027066	0.1983928	0.1789354	0.14971240
Used Only On Site	0.2518611	0.1367922	0.1336460	0.1195913	0.09889279
Used Only By User	0.9074780	0.8219686	0.8179970	0.7982879	0.76175882
Recommended Tag	0.8651171	0.7511914	0.7461285	0.7212886	0.67646666

	system:unfiled	blogs	download	audio	Other
Never Used	0.10858951	0.05765192	0.04056173	0.015621026	0.005614971
Used Only On Site	0.07057044	0.03673210	0.02567437	0.009794199	0.003507223
Used Only By User	0.68868414	0.52628920	0.43430252	0.223707814	0.093005053
Recommended Tag	0.59126789	0.42079478	0.33423697	0.158563735	0.062840806

[[27]]

index.html - JotSpot Wiki (dojomanual)

<http://manual.dojotoolkit.org/index.html>

ea44602f3f307b93679195d4431c225d

Total Number of Users of Site: 218

Number of Users in Fit: 218

Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.24359	0.27958	-22.3321	< 2e-16 ***
used.onSiteTRUE	0.08551	0.30083	0.2843	0.77622
used.byUserTRUE	3.81976	0.33844	11.2863	< 2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-1.15149	0.35141	-3.2768	0.00105 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classdojo	6.302757	0.243974	25.8337	< 2.2e-16 ***
tag_classmanual	4.801595	0.244126	19.6685	< 2.2e-16 ***
tag_classajax	4.531513	0.235560	19.2372	< 2.2e-16 ***
tag_classjavascript	4.349905	0.235664	18.4581	< 2.2e-16 ***
tag_classdocumentation	4.140649	0.249389	16.6032	< 2.2e-16 ***
tag_classreference	3.139983	0.252456	12.4377	< 2.2e-16 ***
tag_classsystem:unfiled	2.797672	0.369620	7.5690	3.760e-14 ***
tag_classframework	2.760640	0.266492	10.3592	< 2.2e-16 ***
tag_classdocs	1.929746	0.471501	4.0928	4.262e-05 ***
tag_class toolkit	1.740600	0.427249	4.0740	4.622e-05 ***
tag_classwebdev	1.696404	0.383986	4.4179	9.967e-06 ***
tag_classapi	1.464293	0.352345	4.1559	3.241e-05 ***
tag_classprogramming	1.371089	0.299739	4.5743	4.779e-06 ***
tag_classlibrary	1.089032	0.382662	2.8459	0.004428 **
tag_classweb2.0	1.032855	0.355004	2.9094	0.003621 **
tag_classweb	0.981484	0.334309	2.9359	0.003326 **
tag_class tools	0.469910	0.433479	1.0840	0.278346
tag_classdevelopment	0.456201	0.433578	1.0522	0.292718
tag_classwiki	0.438994	0.450060	0.9754	0.329356
tag_class tutorial	0.285186	0.406761	0.7011	0.483232
tag_classopensource	0.238845	0.476100	0.5017	0.615900
tag_classdesign	-0.092029	0.476529	-0.1931	0.846862
tag_classwebdesign	-0.107341	0.481981	-0.2227	0.823762

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.95385 +- 0.97665

Goodness of Fit Tests:

Residual Deviance: 3139 on 1.370e+04 degrees of freedom (p-value: 1)

AIC: 3195

Gm: 3150 on 29 degrees of freedom (p-value 0)

R²_L: 0.5009

R²: 0.377

Predictive Power Tests

Lambda_p: 0.2314 (d= 6.896 ; p-value: 5.36e-12) (for prediction models)

Tau_p: 0.5909 (d= 24.85 ; p-value: 2.753e-136) (for classification models)

Phi_p: 0.5377 (d= 21.12 ; p-value: 5.481e-99) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	12682	218
TRUE	423	411

Fitted Probabilities:

	dojo	manual	ajax	javascript	documentation
Never Used	0.5147867	0.1912361	0.1528941	0.1308245	0.1088110
Used Only On Site	0.5361055	0.2048116	0.1643007	0.1408584	0.1173846
Used Only By User	0.9797456	0.9151154	0.8916470	0.8728135	0.8477177
Recommended Tag	0.9433713	0.7878113	0.7391780	0.7026799	0.6571990
	reference	system:unfiled	framework	docs	toolkit
Never Used	0.04295858	0.03089073	0.02980119	0.01320526	0.01095447
Used Only On Site	0.04661482	0.03355589	0.03237548	0.01436719	0.01192074
Used Only By User	0.67175767	0.59238527	0.58341377	0.37893032	0.33553923
Recommended Tag	0.41342379	0.33355714	0.32537650	0.17363719	0.14814663
	webdev	api	programming	library	web2.0
Never Used	0.01048583	0.008331871	0.007596034	0.005739877	0.005428018
Used Only On Site	0.01141125	0.009068932	0.008268543	0.006249088	0.005909727
Used Only By User	0.32575827	0.276969392	0.258697735	0.208365654	0.199250953
Recommended Tag	0.14265539	0.116549466	0.107290311	0.083113232	0.078931250
	web	tools	development	wiki	tutorial
Never Used	0.005157617	0.003098655	0.003056596	0.003004608	0.002577363
Used Only On Site	0.005615465	0.003374346	0.003328558	0.003271959	0.002806805
Used Only By User	0.191181130	0.124125718	0.122643023	0.120803492	0.105396443
Recommended Tag	0.075276369	0.046534795	0.045930329	0.045182162	0.038991981
	opensource	design	webdesign	Other	
Never Used	0.002460939	0.001768909	0.001742075	0.001939094	
Used Only On Site	0.002680044	0.001926519	0.001897299	0.002111837	
Used Only By User	0.101106298	0.074753359	0.073701133	0.081372950	
Recommended Tag	0.037292143	0.027071182	0.026670782	0.029603497	

[[28]]

Basket Note Pads
<http://basket.kde.org/>
 fff546055fbfecdee79bd0281e80d5ca
 Total Number of Users of Site: 124
 Number of Users in Fit: 124
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.13398	0.18703	-27.4499	< 2.2e-16 ***
used.onSiteTRUE	-0.72117	0.23635	-3.0513	0.002279 **
used.byUserTRUE	3.21058	0.23453	13.6896	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-0.32243	0.27802	-1.1597	0.246161

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classkde	5.40483	0.27059	19.9746	< 2.2e-16 ***
tag_classlinux	3.78224	0.26187	14.4433	< 2.2e-16 ***
tag_classsoftware	3.76153	0.26138	14.3913	< 2.2e-16 ***
tag_classproductivity	3.37702	0.27552	12.2569	< 2.2e-16 ***

tag_classnotes	3.14172	0.31933	9.8385	< 2.2e-16	***
tag_classdesktop	3.07802	0.28699	10.7253	< 2.2e-16	***
tag_classtools	3.00970	0.27356	11.0020	< 2.2e-16	***
tag_classopensource	2.96551	0.27037	10.9683	< 2.2e-16	***
tag_classgtd	2.32572	0.34989	6.6471	2.990e-11	***
tag_classnotetaking	1.91783	0.44544	4.3055	1.666e-05	***
tag_classnote	1.88880	0.49355	3.8269	0.0001297	***
tag_classorganization	1.87951	0.41174	4.5648	4.999e-06	***
tag_classapplication	1.70153	0.48925	3.4778	0.0005055	***
tag_classorganizer	1.69789	0.50146	3.3859	0.0007094	***
tag_classmanagement	1.56620	0.39721	3.9431	8.045e-05	***
tag_classpim	1.54397	0.45758	3.3742	0.0007403	***
tag_classresearch	0.82188	0.48147	1.7070	0.0878193	.
tag_classfree	0.67292	0.39701	1.6949	0.0900862	.

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.69026 +- 0.83082

Goodness of Fit Tests:

Residual Deviance: 2550 on 1.225e+04 degrees of freedom (p-value: 1)
AIC: 2596
Gm: 2183 on 24 degrees of freedom (p:value 0)
R^2_L: 0.4612
R^2: 0.3431

Predictive Power Tests

Lambda_p: 0.1339 (d= 3.333 ; p-value: 0.0008577) (for prediction models)
Tau_p: 0.5451 (d= 19.17 ; p-value: 7.017e-82) (for classification models)
Phi_p: 0.4822 (d= 15.79 ; p-value: 3.39e-56) (for selection models)
Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	11506	180
TRUE	331	259

Fitted Probabilities:

	kde	linux	software	productivity	notes
Never Used	0.5673025	0.2055867	0.2022237	0.14717199	0.12001795
Used Only On Site	0.3892856	0.1117582	0.1097181	0.07740593	0.06218551
Used Only By User	0.9701547	0.8651616	0.8627264	0.81055481	0.77176689
Recommended Tag	0.9196668	0.6932246	0.6888012	0.60109225	0.54356774
	desktop	tools	opensource		gtd notetaking
Never Used	0.1134517	0.10675956	0.10261783	0.05687949	0.03856245
Used Only On Site	0.0585727	0.05491717	0.05266807	0.02848638	0.01912740
Used Only By User	0.7603532	0.74768408	0.73925597	0.59924395	0.49860616
Recommended Tag	0.5277255	0.51067278	0.49962601	0.34495611	0.25938330
	note	organization	application	organizer	management
Never Used	0.03750047	0.03716651	0.03129644	0.03118651	0.0274441
Used Only On Site	0.01859030	0.01842152	0.01546448	0.01540927	0.0135337
Used Only By User	0.49134966	0.48902746	0.44475669	0.44385986	0.4116372
Recommended Tag	0.25384571	0.25208968	0.22003198	0.21940924	0.1976888
	pim	research	free		Other
Never Used	0.02685687	0.013228002	0.011418188	0.005858530	
Used Only On Site	0.01324006	0.006475242	0.005584091	0.002856918	

