Extracting Insights from Differences: Analyzing Node-aligned Social Graphs

by

Srayan Datta

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2019

Doctoral Committee:

Associate Professor Eytan Adar, Chair
Associate Professor Mike Cafarella
Assistant Professor Danai Koutra
Associate Professor Clifford Lampe

Srayan Datta

srayand@umich.edu

ORCID iD: 0000-0002-5800-830X

To my family and friends

ACKNOWLEDGEMENTS

There are several people who made this dissertation possible, first among this long list is my adviser, Eytan Adar. Pursuing a doctoral program after just finishing undergraduate studies can be a daunting task but Eytan made it easy with his patience, kindness, and guidance. I learned a lot from our collaborations and idle conversations and I am very grateful for that.

I would like to extend my thanks to the rest of my thesis committee, Mike Cafarella, Danai Koutra and Cliff Lampe for their suggestions and constructive feedback. I would also like to thank the following faculty members, Daniel Romero, Ceren Budak, Eric Gilbert and David Jurgens for their long insightful conversations and suggestions about some of my projects.

I would like to thank all of friends and colleagues who helped (as a co-author or through critique) or supported me through this process. This is an enormous list but I am especially thankful to Chanda Phelan, Eshwar Chandrasekharan, Sam Carton, Cristina Garbacea, Shiyan Yan, Hari Subramonyam, Bikash Kanungo, and Ram Srivatasa.

I would like to thank my parents for their unwavering support and faith in me. This dissertation would not be possible without their constant support throughout the long process of pursuing a Ph.D.

My dissertation is supported by University of Michigan computer science department and school of information. My projects were partially supported by grants from the University of Michigan and NSF. Additionally, I would like to thank Jason Baumgartner for compiling the Reddit dataset, the one I used throughout this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Social media and network research often focus on the agreement between different entities to infer connections, recommend actions and subscriptions and even improve algorithms via ensemble methods. However, studying differences instead of similarities can yield useful insights in all these cases. We can infer and understand inter-community interactions (including ideological and user-based community conflicts, hierarchical community relations) and improve community detection algorithms via insights gained from differences among entities such as communities, users and algorithms. When the entities are communities or user groups, we often study the difference via *node-aligned networks*, which are networks with the same set of nodes but different sets of edges. The edges define implicit connections which we can infer via similarities or differences between two nodes.

We perform a set of studies to identify and understand differences among user groups using Reddit, where the subreddit structure provides us with pre-defined user groups. Studying the difference between author overlap and textual similarity among different subreddits, we find *misaligned* edges and networks which expose subreddits at ideological 'war', community fragmentation, asymmetry of interactions involving subreddits based on marginalized social groups and more. Differences in perceived user behavior across different subreddits allow us to identify subreddit *conflicts* and features which can implicate communal misbehavior. We show that these features can be used to identify some subreddits banned by Reddit. Applying the idea of differences in community detection algorithms helps us identify problematic community assignments where we can ask for human help in categorizing a node in a specific community. It also gives us an idea of the overall performance of a particular community detection algorithm on a particular network input. In general, these improve ensemble community detection techniques. We demonstrate this via CommunityDiff (a community detection and visualization tool), which compares and contrasts different algorithms and incorporates user knowledge in community detection output. We believe the idea of gaining insights from differences can be applied to several other problems and help us understand and improve social media interactions and research.

CHAPTER I

Introduction

Social media plays a major role in our day-to-day life. Online social networks like Facebook and Twitter have billions of users who connect with friends and family, find new friends and post updates about their daily lives. Online discussion forums and news aggregators like Reddit provide hundreds of millions of users a place to share, view and discuss different opinions with like-minded people. People from all over the world jointly contribute to write and maintain over 5 million articles in Wikipedia. Social media sites are often used for different social or political discussions and movement. Social network analysis helps us understand this phenomenon and improve downstream applications. Social media websites are also plagued by user misbehavior in a variety of forms [5, 35, 57, 58, 83, 108] including but not limited to spamming, trolling, flame wars and griefing. In a large number of cases, the extent of this abuse is not well documented. Network modeling of online social media is a powerful tool in understanding social behavior and gaining new insights, improving applications for recommendation systems, online marketing, and providing better usability and user experience for millions of social media users.

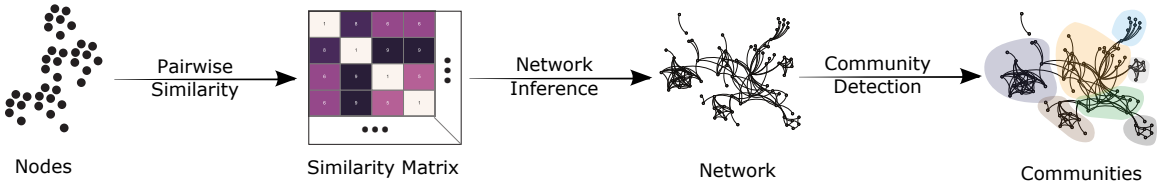

Figure 1.1: A general pipeline for analyzing network-based communities from a set of given nodes/entities. Apart from community detection many other network analysis algorithms like centrality measures can be applied on the inferred network.

A large aspect of social network analysis is defining a link or an edge between two

nodes or entities (an entity can be anything ranging from a user, a post/comment to a discussion topic or a group of users in different contexts) and finding groups or communities of the said entities. Semantically, these links may encode a variety of relationships such as *friendOf* (social media friend networks), *similarTo* (recommender system networks), or *isA* (Wordnet, a semantic network of English words). They can be directional, weighted and temporally varied. Moreover, these links can either be *explicit* (friendship links between two users) or *inferred* (many common users between two user groups) [27]. Where explicit, the links between individuals (*person-to-person* links) are measures of friendship (Facebook), interest (Twitter), or other relationships (family, fan, etc.). When an online system does not have features that explicitly support linking, we rely on inferred connections. For example, we may infer that two people are "linked" if they are part of the same discussion list. Inference is *sometimes* necessary in the case of person-to-person links and *often* in the case of *community-to-community* links where the social media websites rarely provide explicit linking. Inference can be useful on explicit networks also. A friendship network can be 'reduced' based on likes/messages etc.

Apart from social networks, link inference and community detection are used in many other kinds of networks. For example, functional groups in biological networks are identified using community detection [29, 91]. In machine learning and data mining, many non-network problems are converted into a network analysis problem (including community detection problems) by inferring links between relevant nodes/entities. For example word networks in natural language processing (NLP) [139] or user-item recommender models [94] allow us to use community detection to solve problems as diverse as keyword detection or movie recommendations. Most of these problems follow a broad general pipeline depicted in Figure 1.1. This pipeline starts with a set of nodes or entities and some similarity measure is applied to the entities to derive a similarity matrix. A similarity matrix A contains the similarity scores between all pairs of nodes (the cell $A[i, j]$ stores the similarity score between nodes i and j). The similarity matrix is processed into a network and then community detection and further network analysis algorithms are applied to the network. It is worth noting that there are other methods of inferring a connection between two individuals. In social network analysis, sometimes networks are constructed by communication behavior (e.g., retweeting, chat, liking, lending money, etc.). We can modify the pipeline to include these kinds of networks as well.

Community-to-community links are less common compared to links between a pair of individuals and most of the time, these are inferred links. This type of in-

ferred link and community detection are used for deciphering inter-community mobile communications [90]. Inter-community links can be useful for recommender systems when suggesting similar communities to a user in a forum or another social media website. These link might also depict negative relationships. For example, Kumar et al. [109] studied controversial cross-postings in Reddit to identify specific community conflicts.

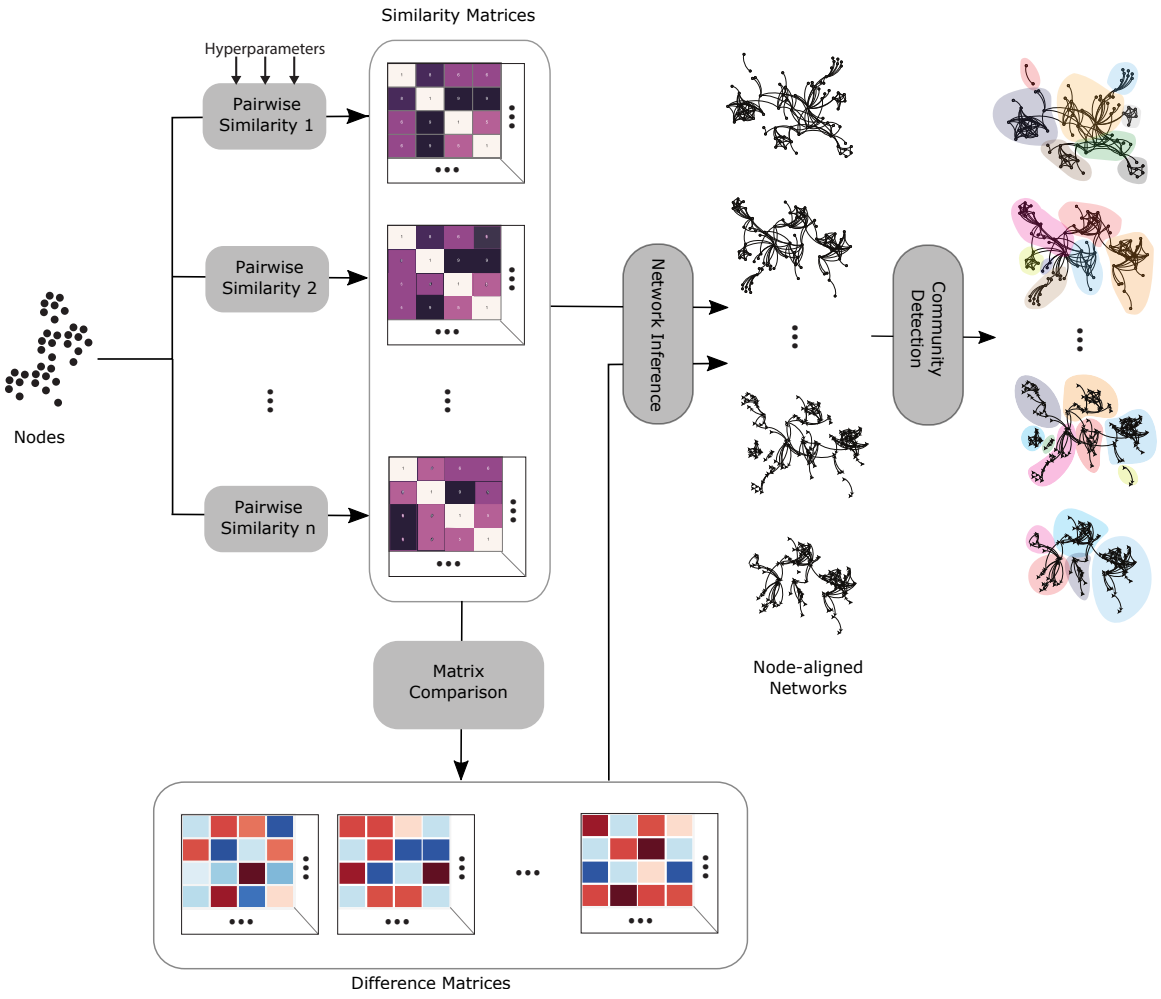

Figure 1.2: An expanded pipeline for analyzing network-based communities from a set of given nodes/entities.

However, inferred links can vary wildly based on the metric used for inference (for example, author overlap vs topical similarity between two user groups) and this *difference* often provides useful insights. Expanding the network analysis pipeline, we can run different pairwise similarity metrics in parallel to infer different kinds of pairwise links and compare them. We can create a similarity matrix for each different

similarity measure and compare the difference between two similarity measures by computing a *difference matrix* using corresponding two similarity matrices. We can create networks from both similarity and difference matrices. These networks share the same set of nodes (the set of nodes we start with) but a different set of links. We call these *node-aligned networks*. We can apply community detection on these networks to get an idea about how nodes are grouped together via a certain kind of similarity (for example, two users who talk about similar topics in social media should be grouped together if we are evaluating users via topical similarity of their posts) and how grouping via similarity measures differ from each other. Figure 1.2 shows a pictorial representation of this modified pipeline. Note that, this pipeline can be further expanded using data from different time periods (e.g. we can create a pipeline by using text/author similarity from few consecutive years to identify the temporal evolution of Reddit or change in behavioral patterns of its users).

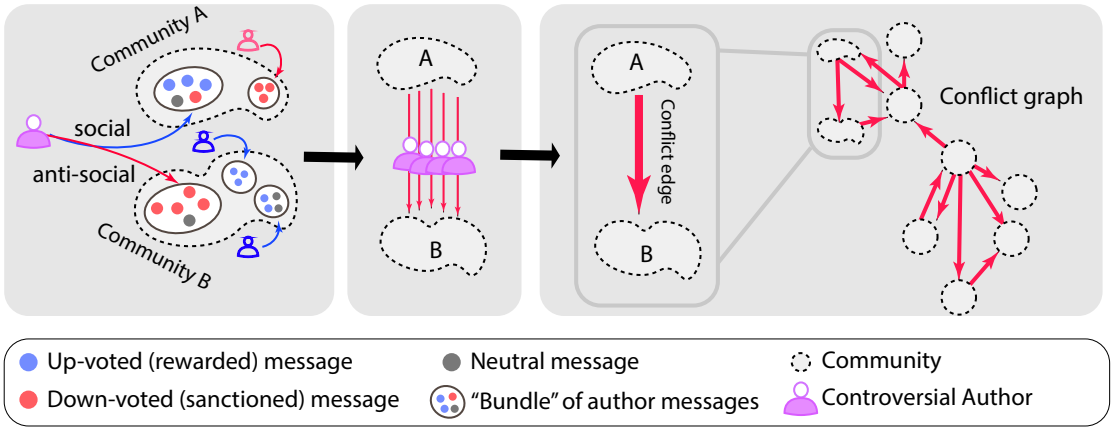

Figure 1.3: A general methodology for identifying conflicts and creating a conflict graph from a set of given user communities.

Inferred links between two nodes are not limited to different similarity metrics either. For two user groups, we can deduce antagonistic relationships (conflicts) if same set of users behave or perceived to behave (based on the groups' reward/sanction mechanism including up/downvotes) differently depending on the group as shown in Figure 1.3. We call these particular users *controversial authors* and the edge as a *conflict edge*. Conglomerating these edges, we can create a conflict graph which documents conflicts given a set of user groups. Based on controversial authors, we can also infer which user groups are 'targeted' together and form a co-conflict graph with the same set of nodes. These node-aligned graphs documents conflict between

different user groups and can be used to identify and understand community conflicts and communal misbehavior. This technique is less general as it makes use of user groups and not any set of entities, but still has wide applicability for any type of social network or social media site with a overlapping community structure like subreddits, Facebook pages and groups, online news communities and Twitter hashtag communities (people who tweeted a particular hashtag are part of that hashtag community).

1.1 Dissertation Overview

In this dissertation, we focus on gaining insights from differences in four different projects. Two other common motifs in these projects are creation of node-aligned networks, and inferring community-to-community relationships. In the first project, we apply the expanded network analysis pipeline on subreddit comment text and authors to identify 'misaligned' links and communities in Reddit. We use author and textual overlap between pairs of subreddits to create a pair of *subreddit similarity networks*: author and term similarity networks. We use the *difference* between author and text similarity to infer *author-coherent* links (high author similarity but low term similarity) and *topic-coherent* (high term but low author similarity) links and create a pair of *misaligned networks*. Note that the subreddit similarity networks and the misaligned networks are node-aligned networks as the nodes are subreddits in all of them. We provide the algorithm to identify these misaligned links and communities and use them to decipher inter-group dynamics (hierarchical links, community fragmentation, communities with opposing viewpoints, satellite subreddits etc.).

In the second project, we identify *differences in perceived commenting behavior* of the same user within multiple subreddits and use it to identify subreddit conflicts. In this project, we build two node-aligned networks — the *subreddit conflict network* which is a directed, weighted network depicting antagonistic subreddit relations and the co-conflict network which shows which subreddits are usually targeted together. We analyze these graphs to identify most instigating and targeted subreddits, find the relationship between subreddit size and conflict intensity, reciprocity of the conflicts and find implication behind certain subreddit bans.

To test the efficacy of these interaction features, we identified more than 1000 banned subreddits with at least 100 comments and used the interaction features along with text-based features to cluster and predict different kinds of banned subreddits. We improve the banned subreddit prediction result against an unbanned sample of the

same size by adding interaction features compared to using only textual features (with a significance level of 0.018). We also implement a banning-by-example paradigm where banned subreddits of a particular category can be identified using other banned subreddits in the same category as examples. We achieve 0.913 mean precision at k (k=10) measure for banning-by-example and this idea can be useful for community moderators in any social media.

Finally, we apply the idea of gaining insights from differences on the problem of choosing suitable community detection algorithms via CommunityDiff, an interactive visualization system that combines visualization and active learning to support the end-user's analytical process. We create a mechanism for visualizing *ensemble spaces* (an abstract space where each point refers to a community detection algorithm), by leveraging *differences* in outputs of most commonly used community detection algorithms. CommunityDiff also features weighted combinations of algorithm outputs, that can identify patterns, commonalities, and differences among multiple community detection algorithms. Among other features, CommunityDiff introduces an active learning mechanism that visually indicates uncertainty about community labels (based on disagreement among different community detection algorithms) to focus end-user attention and supporting end-user control which ranges from explicitly indicating the number of expected communities to merging and splitting communities. Based on this end-user input, CommunityDiff dynamically recalculates communities. We demonstrate the viability of our system through a study of speed of end-user convergence on satisfactory community labels. As part of building CommunityDiff, we describe a design process that can be adapted to other Interactive Machine Learning (IML) applications.

1.2 Motivation

In their recent research Peel et al. [151] showed that there is not always a one-to-one correspondence between the communities determined by the network structure and node metadata. In fact, different node metadata can generate different groupings. Metadata-based clustering can be viewed as a form of link inference where we draw a connection between individuals based on metadata similarity. The fact that there are differences in the generated networks is an argument for studying the difference and find out where this difference is surprising and how it originates. Based of this idea, instead of the classical node-link structure of a network, we can view the relationship between two nodes or entities as a set of edge attributes or edge

metadata, where each edge attribute is based on a different metric. These attributes can be explicit (friendship between two social media users) or inferred (do their posts share enough textual similarity?). This gives rise to a set of node-aligned networks based on different attributes. The pre-existing pipeline for converting data science problems (Figure 1.1) to network analysis problems is easily extended to a highly parallel pipeline (Figure 1.2) which incorporates the idea of node-aligned networks. Note that, the node metadata and hence the edge metadata changes over time. We can create node-aligned networks based on the same similarity metric but at different timestamps. A natural extension of our view of networks is to replace node and edge metadata with a set of time-series data where each time-series corresponds to a different attribute.

Peel et al. also showed that any specific community detection algorithm may not perform well on all networks, which implies that the ability to choose correct algorithm for given task and the human in the loop learning for community detection is important. This is a major motivation behind designing CommunityDiff, an interactive visualization tool to compare and select clustering algorithms which takes user inputs into account.

1.3 Contributions

The contributions of this dissertation are threefold. First, we provide a methodology for understanding *differences in inference* in online user groups. These types on analysis can be used to identify and understand relationships among online communities including community hierarchies, communities with ideologically opposing viewpoints and community fragmentation. We can also identify individual community types (e.g., is the community a mainstream community or is it marginalized) using this methodology. By looking into *differences in user behavior*, we can identify community conflicts and show that some of the features we identify have implications for identifying communal misbehavior. We apply these features on top of textual features to identify specific types of community sanctions in Reddit (banned subreddits) and show significant improvement over using only textual features using a banning-by-example paradigm.

We applied these methodologies on Reddit where subreddits represent user-defined online communities, but our pipelines are generalizable to any online social media with community structure. All social media do not have explicit group structure like Reddit but we can infer communities via metadata (Facebook lists, Twitter hashtag

community) or graph alignment algorithms [86, 85, 192] as a preprocessing step and apply our pipeline. For conflict and communal misbehavior detection, our pipeline is even more generalizable as our user behavior based features are content agnostic and can be applied to communities with languages other than English.

Our final contribution is presenting CommunityDiff, an end-to-end visualization tool to compare and contrast different community detection algorithms to choose the best algorithm and incorporate user knowledge into the final output by looking into *differences in algorithms*. This methodology is applicable to other sets of algorithms where the ground truth is scarce and there is a lot of disagreement among different algorithm outputs. Prime examples of these types of algorithms are clustering and anomaly detection.

We demonstrate that studying differences of inference, user behavior, and algorithms is useful in multiple regards. We hope that our methodology and analyses would help to deal with problems of online communities and make understanding differences in algorithm outputs easier.

CHAPTER II

Network Representation of Data and Community Detection Algorithms

2.1 Network representation of data

In many data mining and machine learning scenarios, data can be represented as networks or graphs. A graph $G(V, E)$ consists of a set of vertices V and a set nodes edges E. The set of vertices V encodes real-world entities which can range from users in a social network, webpages on the internet to words in text network and roads in the road networks. An edge or link e in E represents connection between a pair of vertices v_i and v_j. An edge can be directed/undirected and weighted/unweighted. For example, Facebook *friend-to-friend* links are undirected but Twitter *follower-following* edges (i.e. person A follows person B) are directed. An edge can refer to many different kinds of relationships between nodes. For example, two users in a social media website may have one kind of edge between them because they are friends or have a different kind of edge because they are from the same geographical area.

2.1.1 Link inference

In many cases, links are explicit e.g., friendship links in social networks. Explicit links may not always represent the semantics we want. we might want to remove some edges, keep some or predict entirely new edges based on the problem. However, in many scenarios, we have to infer the complete network structure from the data [27]. Common examples of these kinds of inferred networks are word and sentence networks in NLP [139] and machine translation, collaborative filtering networks in recommender systems [94] etc. Link inference or network inference is highly related to link prediction [127], but they are not the same. Link prediction usually refers to

predicting if there should be a link between two vertices when we already have an established network structure. However, link inference or network inference refers to creating the whole network from a set of free-floating nodes. However, in many cases, link prediction uses very similar methods as link inference (for example, using node metadata and similarity measures).