Used Only By User 0.40626332 0.249453729 0.222615554 0.127482286
 Recommended Tag 0.19418618 0.104787294 0.091613636 0.048938864

[[29]]
 101 Cookbooks
<http://www.101cookbooks.com/>
 b9181c4687575fb372e39af32f8c49e2
 Total Number of Users of Site: 1427
 Number of Users in Fit: 1000
 Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-7.99067	0.12369	-64.6040	< 2.2e-16 ***
used.onSiteTRUE	0.10863	0.14881	0.7300	0.4654
used.byUserTRUE	4.29687	0.15840	27.1267	< 2.2e-16 ***
used.onSiteTRUE:used.byUserTRUE	-1.10583	0.17012	-6.5001	8.024e-11 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)
tag_classcooking	6.77400	0.12355	54.8295	< 2.2e-16 ***
tag_classrecipes	6.71510	0.12334	54.4453	< 2.2e-16 ***
tag_classfood	6.62546	0.12408	53.3982	< 2.2e-16 ***
tag_classrecipe	4.80598	0.15272	31.4699	< 2.2e-16 ***
tag_classsystem:unfiled	4.75186	0.17780	26.7262	< 2.2e-16 ***
tag_classfoodblog	4.73222	0.16489	28.6989	< 2.2e-16 ***
tag_classblog	4.64266	0.13018	35.6636	< 2.2e-16 ***
tag_classblogs	4.00063	0.15456	25.8842	< 2.2e-16 ***
tag_classcookbooks	3.92841	0.23023	17.0631	< 2.2e-16 ***
tag_classcookbook	3.32181	0.28211	11.7748	< 2.2e-16 ***
tag_classfoodblogs	2.94831	0.33613	8.7714	< 2.2e-16 ***
tag_classcuisine	2.92733	0.32695	8.9535	< 2.2e-16 ***
tag_classkitchen	2.89870	0.28472	10.1807	< 2.2e-16 ***
tag_classdaily	2.65502	0.23125	11.4810	< 2.2e-16 ***
tag_classrecipies	2.60528	0.38521	6.7632	1.350e-11 ***
tag_classcook	2.17557	0.45326	4.7998	1.588e-06 ***
tag_classchef	2.16460	0.45334	4.7748	1.799e-06 ***
tag_classbook	2.03836	0.31157	6.5422	6.063e-11 ***
tag_classweblog	2.03392	0.43338	4.6931	2.691e-06 ***
tag_classreference	1.76827	0.22859	7.7357	1.028e-14 ***
tag_classvegan	1.73286	0.45170	3.8363	0.0001249 ***
tag_classforum	1.69895	0.35190	4.8280	1.379e-06 ***
tag_classvegetarian	1.66659	0.45072	3.6976	0.0002177 ***
tag_classdiy	1.62613	0.32036	5.0759	3.856e-07 ***
tag_classresource	1.53836	0.44037	3.4934	0.0004770 ***
tag_classcommunity	1.49756	0.34542	4.3354	1.455e-05 ***
tag_classshowto	1.45756	0.27419	5.3159	1.061e-07 ***
tag_classinspiration	1.20724	0.43415	2.7807	0.0054237 **
tag_classphotography	1.15462	0.28742	4.0172	5.889e-05 ***
tag_classlinks	1.12391	0.38050	2.9538	0.0031393 **
tag_classphotos	1.10720	0.34218	3.2357	0.0012135 **
tag_classguide	1.10133	0.46839	2.3513	0.0187084 *
tag_classinformation	1.05112	0.46840	2.2441	0.0248283 *

tag_classbooks	0.96412	0.31582	3.0527	0.0022678	**
tag_classtools	0.70858	0.35246	2.0104	0.0443885	*
tag_classreviews	0.65708	0.46051	1.4268	0.1536248	
tag_classhealth	0.59107	0.44564	1.3264	0.1847238	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.62729 +- 0.79201

Goodness of Fit Tests:

Residual Deviance: 1.201e+04 on 2.88e+05 degrees of freedom (p-value: 1)

AIC: 1.21e+04

Gm: 1.823e+04 on 43 degrees of freedom (p:value 0)

R²_L: 0.6028

R²: 0.4019

Predictive Power Tests

Lambda_p: 0.2595 (d= 13.45 ; p-value: 2.968e-41) (for prediction models)

Tau_p: 0.6263 (d= 45.92 ; p-value: 0) (for classification models)

Phi_p: 0.4987 (d= 31.5 ; p-value: 9.378e-218) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	285035	302
TRUE	1670	993

Fitted Probabilities:

	cooking	recipes	food	recipe	system:unfiled
Never Used	0.2285228	0.2183049	0.2033954	0.03974572	0.03773093
Used Only On Site	0.2482371	0.2374090	0.2215645	0.04410557	0.04187941
Used Only By User	0.9560685	0.9535271	0.9493897	0.75253423	0.74231892
Recommended Tag	0.8892406	0.8833050	0.8737436	0.52871406	0.51521151
	foodblog	blog	blogs	cookbooks	cookbook
Never Used	0.03702427	0.03396032	0.01816303	0.01691887	0.009295766
Used Only On Site	0.04109837	0.03771047	0.02020521	0.01882384	0.010351459
Used Only By User	0.73854415	0.72088543	0.57611202	0.55838443	0.408061010
Recommended Tag	0.51030454	0.48791896	0.33395331	0.31808561	0.202752631
	foodblogs	cuisine	kitchen	daily	recipies
Never Used	0.006417012	0.006284648	0.006108376	0.004793670	0.004562121
Used Only On Site	0.007148132	0.007000793	0.006804573	0.005340829	0.005082986
Used Only By User	0.321803728	0.317243016	0.311075272	0.261384752	0.251896153
Recommended Tag	0.148972036	0.146332248	0.142792368	0.115477204	0.110492887
	cook	chef	book	weblog	reference
Never Used	0.002973312	0.002940965	0.002593072	0.002581613	0.001980550
Used Only On Site	0.003313384	0.003277350	0.002889780	0.002877014	0.002207326
Used Only By User	0.179722171	0.178110481	0.160374358	0.159777336	0.127246377
Recommended Tag	0.074784060	0.074028493	0.065826679	0.065554148	0.051041813
	vegan	forum	vegetarian	diy	resource
Never Used	0.001911764	0.001848148	0.001789405	0.001718566	0.001574387
Used Only On Site	0.002130681	0.002059795	0.001994339	0.001915402	0.001754738
Used Only By User	0.123364796	0.119744548	0.116375420	0.112278588	0.103823622
Recommended Tag	0.049353358	0.047786645	0.046335549	0.044579976	0.040987570
	community	howto	inspiration	photography	links
Never Used	0.001511536	0.001452355	0.001131105	0.001073184	0.001040761
Used Only On Site	0.001684700	0.001618750	0.001260741	0.001196189	0.001160055

Used Only By User	0.100088098	0.096542521	0.076805855	0.073156625	0.071101444
Recommended Tag	0.039413427	0.037926639	0.029778142	0.028294829	0.027462603
	photos	guide	information	books	
Never Used	0.001023529	0.001017547	0.0009677669	0.0008872037	
Used Only On Site	0.001140850	0.001134183	0.0010787031	0.0009889141	
Used Only By User	0.070005497	0.069624453	0.0664415490	0.0612446637	
Recommended Tag	0.027019732	0.026865903	0.0255839681	0.0235024028	
	tools	reviews	health	Other	
Never Used	0.0006872737	0.0006527954	0.0006111249	0.0003384920	
Used Only On Site	0.0007660814	0.0007276524	0.0006812068	0.0003773210	
Used Only By User	0.0480980573	0.0457941230	0.0429948796	0.0242733964	
Recommended Tag	0.0182995149	0.0173968674	0.0163037956	0.0090941071	

[[30]]

Snipplr - Code 2.0
<http://snipplr.com/>
7a5886991afef3a8cf539d011e58ead3
Total Number of Users of Site: 1137
Number of Users in Fit: 850
Type of Data: bysite

Main Effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-6.52001	0.42436	-15.3645	< 2e-16	***
used.onSiteTRUE	0.43035	0.12925	3.3295	0.00087	***
used.byUserTRUE	3.53595	0.13729	25.7554	< 2e-16	***
used.onSiteTRUE:used.byUserTRUE	-0.14401	0.14510	-0.9925	0.32098	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

Tag Effects (in order of likelihood):