There are many different ways to infer a network [27]. Most of them share a common first step — calculating some kind of similarity between a pair of vertices. The similarity can be based on one attribute or many. An example of a single attribute similarity is the textual similarity between the posts of two social media users. As with many similarity metrics, there are many varied ways of computing text similarity [77] and the suitable metric varies from application to application. In other cases, multiple node attributes are combined together to predict similarity values. For hierarchical metadata, tree-based methods [21] are used for comparing keyword similarity. There are variants of link prediction algorithms which can be used to calculate similarity for link inference. For example, a variant of Adamic-Adar measure [1] (a common neighbor based link prediction algorithm) re-weights common attributes of two nodes based on the inverse log of the attribute's frequency (similar to TF-IDF metric) to calculate similarity. Hashing-based methods [165] can also be used for quickly inferring networks from multiple time-sequecnces.

After we have the pairwise similarity values, we then have to choose which edges to keep. Two very common approaches are *global thresholding* (edges below a certain threshold are pruned) and k-nearest neighbor graphs [62] (each node is connected to its k-nearest neighbors based on similarity value). Both approaches have their advantages and disadvantages. K-nearest neighbor graphs are often disconnected into multiple components and by definition directed graphs. However, we can ignore directions or take only reciprocal links (i.e. if there is a link from vertex A to vertex B and there is a link from B to A) to generate undirected graphs. Global thresholding can ensure a connected graph by using a suitable threshold, but in many cases, the threshold is low enough to create a *hairball* graph with too many edges which obfuscates community structure of the network. Another approach is finding the minimum spanning tree [106] for the network. This approach results in a tree which is not suitable for most network analysis algorithms including community detection. More sophisticated approaches like the backbone extraction algorithm [167] focus on preserving the distribution of weights seen in the original data. This ensures a connected graph which is also much sparser compared to global thresholding.

All the above methods are unsupervised, i.e. we do not need training data to infer

a network. However, if training data (i.e. some set of node metadata and a network originating from only those metadata) is available, we can train a machine learning model to infer networks. A related example is measuring social tie strengths in social media [75] using social behavior and communication patterns in Facebook. In this particular example, Gilbert et al. did not infer a network but instead measured the strength of friendship relations using user reported tie strength as training data, but similar methods can be used if training data is available.

2.1.2 Node aligned networks

We refer to graphs which share the same set of vertices or nodes as node-aligned graphs or networks. These networks represent different relationships among the same set of entities. These links can be explicit (friendship links and links denoting if two persons are from the same geographical area, nodes are social media users), inferred (author overlap and topical similarity between two discussion forums) or even based on the same metric at different timestamps (friendship over different periods of time, nodes are social media users). There are many ways compare the similarity between a pair of node-aligned networks. The simplest one is *edge overlap* [149] which calculated the number of common edges compared to the total number of edges. Papadimitriou et al. [149] used signature similarity for anomaly detection in web graphs. Signature similarity is a SimHash [87] based algorithm (a technique for quick estimation of similarity between two sets using hashing) which takes edge weights into account. Bunke et al. [30] proposed several methods to study changes in communication networks which can be applied to measure the similarity between node-aligned networks. However, the two best approaches among them, *graph edit distance* (calculates the edit distance between adjacency matrices of the graph) and *maximum common subgraph* (find the largest common subgraph based on edges) are both NP-complete. Koutra et al. [105] proposed DeltaCon, which computes similarity matrices based on node affinity (computed using fast belief propagation [104]) and compute their similarity. We use community detection to understand different node clusters originated by different sets of edges in node-aligned networks.

Node-aligned networks are similar to multiplex and multilayer networks [101] as all of them connect the same set of nodes in different ways. However, node-aligned networks are not multilayer as each network has a different set od edges and different types of edges are not layered on top of each other in a single network.

2.2 Community detection algorithms

There are numerous community detection algorithms which are used in practice. Some of the widely used ones are described below. All of these algorithms work on undirected and weighted networks. As node-aligned networks are not multilayer networks, we do not discuss and use multilayer-specific community detection algorithms [18, 96, 186].

2.2.1 Fastgreedy

Fastgreedy is a hierarchical agglomerative algorithm by Newman et al. [42, 147] which follows a bottom-up approach. Fastgreedy, as its name suggests, greedily merges communities iteratively by maximizing modularity, a measure of 'modular strength' of a network. Modularity, Q is captured as:

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v, c_w)$$

Where v and w and two nodes, k_i is the degree of node i, and c_i is the community label for node i, m is the total number of edges in the graph, A is the adjacency matrix representation of the graph (i.e. $A_{vw} > 0$ if an edge exists between v and w), and δ is the Kronecker delta — an indicator for testing if the communities are equal. The intuition for this function is that we are testing the number of edges within a community versus the number of edges expected with random assignment. Stated differently, a strong community contains more edges between its members than expected by chance.

Initially, each vertex is in its own separate community. Neighboring communities are merged iteratively, in the favor of maximum modularity increase, until modularity could not be increased further. This algorithm runs much faster than other usual community detection algorithms; hence it is useful for community detection in large graphs. However, it does not perform particularly well in many cases.

2.2.2 InfoMap

InfoMap [163], on the other hand, follows a very different approach and aims to provide the shortest description length of a random walker trajectory. The description length is measured by the expected number of bits per vertex to encode the random walk path. This algorithm uses the minimum description length principle in information theory and follows the idea that a random walk within a community is

likely to stay within the same community as the number of intra-community edges is higher compared to the number of inter-community edges.

2.2.3 Label propagation

Label propagation [159] follows a straight-forward approach of assigning a vertex the most frequent label from its neighborhood. Initially, every vertex is assigned one of the k labels randomly and these labels are updated according to the most frequent label among the node's neighbors. This method is repeated until no label is changed. The initial assignment of labels can significantly affect the outcome of this algorithm. Moreover, the number of different labels limits the number of communities. This method is very fast and suitable for very large graphs.

2.2.4 Multilevel

This is yet another greedy modularity maximization technique by Blondel et al. [136] which follows a hierarchical approach. First, the method finds 'small' communities based on greedy local optimization. Next, a new network is created where the communities found are treated as nodes. The same technique is applied over and over to achieve modularity maximization.

2.2.5 Spinglass

Spinglass [160] originates from statistical physics and is based on Potts model. In this approach, each vertex has an initial spin state from c specified spin states and edges dictate if two vertices would remain in the same spin state or not. This model is simulated a number of times and vertices having the same spin state are put into the same community. This method uses a predefined number of spin states c, so the total number of communities is bound by c. The initial choice of spin states may significantly affect the outcome of this algorithm. Compared to some other community detection methods, this algorithm is rather slow.

2.2.6 Walktrap

Walktrap [153] is a random-walk based community detection algorithm which follows the same idea as InfoMap that a random walk originated inside a community is likely to stay inside that community. Walktrap employs a short random walk usually consisting of three to four steps to build small communities. These communities are merged in a bottom-up fashion hierarchically to achieve the final partition.

2.3 Summary

In this chapter, we described network-related terminologies and existing research related to link inference and node-align networks. We talked about different types of link inference algorithms and their pros and cons. We defined node-aligned networks and created node-aligned networks via link inference. We also described different community detection algorithms that we make use of throughout this dissertation. We can use other network analysis techniques like centrality measures on the node-aligned networks as well.

CHAPTER III

Reddit: Data and Existing Work

3.1 Choice of the Reddit dataset

There are several social media, social networking websites and news/discussion forums where we can apply the expanded pipeline for network-based analysis. This includes popular social networks like Facebook and Twitter, and social aggregator and discussion forums like Reddit. We focus our studies on Reddit for several different reasons.

A major concern while choosing a suitable dataset is the availability of the data and what percentage of the data is available to us. Facebook data is generally not publicly available and Twitter only allows the use of a maximum of 10% of its feed data to select research organizations for academic purposes. On the other hand, Reddit has a comprehensive publicly available dataset compiled by Baumgartner [17]. This dataset contains various types of metadata (author, subreddit, upvotes, downvotes, time of posting etc.) for both Reddit posts and comments and spans over several years (January 2006 to June 2018 at the time of writing and new data is added periodically). Although this dataset does not have deleted or otherwise moderated posts and comments and miss some data [72], it is much more 'complete' compared to other similar social media datasets. This is one of the primary reasons for focusing our research on the Reddit dataset.

As we focus on community-to-community relationships, Reddit provides another unique opportunity over Facebook and Twitter as it operates as a combination of topic-specific user-groups dubbed subreddits. Each of these subreddits focuses on a specific topic or are based on some kind of social aggregation (e.g., image sharing, video sharing) or discussion. With the exception of default subreddits before 2017, subreddits are usually chosen by a particular user and a user can post, comment and subscribe to any number of public subreddits as long as he/she abides by the rules

specified by the subreddits (rules differ from subreddit to subreddit). Subreddits provide an opportunity to study explicit user-defined communities and their interactions compared to inferred user communities in other social media (e.g., followers of a specific Facebook page and Twitter hashtag communities). Another reason for choosing Reddit is its vast scale. For example, in 2016 alone, we have 9.75 million unique users who commented around 743 million times in different subreddits. Considering all comments from 2010 to 2017 we have 3.8 billion comments from 542.6k subreddits.

For different projects, we use different portions of the Reddit dataset due to its vast size. We describe some relevant statistics of the selected data in their respective chapters.

3.2 Related work

Previous work on Reddit is diverse but in large part has focused on a single subreddit or a small, manually-selected set of subreddits, often as case studies to analyze behavior in a specific context. For example, how the subreddit *nosleep* dealt with a sudden increase in readership [100] and the subreddit *FindBostonBombers* led a botched attempt to crowdsource finding the Boston Marathon Bombers [156]. Leavitt et al. used a very different news event — Hurricane Sandy, a natural disaster — to study how news content was produced and curated in real time [116], and to examine how Reddit's user interface affected the production and curation process [117]. Studies have also examined the effects of Reddit's interface design. For example, Gilbert found that social loafing damaged the site's ability to highlight quality content [74]. Others have examined the role of bots [125], throwaway accounts [115] in Reddit's design, and moderator disruptions in calls for policy change [34, 134].

Reddit is also a popular medium for analyzing language on a particular topic — e.g. smoking cessation [182] or mental health [13, 14, 54, 55, 99] — or studying specific types of user interaction, such as social feedback in weight loss communities [47], seeking support for sexual abuse [11], strategies for persuasive arguments [178, 185] or dogmatism in user comments [66]. Reddit data has also been used to train a model that identifies abusive comments [36] and understand users' moral values using word choice [38].

Less research has focused on the structure of Reddit's network itself [148]. Given the opaqueness of Reddit's structure, which has little explicit structure beyond subreddits, researchers have attempted to classify subreddits using a variety of methods and metrics. Zhang et al. characterized user behavior *within* subreddits by us-

ing comment text to map subreddit topics onto four quadrants: generic-consistent, generic-dynamic, distinctive-consistent, and distinctive-dynamic [191]. Relevant to our study, Hamilton et al. characterized a small number of manually collected subreddits according to the loyalty of their users, finding differences in how much time end-users devote exclusively to a particular subreddit [81]. For example, they found that sports subreddits tended to have loyal users while default subreddits did not. While behaviors within subreddits have clear implications to inter-subreddit behavior, these studies did not extend to analyze linking.

Targeted studies have tried to identify the relationships between subreddits. Hessel et al. focused on highly related communities, identified according to their affixes (e.g. *atheism* and *trueatheism*, or *food* and *foodhacks*) [89]. These pairings often indicated a splintering, either as a result of conflict between users or to afford more specialized discussion. However, these instances only represent a small portion of the Reddit network. Reddit's default subreddits and openness to cross-posting presents an additional challenge, as subreddit networks based on cross-posting are quite dense and require additional filtering. Olson and Neal [148] used author similarity to create a network, then used a backbone extraction algorithm [167] to prune the least important connections. Their analysis of a 2013 dataset found 59 communities with a small-world, scale-free network structure. This power-law distribution was partly attributed to Reddit's UX design, in which new users are subscribed to default subreddits [148].

More recently, Martin [133] also made use of author similarity, in this case for applying topic modelling using an adapted latent semantic analysis. The method indirectly identified topic similarity, as well. For example, the author "subtracted" *politics* from *The_Donald* (a subreddit for Donald Trump supporters) to infer which topics *The_Donald*'s authors contributed most when not talking about politics.

A related study to our own by Hessel et al. [88] combined multiple metrics, using a comparison of author and term similarity to identify obscured interests of users by identifying links according to high user similarity and low term similarity. Using this method, the authors identified several interesting examples, such as the relationship between *LadiesofScience* and *FancyFollicles* (about primarily multicolor hair) and *craftit* (a crafting subreddit). The authors based their analysis only on a limited sample text post submissions (maximum 5000 posts per subreddit), rather than comments or submissions in other media formats (common in many of the popular subreddits). In our work, we extend the idea of finding high-author/low-text coherent subreddits by also identifying other misaligned variants.

3.2.1 Trolling in Reddit

We find that differences in user behavior in different Reddit communities has implications in identifying anti-social behavior in group level. we briefly discuss existing research about individual and communal anti-social behavior in Reddit.

Individual trolling in Reddit is predominantly studied through content analysis (e.g., [137]). A key result for Reddit has been comparing the differences between a smaller number of communities in terms of trolling behavior. For example, Schneider performed a contrastive study on intercultural variation of trolling by two subreddits, *ShitRedditSays* and *MensRights* [166]. Most related to our work is the study by Kumar et al. [109] which found that very few subreddits are responsible for the majority of conflicts. This has implications to the conflict graphs we construct in that we may expect key conflict 'nodes.'

More recently, there has been some research on interventions (e.g., banning) on a case-by-case basis. For example, topic models of been used to study the evolution of (a now banned) subreddit *DarkNetMarkets* [154]. Chandrasekharan et al. [35] studied the effect of banning two particular subreddits, *fatpeoplehate* and *CoonTown*, to combat hate-speech. The work concluded that the bans were likely effective in combating hate-speech. However, this work does not elaborate on subreddit-to-subreddit relations before or after the ban. Subreddit relations are discussed from an ideological frame by identifying subreddits which discuss the same topic from different points of view [53]. However, this approach does not capture conflict explicitly. We study the landscape of subreddit conflicts and banned subreddits as a whole instead of doing it on a case-by-case basis.

CHAPTER IV

Identifying Misaligned Inter-Group Links and Communities in Reddit

4.1 Overview

Network modeling of online social systems is a common approach for the study of social behavior. Where explicit, the links between individuals measure friendship, shared interests, or other relationships (family, followers, fans, etc.). When the online system does not have features that explicitly support or encourage linking, we rely on inferred connections. For example, we may infer that two people are "linked" if they post on the same discussion forum or that two communities are linked if they are similar based on text. Inference is *sometimes* necessary in the case of *person-to-person* links and *often* in the case of *community-to-community* links, where explicit links are rare. For example, subreddits (communities on Reddit) tend not to make explicit connections between each other. Yet, they are connected in many ways. Pairs of subreddits may share topics, share authors, share moderators, link to similar content in web, and so on. While indirect [184], similarities based on these features correlate with—and predict—connections. These connections reflect various social processes and can help model both the current state of the social system and the process by which the relationships emerged.

Choices about which similarity measure(s) and inference algorithm to use (not to mention the hyperparameters of the algorithm, such as normalization and thresholding) must be made carefully, as these choices will influence which links are predicted and how they are to be interpreted. The top of Figure 4.1 depicts a conventional analysis pipeline: similarity measures are applied to a disconnected network to generate a pairwise similarity matrix, and then an inference algorithm determines which values should be considered links and produces a network. On these inferred networks,

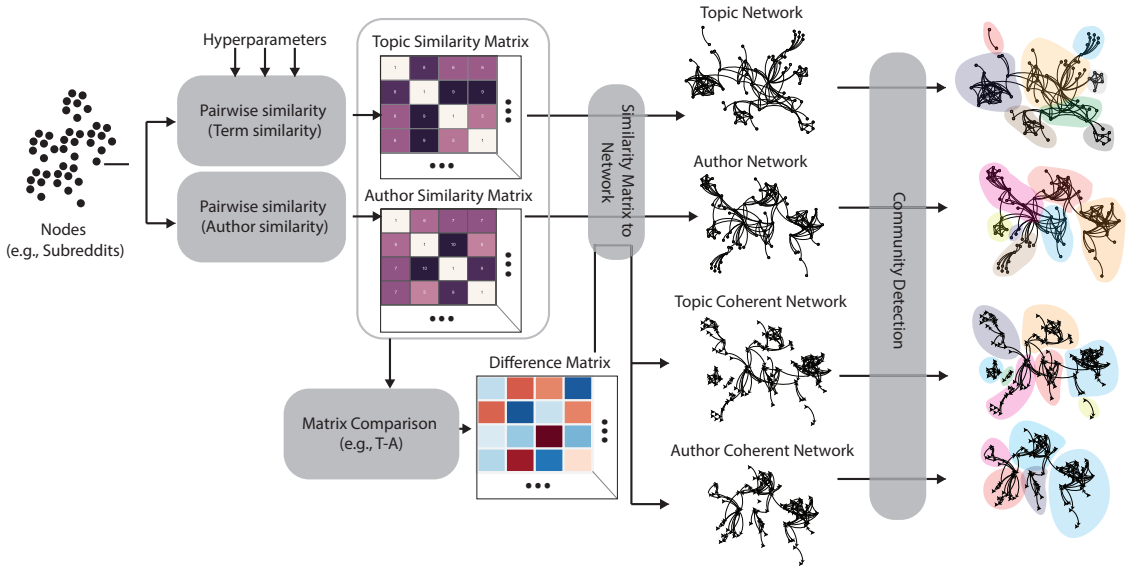

Figure 4.1: Network inference pipeline. Topic and author networks refer to topic and author similarity networks respectively.

downstream analysis such as community detection can be performed (e.g., clustering subreddits into larger communities).

In many situations, different similarity measures are likely to be highly correlated. High author similarity between communities often means topic similarity will also be high. Conversely, low author similarity means we should expect topic similarity to be low, as well. When we see this agreement, it often signals a "good link." Here we treat evidence as additive: if both text and authorship agree, the subreddits should be connected. Many inference algorithms rely on variants of this similarity comparison to infer connections. However, as we demonstrate in the context of Reddit, such correlation can be weak and the many edges that violate this expectation result from behavior and design and may lead to very different outputs.

We argue that *disagreements* between inferences are often as informative as agreements. We define a measure to compare inferred similarity matrices that identifies "misaligned" links (Figure 4.1, bottom). For example, two subreddits may share many authors but discuss entirely different topics; we call these types of links *author-coherent links*. When two subreddits have high text similarity but low author overlap, we call these *topic-coherent links*. To account for the influence of unequal and diverse levels in popularity of different subreddits we develop a score (double-z score, or z^2-score). This score can be used to create directed networks that capture "misaligned" links both locally (in the context of a specific subreddit) and globally.

Our measure of misalignment acts to operationalize, more generally, various structures of interest for social media researchers. For example, social media researchers have targeted phenomena such as "communities at war" [2, 121, 122, 126], community fragmentation (i.e. multiple linked sub-communities instead of a single large community) [71, 89, 190], isolated or niche-interest community links [55], and "strongly linked" communities [89]. Many of these studies have required domain knowledge that is hard to generalize or automate, especially when using common link inference methods. That is, it is difficult to find *multiple* community pairs/groups that have a certain structure (e.g., "communities at war") or to score or rank these found structures for further analysis. Reasons for this include: (1) a single inference algorithm (e.g., text *or* author) does not provide enough "signal" to capture these relationships, (2) algorithms that use multiple inferences (e.g., text *and* author) make naive assumptions about the agreement–or alignment–between the inferred networks, and (3) many algorithms suffer from the presence of a few highly popular communities which tend to be present in a majority of detected links and hide "unexpected" connections. Instead, we demonstrate that *misalignment* between inferred networks can be more generally measured and normalized, and that this measure can be used to find phenomena of interest.

Concretely, we are able to find repeated patterns in these misaligned links. For example, high topic coherence may unearth subreddits that are "at war" with each other (e.g., those with opposing political viewpoints) or have hierarchical relationships (e.g., a niche video game may have a separate community from a more generic gamer subreddit). We also find that subreddits with different ratios of incoming and outgoing links are often out of the mainstream or marginalized. By comparing networks derived through standard similarity measures (e.g., author and text) to our z^2-derived measure, we are able to characterize different types of subreddit-to-subreddit relationships.

Our contributions are twofold. First, we demonstrate a methodology for comparing two inference workflows to identify misaligned links and communities. We introduce a score to compare networks derived from different similarity metrics that can be used to detect properties that are missed when considering only single inference techniques, or those that are additive. Second, we apply these techniques to Reddit to identify subreddit-to-subreddit relationships. We identify key structures (i.e. topic-coherent, author-coherent, and satellite structures). Our analysis classifies how pairs of subreddits interact, how specific subreddits are situated in the broader context of the Reddit ecosystem, and proposes mechanisms by which networks and

higher level communities are formed.

We focus on differences between author overlap and textual similarities to understand community-to-community relations in Reddit and studied a set of node-aligned networks (author and topic similarity networks and author and topic-coherent networks). A paper based on this chapter is published at [53].

4.2 Related work

In addition to work covered in Chapter III, we also identify related materials in the study of politics and social media. The idea of high topic coherence (high text, but low author, similarity) occurs implicitly in the study of political discourse in social networks. Though they discuss similar issues, authors rarely cross-post, leading to fragmentation. Adamic et al. [2] very clearly demonstrated the lack of cross-links between Democratic and Republican bloggers during the 2004 U.S. election. Within more recent social media contexts, Lotan [126] studied Facebook, Twitter and Instagram user networks discussing the topic of the strife at Gaza strip and showed fragmentation within the context of a specific topic. In Twitter, Liu et al. [121] found that users who often mention each other but don't follow each other are "at war." In our work, we demonstrate how warring sub-communities in Reddit can be detected.