	Estimate	Std. Error	z value	Pr(> z)	
tag_classcode	5.2026805	0.4136093	12.5787	< 2.2e-16	***
tag_classprogramming	4.1396694	0.4133828	10.0141	< 2.2e-16	***
tag_classsnippets	2.9388801	0.4247409	6.9192	4.541e-12	***
tag_classreference	2.8021117	0.4140949	6.7668	1.316e-11	***
tag_classsnippet	2.1292389	0.4511837	4.7192	2.367e-06	***
tag_classphp	2.0221716	0.4157046	4.8644	1.148e-06	***
tag_classprogramacion	1.9758039	0.4450899	4.4391	9.033e-06	***
tag_classsystem:unfiled	1.9131317	0.4699627	4.0708	4.685e-05	***
tag_classhtml	1.8426235	0.4183060	4.4050	1.058e-05	***
tag_classrepository	1.8355523	0.4535642	4.0470	5.189e-05	***
tag_classcss	1.7171231	0.4152080	4.1356	3.541e-05	***
tag_classtextmate	1.6472560	0.4584584	3.5930	0.0003269	***
tag_classcodigo	1.6406117	0.4774917	3.4359	0.0005906	***
tag_classprogramaci\303\263n	1.5496807	0.4771130	3.2480	0.0011620	**
tag_classscripts	1.4120243	0.4421945	3.1932	0.0014070	**
tag_classweb	1.3979176	0.4175326	3.3480	0.0008138	***
tag_classcoding	1.3478291	0.4593638	2.9341	0.0033449	**
tag_classrecursos	1.1969001	0.4806457	2.4902	0.0127674	*
tag_classresource	1.1310599	0.4657709	2.4284	0.0151672	*
tag_classsocial	1.0830160	0.4297032	2.5204	0.0117228	*
tag_classwebdev	1.0595880	0.4397234	2.4097	0.0159670	*
tag_classsharing	1.0212176	0.4522077	2.2583	0.0239274	*
tag_classshare	0.9245173	0.4840350	1.9100	0.0561304	.

tag_classexamples	0.8933844	0.4782189	1.8681	0.0617412	.
tag_classdev	0.8426576	0.4995331	1.6869	0.0916244	.
tag_classdevelopment	0.7715536	0.4327064	1.7831	0.0745720	.
tag_classtagging	0.7635115	0.4548410	1.6786	0.0932234	.
tag_classjavascript	0.7558481	0.4235451	1.7846	0.0743303	.
tag_classsource	0.7472162	0.4831385	1.5466	0.1219627	.
tag_classcommunity	0.7072198	0.4443387	1.5916	0.1114694	.
tag_classdesarrollo	0.6907356	0.5291177	1.3054	0.1917403	.
tag_classtags	0.6552763	0.4525164	1.4481	0.1475969	.
tag_classsnipplr	0.6306293	0.5880681	1.0724	0.2835519	.
tag_classtools	0.6071724	0.4269800	1.4220	0.1550216	.
tag_classresources	0.5359458	0.4623369	1.1592	0.2463704	.
tag_classdevelop	0.4756411	0.5726022	0.8307	0.4061623	.
tag_classweb2	0.4044437	0.5571781	0.7259	0.4679132	.
tag_classweb2.0	0.4031038	0.4317507	0.9336	0.3504847	.
tag_classsearch	0.3820099	0.4350219	0.8781	0.3798681	.
tag_classutilidades	0.3376165	0.6030998	0.5598	0.5756145	.
tag_classruby	0.3026161	0.4444763	0.6808	0.4959744	.
tag_classjava	0.2704220	0.4457713	0.6066	0.5440909	.
tag_classtag	0.2097310	0.5127964	0.4090	0.6825436	.
tag_classtutoriales	0.1611149	0.5699259	0.2827	0.7774111	.
tag_classajax	0.1344633	0.4326715	0.3108	0.7559720	.
tag_classtool	0.1226272	0.4732098	0.2591	0.7955278	.
tag_classc	0.0918098	0.5696836	0.1612	0.8719679	.
tag_classfree	0.0784238	0.4417410	0.1775	0.8590894	.
tag_classshowto	0.0023616	0.4468112	0.0053	0.9957828	.
tag_classwebmaster	-0.0512528	0.5961793	-0.0860	0.9314912	.
tag_classperl	-0.0570092	0.4903654	-0.1163	0.9074476	.
tag_classactionscript	-0.0721495	0.5229779	-0.1380	0.8902728	.
tag_classwebdesign	-0.0757332	0.4453126	-0.1701	0.8649571	.
tag_classtips	-0.0988696	0.4579774	-0.2159	0.8290788	.
tag_classdeveloper	-0.1382148	0.5937046	-0.2328	0.8159162	.
tag_classsoftware	-0.1917691	0.4397940	-0.4360	0.6628056	.
tag_classfolksonomy	-0.2198894	0.5517949	-0.3985	0.6902629	.
tag_classarchive	-0.2394456	0.5691992	-0.4207	0.6739953	.
tag_classxhtml	-0.2743048	0.4913357	-0.5583	0.5766505	.
tag_classservice	-0.3062400	0.5408563	-0.5662	0.5712488	.
tag_classwork	-0.3062914	0.5375094	-0.5698	0.5687900	.
tag_classlist	-0.3681731	0.5098054	-0.7222	0.4701817	.
tag_classtutorial	-0.3706259	0.4515223	-0.8208	0.4117394	.
tag_classbookmarks	-0.4071940	0.5284013	-0.7706	0.4409351	.
tag_classpython	-0.4526535	0.5104380	-0.8868	0.3751896	.
tag_classonline	-0.4658716	0.4973998	-0.9366	0.3489572	.
tag_classdel.icio.us	-0.6563270	0.4915726	-1.3352	0.1818247	.
tag_classdesign	-0.7369985	0.4641176	-1.5880	0.1122962	.
tag_classinternet	-0.8521634	0.4955311	-1.7197	0.0854875	.
tag_classlibrary	-0.8874212	0.5380391	-1.6494	0.0990735	.
tag_classdhtml	-0.9522025	0.5899567	-1.6140	0.1065229	.
tag_classlinks	-0.9650278	0.5522653	-1.7474	0.0805682	.
tag_classproductivity	-1.0375777	0.5522515	-1.8788	0.0602699	.
tag_classopensource	-1.0997470	0.5006263	-2.1967	0.0280388	*
tag_classtemplates	-1.1040862	0.5903950	-1.8701	0.0614726	.
tag_classmysql	-1.1295883	0.5507604	-2.0510	0.0402707	*
tag_classOther	-1.4770398	0.4100102	-3.6024	0.0003152	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standard Deviation on normal distribution of user values: 0.88188 +- 0.93909

Goodness of Fit Tests:

Residual Deviance: 2.087e+04 on 3.025e+05 degrees of freedom (p-value: 1)

AIC: 2.104e+04

Gm: 2.033e+04 on 83 degrees of freedom (p-value 0)

R²_L: 0.4934

R²: 0.2622

Predictive Power Tests

Lambda_p: 0.1199 (d= 7.483 ; p-value: 7.255e-14) (for prediction models)

Tau_p: 0.5543 (d= 48.91 ; p-value: 0) (for classification models)

Phi_p: 0.3528 (d= 25.73 ; p-value: 5.664e-146) (for selection models)

Actual vs. Predicted Values:

	FALSE	TRUE
FALSE	298277	479
TRUE	2904	940

Fitted Probabilities:

	code	programming	snippets	reference	snippet
Never Used	0.2112636	0.08468447	0.02709003	0.02370927	0.01223956
Used Only On Site	0.2917351	0.12455518	0.04106084	0.03600109	0.01869891
Used Only By User	0.9019093	0.76053409	0.48870705	0.45463811	0.29842259
Recommended Tag	0.9244892	0.80875721	0.56000088	0.52607563	0.36158865
	php	programacion	system:unfiled	html	repository
Never Used	0.01101050	0.01051687	0.009884299	0.00921758	0.009153224
Used Only On Site	0.01683228	0.01608189	0.015119709	0.01410488	0.014006884
Used Only By User	0.27650035	0.26732137	0.255226677	0.24205680	0.240761843
Recommended Tag	0.33725681	0.32697241	0.313333042	0.29836600	0.296887803
	css	textmate	codigo	programaci\303\263n	
Never Used	0.008139265	0.00759418	0.007544269		0.006893046
Used Only On Site	0.012462017	0.01163083	0.011554701		0.010560981
Used Only By User	0.219782138	0.20803620	0.206943641		0.192417329
Recommended Tag	0.272774468	0.25913707	0.257863520		0.240848412
	scripts	web	coding	recursos	resource
Never Used	0.006011917	0.005928202	0.005640215	0.004853906	0.004546023
Used Only On Site	0.009215334	0.009087421	0.008647297	0.007444905	0.006973828
Used Only By User	0.171926454	0.169927390	0.162978646	0.143421336	0.135521095
Recommended Tag	0.216583201	0.214199207	0.205889124	0.182304096	0.172693842
	social	webdev	sharing	share	examples
Never Used	0.004333702	0.004233777	0.004075052	0.003700836	0.003587801
Used Only On Site	0.006648874	0.006495915	0.006252914	0.005679844	0.005506698
Used Only By User	0.129990417	0.127363764	0.123159817	0.113091733	0.110006461
Recommended Tag	0.165937323	0.162720162	0.157559906	0.145145178	0.141324792
	dev	development	tagging	javascript	source
Never Used	0.003410948	0.003177581	0.003152210	0.003128220	0.003101417
Used Only On Site	0.005235753	0.004878150	0.004839266	0.004802500	0.004761420
Used Only By User	0.105137421	0.098633061	0.097920389	0.097245547	0.096490389
Recommended Tag	0.135280262	0.127176127	0.126286113	0.125442966	0.124499040
	community	desarrollo	tags	snipplr	tools
Never Used	0.002980182	0.002931602	0.002829760	0.002761057	0.002697218
Used Only On Site	0.004575593	0.004501123	0.004344994	0.004239660	0.004141776