Studies comparing text and network structure have also focused on political discourse. For example, Livne et al. [124] studied interactions between political candidates on Twitter during the U.S. 2010 midterm election using both network structure and tweeted. The works notes differences in the strength of correlation between network similarity and language similarity depending on political party. However, the work did not discuss the interaction between the measures. Conver et al. [43] discussed the difference between the mention and retweet network while describing political polarization in Twitter.

4.3 Dataset

We selected Reddit (`www.reddit.com`) due to its popularity and structure. Reddit acts as both aggregator of a diversity of content and as a discussion board. We obtained 10.5 years of Reddit data (posts, authors, comments, etc.) ranging from January of 2006 to June of 2016 [1]. We focus our analysis on the month of June 2016,

[1]The dataset was compiled compiled by Baumgartner [17], available at `files.pushshift.io/reddit/`)

the most recent month at the time of our retrieval. While we find 74,951 subreddits with at least one comment for this period, the distribution is long tail and 22.1% of these subreddits saw only one comment posted. We define a subreddit as "active" if it had more than 500 comments made by more than 100 unique authors in June 2016. Roughly, 500 comments corresponds to the 92.6 percentile in subreddit comment counts and 100 unique authors correspond to 90.45 percentile in subreddit unique author counts. We find 5,193 subreddits that met this criteria. Further filtering out subreddits with "over 18" flags (largely pornographic material), we were left with 4,924 subreddits. Overall, 62.3 million comments (122.7 million sentences) made by 10.6 million unique users were included in our analysis.

Even within this subset, subreddits' activity levels approximate a long-tail distribution. The median number of comments per subreddit was 2083, and the median unique authors was 545. The most active subreddit, *AskReddit*, had about 4.6 million comments made by about 568k unique authors. In contrast, *nashville* — a subreddit well above the median level of commenting activity — had 8573 comments made by 1492 unique authors. This disparity is partly a consequence of Reddit's design. New Reddit users are automatically subscribed to a changing set of "default" subreddits. In our June 2016 dataset, these 56 default subreddits (1.1%) all had more than 2 million subscribers each; no other subreddit had more than 1 million subscribers. These 56 subreddits account for 23.6% of comments.

4.4 Method

The standard analysis pipeline for transforming disconnected entities into a network is illustrated in Figure 4.1. It involves using a similarity metric to create a pairwise similarity matrix. This matrix, often normalized and thresholded, is treated as an adjacency matrix from which a network is constructed. Further analysis, such as community detection, can then be executed on this network. We assume that multiple such pipelines can exist in parallel and that comparing both intermediate and downstream data structures (e.g., similarity matrices, networks, or communities) can lead to interesting findings.

4.4.1 Similarity Metrics

In our analysis, we selected two common similarity measures: *text similarity* of comments and *author overlap*.

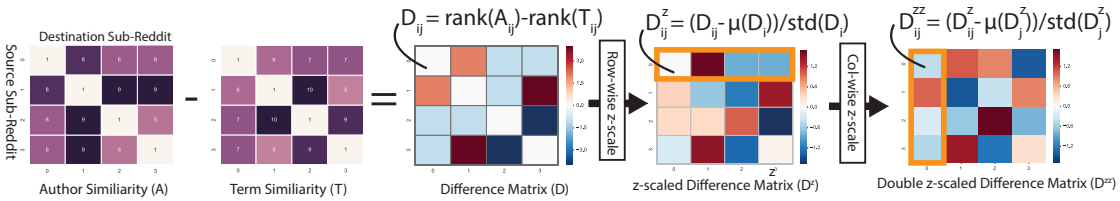

Figure 4.2: Computing the z^2-score.

4.4.1.1 Textual similarity–T_{sim}

Text similarity was calculated by using the angle between term vectors describing each subreddit. Specifically, we applied the standard cosine similarity on a "bag-of-words" model that had been weighted through term frequency-inverse document frequency (TF-IDF) [140]. We applied standard NLP cleaning to all sentences in in a subreddit (stopword and punctuation removal, lowercasing, url removal) and phrase extraction. Sentences of two or fewer words remaining were ignored. Common multi-word tokens (phrases) were detected through a standard algorithm [140]. Specifically, one- to four-grams (all one- to four-word phrases) were extracted from 10% of the text, which we consider training data (roughly 12M sentences). Common "grams" (measured by the number of times the phrase appears relative to the individual terms) were retained. For each subreddit, the count of a particular term (tf_i) was normalized by the maximum frequency for all terms in that subreddit. The IDF frequencies utilized the number of subreddits that contain that term df_i. For calculating document frequency we used all subreddits from June, not only the core 4,924, which removes bias and partially controls for larger values from larger subreddits. The final feature vector for each subreddit contains the TF-IDF score for each term (a term is a single word or a multi-word phrase detected by our phrase detection algorithm) that appears in the corresponding subreddit.

There are many algorithms other than TF-IDF that can be used for measuring textual similarity between two subreddits. These algorithms include word embedding [141] and topic modelling [23, 174]. We opted for a simpler text similarity measure, TF-IDF, as both word embedding and topic modelling approaches are difficult to tune, and are significantly more costly in terms of space and time. This is a concern as our dataset contains 122.7 million sentences. TF-IDF is still a very good measure for measuring text similarity and widely used in information retrieval research.

4.4.1.2 Author similarity–A_{sim}

To calculate author similarity, we similarly calculated the cosine distance to the weighted (TF-IDF) "bag-of-authors." For each subreddit, author frequency (TF) was calculated as the number of times an author posted in the subreddit, normalized by the maximum number of posts made by a single author in that subreddit. Author IDF was determined by the number of subreddits the corresponding author posted on. As with text, for IDF calculation we considered all subreddits that were active in June. Deleted authors were removed.

4.4.2 Matrix Generation

Using the two similarity functions described above, we calculated the pairwise similarity of the 4,924 subreddits to generate two symmetric similarity matrices —A and T— for author and text similarity, respectively. A cell, A_{ij} (or T_{ij}) contained the result of the similarity calculation for subreddits i and j.

4.4.3 Pairwise relationships between subreddits

Once they are constructed, we are able to compare A and T (for example, using Spearman's rank correlation to calculate the correlation between the matrices). The author matrix is sparse, as many subreddits do not share any authors; in contrast, the term matrix contains no 0's as there is invariably some textual overlap. The next steps account for this difference between A and T.

4.4.3.1 Matrix agreement

To compare the likely links, or "edges," that will be formed from the matrices we set thresholds A_{thresh} and T_{thresh} as filters on the corresponding matrix. If the cell value is above the threshold, the cell was set to 1 (an edge exists); otherwise it is 0. Because the author matrix was already sparse, we set A_{thresh} to 0 so all non-zero cells were retained as edges. For the term matrix, T_{thresh} can be varied; we consider this a tunable hyper-parameter of the analysis pipeline. Once we transformed the matrices into binary form, we simply determined the agreement between them as a measure of similarity.

4.4.3.2 Binned comparison

As we have two scores for each subreddit pair, a natural analysis would map each pair onto a standard (though likely binned) x-y plot. One could then easily find pairs matching specific constraints; for example, one could find all subreddits pairs with a 90^{th} percentile score for text and less than the 10^{th} for author similarity. Pairs in this set would roughly correspond to topic-coherent pairs (high term but low author similarity).

This approach has a number of problems, however. First among them is that certain subreddits may dominate the pairings in a particular quadrant. For example, a default subreddit will likely have many author-coherent links, as they have authorship overlap with nearly all other subreddits even when they are topically unrelated. Second, we would ideally like a single score to identify misaligned subreddit connections. Neither the rank differences between similarities nor raw difference produce a satisfactory answer.

4.4.3.3 Double z-score (z^2)

To create our misalignment metric, the z^2-score, we went through a four-step process of calculating and standardizing the differences between the author and term similarity matrices.

To generate a single score comparing the similarity matrix, we might expect to be able to simply calculate a new matrix D where each cell $D_{ij} = norm(A_{ij} - T_{ij})$, meaning each cell in the difference matrix would correspond to the *difference* in the values for that cell in the original similarity matrices. However, because the data distributions for author and term similarity are very different, we chose to calculate the difference matrix using *rank differences* instead of simply subtracting the raw similarity scores.

Thus, the first step was to create *ranked* similarity matrices, where the raw similarity scores for a given source subreddit and each of the 4,923 remaining destination subreddits are ranked against each other. In the original matrices, rows and columns were equivalent, as the raw similarity scores are symmetric. In these new ranked similarity matrices, this is no longer true: for any particular subreddit pair, it is very unlikely that the similarity relationship will be symmetric. For example, a small subreddit is likely to have high author overlap with a large, popular subreddit, simply because of its size; however, this overlap accounts for only a small proportion of the large subreddit's authorship, so the link returning from the large subreddit to the

small one is likely to be ranked much lower.

The second step was to create a single rank-difference matrix D by subtracting the two rank-similarity matrices (the center matrix in Figure 4.2). In this asymmetric matrix, the rows represent the source subreddits; the columns represent the destination subreddits.

Because of the subreddits' diversity, the distributions of rank differences in each row are very different. Therefore, in the third step, we standardize the scores in each row by calculating the z-score (represented by the fourth matrix in Figure 4.2). Represented as an equation: $D_{ij} = norm(rank(A_{ij}) - rank(T_{ij}))$. Recall that the z-score (or standard score) K^z for a set of values K is calculated by subtracting the mean of K, μ_K, from each value k_i in K and dividing by the standard deviation of K, std_K. Thus, $K_i^z = (K_i - \mu_K)/std_K$. The z-score normalized values will be mean-centered on 0 and will capture the number of standard deviations the value is from the mean. In the matrix context, mean and standard deviation can be calculated per row or per column. Therefore, to calculate the single z-score transformed matrix, D^z, we determined the mean and standard deviation of each row D_i of the difference matrix D. Specifically, for any cell D_{ij}^z we computed $(D_{ij} - \mu(D_i))/std(D_i)$. The values in this matrix tell us the difference between author and term ranks for the source and destination subreddit, standardized by the distribution of source similarities.

This has not yet solved the problem of some subreddits simply being similar to all others, however. As described earlier, very large subreddits have this problem because of their size, but it can be caused by other subreddit quirks as well. *CatsStandingUp*, a popular image subreddit, is one example. When comparing its single z-score distribution to that of a second subreddit — say, *pokemongo*— *CatsStandingUp* has high positive z-score in D^z (see Figure 4.3). This is misleading, partly because *CatsStandingUp* has high author similarity with many subreddits, but also because it has unusually low text similarity with most other subreddits: the only word allowed in the comments is the word "cat." Commenting rules such as these can artificially inflate or deflate single z-scores.

To address this, we take our fourth and final step: taking the z-score again, this time *column-wise*. This produces D^{zz}: the double z-score (z^2-score) difference matrix (the rightmost matrix in Figure 4.2). For any cell D_{ij}^{zz} we compute $(D_{ij}^z - \mu(D_j^z))/std(D_j^z)$. Subreddits which have high positive z^2-score have high author coherence: higher author similarity than would be expected, given the term similarity. A high negative z^2-score indicates high topic coherence: higher term similarity than would be expected, given the author similarity. Where term and author

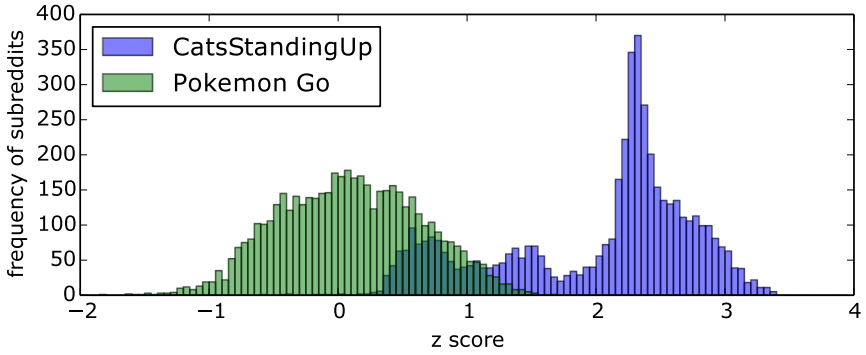

Figure 4.3: Comparison between z-score distribution in rank difference list of all other subreddits for two subreddits: *CatsStandingUp* and *pokemongo*

similarities are about as expected (i.e. aligned), the z^2-score is close to 0.

This final matrix of z^2-scores, D^{zz}, is asymmetric and this matrix can be used to build directed and weighted networks.

4.4.4 Subreddit networks

Given our similarity matrices (author and term) and the z^2-score difference matrix, we are able to produce various network representations. While correlated, the networks have different semantics, each with a different application.

For the **author similarity network**, an easy way to determine which edges should be created would be to use a threshold on author similarity and to create an edge between subreddits if the threshold is exceeded. However, we found that arbitrary thresholds result in a very dense central component with with many disconnected subreddits outside it. This is in part due to default subreddits with high similarity to all subreddits. We instead used an alternative approach popular for producing networks from pairwise similarity values [129]. In this method, we took the top 1% of similarity values for *each* subreddit to create edges. Each threshold is thus unique to the subreddit and ensures a connected graph. To further filter out very weak edges, we included only edges which are in the top 5^{th} percentile of similarity *globally*. The trade-off is that while this does not ensure a completely connected graph, most nodes are connected and edges are largely reliable in community-detection applications. Other approaches for generating networks (e.g., [167]) emphasize other features, such as preserving power-law degree distributions in the final network. The analysis we describe below is applicable when generating networks using alternative strategies.

To produce the **term similarity network** we applied the same technique as above. As with the author network, we did not get a fully connected graph, but we did have a giant component containing majority of the nodes. As we used the same initial nodes (subreddits), we had a direct node-mapping between the author and term networks, though the edges were different.

Using the difference matrix D_{zz} we introduce the notion of **misaligned networks** i.e. networks where edges are defined by misaligned links. Recall that when there is agreement between the author and term similarity for a pair of subreddits, the difference matrix cell will contain a value close to 0. These cells are common, and expected. Extremely high or extremely low values, on the other hand, are what we term "misaligned." Using the difference matrix, we produced two networks: the *author-coherent network*, containing only edges with a z^2-score of 3.0 or more, and the *topic-coherent network*, containing only edges with a z^2-score of -3.0 or less. The first contains only edges when the z^2-score is 3.0 or more. Because the matrices are asymmetric, the produced graphs are directed.

Given these networks, we were able to apply standard metrics such as the clustering coefficient (density of closed triangle compared compared to connected triplets).

4.4.4.1 Community detection and modularity

There are many different community detection algorithms that can be used to detect communities in the undirected similarity networks and the directed misaligned networks. Some of the common algorithms for undirected networks are FastGreedy [147, 42], InfoMap [163], Label Propagation [159], Louvain or Multilevel [136], Spinglass [160] and Walktrap [153]. These algorithms make use of a varied range of underlying techniques for detecting communities. In recent comparisons [6, 112, 157], both Louvain and Infomap are shown to perform well. Louvain or multilevel algorithm [24, 136] is based on modularity maximization for community detection. Recall that modularity is a measure of cohesiveness of a network [145]. Louvain follows a hierarchical approach by first finding small, cohesive communities and then iteratively collapsing them in a hierarchical fashion. Infomap uses a different criteria. Specifically, it is an algorithm based on information theory that assumes that a random walk within a community is likely to stay within that community, as there are more intra-community edges than inter-community edges.

For the author and term similarity networks, we applied all six aforementioned algorithms and compared their results. Random walk based algorithms such as InfoMap and Walktrap produces a large number of very small communities, whereas

Label Propagation and FastGreedy produces one very large community which encompasses most of the network. Louvain produces reasonably-sized (not too small or large) communities compared to other algorithms for these graphs and is scalable for larger graphs. For this reason we use Louvain for community detection in similarity networks.

However, Louvain can only be used on undirected networks. Therefore, for the directed misaligned networks we used Infomap [163]. InfoMap can be applied to both directed and undirected networks, which is a rare property among many community detection algorithms.

4.4.4.2 Measuring difference in the detected communities

To compare the different communities produced by the different networks, we used a metric called Normalized Mutual Information (NMI) [179], which produces a score between 0 and 1 depending on how similar two "partitionings" are. Generally, NMI is used to evaluate two different community partitions given by two different algorithms on the same network. Here, we used NMI to compare community partitions of different networks which shared the same set of nodes. Specifically, we used NMI to give us a single numerical representation of the difference between the communities in the author- and term-similarity networks.

We also calculated the μ-score, which is the proportion of edges that goes outside the community compared to all edges that are touching the community. We can extend this for communities derived from multiple networks. For example, we took all pairs of subreddits in a community and calculated average pairwise author and term similarity. For a "better" community these values should be higher. Especially for communities in the author similarity network, the average pairwise author similarity should be higher than that of the term similarity network; for term similarity network, the average pairwise term similarity should be higher. This gives us an external evaluation (i.e. not dependent on network properties) of the goodness of these communities.

4.5 Results

The z^2-score can give us three types of information: information about individual subreddits, pairs of subreddits, and subreddit networks. In this section, we report descriptive statistics of the matrices, and then illustrate for each information type the characteristics and relationships that can be analyzed using the z^2-score, using

examples from the Reddit data.

4.5.1 Matrices

4.5.1.1 Similarity matrices

An analysis of the similarity matrices produces the expected results, providing a first validation of the z^2-score. As expected, a Spearman's rank correlation found a weak positive correlation between author and term similarity, $r(12120424) = 0.266, p < 0.001$. The relationship between author and term similarity is visualized in Figure 4.4. We note the many subreddit pairs with both high author and high term similarity (represented by the relatively brighter bins in the upper right of the heatmap) and the high values along the diagonal.

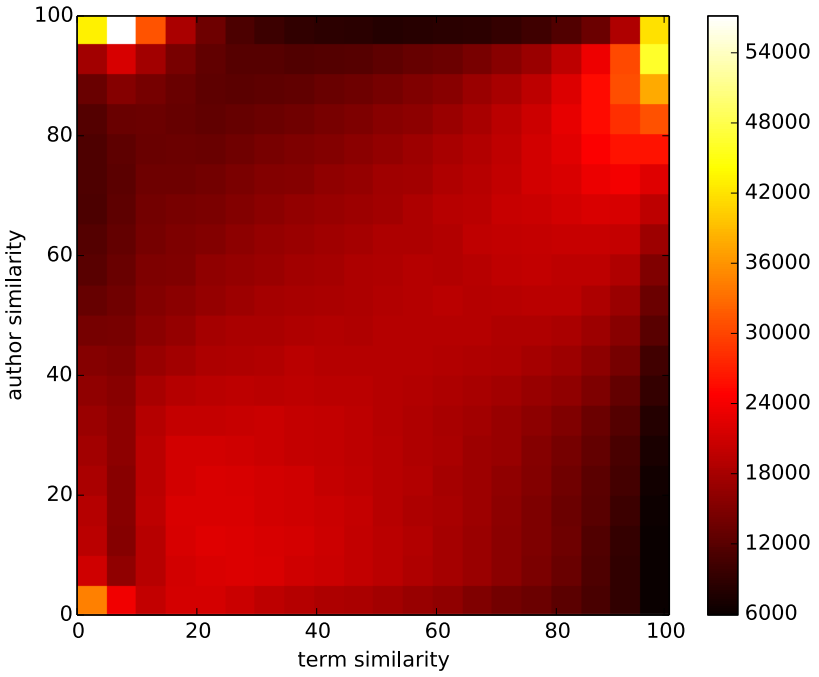

Figure 4.4: Author similarity vs. term similarity for all pairs of subreddits that have non-zero author similarity score. Subreddit pairs are binned in a 20x20 grid according to percentile value. Brighter colors indicate more subreddit pairs in a 2-dimensional bin.

When thresholding the two matrices for creating a simple network representation (i.e. a value above some threshold means an edge should exist) we find the overlap between matrices to range from 43% to 61%. We compute this by thresholding the

author similarity matrix at above 0 and varying the term similarity threshold from 0.05 to 0.95. Edge agreement (based on thresholding) peaks at a threshold of 0.20.

Given the fairly linear fit between author and term similarity we analyzed pairs by ranking outliers. Specifically, we ran a regression analysis on author similarity score and term similarity score and used Cook's distance [44] to identify outliers with undue influence on the regression line. Taking the 1% of subreddit pairs with the highest Cook's distance, we found that some defaults — *AskReddit*, *bestof*, *gifs*, *LifeProTips* — appeared in an unusually high number of pairs. This is consistent with expectations, as Reddit's default memberships induce high author overlap between the defaults and other subreddits.

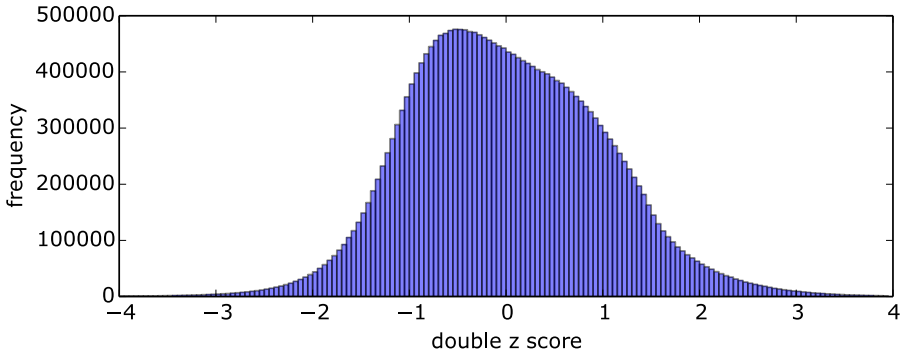

Figure 4.5: Distribution of z^2-scores for all pairs of subreddits.

4.5.1.2 Misaligned matrices

We use the z^2-score difference values to find misaligned links both in terms of *absolute* and *relative* misalignment. We define absolute misalignment as z^2-scores of 3.0 or greater (author coherent) or less than -3.0 (topic coherent), meaning that it is more than 3 standard deviations away from the mean (as z-scores for a majority of subreddits are normally distributed). The distribution of z^2-scores for all subreddit pairs is plotted in Figure 4.5. On average, each subreddit has 18 outgoing author-coherent links and 9 outgoing topic-coherent links. However, there is considerable variation in this value, and in cases where a subreddit has few or no links with z^2-scores high enough to qualify as absolute misalignment, it is still valuable to look at the most misaligned links for that subreddit. We call these types of links *relatively* misaligned links.

4.5.2 Characterizing subreddit pairs

The z^2-score affords analysis of four types of subreddit pairs: strongly similar links (high term similarity and high author similarity), strongly dissimilar links (low term and low author similarity), misaligned author-coherent links and misaligned topic-coherent links. Each of these also have several subtypes. In this section, we report the results of the analysis of the misaligned links.

4.5.2.1 Author-coherent links

The first type of misaligned link is one with high author similarity but low term similarity (a high positive z^2-score). In many cases, this reveals a latent shared interest between the two subreddits. We call these *author-coherent links*. Author-coherent links appear in two types of relationships; we describe both below.

Hierarchical links: Author-coherent links can indicate that the subreddits are part of a hierarchy. One example is the subreddit *pokemon*, which had surprisingly high author similarity with a number of subreddits devoted to other videogames available on the Nintendo 3DS platform (*MyCastleFE*, *EtrianOdyssey*, *monsterhunterclan*, etc.), with z^2-scores ranging from 2.55 to 3.22. Author overlap can be attributed to Nintendo 3DS gamers talking about different games. These subreddits represent niche interests within the same broad topic (Nintendo 3DS games).

In the case of the Nintendo 3DS subreddits, the hierarchy is natural. Others have a hierarchical structure that is enforced through subreddit rules or norms. For example, the subreddit pair *mturk* and *HITsWorthTurkingFor* had a misalignment score of 3.42. Both had a high author overlap, but we expected a similarly high term overlap given the topics (both are forums for Amazon's Mechanical Turk crowdsourcing service).

One strategy for differentiating between natural and enforced hierarchical links is to perform a log-odds analysis [143] of the text in the two subreddits (identifying which terms were more probabilistically likely to appear in one of the subreddits or the other). This makes it possible to identify important differences in terms. Words most likely to appear in *mturk* were related to a general discussion of MTurk (e.g. "work," "pay," "mturk"). The top phrases in *HITsWorthTurkingFor* are from a bot using the same template repeatedly (e.g. "bot action performed automatically"), with very little discussion otherwise. This shows an enforced hierarchical pair, where *mturk* is the general-interest subreddit and *HITsWorthTurkingFor* is a niche subreddit for posting work with primarily bot activity in the comments.

Community fragmentation: Author-coherent links are sometimes an indication of community fragmentation: groups that we would expect to share one community are in fact spread across several communities. For example, *USMilitarySO*, a subreddit for the significant others of U.S. military members, shared high author coherence with subreddits about budget makeup (*drugstoreMUA*, *MakeupRehab*), pregnancy and motherhood (*CautiousBB*, *clothdiaps*), and local subreddits for cities with large military presence (*jacksonville*, *MotoLA*). This is an indication that though many of *USMilitarySO*'s members also post on, for example, *CautiousBB*, they discuss pregnancy primarily on *CautiousBB* and not on *USMilitarySO*. In other words, the community is fragmented across multiple different subreddits.

This interpretation can be validated by "adding" the text of the two subreddits together and calculating term cosine similarity between the combined text (in our example, *CautiousBB* + *USMilitarySO*) and all other subreddits. Subreddits that have similar levels of cosine similarity with *CautiousBB* and *USMilitarySO* individually, and high cosine similarity with *CautiousBB* + *USMilitarySO*, are subreddits represent the topic overlap between the original two. *TheGirlSurvivalGuide* is an example of this: it has a cosine similarity of 0.67 and 0.63 with *USMilitarySO* and *CautiousBB*, respectively, and 0.76 for *CautiousBB* + *USMilitarySO*.

In contrast, subreddits that have high cosine similarity with the combined *CautiousBB* + *USMilitarySO*, but high cosine similarity with only one of the subreddits when taken individually, represent the fragments of the community. Therefore, while *pregnant* has high cosine similarity (0.80) with *CautiousBB* + *USMilitarySO*, it has a much higher similarity with *CautiousBB* than with *USMilitarySO* (0.84 versus 0.53). This validates our initial interpretation that users move to subreddits like *CautiousBB* to discuss pregnancy and do not discuss it frequently on *USMilitarySO*.

In this way, it is possible to identify author-coherent links that indicate community fragmentation without relying on interpretation or domain knowledge of the subreddits in question, and differentiate them from hierarchical author-coherent links.

4.5.2.2 Topic-coherent links

A second type of misaligned subreddit pairs are those with very negative z^2-scores. Here we find high term similarity and low author similarity, which we call *topic-coherent links*.

Communities at war: In some cases, topic-coherent links connect communities that have opposing opinions on the same topic. One example is *TrollXChromosomes* (a feminist subreddit) $\rightarrow$ *MGTOW* (an anti-woman subreddit), which has a z^2-score

of -3.05. Another example is *askscience* (form for discussion of science-related topics) and *theworldisflat*, where participants look for scientific evidence that the world is flat (z^2 of -3.01). In both of examples, the subreddits use similar language but relatively few authors in the one subreddit also post in the other.

One effect of the z^2-score is that when opposing communities cross-post, the score is close to 0. For example, one natural place to look for topic-coherent links is in political subreddits. By June 2016, both Hillary Clinton and Donald Trump had gained the presumptive nomination for their respective parties in the 2016 U.S. presidential election. However, the z^2-scores for the candidates' largest subreddits indicate relatively low topic cohesion. This is because they ranked highly in both term and author similarity. *The_Donald* $\rightarrow$ *hillaryclinton* had a z^2-score near zero, -0.25, indicating the text and author similarities were about what was expected. The z^2-score for *hillaryclinton* $\rightarrow$ *The_Donald* (the mirrored link) was higher, at -1.90, but in both cases they were in each other's top decile in both author and text similarity.

Topic-coherent fragmentation: Other topic-coherent links connect communities that are not necessarily antagonistic. A number of subreddits about different programming languages have high topic coherence, e.g. *javascript* and *matlab*. A number of programming subreddits — *java*, *javahelp*, *programming* — have high topic coherence with subreddits for students taking computer science courses, such as *OSUOnlineCS* and *cs50* (a Harvard University programming class). Interestingly, some of these links are approximately symmetric, often not the case for these misaligned links: *javahelp* $\rightarrow$ *cs50* has a z^2-score of -3.76, while *cs50* $\rightarrow$ *javahelp* has a z^2-score of -3.02. With mutually low author overlap, this example suggests that the students of *cs50* are not utilizing Reddit as a resource for programming help; however, the z^2-score alone cannot differentiate between antagonistic and non-antagonistic topic coherence.

4.5.3 Characterizing individual subreddits

In the previous section, we discussed examples primarily in terms of relative misalignment. If we focus only on the most misaligned links overall—those greater than 3.0 or less than -3.0 — we see that some subreddits have many more outgoing misaligned links than others, or many more incoming misaligned links. For example, *USMilitarySO* has 55 outgoing links outside of the -3.0 to 3.0 range, while *The_Donald* has only one.

Intuitively, we might expect a subreddit with *many outgoing* links with high author coherence (high mean z^2) to mean that the authors in that subreddit also post

in many other parts of Reddit about unrelated topics. *Few incoming* high author coherence links, that might mean that Reddit's "mainstream" is not interested in the subreddit. For example, a mother might post on many subreddits, but mostly talk about motherhood on subreddits devoted to the topic, such as *Mommit*. Similarly, having many outgoing topic-coherent links with few incoming topic-coherent links could indicate that the subreddit serves a niche audience for a general topic that is part of the Reddit mainstream, such as a subreddit for trading skins and other equipment within a specific popular video game (*csgotrade* is one example of this). Conversely, we would expect subreddits with many incoming links and not a lot of outgoing links to be gathering places on Reddit. If there are both author- and topic-coherent links incoming, we might be able to characterize these subreddits as Reddit's "mainstream": places where many different authors with many different interests gather to discuss topics of universal interest. By the same token, a subreddit with many more outgoing than incoming links that are both author- and topic-coherent might represent a community that exists on the margins of the Reddit mainstream: not isolated, but not fully accepted, either.

4.5.3.1 Mainstream vs. marginalized subreddits

To investigate this, we looked at the proportion of outgoing versus incoming links for each subreddit. This proportion of outgoing versus incoming links is similar to what is called the *hub* and *authority* scores [102], where hubs have many outgoing links and authorities have many incoming links. For this analysis, we combined author- and topic-coherent links (i.e. edges were considered if they had a z^2 above 3 or below -3). This made it possible to filter outliers like *CatsStandingUp*, which has an artificially high number of outgoing author-coherent links because its text is limited to the single word "cat." No comprehensive categorization of subreddits exists, so we focused on a small sample of the subreddits at the extremes: those with many more outgoing links, and with many more incoming links. Specifically, we selected subreddits which had above the median number of total links and were in the highest 5% of subreddits with more outgoing than incoming links, and vice versa. We then assigned each subreddit to one of several broad categories according to topic.

This expected pattern is reflected in the data. Subreddits displaying authority behavior (i.e. many more incoming than outgoing links) tend to have content that is appealing to a general Reddit audience. Subreddits displaying hub behavior (i.e. many more outgoing than incoming links) tend to cater to identities that are marginalized from the Reddit mainstream.

We categorized 122 subreddits with the highest proportion of incoming to outgoing links. Of these, 30 (24.6%) were default subreddits, which tend to appeal to general audiences. Another 22 (18.0%) were image boards such as *nonononoyes* and *reactiongifs*. Much of the remaining content covered topics of technology, gaming, and comics. In total, these subreddits — we refer to them broadly as "internet culture" — comprise 63.1% of these 122 subreddits. Of the remaining subreddits, 5 were far-right political forums, 1 was anti-capitalist, and 3 were advice forums targeted at a male audience; the others did not fall into any particular category.

The 121 subreddits with the highest proportion of outgoing to incoming links had a different composition. Of these, only 28 (23.1%) were internet culture subreddits, and tended to have a more specific focus (e.g. *xmen* instead of *comics*). In the remaining subreddits, 28 (23.1%) targeted specific identities that are less well-represented in Reddit's readership: women, people of color, people over 30, and local subreddits for specific cities. Two far-right and four far-left forums were also in this group. The remaining subreddits did not fall into any particular category.

To validate our analysis, we calculated hub and authority scores using the HITS algorithm [102] in a subreddit network with the most misaligned links as directed edges. The score is commonly used in network analysis and identifies central nodes based on both the number of incoming links (authorities) and outgoing links (hubs). The correlations were fairly strong for both: the authority score and number of incoming links had a correlation of $r(2388) = 0.71, p < 0.001$, and the hub score and number of outgoing links had a correlation of $r(2388) = 0.44, p < 0.001$. In addition, a few selected case studies from throughout the proportion distribution were also consistent. Since Reddit contributors are more likely to be male than female, we would expect subreddits targeted toward men to have more incoming than outgoing links, with the opposite being true for subreddits targeted toward women. *AskMen* has 197 incoming links and 0 outgoing links (an authority); in contrast, *AskWomen*, a subreddit with an almost identical number of subscribers, has 1 incoming link and 21 outgoing links (a hub). Further, in a selection of subreddits targeted toward a male or female audience (six each), three of the male-targeted subreddits were hubs, while for female-targeted subreddits only one qualified as a hub, the default subreddit *TwoXChromosomes*. Six LGBTQ subreddits were more evenly split: three hubs, three authorities. However, subreddits targeted specifically toward trans and genderqueer folk were much more likely to be hubs; of the 10 subreddits tested, eight had more outgoing than incoming links.

4.5.3.2 Satellite subreddits

Most misaligned pairs have a high author similarity rank and low term similarity rank, or vice versa. That is not always the case, however, as the z^2-score is not simply a measure of rank difference. Some links have a high z^2-score even though both their author and term similarity ranks are both very high, i.e. in the top 10% for the origin subreddit for both similarity types. These links are interesting because they identify relatively close relationships for subreddits that are otherwise unusually isolated in authorship or shared terms. We call these *satellite pair links*.

One example of this type of subreddit is *Divorce*, an advice and support forum with relatively high term similarity but very low author similarity with the rest of Reddit. It has a median raw term similarity score of 0.33, compared to an overall median of 0.22 for all subreddit links, and yet shares an author in common with only 27.4% of other subreddits, compared to 60.7% for all subreddits. Consequently, the pair *Divorce* $\rightarrow$ *datingoverthirty* has surprisingly high author coherence (z^2-score of 3.11) even though the raw authorship similarity score is low. This suggests that the authors in these subreddits are usually isolated from the rest of the network compared to other subreddits that discuss similar topics.

Subreddits that are unusually isolated in authorship but have multiple outgoing links of high author coherence tend to share common characteristics. Like *Divorce*, most of these subreddits are advice and support subreddits for relationships, mental health issues, or other medical problems. Of the 50 subreddits that have five or more of these high-author satellite pair links, 44 fall under this category. Examples include *selfharm*, *TryingForABaby*, *Fibromyalgia*, and *rapecounseling*. These 50 subreddits have a median text similarity considerably higher than the median for all subreddits (0.31 versus 0.22), but have lower median shared authorship than Reddit at large (median 47.5% versus 60.7%). Further, the majority of these links (55.6%) are to other isolated advice and support subreddits.

Less common are subreddits that are unusually isolated in term similarity with surprisingly high topic coherence. Only 139 of these pairs exist, compared to 2258 for authorship satellite pairs, which make these subreddits more difficult to characterize.

4.5.4 Characterizing subreddit networks

4.5.4.1 Author similarity network

Community detection produced 8 large communities (Table 4.1) in the author similarity network that include all but 6 of the 4,924 nodes (those remaining were

sorted into communities of less than three nodes).

Top 3 subreddits	Size	μ-score	cc	Aut. score	Term score
funny todayilearned pics	1128	0.44	0.16	0	0.21
AskReddit worldnews announcements	988	0.52	0.19	0	0.18
IAmA videos Music	887	0.35	0.16	0.004	0.17
gaming leagueoflegends Games	740	0.49	0.17	0	0.19
space technology europe	595	0.55	0.24	0	0.21
sports BlackPeopleTwitter nfl	360	0.57	0.22	0	0.34
Fitness trees Drugs	160	0.62	0.44	0	0.32
Art ArtisanVideos montageparodies	60	0.56	0.74	0.001	0.23

Table 4.1: Top communities in the author similarity network. The community's three largest subreddits are listed, along with the size and a number of metrics that measure the quality of the community structure: μ-score, clustering coefficient (CC), and median raw pairwise similarity scores for both authors and terms.

As shown in the table, author communities tend to have a relatively high μ-score (median = 0.54) and a relatively low clustering coefficient (median = 0.20), indicating that community structure is fairly poor. This partly a function of the size of the communities. As measured by the clustering coefficient, cohesiveness improves in the smaller communities. Pairwise raw similarity scores are another way to evaluate the community structure. While at first glance, the median raw pairwise author similarities seem very low, this is because these scores are very low overall, with an overall median of 5.24×10^{-6}. The author communities have a median pairwise author similarity that is about 10 times higher than the overall median, an indication that the identified communities are reflecting a true community structure, even if the μ-score and clustering coefficient indicate that it is not strong.

As with the author similarity *matrix*, it is likely that the muddy structure of the author similarity *network* is a reflection of Reddit's design. Because all new users begin with a similar set of default subreddits from which they explore other parts of Reddit, those defaults have author connections to many other subreddits. This would explain why all of the identified communities in the author similarity network include at least one default subreddit, and the largest communities include multiple defaults. Even so, there is a coherent theme in many of these communities: *gaming*, *leagueoflegends*, and *Games*, as an example, or *trees* and *Drugs* (*trees* is a subreddit for users of cannabis).

Top 3 subreddits	Size	μ-score	cc	Auth. score	Term score
gaming sports leagueoflegends	845	0.24	0.27	0	0.31
IAmA blog tifu	728	0.44	0.25	0	0.22
AskReddit funny todayilearned	621	0.60	0.18	0	0.21
worldnews announcements news	516	0.39	0.30	0	0.39
LifeProTips Futurology technology	503	0.53	0.31	0.001	0.45
pics travel beer	289	0.42	0.32	0	0.59
hiphopheads Guitar tipofmytongue	155	0.12	0.44	0.001	0.43
nonononoyes cars bicycling	143	0.13	0.71	0.002	0.46
hearthstone magicTCG CompetitiveHS	45	0.01	0.72	0	0.44
gardening Aquariums Fishing	30	0.08	0.66	0.001	0.39

Table 4.2: Top 10 communities in the term similarity network. The community's three largest subreddits are listed, along with the size and a number of metrics that measure the quality of the community structure: μ-score, clustering coefficient (CC), and median raw pairwise similarity scores for both authors and terms.

4.5.4.2 Term similarity network

Compared to the author similarity network, the term similarity network produced many more communities (Table 4.2): 26 in total, with more variation in size (3 to 845 subreddits). The term similarity network also produced many more communities of one to two nodes, totaling 19.8% of subreddits.

The term similarity network had low μ-scores and high clustering coefficients: communities in the term network have a median μ-score of 0.007 and median clustering coefficient of 0.71. This indicates a good community structure, particularly in comparison to the author similarity network. As with the author communities, cohesiveness improves as the community shrinks. The μ-score also tends to get lower as the communities get smaller, an indication that the community structure is improving. Looking at the pairwise term similarity scores, the term communities' median score of 0.30 is higher than the overall median of 0.22.

As with the communities in the author network, many of the term communities include defaults — unsurprising, given that the defaults are chosen to have widespread appeal — but the influence is not as strong so not all communities include defaults. As would be expected, the themes of each community are clearer than in the author communities, and we can clearly make out different gaming, news, technology, music, and hobby-related communities.

4.5.4.3 Comparing similarity networks

We find the NMI score between the term and author derived networks to be fairly low at 0.284. In this particular pairing the differences are likely due to the existence of default subreddits. Communities in author similarity network are larger and mostly centered around these defaults, whereas communities in term similarity network vary greatly in size and are topically more cohesive. However, we note that the largest communities in both networks share many common nodes.

4.5.4.4 Misaligned networks

We computed two misaligned networks using the z^2-scores, one with high author-coherent links ($z^2 \geq 3$) and one with high topic-coherent links ($z^2 \leq 3$). Both are composed of one giant component (containing 2670 and 2023 nodes for the author- and topic-coherent networks, respectively), and many small connected components.

After running the InfoMap community detection algorithm, we found the clearest structure in the topic-coherent network. Disregarding communities with fewer than 5 subreddits, the algorithm found one large community of 1940 subreddits and many 127 smaller communities ranging in size from 5 to 143. These communities usually have very low clustering coefficient (median 0.0) and very high *mu*-score (median 0.7), but have median pairwise author and term similarity scores that are much higher than the communities in the similarity networks (3.3×10^{-4} the median author score and 0.31 median term score in the similarity networks). Table 4.3 shows some examples of these topic-coherent communities. We observe that they consist of subreddits related by broad general category (e.g., "sports") with different sub-topics that often have very little to no overlap in participation (*nfl*, *soccer*, *nba*).

Top 3 subreddits	Size	Broad category
listentothis tipofmytongue electronicmusic	143	Music and music/video recommendation
hearthstone magicTCG boardgames	91	Board games and card video games
Python perfectloops dailyprogrammer	81	Programming
Diablo diablo3 Smite	68	Online multiplayer gaming
Gunners LiverpoolFC Seahawks	48	Sports fan clubs
EatCheapAndHealthy keto fitmeals	46	Food and health
nfl soccer nba	45	Sports
compsci jobs cscareerquestions	44	Education and career
bicycling motorcycles MTB	39	Motorcycles and bicycles

Table 4.3: Some interesting misaligned communities in the topic-coherent network. The community's three largest subreddits are listed, along with its size and broad categories.

4.5.5 Summary

The z^2-score method makes it possible to characterize communities and the relationships between them in a dense network with little explicit structure. By using multiple measures of similarity — author overlap and term similarity — we are able to identify relationships that would be invisible if one used a single measure, and use those relationships to analyze elements in the network that are of interest to researchers. Below, we summarize how the z^2-score can be used to analyze different elements in the network.

4.5.5.1 Individual subreddits

- **Mainstream vs. marginalized subreddits**: By calculating the proportion of the total number of incoming and outgoing links with extreme z^2-scores (similar to *hub* and *authority* scores in network analysis) one can differentiate between communities that are part of the network's mainstream and communities that participate in the mainstream but are pushed to the margins. In this way, one can also learn about the network as a whole: what is mainstream, what is marginalized, and what is isolated.

- **Satellite subreddits**: Satellite subreddits have authors that are more isolated from the rest of the network than would be expected, given the subreddit topic, or vice versa; in our analysis, these communities tended to cater toward vulnerable users (e.g. *selfharm*). These are distinct from subreddits that are just isolated, which can be found simply by using the similarity matrices to find subreddits that have low term and low author similarity. Satellite subreddits can be identified using a particular subtype of author- and topic-coherent links we call *satellite pair links*, in which two subreddits have *both* very high author and very high term similarity but still have a high z^2 because both subreddits are otherwise unusually isolated from other subreddits.