Used Only By User	0.093059333	0.091677382	0.088767038	0.086793502	0.084952236
Recommended Tag	0.120204540	0.118472134	0.114818701	0.112337369	0.110019492
	resources	develop	web2	web2.0	search
Never Used	0.002512252	0.002365577	0.002203368	0.002200424	0.002154594
Used Only On Site	0.003858131	0.003633164	0.003384331	0.003379815	0.003309501
Used Only By User	0.079576596	0.075270123	0.070461887	0.070374183	0.069006623
Recommended Tag	0.103236596	0.097785738	0.091681809	0.091570294	0.089830639
	utilidades	ruby	java	tag	tutoriales
Never Used	0.002061229	0.001990474	0.001927535	0.001814236	0.001728293
Used Only On Site	0.003166250	0.003057679	0.002961095	0.002787214	0.002655303
Used Only By User	0.066208578	0.064077256	0.062173416	0.058727282	0.056096811
Recommended Tag	0.086266472	0.083547235	0.081115046	0.076704891	0.073331914
	ajax	tool	c	free	howto
Never Used	0.001682916	0.001663148	0.001612757	0.001591346	0.001474966
Used Only On Site	0.002585650	0.002555304	0.002477949	0.002445081	0.002266406
Used Only By User	0.054702193	0.054093367	0.052538021	0.051875671	0.048259582
Recommended Tag	0.071541291	0.070759077	0.068759376	0.067907179	0.063248132
	webmaster	perl	actionscript	webdesign	tips
Never Used	0.001398077	0.001390063	0.001369204	0.001364313	0.001333151
Used Only On Site	0.002148348	0.002136043	0.002104014	0.002096503	0.002048652
Used Only By User	0.045855835	0.045604631	0.044950167	0.044796572	0.043816929
Recommended Tag	0.060145003	0.059820429	0.058974563	0.058775996	0.057509047
	developer	software	folksonomy	archive	xhtml
Never Used	0.001281783	0.001215025	0.001181374	0.001158521	0.001118876
Used Only On Site	0.001969769	0.001867246	0.001815564	0.001780466	0.001719574
Used Only By User	0.042197754	0.040085537	0.039017389	0.038290702	0.037027486
Recommended Tag	0.055413227	0.052675887	0.051290173	0.050346885	0.048706076
	service	work	list	tutorial	bookmarks
Never Used	0.001083747	0.001083692	0.001018730	0.001016237	0.0009797822
Used Only On Site	0.001665617	0.001665532	0.001565746	0.001561916	0.0015059162
Used Only By User	0.035905474	0.035903698	0.033822131	0.033742067	0.0325699341
Recommended Tag	0.047247542	0.047245232	0.044536499	0.044432239	0.0429052396
	python	online	del.icio.us	design	
Never Used	0.0009362798	0.0009239968	0.000763882	0.0007047204	
Used Only On Site	0.0014390871	0.0014202171	0.001174216	0.0010833089	
Used Only By User	0.0311675854	0.0307709142	0.025571154	0.0236361485	
Recommended Tag	0.0410767830	0.0405592766	0.033763070	0.0312280361	
	internet	library	dhtml	links	
Never Used	0.0006281085	0.0006063618	0.0005683478	0.0005611092	
Used Only On Site	0.0009655795	0.0009321596	0.0008737386	0.0008626138	
Used Only By User	0.0211192900	0.0204025717	0.0191472733	0.0189078852	
Recommended Tag	0.0279258894	0.0269845455	0.0253347548	0.0250199812	
	productivity	opensource	templates	mysql	
Never Used	0.0005218629	0.0004904224	0.0004883000	0.0004760106	
Used Only On Site	0.0008022960	0.0007539731	0.0007507109	0.0007318221	
Used Only By User	0.0176079976	0.0165642347	0.0164936964	0.0160850711	
Recommended Tag	0.0233098801	0.0219356694	0.0218427650	0.0213044890	
	Other	Other			
Never Used	0.0003363419	0.0003363419			
Used Only On Site	0.0005171333	0.0005171333			
Used Only By User	0.0114177874	0.0114177874			
Recommended Tag	0.0151460396	0.0151460396			

Appendix D

Proofs of Major Theorems in Chapter 5

D.1 Proof of Lemma 5.1

Since person i 's decision will depend on what everyone else contributes, we need to first start by making an assumption about what person i expects others to do. She assumes that others will contribute a total of x_{-i} information to the pool. Eventually, we will find a *fulfilled expectations equilibrium*, which is an equilibrium where everyone decides to choose exactly what is expected. But for now, we are just concerned with a single person's choice given their expectations.

First, we calculate what person i would choose if there were no threshold constraint. In that case, she would choose x_i that maximizes her utility:

$$x_i^0(x_{-i}) = \operatorname{argmax}_{x_i} v_i(\alpha x_i + x_{-i}) - c_i(x_i)$$

We call the choice of contribution that maximizes this utility $x_i^0(x_{-i})$. The threshold t is a constraint; if $x_i^0(x_{-i}) > t$ then person i will happily choose to contribute this amount.

However, if her optimal choice of contribution falls below the threshold t , then she must either raise her contribution up to at least t , or she must be willing to forego the benefits of accessing the information pool. If she chooses to contribute below t and not receive any benefits from the information pool, then her utility is simply $-c_i(x_i)$. Since $c_i(\cdot)$ is increasing and positive, the best she can do is to contribute nothing ($x_i = 0$). If she chooses to contribute $x_i \geq t$, her utility is $v_i(\alpha x_i + x_{-i}) - c_i(x_i)$. This expression is decreasing in x_i for $x_i \geq t > x_i^0$ by the assumptions on $v_i(\cdot)$ and $c_i(\cdot)$. Therefore, the best choice would be to choose $x_i = t$ since that is the smallest contribution that satisfies the threshold constraint.

Person i 's final choice then depends on how valuable the information pool is relative to the cost of contributing at the threshold level. If we assume that the cost of contributing

nothing is zero ($c_i(x_i) = 0$), then she should choose to contribute exactly the minimum threshold t if and only if she would prefer to contribute below the threshold ($x_i^0(x_{-i}) < t$) and the utility from contributing the threshold ($v_i(\alpha t + x_{-i}) - c_i(t)$) is greater than the utility from contributing nothing ($c_i(0) = 0$).

D.2 Proof of Proposition 5.1

Before we begin this proof, we must first repeat a famous result from Milgrom and Shannon (1994):

Milgrom and Shannon (1994) define a function $f(x, i)$ to have *increasing differences* (ID) if for all $x' > x'', i' > i'', f(x', i') - f(x'', i') > f(x', i'') - f(x'', i'')$. Another way of saying this is that for $x > y$, $f(x, i) - f(y, i)$ is increasing in i . For continuous and differentiable functions, this is similar and related to the property that the cross derivative is positive. Milgrom and Shannon (1994) were then able to prove the following theorem:

Theorem D.1 (Milgrom and Shannon (1994)) *If $f(x, i)$ is supermodular in x , and $f(x, i)$ has increasing differences in (x, i) then $\operatorname{argmax}_x f(x, i)$ is non-decreasing in i .*

This theorem allows us to describe properties of a user choice (choosing x to maximize $f(x, \cdot)$) as a function of an external parameter i . We also use one more simple result from that same paper that relates more common properties to the notion of increasing differences:

Lemma D.1 (Milgrom and Shannon (1994)) *if $f(x, i)$ is continuous and differentiable in x , then $f(x, i)$ has increasing differences if and only if $\frac{\partial}{\partial x} f(x, i)$ is weakly increasing in i . If $f(x, i)$ is twice continuously differentiable, then $f(x, i)$ has increasing differences if and only if*

$$\frac{\partial^2 f(x, i)}{\partial x \partial i} \geq 0$$

We will prove this proposition by calculating a fulfilled expectations equilibrium. In this type of equilibrium, everyone forms an expectation about everyone else's behavior. We define \bar{x}_i to be the expected contribution from user i . Users will then make contribution decisions based on the expected contributions from everyone else. We then calculate a set of contributions x_i^* where each person will choose to contribute exactly what is expected of them ($x_i^* = \bar{x}_i$), thus fulfilling expectations and providing a stable equilibrium.

Before we can prove this theorem, we must prove a couple of helpful lemmas:

Lemma D.2 *The function $f_j(x, i) = v_j(\alpha x + \bar{x}_{-i})$ has increasing differences in (x, i) .*

Proof: First look at the assumption that users expect $\bar{x}_{-i} \leq \bar{x}_{-j}$ for $i > j$. This is basically saying that \bar{x}_{-i} is decreasing in i . The first derivative $f'_j(x, i) = \alpha v'_j(\alpha x + \bar{x}_{-i})$ is weakly increasing in i since \bar{x}_{-i} is decreasing in i , and $v'_j(y)$ is decreasing in y (by the concavity of $v(\cdot)$). Another way of seeing this is by looking at the continuous analog: $\frac{\partial \bar{x}_{-i}}{\partial i} \leq 0$, and

$$\frac{\partial^2}{\partial x \partial i} f(x, i) = \alpha v''_j(\alpha x + \bar{x}_{-i}) \frac{\partial \bar{x}_{-i}}{\partial i} \geq 0$$

since $v''_j(\cdot) \leq 0$ by the concavity assumption. ■

Lemma D.2 means that $v_j(\alpha x^H + \bar{x}_{-iH}) - v_j(\alpha x^L + \bar{x}_{-iH}) \geq v_j(\alpha x^H + \bar{x}_{-iL}) - v_j(\alpha x^L + \bar{x}_{-iL})$. This allows us to separate the individual value and cost functions from the changes in expected contributions as i changes.