4.5.5.2 Subreddit pairs

- **Author-coherent links**: These relationships indicate that a community (in this analysis, a subreddit) shares more authors with another community than would be expected, given their level of textual similarity. These links can identify two different phenomena: *hierarchical links* and *community fragmentation*. Hierarchical links occur between subreddits that represent niche interests within the same

broad topic; these can occur naturally, or can occur as a consequence of subreddit rules. Community fragmentation occurs when a single group of users that we would expect to share one community instead spread across multiple communities to discuss different topics. These links are interesting because they can reveal a shared interest in topics that do not always seem related at first glance.

- **Topic-coherent links**: In these relationships, a community has more textual similarity with another community than would be expected, given their author overlap. This usually means that two different groups of people are discussing the same topic, but not talking to each other. These can take several different forms. Some are links between *communities "at war"* and not speaking to each other; conversely, links with low z^2-scores between two antagonistic communities indicate that they are cross-posting on each others' forums. Topic-coherent links are not always an indication of antagonism, however; in *topic-coherent fragmentation*, the linked communities may have ambivalent or neutral opinions of the other.

4.5.5.3 Subreddit networks

The z^2-score can also be used to characterize a network as a whole. Unlike the results for individual subreddits and subreddit pairs, where we reported only the misaligned results, we analyzed networks constructed from the similarity matrices as well for additional face validity. In the undirected similarity networks, both author and term networks show definite community structure in Reddit, although the communities in author similarity network are very much focused on default subreddits as an artifact of Reddit design (all new users are automatically subscribed to default subreddits). For the directed misaligned networks that used z^2-scores, the author-coherent network did not show a pronounced network structure when a directed community detection algorithm is run, but topic-coherent network produced small communities based on a common, broad interest with the individual subreddits different enough from each other that they have few users in common.

4.6 Discussion

In this chapter we demonstrate how the z^2-score based methodology can be used to find misaligned links and communities. The z^2-score makes it possible to find particular relationships that have been identified as interesting by previous research but have been difficult to find and characterize systematically. Hierarchical communities

and community fragmentation, both of which can be identified using the z^2-score, are important in understanding how a network and its communities develop over time. In addition, the z^2-score may have other practical applications, such as identifying vulnerable users who may be at risk of harm. For example, high-author satellite subreddits are populated by users who are unusually isolated from the rest of Reddit; as this tends to correspond with subreddits such as *SuicideWatch* and *selfharm*, these users may be in particular need of support.

4.6.1 Limitations

The z^2-scores alone can not distinguish between all relationship types; in many cases, human input and domain knowledge is required to interpret the z^2-score results. For example, we can not distinguish "communities at war" from other topic-coherent links. Human input or additional NLP techniques (e.g., sentiment analysis) are necessary. Though rare we also find that if there is significant cross-posting between warring subreddits, then both author and term similarity between the subreddits are high. Thus, z^2-score alone can not detect these pairs. For example, *The_Donald* and *hillaryclinton* are warring subreddits, but many authors from *The_Donald* posted in *hillaryclinton* in the month of June. This resulted in high author similarity between both. We also need to employ additional measures to identify misaligned links that are produced by subreddit moderation like *mturk* $\rightarrow$ *HITsWorthTurkingFor*. We demonstrated some techniques that can automatically distinguish between relationship types — for example, the subreddit "addition" that makes it possible to differentiate between hierarchical and fragmented communities — but some level of human interpretation was still necessary in many of our reported results.

We also need to keep in mind that z^2-score makes use of pairwise similarity values that necessities access to metadata of all social entities in the process. Getting access to metadata is difficult in many cases. For example, we do not have access to moderator list of all public subreddits. This limits the usefulness of z^2-scores when using common moderators as a similarity measure in Reddit.

4.6.2 Future extensions

4.6.2.1 Expanding text analysis

Additional quantitative methods such as log-odds ratios or sentiment analysis could shore up validity in future analyses. For example, sentiment analysis might be able to differentiate between communities at war and topic-coherent fragmenta-

tion. Developing additional techniques that can automatically differentiate between different types of relationships will increase the method's validity and reduce the need for subjective interpretation. It would also be useful in z^2-score analyses that look for additional hidden structures in the network. For example, Reddit has a small network of non-English-language subreddits. Unsurprisingly, these tend to be more isolated from the rest of Reddit. However, links of high author coherence pointing to these subreddits from English-language subreddits can show points of crossover, e.g. between English and non-English soccer subreddits.

4.6.2.2 Additional similarity measures

Useful insights can be gained from using other kinds of similarity measures between subreddits, such as a moderator network where edges between subreddits indicate that they share at least one common moderator. In a preliminary analysis, we found a community of Internet meme subreddits (*dankmemes*) and subreddits of popular animes in meme culture (*KillLaKill*, *cowboybebop*). This is not apparent when observing only author and term similarity networks.

4.6.2.3 Application in other social media

The concept of z^2-scores can readily be applied to other social media and social networking websites like Twitter or Facebook. For example, differences tweet hashtag use and @-mentions can reveal nuances in communication in Twitter. Our pipeline does not restrict the user from using any kind of similarity measure between two entities. Moreover, choice of similarity measurement algorithms or community detection algorithms can be fine-tuned. We believe with appropriate choice of similarity measures and algorithms z^2-scores can be used to detect "communities in war", community fragmentation or isolated groups in other social media. However, as discussed before, when we have incomplete data, z^2-scores have limited usefulness as they depend on pairwise similarity values.

4.7 Conclusion

We described a method for inferring network structure using different similarity metrics for social media data. Rather than focusing on the agreement between different scores, we identified the importance of differences in capturing uncommon structures and behaviors. We provided a method for comparing the pairwise simi-

larity matrices and a normalization (the z^2-score) that identifies 'misaligned' connections. Both extremely high and extremely low values of z^2 can be used to produce 'misaligned networks' that display topical or author coherence. We applied these methods to the study of subreddits and demonstrated that they were able to identify (and help in classifying) different types of behavioral patterns as well as artifacts of UX design. We believe that our technique can be applied in other scenarios where network inference is employed in the study of social media.

In the next chapter, we look into differences in local user behavior in subreddits instead of differences in inference. We find that there are implications for anti-social behavior if we study users who behave differently in different subreddits.

CHAPTER V

Extracting Inter-community Conflicts in Reddit

5.1 Overview

Anti-social behavior in social media is not solely an individual process. Communities can, and do, antagonize other groups with anti-social behaviors. Similarly, both individuals and communities can be sanctioned in reaction to this behavior. On Reddit, for example, individuals can be banned or otherwise be sanctioned (e.g., have their posts down-voted). Likewise, entire subreddits can also be sanctioned when multiple individuals use the community as a platform for generating conflict, in violation of general Reddit norms. We look into anti-social behavior in Reddit through the lens of difference in user behavior in different subreddits.

Critically, the form of anti-social behavior at the community level can be quite varied. The ability to identify and coordinate with others means that actions considered anti-social for the individual can be expanded to group settings. A group can thus act anti-socially — producing mass spamming and trolling, flame wars, griefing, baiting, brigading, fisking, crapflooding, shitposting, and trash talking — against both individuals and other subreddits [109, 166]. On Reddit, as in other discussion boards, the ability to create (multiple) accounts under any pseudonym can further exacerbate such behaviors. Although a vast majority of users are generally norm-compliant, anonymity can lead to less inhibited behavior from users [176]. In aggregate, the result is an entire embedded network of subreddit-to-subreddit conflicts inside of the Reddit ecosystem. Research has found specific instances of these conflicts. Our goal is to inferentially identify the structure and dynamics of this *community-to-community* conflict network at scale.

To achieve this, we address a number of challenges. First among them is the lack of explicit group membership. Group 'membership' in Reddit, and systems like it, can

be vague. While subscriptions are possible, individuals can display member-like behaviors by posting to subreddits they are not part of. Such behaviors — subscription and posting — are not, however, a clear indication of the individual's 'social homes.' An individual can display both social and anti-social behaviors within the community via posting. Instead, what we seek is not simply to identify an individual's 'home' but to further discriminate between *social homes* and *anti-social homes*.

To achieve this separation, we apply a definition that extends Brunton's construct of spam: a community defines spam as (messaging) behavior that is not consistent with its rules and norms [28]. That is, we seek to separate *norm-complaint* behaviors that indicate social membership and those that are *norm-violating* (indicating an anti-social home). Rather than relying on a global definition of norms, we utilize the sanctioning and rewarding behavior of individual subreddits in response to norm violation and compliance respectively. An explicit measure we leverage is up- and down-voting on posted comments. While these are not the only kind of sanctions and rewards, they are (a) consistently used, and (b) can be aggregated both at the individual and community levels. As we demonstrate below, inference based on these lower level signals can help identify broader conflicts.

A further appeal of the bottom-up approach is that the converse, top-down identification of sanctions at the subreddit level, does not provide a clear indication of conflict. First, this signal is sparse as the banning of subreddits remains rare. Except for explicit *brigading*, which are (hard to detect) coordinated attacks on another subreddit, community-based anti-social behaviors may not result in a community being sanctioned. Second, even when a sanction is employed it may be due to other reasons than community-on-community attacks. For example, subreddits such as *fatpeoplehate* (a fat-shaming subreddit) and *europeannationalism* (a Nazi subreddit) have been banned but not necessarily due to any specific 'attack' but rather non-compliance with Reddit-wide norms on hate speech.

Our bottom-up inference is different in that we can identify pairs of social and anti-social homes and aggregate these to find conflicts. Specifically, we can find authors implicated in conflicts — which we call *controversial authors* — by identifying those that have both social *and* anti-social homes. From this, we can say that if multiple authors have a social-home in subreddit A *and* an anti-social home in subreddit B, then there is a directed conflict between A and B. By finding aggregate patterns using all Reddit comments from 2016 (9.75 million unique users and 743 million comments), we can construct the subreddit *conflict graph* at scale. Furthermore, we demonstrate how the directed edges in our graph can be weighted as a measure of *conflict intensity*.

The process of identifying conflict edges and their associated weights is complicated by the inherent noise in behaviors and high-variance of community sizes. A specific contribution of this chapter is the use of different aggregation and normalization techniques to more clearly identify the conflict graph.

Using this graph, we can determine not only the broad landscape of community-to-community conflicts but can answer specific questions as well: Which subreddits are most often instigators of conflict (versus targets)? Are conflicts reciprocal and are they proportional in intensity? Does 'attacking' multiple subreddits imply broad misbehavior by members of that subreddit or the work of just a few individuals? Are certain subreddits targeted 'together?' Do conflicts shift over time?

Briefly, we find that subreddit conflicts are often reciprocal, but the conflict intensity is weakly negatively correlated with the intensity of the 'response.' We also find the larger subreddits are more likely to be involved in a large number of subreddit conflicts due to their size. However, our analysis of the fraction of users involved can isolate situations where both relative and absolute counts of involved authors are high. Additionally, we find different patterns of conflict based on intensity. For example, a single subreddit targeting many others may divide its attention, resulting in decreased intensity across the targets. On the other hand, we find anecdotal evidence that subreddits which act as social homes to many controversial authors and have high average conflict intensity against other subreddits often display communal misbehavior. Because of the longitudinal nature of our data, we are also able to perform a dynamic analysis to isolate temporal patterns in the conflict graph. We find, for example, that subreddits that conflict with multiple other subreddits change their main focus over time.

Our specific contributions are mapping the static and dynamic subreddit conflict networks across Reddit. We identify group membership and define the concept of social and anti-social homes as a way of defining conflicts. By analyzing the different static and temporal patterns in subreddit conflicts, we provide evidence for mechanisms that can identify communal misbehavior. We provide a baseline for quantifying conflicts in Reddit and other social networks with 'noisy' community structure and where individuals can behave (and misbehave) in a communal fashion. This chapter has implications in identifying community features which can be used to automatically monitor community (mis)behavior in such social networks as an early warning system.

In this chapter, we focus on differences in user behavior in different subreddits to identify and understand patterns in community-to-community conflicts in Reddit.

We make use of two node-aligned networks the conflict graph and the co-conflict graph for this purpose. A paper based on this chapter is published at [51].

5.2 Related Work

In addition to relevant related works on trolling in Reddit in Chapter III, we describe works pertaining to conflicts in social media and signed social networks in this section.

5.2.1 Conflicts in Social Media

Undesirable behavior in online communities are widely studied in social media research. Qualitative analysis often focuses on identifying and characterizing different types of inappropriate online behaviour or provides case studies in different forums. Analysis of Usenet news, for example, helped explore identity and deception [58]. General anti-social behaviors [83] can also manifest in 'site-specific' ways as in the trolling and vandals on Wikipedia [169], or griefing and combative strategies in Second Life [41].

Predicting trolls and other anti-social behavior is another well explored research area. For example, researchers have studied the connection between trolling and negative mood which provided evidence for a 'feedback' loop that contributes to further trolling [39]. In the context of *prediction*, several studies focused on detecting certain anti-social behaviors on specific sites (e.g., vandalism in Wikipedia [5, 110]). Others have attempted to predict both anti-social behaviors or sanctions. Examples of the former include finding sockpuppets (same user using multiple accounts) on discussion sites [108] and Twitter [73]. Within Reddit, Kumar et al. [109] studied controversial hyperlink cross-postings between subreddits to identify community conflict. Examples of the latter (sanction prediction) include future banning based on comments [40] and using abusive content on one forum to predict abuse on others [36]. The bulk of research has emphasized the behavior of individuals rather than inter-community anti-social behavior (rare exceptions emphasized specific types of anti-social behavior). While we draw upon this literature to understand individual trolling, our focus is on a broad definition of higher-order inter-community conflicts. That is, our aim is to identify inter-community conflict (rather than individual-on-individual or individual-on-community) by developing behavioral mapping mechanisms in the context of the broader network.

5.2.2 Signed Social Networks

We analyze subreddit conflicts by creating a subreddit conflict graph, which can be viewed as a *signed graph* (where all the edges are marked negative). Use of signed graphs for trolling detection is uncommon but has been explored in past research. Kunegis et al. [111] predicted trolls and negative links in Slashdot (a technological news website and forum where users are able to tag other users as 'friend' or 'foe'). Multiple studies [170, 187] proposed models to rank nodes in signed social networks. Signed networks incorporate both positive and negative edges. In our case, it is difficult to make claims about positive relations in the conflict graph. Because most individuals are norm-compliant, edges constructed between two *social* homes may be an artifact of authors being largely norm-compliant and simply reflect correlated interests. In contrast, an author that displays both norm-compliant and norm-violating behaviors provides a better indication of likely conflict.

5.3 Dataset

For the analysis presented here, we used all publicly available Reddit comments from 2016. This was a subset of the multi-year Reddit data (posts, authors, comments, etc.) compiled by Baumgartner [17]. We specifically mined commenting behavior (rather than posting) for building conflict graphs. Comments are much more prevalent than posts, and anti-social behavior in Reddit often involves inflammatory comments rather than posts. For each comment, we make use of the following metadata: author of the comment, which subreddit the comment was posted on and how many upvotes and downvotes the comment received. We found that there are 9,752,017 unique authors who commented at least once in Reddit in our sample. Though largely a 'human population,' bots can also be programmed to generate comments. Of the 9.7M authors, 1,166,315 were 'highly active,' posting more than 100 comments throughout the year. On average, a Reddit user posted in 7.2 subreddits and commented 76.2 times in 2016. As may be expected, most Reddit users are pro-social. In 2016, we find that 79.2% of authors (across all of Reddit) have at least 90% of their comments upvoted.

5.4 Identifying Inter-community Conflicts

To define the *conflict graph* between subreddits we need first to identify edges that capture community-on-community 'attacks.' We would like these edges to be

directed (as not all conflict is reciprocated) and weighted (to indicate the strength of the conflict). Our goal is not to only identify 'passive' ideological opposition but also behaviors where one subreddit actively engages with the other.

This distinction is important as there are instances where two subreddits are discussing the same topic through different ideologies (as determined through text analysis), but have very low author overlap. For example, the *askscience* (discussion forum for science-related topics) and *theworldisflat* (forum for scientific evidence that the world is flat) could be considered to be ideologically opposed [53]. However, there are very few authors who post in both subreddits, meaning there is no engagement and no 'conflict' by our definition. These individuals do not agree but largely leave each other alone.

Instead, we focus on identifying individuals that post to multiple subreddits and behave *differently* depending on the subreddit. In our model, behaviors, such as commenting, can be norm-compliant or norm-violating. Norm-compliant are those behaviors that the community finds agreeable in that they are consistent both with the way behaviors (e.g., message posting) should be done and/or the content of the message itself. Norm-violating are those behaviors that are disagreeable in the same way (how they're posted or what is in them). Norm-violating behavior can include traditionally anti-social behaviors: flame wars, griefing, spamming, trolling, baiter, brigading, baiting, fisking, crapflooding, shitposting, and trash talking. This, again, is consistent with Brunton's spam definition [28]. The appeal of this localized definition of spam is that each community can assert what they consider social or anti-social behavior (i.e. norm-compliant and norm-violating) and can make local decisions to reward or sanction such behaviors respectively.

Our inferential goal is to operationalize social and anti-social behavior by leveraging reward and sanction behaviors as indicators. For this purpose, we use up- and down-votes. Obviously, not all compliant behaviors are rewarded through up-votes, nor are all norm-violating sanctioned through down-votes (banning being a notable alternative). Other metrics for norm-violation may include identifying banned users or users whose comments are regularly removed by moderators. Unfortunately, such data is not readily available (removed comments and authors will be missing from the dataset). Posts can also be marked as 'controversial' to signal undesirable behavior, but these are not always anti-social per se. Additionally, both banning and controversial post 'tagging' may not be reliably imposed. Upvoting and downvoting, however, are specifically part of the incentive structure for Reddit and are both uniformly applied and ubiquitous.

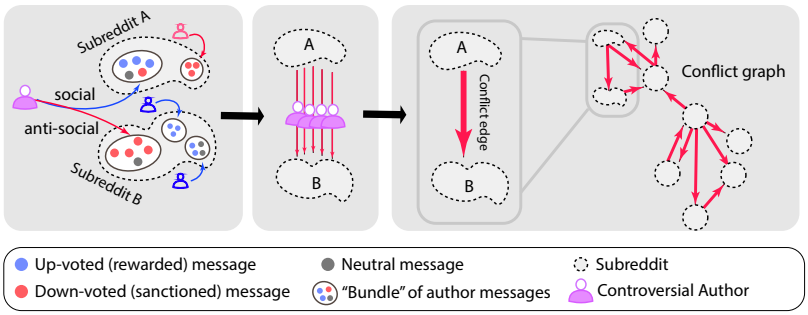

Figure 5.1: General methodology for identifying conflicts and creating the conflict graph.

An individual who reliably produces enough *measurable norm-compliant behavior* (e.g., many upvoted messages) can be said to have a *social home* in that community. Likewise, an individual that produces a substantial amount of *measurable norm-violating behavior* (i.e. many downvotes) is said to have an *anti-social home* in that community. An individual can have multiple social and multiple anti-social homes. Because our goal is to find conflict *edges* we do not consider authors that are *only* social or *only* anti-social. Those who are globally norm-violating (e.g., spammers, malicious bots, etc.) and are negatively treated in all subreddits in which they post are removed from consideration. Figure 5.1 (left) illustrates this idea.

A second key aspect in building the conflict graph is in aggregation. One *particular* individual may have a social home and an anti-social home. However, from the single example, we can not infer that the other members of that person's social home would endorse the messages the person is posting to the other subreddit. Instead, we look for signals in the aggregate. If there are many individuals, who cross-post to two subreddits — where one subreddit is clearly a social home, and the other is clearly an anti-social home — we infer that a conflict exists. This conflict need not be reciprocated, but as we show below, it often is.

We can roughly quantify the anti-social behavior of a user within a subreddit if he/she has more downvoted comments compared to upvoted comments. Note that a single comment can have multiple upvotes and downvotes. Reddit automatically upvotes a user's own comment (all comments in Reddit start with one upvote). We consider the user's upvote as a 'baseline' as we assume the author views his or her own comment positively[1]. We say a comment is *downvoted* (in aggregate) if the

[1]The algorithms described in this project can be applied with or without a individuals' personal upvotes. However, we note that exclusion of this number may slightly change the descriptive statistics we report.

total number of downvotes for the comments exceeds upvotes, and *upvoted* when upvotes exceed downvotes. Similarly, we determine that a user has shown social behavior if they have more upvoted comments (rewarded, norm-compliant) compared to downvoted ones (sanctioned, norm-violating) within a subreddit.

To distinguish between an author's 'home' and simply a 'drive-by' comment, we enforce a threshold (we call this *significant presence*) of more than ten comments in the subreddits over the course of the year (2016). This threshold also ensures that we can observe enough up and down votes for any particular author. Additionally, new authors in a subreddit might break some unfamiliar rules, and receive downvotes initially. Our threshold gives sufficient data to determine if they 'learned.' We also enforce that the user has more than 100 total posts in 2016, which ensures that they have an overall significant presence on Reddit.

As authors were automatically assigned to *default subreddits* (*AskReddit*, *news*, *worldnews*, *pics*, *videos*), many Reddit authors began by posting in these groups [2]. Norms (and norm-compliance) in these subreddits may be significantly different from rest of the subreddits. Using our definition of social homes, a large number of users have at least some default subreddit as their social home or anti-social home just because they started by posting in these forums. For this reason, we exclude default subreddits from our analysis.

5.4.1 Controversial Authors

We denote an author with at least one social and one anti-social home as a *controversial author* (the purple figures in Figure 5.1). In 2016, 1,166,315 authors had more than total 100 comments over the year. After filtering for significant presence in subreddits, we found 23,409 controversial authors. This indicates that only about 2% of the more prolific Reddit users fall into this category. Among the controversial authors, 82% have only a *single* anti-social home. The vast majority (92.5%) of controversial authors have *more* social than anti-social homes. This indicates that these authors differ from the conventional idea of a "troll" who misbehaves in every forum they participate in. This also means that a typical controversial author focuses his/her 'misbehavior' on a small number (usually 1) subreddits. This result is consistent with Reddit users being loyal (in posting) to a small set of subreddits [81].

In aggregate, if there are many controversial authors that have a social home in subreddit A and anti-social home in subreddit B, we view this to be a directed conflict

[2]Though this does not impact our analysis (for 2016 data), we note that default subscription was replaced in 2017 with a dynamic popular subreddit homepage.

from A against B. We call this a *conflict edge*. The sum of all these edges, after some additional filtering, captures the *conflict graph* (Figure 5.1, right).

Our approach has the benefit that aggregation can eliminate various types of noise. While upvoting/downvoting is noisy at the level of any particular message, aggregation at the author level allows us to look for *consistent* behaviors (i.e. are messages from an author always rewarded in one place and sanctioned in another?). Noise at the level of a *particular* controversial author is similarly mitigated by aggregation (are there *multiple* individuals being rewarded in one place and sanctioned in the other?).

5.5 The Subreddit Conflict Graph

5.5.1 Constructing the Conflict Graph

To construct the conflict graph, we apply the following strategy. If k authors have a social-home in subreddit A, and an anti-social home in subreddit B, we can create a weighted directed *conflict edge* from A to B. If we create these edges for all subreddit pairs, we have a graph of antagonistic subreddit relations. Weights for these edges must be normalized as different subreddits have a different number of users. Thus, a raw author count (i.e. common authors with a social home in A and anti-social home in B) is misleading. Larger subreddits would dominate in weights as more authors often means more controversial authors. For convenience, we refer to the 'source' of the edge as the *instigating* subreddit and the 'destination' as the *targeted* subreddit.

We normalize the raw controversial author counts by the number of common authors in both subreddits. Furthermore, for each subreddit pair, we require that there are at least five controversial authors between them to ensure that we are not misidentifying a conflict due to very few controversial authors (i.e. if there is k_1 authors with social home in subreddit A and anti-social home in subreddit B, and k_2 authors with social home in subreddit B and anti-social home in subreddit A, $k_1 + k_2$ must be at least five). We emphasize that the weight, direction, or even existence, of an edge from subreddit A to B, is very different from an edge from B to A.

5.5.2 Eliminating Edges Present due to Chance

While defining conflict between a pair of subreddits, we need to make sure that users are not perceived negatively in the attacked subreddit by chance. For two subreddits A and B with n_{common} common users and n_{actual} users perceived positively in subreddit A but negatively in subreddit B (we only consider users who posted more

than 10 times in both subreddits), we calculate the number of users who can be perceived negatively in subreddit B by chance. First, we define an empirical multinomial distribution of comment types for subreddit B, i.e. we calculate the probabilities of a random comment in subreddit B being positive (upvoted), negative (downvoted) or neutral. To create this multinomial distribution, we only use comments from users who posted more than ten times in subreddit B as these are the users we consider when declaring controversial authors. For a common user i, if i posted n_i times in subreddit B, we sample n_i comments from the probability distribution and calculate if he/she is perceived negatively in the sample. We sample all common users and count the total number of users perceived negatively in subreddit B. We repeat this experiment 30 times to create a sampling distribution of the expected number of negatively perceived users and calculate the z-score of n_{actual} using this sampling distribution. We only retain conflicts from A to B, where this z-score is greater than 3, i.e. the number of users perceived negatively in the attacked subreddit is significantly higher than the number expected from random chance. The final set of subreddits (nodes) and associated edges are the *conflict graph*.

5.5.3 Conflict Graph Properties

The final subreddit conflict graph for 2016 consists of 746 nodes and 11,768 edges. This is a small fraction of active subreddits in 2016 (around 76,000) which is, in part, due to the low amount of 'multi-community posting' on Reddit [81] (i.e. very few authors regularly post to more than one 'home' community). As we require multi-community posts to create an edge, the result is that many subreddits are 'free floating' and are removed from consideration. Of the 746, nine were banned sometime between the end of 2016 and April of 2018: *PublicHealthWatch* (a subreddit dedicated to documenting the 'health hazards' of, among others, LGBTQ groups), *altright*, *Incels* (involuntary celibate), *WhiteRights*, *european*, *uncensorednews*, *europeannationalism*, *DarkNetMarkets* and *SanctionedSuicide*. An additional six became 'private' (requiring moderator approval to join and post), which includes a couple of controversial subreddits: *Mr_Trump* and *ForeverUnwanted*.

The conflict graph consists of 5 components, with the giant component containing 734 nodes. The next largest component consists of only 6 nodes representing different sports streaming subreddits (*nflstreams*, *nbastreams*, *soccerstreams* etc.). Through manual coding of subreddits we identify the following high-level categories: political subreddits (e.g. *politics*, *The_Donald*, *svenskpolitik*) discussion subreddits, video game subreddits (e.g. *Overwatch*, *pokemongo*), sports fan clubs, location-focused

subreddits (e.g. *canada*, *Seattle*, *Michigan*, *Atlanta*), subreddits for marginalized groups (e.g. *atheism*, *DebateReligion*, *TrollXChromosomes*, *lgbt*, *BlackPeopleTwitter*), *porn subreddits (these are image sharing subreddits with their name ending with porn, they are not pornographic in nature — e.g. *MapPorn*, *HistoryPorn*) and NSFW subreddits (e.g. *nsfw*, *NSFW_GIF*). Because of our use of 2016 data and the associated (and contentious) election, political subreddits are heavily represented in the conflict graph. Figure 5.2 shows the ego network for the subreddit *Liberal* in the conflict graph.

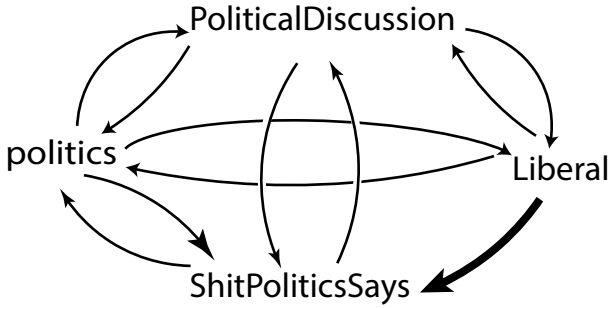

Figure 5.2: Ego network for the subreddit *Liberal*. Thicker edges denote higher conflict intensity.

Edge weights in the conflict graph are often low. On average, only 3.57% (median is 1.70%) of authors in the 'conflict source' subreddit (i.e. their social home) post to the target subreddit (i.e. their anti-social home). There are, however, edges with extremely high weights. The highest edge weight in our data is 85.71% from *The_Donald* to *PanicHistory*. However, in this case, this is due to the disproportionate difference in size of the two (they share only seven common authors). Thus, a high conflict intensity does not necessarily mean that a large fraction of originating subreddit users are antagonistic to the target subreddit. Nonetheless, it does point to the fact that larger subreddits with many controversial authors can overwhelm a smaller subreddit. The high edge-weight here indicates the degree to which this happens. Using the subreddit conflict graph, we can isolate the main source and targets of conflicts and understand where conflicts are one-sided or mutual.

Are conflicts reciprocal? We find that if a conflict edge exists between subreddit A and B, in 77.2% cases the inverse edge will *exist*. Calculating the Spearman correlation between conflict intensities of pairs of reciprocated edges, $\rho(5126) = -0.111, p < 0.0001$, we observe a weak (but significant) negative relationship. Figure 5.3 depicts the outgoing conflict (source) intensity versus incoming conflict (the

conflict target) intensity. This indicates that a targeted subreddit usually reciprocates, but the intensity is usually not proportional.

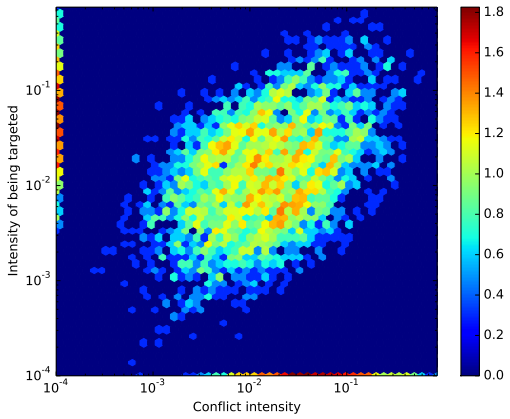

Figure 5.3: Conflict intensity vs intensity of reciprocation in the subreddit conflict graph (log-scale). Un-reciprocated edges appear at the bottom and left edge.

Which subreddits are most targeted in 2016? The indegree of a subreddit roughly indicates the number of other subreddits targeting it. The weighted sum of these edges (weighted indegree) corresponds to the intensity. The top 10 most targeted subreddits by indegree are *politics*, *SubredditDrama*, *AdviceAnimals*, *EnoughTrumpSpam*, *atheism*, *SandersForPresident*, *The_Donald*, *PoliticalDiscussion*, *technology* and *KotakuInAction* respectively. However, when we order subreddits by *total incoming conflict intensity* (see Table 5.1) the list is somewhat different. In both lists, we observe that the most targeted subreddits are social and political discussion forums as well as forums that discuss Reddit itself. The heavy presence of political forums can be attributed to the 2016 US presidential election. We try to deduce if, in general, the most targeted subreddits by degree are also the most targeted subreddits by average incoming intensity (total intensity/number of sources) and vice versa. When contrasting indegree to average intensity for subreddits that are targeted by at least one subreddit (we have 673 such subreddits), we find a weak positive correlation with Spearman $\rho(673) = 0.242, p < 0.0001$. A subreddit targeted by many subreddits is not necessarily targeted with high intensity. Conversely, subreddits targeted by only a few others can nonetheless be targeted with high intensity.

Which are the most conflict 'instigating' subreddits in 2016? By using the conflict graphs outdegree (weighted or not) we can similarly find the largest conflict sources. The top-10 subreddits ranked by outdegree are *politics*, *AdviceAnimals*,

Subreddit	Indegree	Weighted indegree
SubredditDrama	272	19.51
EnoughTrumpSpam	217	13.25
BestOfOutrageCulture	46	10.59
ShitPoliticsSays	48	10.29
Enough_Sanders_Spam	48	9.80
sweden	81	9.62
KotakuInAction	168	9.19
ShitAmericansSay	94	8.83
PoliticalDiscussion	185	8.08
vegan	71	7.26

Table 5.1: Top-10 targeted subreddits ranked by total incoming intensity.

The_Donald, *SandersForPresident*, *WTF*, *technology*, *atheism*, *SubredditDrama*, *EnoughTrumpSpam* and *PoliticalDiscussion*.

When ordered by total intensity, the top-10 list changes to include more news, politics, and controversy focused subreddits (Table 5.2). This list also includes a now banned subreddit (*uncensorednews*). However, we observe that most of these subreddits have low average conflict intensity (i.e. intensity per edge is low). If we order by average intensity (Table 5.3), we find that subreddits targeting very few others (usually 1 or 2 subreddits) show up at top spots. However, we find that the subreddits at the first, third and ninth position of this list (*europeannationalism*, a Nazi subreddit, *PublicHealthWatch*, an anti-LGBT subreddit and *WhiteRights*) are banned by Reddit. A controversial now private subreddit (*ForeverUnwanted*) also appears in this list. This may have implications for identifying problematic subreddits.

As before, we can check if the subreddits most often at the source of a conflict (by outdegree) are also the most instigating (by average conflict intensity). Using 719 'source' subreddits in our conflict graph, we find a weak positive correlation between the number of targeted subreddits and the average outgoing conflict intensity (Spearman $\rho(719) = 0.189, p < 0.0001$), which falls in line with our previous discussion.

Do larger subreddits get involved in more conflicts due to their size? We find that larger subreddits are more likely to get involved in both incoming and outgoing conflicts. Using number of unique authors who posted more than 10 times in 2016 in the subreddit as a measure of subreddit size, we find moderate positive correlation between both size and number of incoming conflicts (Spearman $\rho(673) = 0.403, p < 0.0001$), and size and outgoing conflicts (Spearman $\rho(719) = 0.457, p < 0.0001$). However, taking conflict intensities into account, we find size and average

Subreddit	Outdegree	Weighted outdegree
The_Donald	260	17.75
politics	542	10.15
conspiracy	141	7.39
KotakuInAction	152	7.35
uncensorednews	113	7.14
AdviceAnimals	268	6.76
SandersForPresident	211	6.62
CringeAnarchy	148	6.12
ImGoingToHellForThis	114	5.99
Libertarian	92	5.86

Table 5.2: Top 10 subreddits (conflict source) ranked by total conflict intensity.

Subreddit	Outdegree	Average outdegree
europeannationalism	1	0.75
OffensiveSpeech	1	0.62
PublicHealthWatch	1	0.62
askMRP	1	0.57
ForeverUnwanted	2	0.50
theworldisflat	1	0.43
FULLCOMMUNISM	2	0.42
marriedredpill	1	0.38
WhiteRights	2	0.34
SargonofAkkad	2	0.33

Table 5.3: Top 10 subreddits (conflict source) ranked by average conflict intensity.

incoming conflict intensity is moderately negatively correlated (Spearman $\rho(673) = -0.594, p < 0.0001$). Similarly, size and average outgoing conflict intensity is also weakly negatively correlated (Spearman $\rho(719) = -0.222, p < 0.0001$). This tells us that subreddits with larger size are more likely to be involved in conflicts just because there are more authors commenting in them, but average conflict intensity is not indicative of subreddit size.

5.5.3.1 Node properties

Edge weights alone do not tell us if controversial authors are particularly prevalent in a specific subreddit. Rather, it only indicates the fraction of common users who are sanctioned (norm-violating) in the target subreddits. However, these common users might represent only a small fraction of users of a subreddit. This is especially possible for the larger subreddits. To determine which subreddits are the social home

for *many* controversial authors, we use three additional metrics: *con_author_percent* is the percentage of controversial authors who make their social home in a subreddit relative to the number of authors who posted in that subreddit (more than 10 times in the year); *avg_subs_targeted* and *median_subs_targeted* are the average and median of number of subreddits that these controversial authors 'target.' These numbers can tell us (a) what fraction of a subreddit are engaged in conflict, and (b) are they engaging in broad (across many subreddits) or focused conflicts. We limit our study to subreddits with at least 20 controversial authors who have a social home on that subreddit (overall, we find 698 subreddits meet this criterion). Removing smaller subreddits minimally affects the top-10 subreddits (see Table 5.4) by con_author_percent (only *theworldisflat*, with 11 controversial authors, is removed from the list). We refrain from listing one pornographic subreddit in the table at rank 8.

Subreddit	**Con_author_%**	**Average**	**Median**
PublicHealthWatch	35.25	2.44	2.0
OffensiveSpeech	34.78	2.86	2.0
WhiteRights	32.60	3.19	2.0
ThanksObama	32.59	2.34	1.5
europeannationalism	32.43	2.77	2.0
subredditcancer	27.51	2.33	2.0
subredditoftheday	24.90	1.94	1.0
POLITIC	23.94	1.91	1.0
undelete	23.04	2.13	1.0
SRSsucks	22.18	2.07	1.0

Table 5.4: Top 10 subreddits with highest percentage of positively perceived controversial authors (with at least 20). The average and median columns correspond to *avg_subs_targeted* and *median_subs_targeted* respectively.

Most subreddits in top 10 list are either political forums or somewhat controversial in nature. To lend further credence to this measure, *PublicHealthWatch* (an anti-LGBT subreddit, rank 1), *WhiteRights* (rank 3) and *europeannationalism* (a Nazi subreddit, rank 5) score highly with our metric and were recently banned by Reddit. It is also important to note that most controversial authors have only one anti-social home. Thus in almost all cases, median_subs_targeted is 1. The only exceptions are the first six subreddits in the table 5.4, *The_Farage* (median is 2) and *sjwhate* (median is 2). Note that all banned subreddits shown in this table have a median of 2. The median con_author_percent for all 698 subreddits is 4.09%, and the lowest is 0.36%. It is worth noting that, the three banned subreddits in this list targeted only one or two subreddit each but with very high conflict intensity (e.g., *europeannationalism*

attacked *AgainstHateSubreddits* with conflict intensity of 0.75). All three subreddits also show up in the list of most conflict-source subreddits by average conflict intensity. This shows that a subreddit does not have to target multiple other subreddits to be problematic.

Compared to the top-10 subreddits by con_author_percent, large political subreddits in the most instigating subreddit list (conflict source) had a lower percentage of controversial authors who engage in conflict with other subreddits (e.g., *The_Donald* (8.09%), *SandersForPresident* (6.75%), *politics* (5.71%)). However, in many cases, these values are higher than the median.

5.5.3.2 Banned subreddits

Three out of nine banned subreddits in the conflict graph rank within the top 10 when ranked by con_author_percent and average conflict intensity. Table 5.5 show rank (and value) by con_author_percent and average conflict intensity for all nine banned subreddits (lower ranks means higher con_author_percent and higher average intensity respectively). We observed that moderately low ranks in both measures for three other banned subreddits. Two controversial moderated subreddits *Mr_Trump* (rank 37 by con_author_percent and rank 20 by average intensity) and *ForeverUnwanted* (rank 74 by con_author_percent and rank five by average intensity) also rank low when ranked by both measures. High con_author_percent means that a large fraction of the corresponding subreddit is participating in norm-violating behavior and high average intensity means that a large fraction of common authors between the source and target subreddits are norm-violating. Low ranks by both these measures should indicate that the corresponding subreddit is misbehaving as a community. This is supported by the fact that 6 out of 9 banned subreddits and two controversial subreddits (both set to private by moderators of the respective subreddits) display this behavior. We emphasize again that subreddits can be banned due to their content and not due to the conflict they caused. Such subreddits will not rank low in these two measures.

5.6 Co-Conflict Communities

5.6.1 Creating the Co-conflict Graph

Although most controversial authors have only one anti-social home, there are nonetheless patterns of conflict directed from one subreddit against multiple others.

Subreddit	Con_author_% rank (value)	Average conflict intensity rank (value)
PublicHealthWatch	1 (35.25)	2 (0.62)
europeannationalism	5 (32.43)	1 (0.75)
WhiteRights	3 (32.60)	9 (0.34)
altright	23 (18.18)	19 (0.24)
european	31 (17.41)	24 (0.20)
Incels	183 (7.87)	57 (0.11)
uncensorednews	13 (19.94)	123 (0.06)
DarkNetMarkets	635 (1.59)	494 (0.02)
SanctionedSuicide	475 (2.94)	199 (0.04)

Table 5.5: Banned subreddits and their ranks and values by average conflict intensity and con_author_percent.

Subreddits targeted by same set of authors gives us further insight about these authors and the subreddits they call home. Using all subreddits from the conflict graph, we can create graphs that map the subreddits that are co-targeted. In the co-conflict graph, nodes are still subreddits. Edges are determined by generating a weighted edge between two subreddits A and B if the Jaccard coefficient between the set controversial authors, who have anti-social homes in A and B, is positive. The Jaccard coefficient denotes how many of such authors A and B have in common compared to distinct negatively perceived controversial authors in both subreddits. If X and Y denotes the set of such authors in subreddit A and B respectively, the Jaccard coefficient between X and Y is defined as:

$$Jaccard(X, Y) = \frac{X \cap Y}{X \cup Y}$$

We also make sure that there are at least 2 common negatively perceived controversial authors between subreddits A and B, so that we do not misidentify an edge due to one single author.

5.6.2 Co-conflict Graph Properties

As majority of controversial authors misbehave in only one subreddit, the co-conflict graph has many disconnected components. We only focus on the largest connected component (i.e. the giant component) which consists of 237 nodes and 780 edges. Unlike the subreddit conflict graph, the co-conflict graph is undirected. Furthermore, edge semantics are different as edges denotes the similarity between two subreddits. Common network analysis algorithms can be applied to this graph more intuitively. Use of community detection, for example, can help us determine

which groups of subreddits (rather than pairs) are 'co-targeted.' There are multiple algorithms for community detection in undirected networks [69] (e.g., FastGreedy, InfoMap, Label Propagation, Louvain or Multilevel, Spinglass and Walktrap). The algorithms have different trade-offs [6, 112, 157], though generally both Louvain and Infomap are shown to perform well. Louvain or multilevel algorithm [24, 136] is based on modularity maximization, where modularity is a measure of cohesiveness of a network. An attractive property of Louvain is that it follows a hierarchical approach by first finding small, cohesive communities and then iteratively collapsing them in a hierarchical fashion. This approach on the co-attacked graph produced reasonably sized communities and the results of the community detection algorithm were very stable (i.e. do not change much on different runs). Note that, we use the weighted Louvain algorithm for this purpose.

5.6.3 Community Detection Results

We evaluate the communities using μ-score and clustering coefficient (CC). μ-score is defined as fraction of edges from within the community to outside the community compared to all edges originating from the community. The clustering coefficient of a node is the fraction of connected neighbor pairs compared to all neighbor pairs. For a community, the CC is the average of clustering coefficients of all nodes in the community. In general, low μ-score and high CC denotes a 'good' community. Using the weighted multilevel algorithm on the co-attacked graph we find 15 distinct communities. Table 5.6 shows exemplar subreddits per community, size of the community, μ-score and clustering coefficient for subreddits with at least 10 nodes in them.

Figure 5.4 shows the co-conflict graph and its communities. In general, most communities show low μ-score and low clustering coefficient due to presence of star-like structures (i.e. a large number of nodes are connected to one single node). For example, *politics* is connected to 103 other subreddits. Smaller subreddit communities are topically more cohesive compared to larger communities. For example, community 6 (video game subreddits) and 7 (gun-related subreddits) in table 5.6 are both topically very cohesive.