Given total expected contributions \bar{x}_{-i} from everyone else, each user i will choose her contribution to maximize her personal utility function:

$$g(x, i) = U_i(x, \bar{x}_{-i}) = v_i(\alpha x + \bar{x}_{-i}) - c_i(x)$$

Now we can state the main lemma that we need to prove this proposition:

Lemma D.3 *If users expect that $\bar{x}_{-i} \leq \bar{x}_{-j}$ for all $i > j$, then $g(x, i)$ has increasing differences.*

Proof: To show this, we must prove that if $x^H > x^L$, $i > j$ then $g(x^H, i) - g(x^L, i) \geq g(x^H, j) - g(x^L, j)$:

$$\begin{aligned} & g(x^H, i) - g(x^L, i) \\ &= (v_i(\alpha x^H + \bar{x}_{-i}) - c_i(x^H)) - (v_i(\alpha x^L + \bar{x}_{-i}) - c_i(x^L)) \\ &\geq (v_i(\alpha x^H + \bar{x}_{-j}) - c_i(x^H)) - (v_i(\alpha x^L + \bar{x}_{-j}) - c_i(x^L)) \\ &\geq (v_j(\alpha x^H + \bar{x}_{-j}) - c_j(x^H)) - (v_j(\alpha x^L + \bar{x}_{-j}) - c_j(x^L)) \\ &= g(x^H, j) - g(x^L, j) \end{aligned}$$

The first equality is by definition of $g(x, i)$. The next line is a direct result of Lemma D.2. The next line is a consequence of our assumption on the ordering of users; the first derivative of $v_i(\alpha x + y) - c_i(x)$ with respect to x is increasing in i and therefore has increasing differences in (x, i) . Finally, the last equality is by definition. ■

A straightforward corollary of Theorem D.1 and Lemma D.3 states that the optimal choice of contribution x_i^* is weakly increasing in i . This means that users with a higher marginal benefit of contribution will voluntarily choose to contribute more information.

Corollary D.1 *If users expect $\bar{x}_{-i} \geq \bar{x}_{-j}$ when $i > j$, then x_i^* is weakly increasing in i .*

Finally, to complete the proof we combine Lemma 5.1 and Corollary D.1. We assume that everyone has identical expectations that users will contribute:

$$\bar{x}_i^* = 0 \quad \text{if } i \leq i^0 \quad (\text{D.1})$$

$$\bar{x}_i^* = t \quad \text{if } i^0 < i < i^* \quad (\text{D.2})$$

$$\bar{x}_i^* = x_i^0(\bar{x}_{-i}) \quad \text{if } i > i^* \quad (\text{D.3})$$

First note that this schedule of contributions is weakly increasing in i : no user i contributes less than any user numbered less than i . If users expect each other to contribute according to this schedule of contributions, then the precondition for Lemma D.3 is fulfilled.

Let us begin with line D.3. Assume that for some i , $x_i^* = x_i^0$, meaning that user i chose to contribute his optimal amount, which is greater than the threshold t by Lemma 5.1. Then all users $j > i$ will also want to contribute their optimal amount x_j^0 , since by Corollary D.1, $x_j^0 > x_i^0$ and the user's optimal choice according to Lemma 5.1 is to contribute x_j . Define i^* to be the smallest i that contributes x_i .

Next we move to line D.1. If, given the expectations \bar{x}_i , no user will choose $x_i^* = 0$ by Lemma 5.1, then $i^0 = 0$. If at least one person chooses $x_i^* = 0$ then by Lemma 5.1, we know that $x_i^0 < t$ and $v_i(\alpha t + \bar{x}_{-i}) < c_i(t)$. This last statement is equivalent to saying $g(t, i) < 0$. Then all users $j < i$ will also want to contribute 0: We know that $x_j^* \leq x_i^*$ by Corollary D.1 and the only possible optimal choice from Lemma 5.1 is $x_j^* = 0$. Let i^0 be the largest i that contributes exactly 0.

Line D.2 is all that is left, and is fairly straightforward now. Choose an i such that $i^0 < i < i^*$. We know that $x_i^* \leq t$ since $i < i^*$. We know that $v_i(\alpha t + \bar{x}_{-i}) - c_i(t) > 0$ since $i > i^0$. Therefore, by Lemma 5.1, person i will choose to contribute t .

D.3 Proof of Lemma 5.2

Everyone who contributes greater than t is choosing their contribution to maximize their utility function $U_i(x_i, x_{-i}) = v_i(\alpha x_i + x_{-i}) - c_i(x_i)$. The first order condition for this maximization states that

$$\alpha v_i'(\alpha x_i + x_{-i}) - c_i'(x_i) = 0$$

Using the implicit function theorem, we find that

$$\frac{\partial x_i}{\partial x_{-i}} = -\frac{\alpha v_i''(\alpha x_i + x_{-i})}{\alpha^2 v_i''(\alpha x_i + x_{-i}) - c_i''(x_i)}$$

This derivative is always negative (since $v_i''(\cdot) < 0$ and $c_i''(\cdot) > 0$ by assumption), and furthermore has the same sign for all $i \geq i^*$. Therefore, as the total contribution from other people (x_{-i}) increases, all users who voluntarily contribute more than t will decrease their contribution slightly; however this decrease will not be enough to decrease the total size of the pool.

D.4 Proof of Proposition 5.2

Let \bar{X}_t be the expected total contributions of everyone in this equilibrium. As long as everyone contributes, we know that $\bar{X}_t \geq Nt$, where N is the total number of users. User 1, the person with the lowest marginal net benefit, will be willing to contribute t as long as his or her net benefit is positive. This net benefit is:

$$U_1(t) = v_1(\alpha t + \bar{x}_{-i}) - c_1(t) \geq v_1((N-1)t) - c_1(t)$$

As N increases, the total value to user 1 also increases since $v_1(\cdot)$ is increasing. As long as the threshold is low enough that

$$c_1(t) < \lim_{X \rightarrow \infty} v_1(X)$$

then there will exist an \underline{N} such that any population size greater than \underline{N} will lead to enough value that user 1 is willing to contribute t . By Proposition 5.1, if user 1 is willing to contribute t then so are all of the other users, and this is a Nash equilibrium.

In an informative pool without a threshold, the dominant strategy equilibrium is for everyone to contribute nothing, and consequently the pool will be of size 0. In this equilibrium everyone has zero utility since there is nothing in the pool and no one contributes. In the threshold equilibrium described above, all users have voluntarily chosen to contribute t , and to do so they must have a net utility that is greater than 0. Therefore each user has greater utility than in the no-threshold equilibrium and using a threshold is a Pareto improvement in welfare.

In a collaborative pool, users will voluntarily contribute some information even without a threshold, leading to a non-zero pool size X_0 . User i was receiving non-zero utility $U_i(x_i^0) = v_i(X_0) - c_i(x_i^0)$. Once a threshold is introduced, user i will have to increase their contribution if they were below the threshold. Their new utility $U_i(t) = v_i(X_t) - c_i(t)$ is positive because they are willing to contribute, but this utility may be smaller than the utility they received without a threshold. However, not everyone loses utility upon the

introduction of the threshold; users who voluntarily contribute above the threshold see their utility increase. System welfare, the sum of everyone's utility, increases when $W_t - W_0 > 0$.

$$W_t - W_0 = \sum_{i=1}^N (v_i(X_t) - c_i(\max\{t, x_i^*\})) \quad (\text{D.4a})$$

$$- \sum_{i=1}^N (v_i(X_0) - c_i(x_i^0)) \quad (\text{D.4b})$$

Since no one has dropped out, $X_t \geq X_0$. By Lemma 5.2, users who contribute above the threshold will voluntarily decrease their contribution. Therefore, any user who voluntarily contributes above the threshold increases total welfare.

Only users who contribute exactly t can cause a decrease in welfare (due to the increased costs of contributing t). However, if X_0 is sufficiently small, then this decrease can be offset by the increased value from having a larger pool. Specifically, this happens when

$$\sum_{i=1}^N v_i(X_0) \leq \sum_{i=1}^N v_i(X_t) - \sum_{i=i^*}^{i^*} c_i(t)$$

We know the each element of the summation on the right hand side is positive since no one has dropped out. The left hand side is continuous in X_0 ; therefore, there exists a maximum \bar{X}_0 that makes this an equality. We can ignore the costs of the voluntary contributions because they just make this condition weaker ($v_i(X_0) - c_i(x_i^0) \leq v_i(X_0)$). As long as the voluntary equilibrium is sufficiently bad ($X_0 \leq \bar{X}_0$), then introducing a threshold t leads to an increase in total welfare.

D.5 Proof of Proposition 5.3

We begin by making two assumptions. First we assume that the system designer increases the threshold to t from $t' < t$. Second, we assume that this increase causes the total size of the information pool to decrease, from $X_{t'}$ to X_t . We observe that if t increases but X decreases, then it must be the case that some users stopped contributing and t^0 increased.