It is worth re-emphasizing that the co-conflict graph does not necessarily mean that a pair of subreddits in the same community are 'friendly' and do not have a conflict with each other. For example, *Christianity* and *atheism* belong to same community and there are many authors who have a social home in *Christianity* and anti-social home in *atheism*. Similarly, *SandersForPresident* and *Enough_Sanders_Spam* are in the same community and are very much "at war". This is mostly due to pres-

No	Example subreddits	Size	μ-score	cc	description
1	*politics*, *PoliticalDiscussion*, *hillaryclinton*, *SandersForPresident*, *EnoughTrumpSpam*, *Enough_Sanders_Spam*, *AskTrumpSupporters*, *SubredditDrama*	74	0.33	0.41	mostly politics and political discussion subreddits
2	*KotakuInAction*, *conspiracy*, *undelete*, *MensRights*, , *PublicFreakout*, *WikiLeaks*, *worldpolitics*, *Political_Revolution*, *europe*, *The_Donald*	39	0.20	0.29	Political subreddits, controversial subreddits
3	*nsfw*, *NSFW_GIF*, *woahdude*, *cringepics*, *trashy*, *WatchItForThePlot*	34	0.23	0.38	mostly NSFW subreddits, subreddits making fun of others
4	*nba*, *nfl*, *baseball*, *Patriots*, *canada*, *toronto*, *ontario*	16	0.17	0.15	sports subreddits, Canada related subreddits
5	*TopMindsofReddit*, *AgainstHateSubreddits*, *SRSsucks*, *worstof*,*ShitAmericansSay*, *TrollXChromosomes*	15	0.19	0.24	Subreddits focusing on other subreddits
6	*Overwatch*, *DotA2*, *GlobalOffensive*, *NoMansSkyTheGame*, *leagueoflegends*	11	0.06	0.00	video game related subreddits
7	*guns*, *progun*, *Firearms*, *gunpolitics*, *shitgunconcontrollerssay*	10	0.06	0.23	gun-related subreddits
8	*relationships*, *OkCupid*, *AskMen*, *AskWomen*, *niceguys*, *instant_regret*, *sadcringe*, *TheBluePill*	10	0.39	0.51	relationship subreddits, satirical subreddits

Table 5.6: Communities in co-conflict network with at least 10 nodes. For each community, exemplar subreddits, size of the community, μ-score and clustering coefficient(cc) is shown

ence of aforementioned star-like structures. For example, *Republican* and *democrats* both are only connected to *politics* and thus belong in the community containing *politics*. This does not mean that *Republican* and *democrats* have a common group of people perceived negatively.

5.7 Conflict Dynamics

One interesting question for our conflict graphs is how they change over time? It is possible that controversial authors maintain the same social and anti-social homes over time. Conversely, a subreddit with controversial authors may 'shift' its negative behaviors to different subreddits over time. To better understand these dynamics, we study this in both an aggregate manner (i.e. does the most targeted and most instigating subreddits vary each month or do they remain mostly static?), and from the perspective of a few individual subreddits (how does rank of a particular subreddit among most targeted and most instigating subreddits vary over time?). To do so, we created conflict graphs for each month in 2016. These monthly graphs use the same set of subreddits and the same set of controversial authors used in constructing the yearly conflict graph.

We focus this preliminary analysis on subreddits that targeted five or more other subreddits over the year and model how their 'conflict focus' varies. That is, do they

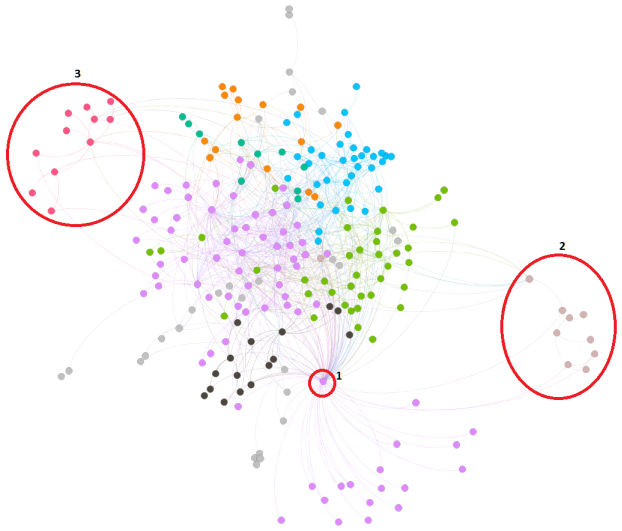

Figure 5.4: The Co-conflict Graph. Different communities are shown in different colors. 1 denotes is *politics*, a subreddit demonstrating star pattern. 2 denotes the gun-related subreddit community and 3 denotes community of video game subreddits.

specifically focus on a single subreddit over all months, or does their most targeted subreddit in a specific month vary from month to month? To determine this, we count the number of times the most targeted subreddit for each conflict source subreddits change from one month to the next. We call this the *change_count* for the attacking subreddit. By definition, change_count can vary from 0 (most targeted subreddit did not change in all 12 months) to 11 (most targeted subreddit changed every month). If a subreddit did not target any other in a particular month, but targeted some subreddit in the next (or vice versa), we count that as a change. Figure 5.5 shows the distribution of change_count for source subreddits.

On average, we find that change_count is 6.91 (median of 7), which means that most subreddits shifts their primary focus over time. We find only 2 subreddits did not change their target at all in 12 months. One example of this is *CCW* (concealed carry weapons subreddit) targeting *GunsAreCool* (a subreddit advocating for gun control in USA).

Because of the 2016 US election, the monthly 'most targeted' and 'most instigating' subreddits are still predominantly political. However, some subreddits only appear in the beginning of the year (e.g. *SandersForPresident* is in the list of top 3 most instigating subreddits for the first four months, *The_Donald* is the top 3 most targeted

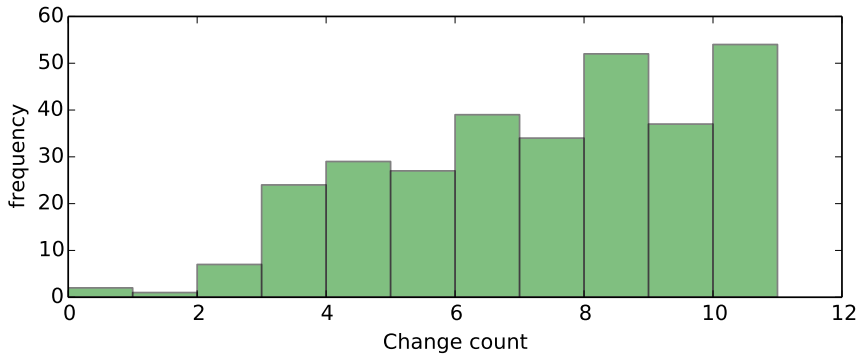

Figure 5.5: Change count for source subreddits who targeted at least 5 subreddits

subreddit list for the first 3 months) or end of the year (e.g. *EnoughTrumpSpam* is in the list of top 10 most targeted subreddits for the last 7 months and during that time, it is the most targeted subreddit). On the other hand, some subreddits show remarkable consistency — *The_Donald* is always the most instigating subreddit (for all 12 months) and *politics*, *SubredditDrama* are always in the top 10 most targeted subreddits list.

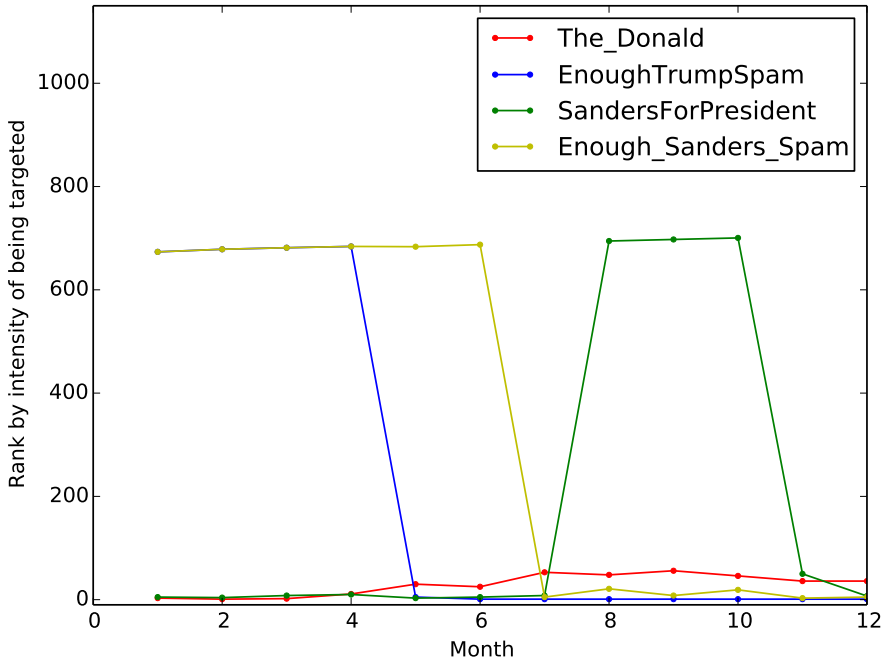

Figure 5.6: Rank by intensity of being targeted for four political subreddits over 2016.

Figures 5.6 and 5.7 illustrate the rank of four political subreddits related to the

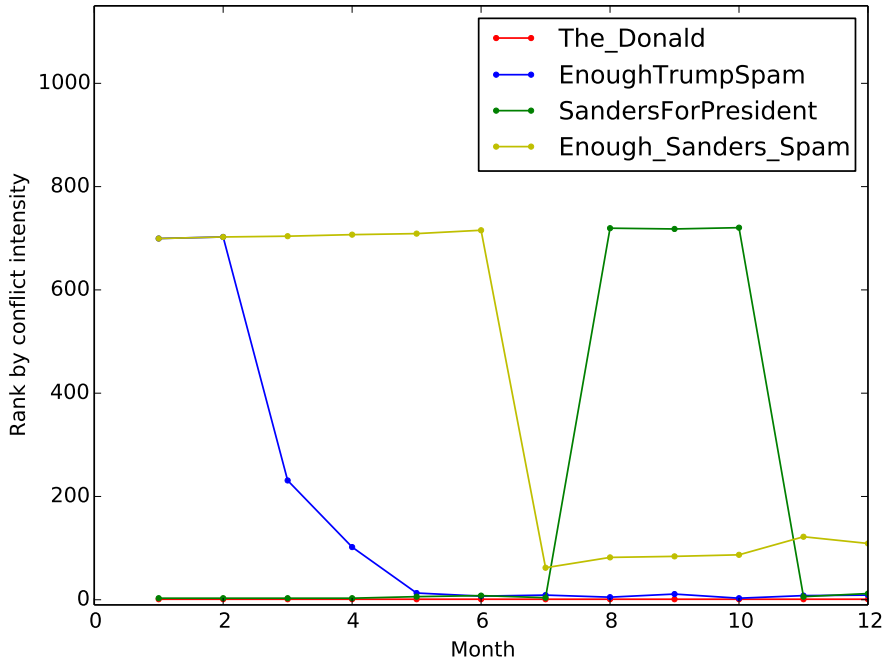

Figure 5.7: Rank by conflict intensity for four political subreddits over 2016.

US presidential election (*The_Donald*, *EnoughTrumpSpam*, *SandersForPresident* and *Enough_Sanders_Spam*). The figures capture the rank of these in the most targeted (largest indegree in the conflict graph) and most instigating lists (largest outdegree) respectively. These demonstrate both the pattern of stable conflict as well as varying ones.

Perhaps the most important observation from these plots is how mirrored they are. *The_Donald* is always the most instigating subreddit and it is consistently targeted back. *EnoughTrumpSpam* gained popularity during March 2016 and gradually became more instigating in the next two months. For the last seven months of 2016, *EnoughTrumpSpam* is the most targeted subreddit. *SandersForPresident* is near the top in both most targeted and most instigating list until the end of July 2016 and from November 2016. However, in three months between July and November, this subreddit did not have any antagonistic relation with any other subreddit as it was shutdown after US political conventions in July and subsequently brought back after in November. *Enough_Sanders_Spam* was formed in July 2016 and instantly became highly targeted due to its content. This shows that, a subreddit instigating/targeted can be highly dependant on external events.

5.8 Discussion

In this chapter, we demonstrate a quantitative method for identifying community-to-community conflicts by aggregating users who behave differently depending on the community they interact with. We define social and anti-social homes of a user based on a local perception of norm-compliance and norm-violation (which we measure by reward and sanction through voting). This method allows us to find conflict in any social network with 'noisy' community structure. Though we focus on Reddit in a specific year (2016), we believe the work is more broadly usable both across time and other social media sites.

Before discussing in which situations our approach may or may not be usable, we briefly summarize our key findings. We find that community-to-community conflicts are usually reciprocal but mutual conflict intensities usually do not match up. We identify which subreddits generated most conflict and which subreddits were most targeted. By analyzing subreddits banned by Reddit in relation to our measures (e.g., average conflict intensity, a high percentage of positively perceived controversial authors, etc.) we illustrate how our technique may be useful for identifying problematic subreddits. Co-conflict subreddit communities show that subreddit conflicts are not random in nature, as we observe topically similar subreddits usually belong to the same co-conflict community. We perform a preliminary analysis of temporal patterns in subreddit conflicts and find that the conflict focus usually shifts over time.

Below we focus on the generalizability and limitations of our findings and approach. Specifically, we discuss the appropriateness and alternatives to using up/downvotes to determine conflicts, contrast of a subreddit conflict with topically opposite subreddits, robustness of the threshold parameters, the potential of communal misbehavior versus behavior of only a few members of a given community and the co-conflict graph.

5.8.1 Downvotes for determining community conflicts

A downvoted comment in a particular subreddit may be a reaction to a number of factors ranging from innocuous norm-violation, being off-topic, presenting a non-conforming viewpoint, low-effort posts (e.g. memes), reposts, and truly malicious behavior. Furthermore, social news aggregation websites like Reddit generally skew towards positive feedback, and are susceptible to social influence effects [144]. Because any individual comment, or even author, may receive up- or down-votes due to these factors, we rely on aggregate signals in our analysis. Thus, a user having many

comments with downvotes (these comments may have more upvotes compared to downvotes) or downvoted comments at all, might point to existence of an anti-social home of the user. However, we opt for a more stringent definition of anti-social home due to two reasons. First, Reddit provides a comment score which is simply the number of upvotes minus the number of downvotes, but does not provide the exact number of up and downvotes as a measure of reducing spam-bot activity [3]. Thus, we can not use the number of up and downvotes of a comment provided by Reddit as a reliable metric. Furthermore, a user in a subreddit might have downvotes (and a few downvoted comments) due to being new in the subreddit (i.e. not knowing all the rules) or brigading, where users from antagonistic subreddits downvote random or targeted comments as a 'downvote brigade'. However, we would like to point out that our definition and threshold parameters of social and anti-social homes are not set in stone and can be easily adapted for similar definitions or threshold parameters, without changing the rest of the algorithmic pipeline to determine the conflict graph.

It is also worth repeating that a downvote does not provide a *global* quality assessment of a comment. Rather, a downvoted comment within a subreddit signifies that this particular subreddit perceives the comment as low quality. This is a localized definition of quality defined by the subreddit and it is consistent with Brunton's model of spam [28]. Globally, these comments might not be seen as norm-violating or low-quality. We acknowledge the fact that users may receive negative feedback not for their own antisocial behavior, but for the antagonistic stance of the receiving community. We do not assume that, for a conflict edge, the instigating community is a 'community of aggressors'. In fact, depending on the viewpoint, it might be viewed as a 'refuge for social outcasts.' New users in a subreddit are more susceptible to innocuous norm-violation due to them not knowing all rules of a new subreddit, but with time they tend to learn. To eliminate these users from the list of controversial users, we enforce a minimum threshold of comments in a subreddit. Excluding these users, we use downvote within a subreddit to determine subreddit conflicts and not as an indicator of the global quality of the comment.

5.8.2 Subreddit conflict due to ideological differences

Many subreddit conflicts in the subreddit conflict graph are between subreddits with topically or ideologically opposing viewpoints. This is expected given the high presence of political subreddits in the graph. However, identifying subreddits with

[3] `https://www.reddit.com/wiki/faq`

ideologically differing viewpoints does not signify a subreddit conflict. Similarly, topical differences do not explain all subreddit conflicts. Two subreddits with opposing ideologies may engage in a civil discussion about the topic, may not engage at all or be part of a subreddit conflict. In many cases, engagement is very low or non-existent. An example of this is *askscience* and *theworldisflat* [53]. Similarly, the conflict edge from *The_Donald* to *PanicHistory* can not be fully explained by topical opposition. On the other hand, the presence of subreddit conflicts between many ideologically antagonistic subreddits works as a sanity check and provide insight into what kind of ideological opponents are more likely to engage in community level conflicts.

5.8.3 Robustness of threshold parameters

We employ multiple thresholds to ensure proper conflict identification. Although some of the thresholding can be eliminated via other methods [132], we use this approach for computational simplicity and effectiveness. Nonetheless, threshold parameters must still be tuned for the particular dataset and application. We discuss our philosophy behind choosing different parameters and justify our choice via a set of small-scale sensitivity analyses. We focus on key differences for different threshold values.

The first threshold is the number of comments by a user. We only consider users who commented more than 100 times to ensure that we perform our analysis on active users. However, we find that our results are quite robust to change in the threshold. Considering users who commented more than 50 times we see around 10% increase in the number of controversial authors. On the other hand, a stricter threshold of 200 reduces the number by 16%. The conflict graphs generated using these thresholds also show little change. We observe 3.1% increase in conflict graph nodes and 1.1% increase in conflict edges using threshold 50, and 4.2% decrease in nodes and 2.7% decrease in edges using a threshold 200. Removing overall low activity accounts from consideration removes malicious sockpuppet accounts (i.e. a single user uses multiple accounts usually unlinked with each other). Unfortunately, we can not directly account for these users as we do not have the data. However, with knowledge of sockpuppets, we can merge multiple accounts before thresholding, which retains the behavior of the sockpuppet account in the aggregate.

The next major threshold we use is determining the minimum number of comments for a user to have a social or anti-social home. We settled for users having more than 10 comments in a subreddit. This threshold works as a trade-off between adding genuine social and anti-social homes for users with lower activity, possibly

in smaller subreddits and falsely identifying social and anti-social homes due to low user activity. Using a lower threshold of more than five comments adds 131% more controversial authors, which in turn adds 124% more nodes and 378% more edges in the conflict graph. On the other hand, using a stricter threshold of more than 20 comments, eliminated 63% controversial authors and 62% nodes and 86% edges in the conflict graph. This stricter threshold also eliminates five of nine banned subreddits from the conflict just because they are not very large in size. It can be argued that different thresholds should be used for social and anti-social homes as one-off malicious comments from many users can overwhelm subreddits. However, due to ease of account creation in Reddit, these one-off comments are often done via 'throwaway accounts' created for the explicit purpose of anti-social behavior and it is difficult to link these sockpuppets to specific communities. Moreover, it is very difficult to identify truly malicious one-off comments from innocuous norm-violations or low-effort posting as we do not have labeled data and deciphering the true intention behind these comments are often context-sensitive (i.e. same comment can be perceived as malicious or non-malicious depending on the context). Our experience is lowering the threshold for determining anti-social homes add many false conflict edges. We choose to err on the side of caution by having a somewhat strict threshold without eliminating most smaller low activity subreddits. We acknowledge that while this approach finds social and anti-social home for long-term misbehaving users, it does not capture sudden conflicts risen from strong external stimuli (e.g., *2015 AMAgeddon*) or when long-standing contributors to a community suddenly starts 'misbehaving' [83].

The thresholds for determining conflict edges (at least five controversial authors behaving differently in a pair of subreddits) and co-conflict graph edges (at least two controversial authors perceived negatively in a pair of subreddits) are somewhat lenient, as our definition of a controversial author is quite strict. We observe, a stricter threshold in both above-mentioned cases, eliminates smaller low-activity subreddits from consideration.

5.8.4 Identifying communal misbehavior

Many conflict edges in the conflict graph have low intensity and most subreddits have low con_author_percent value. In other words, only a few individuals in a subreddits compared to subreddit size are controversial authors. We can infer that getting involved in subreddit conflicts does not imply communal misbehavior. In fact, larger subreddits are more likely to be involved in more conflicts due to their size. However, there is an important distinction in a conflict edge compared to a

few pathological individuals behaving badly. We determine conflicts via controversial authors, which means the users who are perceived negatively in the target subreddit are perceived positively in the source subreddit. This implies either these controversial authors behave very differently in the source compared to the target, or users in the source subreddit support the controversial author's behavior. We do not look for such distinctions in this project. However, this distinction may be useful to isolate in future work. To determine community-wide misbehavior we primarily look into what percentage of active subreddit members are positively perceived controversial authors (con_author_percent). For many banned subreddits (where we infer communal misbehavior because these were banned) we observe that more than 30% of subreddit users fall into this category. Many banned subreddits also show high average outgoing conflict intensity. In general, Reddit users restrict themselves to posting in only a few subreddits [81]. Thus, a conflict edge with high intensity shows that whatever little interaction the participating subreddits have, is toxic. Notably, due to high variance of subreddit sizes, only a few people from a large subreddit can potentially overwhelm a smaller subreddit even if the number of misbehaving users is very low compared to the size of the larger subreddit. We believe that both high con_author_percent and high average outgoing conflict intensity implies communal misbehavior. We would emphasize that this is not the only way a community can misbehave. Abusive language, anti-social or unlawful behavior within the subreddit can also point to communal misbehavior and can lead to subreddit bans.

Moreover, we would like to encourage discussion about communal behavior versus behavior of only a few members in the community in a general sense. It is not always clear what threshold one should abide by when declaring a particular behavior as 'communal' (e.g., what percentage of community members must behave in a certain way to consider that behavior as communal). This discussion applies to Reddit as well as many other online social platforms which exhibit community patterns. We believe that the con_author_percent measure for banned subreddits can be used as a starting point of identifying community-wide misbehavior at least for different subreddits.