Now, we can compute the change in aggregate welfare:

$$W_t - W_{t'} = \sum_{i^0}^N (v_i(X_t) - v_i(X_{t'})) \quad (\text{D.5a})$$

$$- \sum_{i^0}^{i_0^*} (v_i(X_{t'} - c_i(t'))) \quad (\text{D.5b})$$

$$+ \sum_{i^0}^{\min i^*, i_0^*} (c_i(t) - c_i(t')) \quad (\text{D.5c})$$

$$+ \sum_{\max i^*, i_0^*}^N (c_i(x_i^*) - c_i(x_i^*)) \quad (\text{D.5d})$$

$$+ \sum_{i_0^*}^{i^*} (c_i(t) - c_i(x_i^*)) \quad (\text{D.5e})$$

$$+ \sum_{i^*}^{i_0^*} (c_i(x_i^*) - c_i(t')) \quad (\text{D.5f})$$

(Note, only one of (D.5e) and (D.5f) will be non-zero, depending on whether users switch to contributing the threshold from contributing above the threshold, or vice versa.)

(D.5a) is negative, and indicates a welfare loss; the value of the pool to each user decreases because the total size of the information pool decreases. (D.5b) is also a welfare loss from the users who have left the system because of the threshold increase. (D.5c) is a welfare loss as everyone who contributes the threshold has increased their costs due to the threshold increase. (D.5d) is also a welfare loss; users who contribute above the threshold will choose to increase their contributions to compensate for the decrease in pool size, and this increase will lead to a corresponding increase in costs. Finally, both (D.5e) and (D.5f) are both negative because whichever direction the user switches, they do so because it is a higher contribution, and therefore a higher cost. Since all components of the expression are negative, the total change in welfare is negative.

The second half of the proposition can be shown by reversing the direction of the change of both t and X . It is straightforward to show that this reverses the sign on everything in (D.5a)-(D.5f) leading to a welfare increase.

Bibliography

- Adams, Anne and Martina Angela Sasse. 1999. Users are not the enemy, *Communications of the ACM*, 42(12), 40–46, URL <http://portal.acm.org/citation.cfm?id=322796.322806>.
- Adar, Eytan and Bernardo Huberman. 2000. Free riding on gnutella, *First Monday*, 5(10).
- Agresti, Alan. 2007. *An Introduction to Categorical Data Analysis*, Wiley, second edn.
- Al-Shaer, E. S. and H. H. Hamed. 2004. Discovery of policy anomalies in distributed firewalls, in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2605–2616.
- Ames, Morgan and Mor Naaman. 2007. Why we tag: Motivations for annotation in mobile and online media, in *CHI '07*, 971–980.
- Anderson, Lisa and Charles Holt. 1997. Information cascades in the laboratory, *American Economic Review*, 87(5), 847–862, URL [http://links.jstor.org/sici?sici=0002-8282\(199712\)87%3A5%3C847%3AICITL%3E2.0.CO%3B2-9](http://links.jstor.org/sici?sici=0002-8282(199712)87%3A5%3C847%3AICITL%3E2.0.CO%3B2-9).
- Anderson, Ross. 1993. Why cryptosystems fail, in *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, ACM Press, 215–227, URL <http://portal.acm.org/citation.cfm?id=168615>.
- Andreoni, J. 1998. Toward a theory of charitable fund-raising, *Journal of Political Economy*, 106, 1186–1213.
- Andreoni, J and A A Payne. 2003. Do government grants to private charities crowd out giving or fund-raising?, *American Economic Review*, 93, 792–812.
- Andreoni, James. 2006. Philanthropy, in S-C. Kolm and J. Mercier Ythier, (Eds.) *Handbook of Giving, Reciprocity and Altruism*, Amsterdam: North Holland, 1201–1269, URL <http://econ.ucsd.edu/~jandreon/WorkingPapers/Philanthropy.pdf>.
- Asgharpour, Farzaneh, Debin Liu, and L. Jean Camp. 2007. Mental models of computer security risks, in *Workshop on the Economics of Information Security (WEIS)*, URL <http://weis2007.econinfosec.org/papers/80.pdf>.
- Axtell, Robert, Robert Axelrod, Joshua M. Epstein, and Michael D. Cohen. 1996. Aligning simulation models: A case study and results, *Computational and Mathematical Organization Theory*, 1(2), 123–141.
- Bacher, Paul, Thorsten Holz, Markus Kötter, and Georg Wicherski. 2005. Know your enemy: Tracking botnets, URL <http://www.honeynet.org/papers/bots/>. From the Honeynet Project.
- Bag, Parimal Kanti and Eyal Winter. 1999. Simple subscription mechanisms for excludable public goods, *Journal of Economic Theory*, 87, 72–94.

- Bagnoli, Mark and Barton Lipman. 1989. Provision of public goods: Fully implementing the core through private contributions, *The Review of Economic Studies*, 56(4), 583–601, URL [http://links.jstor.org/sici?sici=0034-6527\(198910\)56%253A4%253C583%253APOPFGFI%253E2.0.CO%253B2-P](http://links.jstor.org/sici?sici=0034-6527(198910)56%253A4%253C583%253APOPFGFI%253E2.0.CO%253B2-P).
- Barford, Paul and Vinod Yegneswaran. 2006. An inside look at botnets, in *Special Workshop on Malware Detection*, Springer-Verlag, Advances in Information Security, URL <http://www.cs.wisc.edu/~vinod/botnets.pdf>.
- Barreau, D. and B. A. Nardi. 1995. Finding and reminding: File organization from the desktop, *SIGCHI Bulletin*, 27(3), 39–45. BarreauNardi1995.
- Bikhchandani, Sushil, David Hirshleifer, and Ivo Welch. 1992. A theory of fads, fashion, and cultural change as information cascades, *Journal of Political Economy*, 100(5), 992–1026, URL <http://www.jstor.org/view/00223808/di980598/98p00557/0>.
- Bikhchandani, Sushil, David Hirshleifer, and Ivo Welch. 1998. Learning from the behavior of others: Conformity, fads, and informational cascades, *Journal of Economic Perspectives*, 12(3), 151–170, URL <http://www.jstor.org/view/08953309/di014715/01p0058j/0>.
- Brin, Sergey and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine, in *7th International World Wide Web Conference*, Brisbane, Australia.
- Bruce, H., W. Jones, and S. Dumais. 2004. Keeping and re-finding information on the web: What do people do and what do they need?, in *ASIST 2004: Proceedings of the 67th ASIST annual meeting*, Chicago, IL.
- Bryant, Susan, Andrea Forte, and Amy Bruckman. 2005. Becoming wikipedian: Transformation of participation in a collaborative online encyclopedia, in *Group 05 Workshop: Sustaining Community: The role and design of incentive mechanisms in online systems*.
- Burke, Moira and Robert Kraut. 2008. Mopping up: Modeling wikipedia promotion decisions, in *Computer Support Cooperative Work (CSCW)*.
- Burke, Moira, Cameron Marlow, and Thomas Lento. 2009. Feed me: Motivating newcomer contribution in social network sites, in *ACM Conference on Human Factors in Computing (CHI)*.
- Camp, Jean L. 2006. Mental models of privacy and security, URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=922735. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=922735.
- Camp, L Jean and Catherine Wolfram. 2000. Pricing security, in *Proceedings of the Information Survivability Workshop*, URL <http://www.springerlink.com/index/m44317165u727779.pdf>.
- Chang, S-J and R.E. Rice. 1993. Browsing: A multidimensional framework, *Annual Review of Information Science and Technology*, 28, 231–271. ChangRiceARIST1993.

- Cheswick, William, Steven Bellovin, and Aviel Rubin. 2003. *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley Professional.
- Clarke, Edward H. 1971. Multipart pricing of public goods, *Public Choice*, 11(1), 17–33.
- Clauset, Aaron, Cosma Rohilla Shalizi, and Mark E. J. Newman. 2007. Power-law distributions in empirical data. <http://arxiv.org/abs/0706.1062v1>.
- Cohen, Bram. 2003. Incentives build robustness in bittorrent, in *Workshop on the Economics of Peer-to-Peer Systems*, URL <http://bittorrent.org/bittorrentecon.pdf>.
- Collins, Allan and Dedre Gentner. 1987. How people construct mental models, in Dorothy Holland and Naomi Quinn, (Eds.) *Cultural Models in Language and Thought*, Cambridge University Press, URL <http://books.google.com/books?id=rJ6z1o365YEC&pg=RA6-PA243&lpg=RA6-PA243&dq=cultural+models+in+language+and+thought&sig=dWVgqSU8w9bf2tzfkAHaQhezX1M>.
- Contu, Ruggero and Matthew Cheung. 2009. Market share: Security market, worldwide 2008, Gartner Report: <http://www.gartner.com/it/page.jsp?id=1031712>.
- Cosley, Dan, Dan Frankowski, Loren Terveen, and John Riedl. 2007. Suggestbot: Using intelligent task routing to help people find work in wikipedia, in *International Conference on Intelligent User Interfaces*, ACM Press, 32–41.
- Cranor, Lorrie and Simson Garfinkel. 2005. *Security and Usability: Designing Secure Systems That People Can Use*, O'Reilly Media, Inc., URL <http://www.amazon.com/Security-Usability-Designing-Secure-Systems/dp/0596008279>.
- Cranor, Lorrie Faith. 2008. A framework for reasoning about the human in the loop, in *Usability, Psychology, and Security Workshop*, USENIX.
- Cutrell, Edward, Daniel Robbins, Susan Dumais, and Raman Sarin. 2006. Fast, flexible filtering with phlat, in *CHI '06*, 261–270.
- D'Andrade, Roy. 2005. *The Development of Cognitive Anthropology*, Cambridge University Press, URL <http://www.amazon.com/Development-Cognitive-Anthropology-Roy-DAndrade/dp/0521459761>.
- Day, Jennifer Cheeseman, Alex Janus, and Jessica Davis. 2005. Computer and internet use in the united states, US Census Bureau, URL <http://www.census.gov/prod/2005pubs/p23-208.pdf>.
- Deb, Rajat and Laura Razzolini. 1999. Auction-like mechanisms for pricing excludable public goods, *Journal of Economic Theory*, 88(2), 340–368.

- Dourish, Paul, Rebecca Grinter, Jessica Delgado de la Flor, and Melissa Joseph. 2004. Security in the wild: User strategies for managing security as an everyday, practical problem, *Personal and Ubiquitous Computing*, 8(6), 391–401, URL <http://portal.acm.org/citation.cfm?id=1037315>.
- Dwork, Cynthia and Moni Naor. 1993. Pricing via processing or combatting junk mail, in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, London, UK: Springer-Verlag, 139–147, URL <http://research.microsoft.com/research/sv/PennyBlack/junk1.pdf>.
- Ellison, Nicole, Charles Steinfield, and Cliff Lampe. 2007. The benefits of facebook "friends:" social capital and college students' use of online social network sites, *Journal of Computer Mediated Communication*, 12(4).
- Ephrati, Eithan, Gilad Zlotkin, and Jeffrey S. Rosenschein. 1994. Meet your destiny: A non-manipulable meeting scheduler, in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 359–371.
- Farooq, Umer, Thomas G. Kannampallil, Yang Song, Craig H. Ganoe, John M. Carroll, and C. Lee Giles. 2007. Evaluating tagging behavior in social bookmarking systems: Metrics and design heuristics, in *GROUP '07*, 351–360.
- Feldman, Michal, Christos Papadimitriou, John Chuang, and Ian Stoica. 2006. Free-riding and whitewashing in peer-to-peer systems, *IEEE Journal on Selected Areas in Communications*, 24(5), 1010–1019.
- Friedman, Eric and Paul Resnick. 2001. The social cost of cheap pseudonyms, *Journal of Economics and Management Strategy*, 10(2), 173–199.
- Furnas, George, Tom Landauer, L Gomez, and Susan Dumais. 1987. The vocabulary problem in human-system communication, *Communications of the ACM*, 30(11), 964–971.
- Furnas, G.W., T.K. Landauer, L.M. Gomez, and S.T. Dumais. 1983. Statistical semantics: Analysis of the potential performance of key-word information systems, *The Bell System Technical Journal*, 62(6), 1753–1806.
- Gazzale, Robert S. and Jeffrey K. MacKie-Mason. 2008. User cost, usage and library purchasing of electronically-accessed journals, in Jeffrey K. MacKie-Mason and Wendy Lougee, (Eds.) *Byting the Bullet: Economics and the Usage of Digital Libraries*, Ann Arbor, MI: University of Michigan Scholarly Publishing Office, chap. 6.
- Glazer, A and A K Konrad. 1996. A signaling explanation for charity, *American Economic Review*, 86, 1019–1028.
- Goecks, Jeremy, W. Kieth Edwards, and Elizabeth D. Mynatt. 2009. Challenges in supporting end-user privacy and security management with social navigation, in *Proceedings of SOUPS: Symposium On Usable Privacy and Security*.

- Golder, Scott and Bernardo A. Huberman. 2006. Usage patterns of collaborative tagging systems, *Journal of Information Science*, 32(2), 198–208.
- Granovetter, Mark. 1973. The strength of weak ties, *American Journal of Sociology*, 78(6), 1360–1380.
- Grinter, Rebecca E., W. Kieth Edwards, Mark W. Newman, and Nicolas Ducheneaut. 2005. The work to make a home network work, in *Proceedings of the 9th European Conference on Computer Supported Cooperative Work (ECSCW '05)*, 469–488, URL <http://www.cc.gatech.edu/~beki/c27.pdf>.
- Gross, Joshua and Mary Beth Rosson. 2007. Looking for trouble: Understanding end user security management, in *Symposium on Computer Human Interaction for the Management of Information Technology (CHIMIT)*, URL <http://portal.acm.org/citation.cfm?id=1234786>.
- Groves, Theodore and John Ledyard. 1977. Optimal allocation of public goods: A solution to the "free rider" problem, *Econometrica*, 45(4), 783–809.
- Grudin, Jonathan. 2006. Enterprise knowledge management and emerging technologies, in *HICSS '06*.
- Halpin, Harry, Valentin Robu, and Hana Shepherd. 2007. The complex dynamics of collaborative tagging, in *WWW '07*.
- Harbaugh, W T. 1998. The prestige motive for making charitable transfers, *American Economic Review*, 88, 277–282.
- Jian, Lian and Jeffrey K. MacKie-Mason. 2006. Why share in peer-to-peer networks?, in *Workshop on the Economics of Networks Systems (NetEcon 06)*, URL <http://www.cs.duke.edu/nicl/netecon06/papers/ne06-reciprocity.pdf>.
- Jian, Lian and Jeffrey K. MacKie-Mason. 2008. Why leave Wikipedia?, in *iConference*, UCLA. Extended abstract and poster.
- Johnson-Laird, P. N. 1980. Mental models in cognitive science, *Cognitive Science: A Multidisciplinary Journal*, 4(1), 71–115, URL http://www.leaonline.com/doi/abs/10.1207/s15516709cog0401_4.
- Johnson-Laird, P.N., Vittorio Girotto, , and Paolo Legrenzi. 1998. Mental models: a gentle guide for outsiders, URL <http://www.si.umich.edu/ICOS/gentleintro.html>. Available at <http://www.si.umich.edu/ICOS/gentleintro.html>.
- Judge, George G., William E. Griffiths, R. Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee. 1985. *The Theory and Practice of Econometrics*, Ch. 22, Wiley Series in Probability and Statistics, New York: Wiley, 2nd edn.
- Karau, Steven and Kipling Williams. 1993. Social loafing: A meta-analytic review and theoretical integration, *Journal of Personality and Social Psychology*, 65(4), 681–706.

- Kempton, Willett. 1986. Two theories of home heat control, *Cognitive Science: A Multidisciplinary Journal*, 10(1), 75–90, URL http://www.leaonline.com/doi/abs/10.1207/s15516709cog1001_3.
- Krishnan, Ramayya, Michael D. Smith, Zhulei Tang, and Rahul Telang. 2004. The virtual commons: Why free-riding can be tolerated in file sharing networks, Tech. rep., Carnegie Mellon University, URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=450241.
- Kuzel, A. J. 1992. Sampling in qualitative inquiry, in B.F. Crabtree and W. L. Miller, (Eds.) *Doing Qualitative Research*, Sage Publications, Inc., chap. 2, 31–44.
- Lampe, Cliff and Paul Resnick. 2004. Slash(dot) and burn: Distributed moderation in a large online conversation space, in *Conference on Human Factors in Computing Systems (CHI)*.
- Lansdale, M.W. 1988. The psychology of personal information management, *Applied Ergonomics*, 19(1), 55–66.
- Laurie, Ben and Richard Clayton. 2004. Proof of work proves not to work, in *Workshop on the Economics of Information Security*, URL <http://www.cl.cam.ac.uk/~rncl/proofwork.pdf>.
- Ling, K., G. Beenen, P. Ludford, X. Wang, K. Chang, D. Cosley, D. Frankowski, L. Terveen, A. M. Rashid, P. Resnick, and R Kraut. 2005. Using social psychology to motivate contributions to online communities, *Journal of Computer-Mediated Communication*, 10(4), URL <http://jcmc.indiana.edu/vol10/issue4/ling.html>.
- Liu, Debin and L Jean Camp. 2006. Proof of work can work, Tech. rep., NET Institute, URL <http://ssrn.com/abstract=941190>. Working Paper No. 06-18.
- Locke, Edwin and Gary Latham. 2002. Building a practically useful theory of goal setting and task motivation, *American Psychologist*, 57(9), 705–17, URL <http://content.apa.org/journals/amp/57/9/705>.
- Loder, Thede, Marshall van Alstyne, and Rick Wash. 2006. An economic response to unsolicited communication, *Advances in Economic Analysis and Policy*, 6(1), URL <http://www.bepress.com/bejeap/advances/vol6/iss1/art2/>.
- Ma, Hao, Raman Chandrasekar, Chris Quirk, and Abhishek Gupta. 2009. Page hunt: Improving search engines using human computation games, in *SigIR*.
- MacKie-Mason, Jeff, Scott Shenker, and Hal Varian. 1996. Service architecture and content provision: The network provider as editor, *Telecommunications Policy*, 20(3), URL <http://www-personal.umich.edu/~jmm/papers/clutter.pdf>.
- Markoff, John. 2007. Attack of the zombie computers is a growing threat, experts say, *New York Times*, URL <http://www.nytimes.com/2007/01/07/technology/07net.html>.

- Markus, Lynne M. 2001. Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success, *J. of MIS*, 18(1), 57 – 93.
- Marlow, Cameron, Mor Naaman, danah boyd, and Marc Davis. 2006. Position paper, tagging, taxonomy, flickr, article, toread, in *WWW '06 Collaborative Tagging Workshop*.
- Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green. 1995. *Microeconomic Theory*, Oxford University Press.
- Medin, Douglas, Norbert Ross, Scott Atran, Douglas Cox, John Coley, Julia Proffitt, and Sergey Blok. 2006. Folkbiology of freshwater fish, *Cognition*, 99(3), 237–273.
- Menard, Scott. 2002. *Applied Logistic Regression Analysis*, Quantitative Applications in the Social Sciences, Sage University Press.
- Miles, Matthew B. and Michael Huberman. 1994. *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publications, Inc., 2nd edition edn. MilesHuberman1994.
- Milgrom, Paul and Chris Shannon. 1994. Monotone comparative statics, *Econometrica*, 62(1), 157–180.
- Millen, David, Meng Yang, Steven Whittaker, and Jonathan Feinberg. 2007. Social bookmarking and exploratory search, in *ECSCW '07*, 21–40, URL www.springerlink.com/index/p46015057028m781.pdf.
- Millen, David R., Jonathan Feinberg, and Bernard Kerr. 2006. Dogear: Social bookmarking in the enterprise, in *CHI '06*.
- Morgan, J and M Sefton. 2000. Financing public goods by means of lotteries: Experimental evidence, *Review of Economic Studies*, 67, 785–810.
- Morgan, John. 2000. Financing public goods by means of lotteries, *Review of Economic Studies*, 67(4), 761–784.
- Moulin, Herve. 1994. Serial cost-sharing of excludable public goods, *The Review of Economic Studies*, 61(2), 305–325.
- Myerson, R and M Satterthwaite. 1983. Efficient mechanisms for bilateral trading, *Journal of Economic Theory*, 29, 265–281, URL http://www.sciencedirect.com/science?_ob=MIImg&_imagekey=B6WJ3-4CYGD4J-13J-1&_cdi=6867&_user=99318&_orig=browse&_coverDate=04%2F30%2F1983&_sk=999709997&view=c&wchp=dGLbVzb-zSkWW&md5=2f6a1cecc403c7d6b6159a50cdf9685f&ie=/sdarticle.pdf.
- Nan, Ning, Erik W. Johnston, Judith S. Olson, and Nathan Bos. 2005. Beyond being in the lab: using multi-agent modeling to isolate competing hypotheses, in *CHI '05*, 1693–1696.
- Newman, M. E. J. 2005. Power laws, pareto distributions and zipf's law, *Contemporary Physics*, 46, 323–351.

- Nonnecke, Blair, Dorine Andrews, and Jenny Preece. 2006. Non-public and public online community participation: Needs, attitudes, and behavior, *Electronic Commerce Research*, 6(1), 7–20.
- Onwuegbuzie, Anthony J and Nancy L Leech. 2007. Validity and qualitative research: An oxymoron?, *Quality and Quantity*, 41, 233–249.
- Page, Scott. 2006. Path dependence, *Quarterly Journal of Political Science*, 1(87-115).
- Pirolli, Peter. 2005. Rational analyses of information foraging on the web, *Cognitive Science*, 29(3), 343–373.
- Preece, Jennifer and Ben Schneiderman. 2009. The reader-to-leader framework: Motivating technology-mediated social participation, *AIS Transactions on Human-Computer Interaction*, 1(1), 13–32.
- Putnam, Robert. 2000. *Bowling Alone*, New York: Simon and Schuster.
- Rashid, Al, Kimberly Ling, Regina Tassone, Paul Resnick, Robert Kraut, and John Reidl. 2006. Motivating participation by displaying the value of contribution, in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, URL <http://doi.acm.org/10.1145/1124772.1124915>.
- Ratliff, Evan. 2006. The zombie hunters, *The New Yorker*, URL http://www.newyorker.com/fact/content/articles/051010fa_fact.
- Rivadeneira, A.W., Daniel M. Gruen, Michael J. Muller, and David R. Millen. 2007. Getting our head in the clouds: Toward evaluation studies of tagclouds, in *CHI 2007*, San Jose, CA, 995–998. RivadeneiraCHI2007.
- Russell, Dan, Stuart Card, Peter Pirolli, and M Stefik. 1993. The cost structure of sensemaking, in *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing system*.
- Samuelson, Paul A. 1954. The pure theory of public expenditure, *Review of Economics and Statistics*, 36(4), 387–389.
- Scarfone, Karen and Peter Mell. 2007. Guide to intrusion detection and prevention systems (idps), Tech. Rep. SP 800-94, National Institute of Standards and Technology, URL <http://csrc.nsl.nist.gov/publications/nistpubs/800-94/SP800-94.pdf>.
- Sen, Shilad, F. Maxwell Harper, Adam LaPitz, and John Riedl. 2007. The quest for quality tags, in *GROUP '07*.
- Sen, Shilad, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. 2006. tagging, communities, vocabulary, evolution, in *CSCW '06*, 181–190.

- Sieberg, Daniel. 2006. Experts: Botnets no. 1 emerging internet threat, *CNN.com, Technology Section*, URL <http://www.cnn.com/2006/TECH/internet/01/31/furst/>.
- Stone, Brad. 2006. Spam doubles, finding new ways to deliver itself, *New York Times*, URL <http://select.nytimes.com/gst/abstract.html?res=F10812FD3D550C758CDDAB0994DE404482>.
- Storey, M. A., L.T. Cheng, I. Bull, and P. Rigby. 2006. Shared waypoints and social tagging to support collaboration in software development, in *Computer Support Cooperative Work (CSCW)*, Banff, Alberta, Canada.
- Tang, John C., Eric Wilcox, Julian A. Cerruti, Hernan Badenes, Stefan Nusser, and Jerald Schoudt. 2008. Tag-it, snag-it, or bag-it: combining tags, threads, and folders in e-mail, in *CHI '08*, 2179–2194.
- Teevan, Jaime, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search, in *CHI '04*, 415–422.
- Trend Micro. 2006. Taxonomy of botnet threats. Whitepaper.
- Varian, Hal. 1994. A solution to the problem of externalities when agents are well-informed, *American Economic Review*, 84(5), 1278–1293. A Solution to the Problem of Externalities When Agents Are Well-Informed.
- Vesterlund, L D. 2003. The informational value of sequential fund-raising, *Journal of Public Economics*, 87, 627–657.
- von Ahn, Luis, Manuel Blum, Nicholas Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security, in *Proceedings of EUROCRYPT 03*, Lecture Notes in Computer Science, URL <http://www.springerlink.com/index/P8T2Q8Q6BXEY8TVX.pdf>.
- von Ahn, Luis and Laura Dabbish. 2004. Labelling images with a computer game, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 319–326, URL <http://portal.acm.org/citation.cfm?id=985692.985733>.
- von Ahn, Luis, Mihir Kedia, and Manuel Blum. 2006a. Verbosity: A game for collecting common sense facts, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 75–78, URL <http://portal.acm.org/citation.cfm?id=1124772.1124784>.
- von Ahn, Luis, Ruoran Liu, and Manuel Blum. 2006b. Peekaboom: A game for locating objects in images, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, URL <http://www.cs.cmu.edu/~biglou/Peekaboom.pdf>.
- Wash, Rick and Jeff MacKie-Mason. 2008. A social mechanism for home computer security, in *Workshop on Information Systems and Economics (WISE)*.

- Wash, Rick and Jeff MacKie-Mason. 2009. Using a minimum threshold to motivate contributions. Working Paper.
- Wash, Rick and Jeffrey K. MacKie-Mason. 2006. Incentive-centered design for information security, in *USENIX Hot Topics in Security (HotSec 06)*, Vancouver, BC: USENIX.
- Wash, Rick and Jeffrey K. MacKie-Mason. 2007. Security when people matter: Structuring incentives for user behavior, in *Ninth International Conference on Electronic Commerce (ICEC-07)*, New York, NY, USA: ACM, 7–14, URL <http://www-personal.siu.umich.edu/~rwash/pubs/icec702w-wash.pdf>.
- Wash, Rick and Emilee Rader. 2007. Public bookmarks and private benefits: An analysis of incentives in social computing, in *American Society for Information Science & Technology*.
- Whittaker, Steve and Candace Sidner. 1996. Email overload: exploring personal information management of email, in *CHI '96: Human factors in computing systems*, Vancouver, British Columbia, 276–283. WhittakerSidner1996.
- Xu, Z., Y. Fu, J. Mao, and D. Su. 2006. Towards the semantic web: Collaborative tag suggestions, in *WWW 2006 Collaborative Web Tagging Workshop*, Edinburgh, Scotland.
- Young, H. Peyton. 1998. Cost allocation, demand revelation, and core implementation, *Mathematical Social Sciences*, 36(3), 213–228.
- Zhuge, Jianwei, Thorsten Holz, Xinhui Han, Jinpeng Guo, and Wei Zou. 2007. Characterizing the irc-based botnet phenomenon, Tech. Rep. TR-2007-010, The HoneyNet Project, URL <http://honeyblog.org/junkyard/reports/botnet-china-TR.pdf>.