5.8.5 Co-conflict graph

The co-conflict graph embodies anti-social home to anti-social home relationships among the same set of subreddits as the conflict graph. As 82% of the controversial authors have only a single anti-social home, the co-conflict graph is sparser compared to the conflict graph. Communities in the co-conflict graph identify which meta-subreddit groups are targeted together. As one might expect, some of the co-

conflict graph communities are extremely topic-coherent (gun-related and video-game subreddits). However, other groupings provide additional insights. For example, a larger subreddit, when targeted together with many comparatively smaller subreddits, forms a star pattern. The meta-subreddit groups are not necessarily targeted by another meta-subreddit groups as we observe a lot of conflicts are generated within the co-conflict subreddit communities. Interestingly, a social home to social home relationship graph form a very dense structure and does not exhibit community behavior, which means that we can not readily classify conflict between different subreddit communities using this method.

5.8.6 Limitations

Using controversial authors to find subreddit conflicts has some limitations. First, this method does not take into account comments deleted by users or moderators (this data is not available for collection). Some subreddits are especially aggressive in deleting downvoted or moderated comments. In some cases, misbehaving authors in a subreddit are banned from further posts. As with comments, we do not have records of this type of moderation. When a subreddit aggressively bans many people, it can change the conflict graph from a static and dynamic perspective.

A clear example of this is *The_Donald*, which banned thousands of individuals over its lifetime (these banned individuals formed a subreddit *BannedFromThe_Donald*, with a subscriber count of 2,209 in November of 2016 and over 27,000 in July of 2018). These individuals do not show up as controversial authors as their comments are gone. We also do not account for sockpuppetry, i.e. having multiple accounts, one for normal posting behavior on Reddit and others for misbehaving. Presence of many users with sockpuppets can skew the estimation of controversial authors in different subreddits.

If data such as bans on the source of sockpuppet accounts can be determined, this data could easily be incorporated in our algorithmic pipeline by updating the definition of anti-social homes. For example, if we know the users who are banned from a particular subreddit, we declare that these users have an anti-social home in the subreddit they are banned from.

In our current analysis, we do not filter bots (software applications that generate comments) from our list of authors. However, strictly malicious bots — those with *only* anti-social homes — do not change our conflict graph as they are not counted among the controversial authors. Occasionally, bots can show up as controversial authors. These include moderator bots (e.g., *AutoModerator*). It is worth a future

study to understand why a bot can be perceived positively or negatively depending on the subreddit. This might mean that bots can intentionally, or not, violate norms for some subreddits while complying with others. Though bots represent a small fraction of Reddit users, this behavior would be interesting for future study.

One final limitation of our model is that *correlated* multi-community posting may appear as a conflict edge. For example, members of community A (a subreddit for a specific computer game) are found to conflict with community B (a feminist subreddit). However, it may not be appropriate to say that A conflicts with B. The topics of the two communities are completely orthogonal. In this situation, it might be due to the presence of a third subreddit, C (e.g., an anti-feminist subreddit) that conflicts with B. It simply happens that many members of A (the game) also have a social-home on C (the anti-feminist subreddit). It would thus be more accurate to say that C conflicts with B. One approach for handling this is to ensure that there is some topical correspondence between the communities we are considering (based on text). This eliminates the $A - B$ edge but retains $C - B$. It is nonetheless possible that we may want to know that the $A - B$ link exists. Moderators of subreddit A might want to be made aware of this correlation and take action.

5.8.7 Implications

Although we perform our analysis on Reddit, our analysis is equally applicable in other social media with inherent or inferred community structure with associated community feedback. For example, we can perform a similar analysis on Facebook pages and groups, online news communities and Twitter hashtag communities (people who tweeted a particular hashtag are part of that hashtag community). We quantify user behavior based on upvotes and downvotes in a particular community, and this data is more easily available for many social media websites compared to a list of banned or otherwise sanctioned users from a particular community. Our approach is highly adaptable and can incorporate new information (e.g., banned and sockpuppet accounts). The analysis is also fully automated and highly parallelizable which increases the adaptability for a very large amount of data.

In addition to providing insight into communities, we also believe that our work can be used for moderation purposes. We observe that several banned subreddits rank very high on particular metrics for measuring conflict. We can calculate these measures for monthly (or otherwise temporal) subreddit conflict graphs and see how different subreddits rank in these measures over time. This observation can be used to monitor problematic subreddit behavior as a whole or create an early-warning

system based on machine learning where we treat currently banned subreddits as positive examples of communal misbehavior and use the metrics above as features.

5.9 Conclusion

In this chapter, we studied community-on-community conflict. We described a mechanism for determining the social and anti-social homes for authors based on commenting behavior. From these, we constructed 'conflict edges' to map the conflicts on Reddit. Using our approach, we allow for a contextual definition of anti-social behavior based on local subreddit behavior. This provides a different perspective than studying global-norm violating behaviors.

We found that most conflicts (77.2%) are reciprocated, but the intensities from both sides did not necessarily match up. Larger subreddits were more likely to be involved in more subreddit conflicts due to their large user-base, but most of these conflicts were minor, and this does not imply large-scale communal misbehavior. On the other hand, we found that high average conflict intensity and a large fraction of subreddit users perceived negatively in other subreddits may have implications for communal misbehavior. Finally, we explored temporal patterns in conflicts and found that subreddits that target multiple others, will shift their main conflict focus over time. We believe that this analysis can be applied to other social media sites which display community structure, create early warning systems for norm-violating communities and help encourage discussion about community-wide misbehavior in social media.

In the next chapter, we specifically look into banned subreddits (a proxy for communal misbehavior) and explore the efficacy of features derived from differences in user behavior to predict communal misbehavior.

CHAPTER VI

Identifying, analyzing and predicting banned subreddits

6.1 Overview

Differences in user behavior can point to communal misbehavior, which is sometimes subjective and hard to identify. We can use community-wide sanctions which translates to banning of entire subreddits from Reddit ecosystem as proxy of communal behavior. In this chapter, we identify such subreddits and use features derived from user behavior differences along with textual features to understand and predict communal misbehavior.

There are a number of reasons why individuals and sub-communities can be sanctioned on social media sites. Anti-social behavior, for example, manifests in both individual and communal fashion with the consequence, on a site such as Reddit, being banning. Many of these behaviors are effectivelly norm-violating and can include trolling, griefing, spamming, brigading, baiting, fisking, crapflooding and shitposting. Communal norm-violating behaviors are particularly interesting as they can be coordinated. In response, a site such as Reddit will ban both individuals and specific Subreddits (sub-communities). However, not all banning is the result of anti-social norm-violation. Other forms of rule-violation, for example posting of illegal content or not complying with moderator policies, may also result in sanctioning.

Over the course of its existence, Reddit has banned multiple subreddits for a myriad of reasons. Where it has been possible to determine, these have ranged from the use of hate speech, violent content, inciting harm to others, providing prohibited goods and services, doxing (release of personal and confidential information), spamming, copyright violation and to involuntary pornography. There are other, more prosaic reasons for banning (e.g., lack of an active moderator). While individual anti-

social behavior, and a site's response, are more fully studied, community banning is often studied on a case-by-case basis and not as a whole. Our goal is to identify and study these communities to understand the landscape of subreddit bans.

Understanding bans, and the process of banning, is increasingly important for social media site design. As social media websites are growing in size, it has become increasingly difficult to monitor and sanction undesirable content. Reddit users posted more than 3.8 billion comments from 2010 to 2017, making human moderation very difficult. As we describe in our study, banning behavior is often temporally and topically clustered. Put another way, banning activities appear to be highly focused. This suggests that moderators may benefit from tools that identify other possible ban targets (what we term: *ban by example*) when the site is focused on this activity. In general, identifying banned subreddits and extracting their content is a very difficult task. We can find a banned subreddit by querying Reddit but in many cases it is impossible to get any content from the subreddit this way as all of its content is already deleted. In this chapter we focus on a large historical archive of Reddit data where at least a portion of content for banned subreddits is maintained. By examining over 1000 banned subreddits over many years, we are able to identify the features (text and network) that are indicators of misbehavior.

A central difficulty in studying bans is the lack of discernible reason for that ban. Of the subreddits we study, 42% do not have an obvious indicator for why the subreddit was removed. Even when we do have some meta-data, it is often unstructured or vague. This is perhaps intentional to prevent banned subreddits from returning. However, this presents a research difficulty for studying banning. By analyzing both textual and interaction features, we are able to group subreddits and identify common ban reasons. We find a fairly unequal distribution to these clusters, with 47% of subreddits in our dataset belonging to one cluster. Using a sample of unbanned subreddits, we demonstrate that it is possible to predict banning (with a 0.841 F_1-score for the central cluster).

A qualitative finding of this chapter is in isolating the main reasons for banning and producing unique features to detect these. We find that one single set of features does not capture all reasons for a ban. We find that bans fall into different categories: *meta*, *internal*, *external*, and and that each presents a different challenge for prediction. *Meta* bans are those that have to do with violating the organizational rules of Reddit (e.g., lack of active moderation). *Internal* bans are those where there is very little social interaction between the banned subreddit and other parts of Reddit. Obvious banning due to violent or toxic content or involuntary pornography

fall into this category. However, more subtle content includes the sale of prohibited goods or services, where copyright infringement is rampant, or where user agreements are violated. These are not necessarily anti-social behaviors and may require certain domain expertise (e.g., legal expertise) to identify. *External* bans include more classically anti-social behaviors. These subreddits display norm-violating behaviors against others on the site and are often reflected in the posting behaviors of across Reddit. We find that all three ban types — meta, internal, and external — require different features for detection. For example, internal bans are often detectable through text analysis but external bans can be found through interaction features [51] (and often, a combination is what works best).

We find that subreddit bans follow both temporal and 'reason' patterns. Subreddits are banned in a sporadic fashion over time and subreddits with similar types of misbehavior are banned in batches. We use this information to implement a banning-by-example prediction scheme that seeks to model how a banning 'workflow' might proceed. By selecting one or more 'seed' subreddits, we demonstrate that we can rank other subreddits as possible ban targets (with a $P@10$ of .913 for a balanced dataset).

Our contribution in this chapter are two-fold. First, we identify and categorize a large set of banned subreddits. We believe this is the largest dataset of its kind. We analyze the major characteristics of these subreddits via data analysis and clustering. Second, we show that banned subreddits can be predicted and introduce a banning-by-example task. We believe that such an approach may be useful to moderators who are banning groups of related subcommunities. A paper based on this chapter is under review [52].

6.2 Data Collection

We focus our analysis of community-wide misbehavior on Reddit (`www.reddit.com`), a social aggregator and new forum website with inherent community structure. Reddit has banned several of its sub-communities, or subreddits, over a number of years. To establish a ground-truth dataset we aim to identify these banned subreddits and the reason for their banning if provided by Reddit. In our analysis, we utilize the Reddit dataset compiled by Baugartner [17] of all publicly available subreddits. This dataset is a multi-year dataset consisting of publicly available posts, comments, authors and other miscellaneous subreddit metadata. We specifically focus on comments instead of posts as a large number of Reddit posts are images, videos or links

Reason of banning	No	Top 5 largest subreddits
no reason provided	442	*fakeid*, *Steroidsourcetalk*, *DNMUK*, *TheXanaxCartel*, *opiaterollcall*,
spam	190	*BabyFart*, *streamsoccer*, *HealthProject*, *milf_nowandforever*, *nsfw_showerthoughts*
unmoderated	124	*european*, *CandidFashionPolice*, *FreeKarmas*, *oastme*, *Nude_Selfie*
content policy violation	108	*uncensorednews*, *CoonTown*, *SanctionedSuicide*, *Dream_Market*, *AlphaBayMarket*
prohibited goods or services	74	*juiceswap*, *JuulMarket*, *Kratom_Vendors*, *sugardaddydatingsites*, *ResearchChemBarter*
violent content	47	*Incels*, *WhiteRights*, *selfharmpics*, *Womad*, *zoophilia*
involuntary pornography	36	*xray*, *doppelbangher*, *CelebFakes*, *SluttyStrangers*, *CelebCumSluts*
inciting harm	7	*GasTheKikes*, *RapingWomen*, *PhilosophyOfRape*, *beatingwomen2*, *beatingtrannies*
multiple violations	7	*pedofriends*, *leftwithsharpedge*, *nsfwshoops*, *IncelHeaven*, *candid*,
personal and confidential information	6	*altright*, *TapSportsBaseball*, *picsofcaninedicks*, *cheatingrevenge*, *exgf*
copyright violation	5	*CrackedSoftware*, *SocialMediaMarketing*, *PatreonBabes*, *BusinessAdviceTeam*, *NikiSkyler*
to keep everyone safe	5	*fatpeoplehate*, *NeoFAG*, *TalesofFatHate*, *HamPlanetHatred*, *Trans_fags*
harassment	4	*IDontLikeRPolitics*, *FuckBoyRiotSquad*, *lukecis*, *ClubSorel*
user agreement violation	3	*illegaltorrents*, *Pickpocket*, *shopliftingadvice*
lack of active moderators	2	*BrockTurnerInnocent*, *NewsReviewNow*

Table 6.1: Common reasons for banning a subreddit, number of subreddits banned for the reason and the top 5 largest banned subreddits in that category.

to different articles or media across the internet. We restrict ourselves to comments from 2010 to 2017, which includes 3.8 billion comments from 542.6k subreddits. We only consider subreddits with at least 100 comments, which restricts us to 73.8k subreddits. We identify vanished subreddits by querying the Praw Reddit API. These subreddits include banned subreddits and subreddits that became private or otherwise restricted. For each vanished subreddit, we crawl their homepages and checked if the *subredditBanned* flag is set to *True*, which indicates that the subreddit was banned by Reddit. We also extracted the reason of banning as provided by Reddit from the *subredditBanMessage* field. For the data between 2010 and 2017 we found 1060 banned subreddits with at least 100 comments (these were banned on or before June 2018, the date of our crawl).

We emphasize that this dataset does not contain all subreddits banned by Reddit before June 2018 (e.g.- *jailbait*, *TheFappening*, *pizzagate*, *Deepfakes*). As we base our analysis on data collected by Baumgartner, any subreddits entirely removed by Reddit from public access *before* this crawl are not present in this data. For the purpose of our analysis, we did not use these subreddits in our study. It is possible that this introduces a bias. Nonetheless, we are confident that we capture those banned subreddits that were available in the crawl.

By enforcing a threshold of 100 comments we ensure that we have enough data form which to extract features. However, the reason for a ban may not be obvious from the subreddit content alone. A prime example of this are, 'copycat' subreddits which emerge in response of banning a large subreddit (e.g. *fatpeoplehate2* and *fatpeoplehate3* were created in response to banning *fatpeoplehate* and were promptly banned soon afterwards). These 'copycat' subreddits do not have enough content and banned due to violating Reddit policy of 'ban evasion' which means creating a

community with the same purpose as a banned community. This is very difficult to glean from subreddit content and interactions alone. We do not exclude moderator-removed comments when considering this threshold as these subreddits can still have interaction features.

6.3 Properties of the Banned

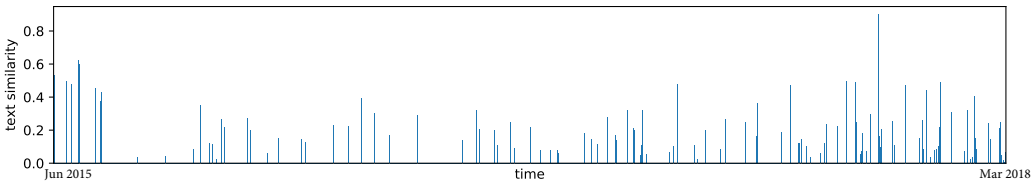

Figure 6.1: Average text similarity with banned subreddits banned within one day window of self-ban binned daily from Jun 2015 to Mar 2018.

Subreddits can be banned for a wide array of reasons including both norm- and rule-violating. Examples range from toxic and violent content, misbehavior in other subreddits, buying/trading of prohibited goods and services, advocating illegal or otherwise reckless behavior, harassment, involuntary pornography, being unmoderated, spam, and so on. In many cases the reason is not provided directly by Reddit or the reason is vague. Table 6.1 summarizes the reasons provided by Reddit and example subreddits for that category. In a large number of cases (442), the ban reason is either blank or generic (e.g., 'this subreddit has been banned' or 'due to violating Reddit rules'). In 223 cases, Reddit cites content policy violations. However, in only 115 of those is there a specific note about which policy (e.g., violent content, prohibited goods or services, involuntary pornography etc.). In Table 6.1 only those policy violations that are in the general form are recorded in that row (specific reasons are recorded elsewhere).

By manually reviewing the content of a subreddit, it is sometimes possible to identify the reason for a ban. However, even human annotators may struggle with this as subreddits can obfuscate policy-violating behaviors. The lack of specific and structured annotations presents a challenge for analysis. Furthermore, because the reason for banning may involve 'management' issues (meta), internal-facing commenting behavior (internal), or intra-subreddit behaviors (external), it is necessary to develop a broad set of features — network, text, temporal, and metadata. We explore vari-

ous properties for banned subreddits starting with the data that is most prevalent: comments.

6.3.1 Banned subreddit comment properties

Banned subreddits vary drastically in size. The top 5 largest subreddits by comment count in our dataset are *fatpeoplehate*, *fakeid*, *Incels*, *uncensorednews* and *european*. The largest banned subreddit *fatpeoplehate* has 1.59 million comments, while the 10th largest *altright* has 166.4k comments. Only 215 subreddits have at least 1000 comments and the median number of banned subreddit comments in our dataset is 332. Figure 6.2 shows the distribution of banned subreddit comment counts where comments are logarithmically binned into 100 bins. From the figure, we observe that the comment distribution is long-tailed.

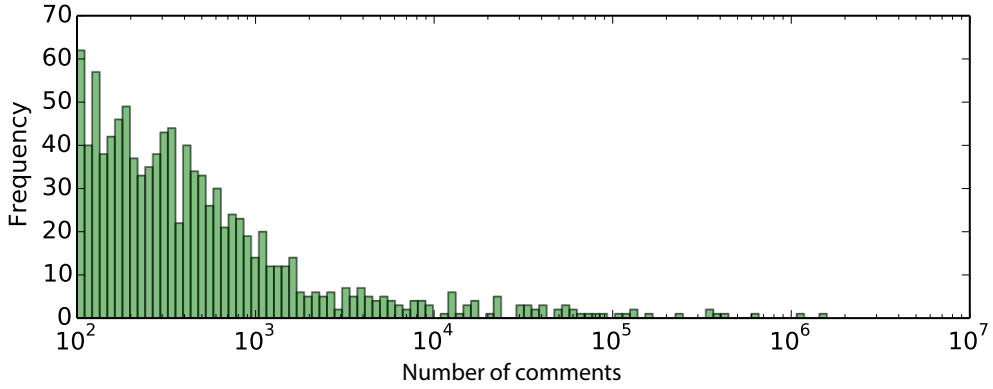

Figure 6.2: Banned subreddit comment count histogram with 100 bins. The x-axis is log-scaled.

A significant portion of comments in these banned subreddits are either deleted or removed (by a moderator). On average, we find that 18.87% of all banned subreddit comments are unavailable (median 11.82%). Thirteen subreddits have all of their comments deleted. On the other extreme, three subreddits had none of their comments deleted.

To compare these values to unbanned subreddits, we sampled the same number of subreddits from the entire subreddit population. To produce a matched sample, we used the comment distribution of the banned subreddits to select similarly sized subreddits (by comment count). That is, for each banned subreddit we find an unbanned one with the same number of comments (ties broken randomly). If there is

no subreddit with the same exact comment count as a banned subreddit, we choose the next largest subreddit. We use this unbanned sample to contrast the properties of banned subreddits throughout this paper. For unbanned subreddits, we find that on average only 9.13% comments are deleted (median 6%). There are no subreddits in our sample which have all of their comments deleted (though one particular subreddit had nearly 97% of comments removed). We have eight subreddits where none of the comments are deleted. Figure 6.3 shows deleted comment proportions (in percentages) of banned and unbanned subreddits in our dataset.

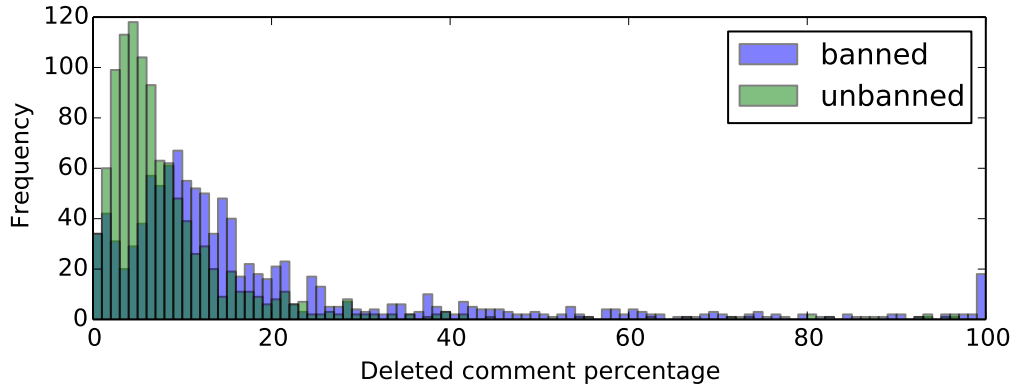

Figure 6.3: Banned and unbanned subreddit deleted comment percentages.

Because comment voting can indicate reward and sanctioning of behavior (norm-compliant and norm-deviating, respectively) we also looked at the percentage of downvoted comments in both groups of subreddits. Because comment metadata (including up and downvotes) are stored separately than comment content, we are able to perform this calculation on all comments, even those that have been deleted. If a comment has more downvotes compared to upvotes, we say that the comment is downvoted. Reddit automatically upvotes a user's own comment, but we assume that a user always views his/her comments positively. For this reason, we only mark a comment with a negative score as downvoted. We find that on average both banned and unbanned subreddits have very low percentages of downvoted comments (1.35% for banned subreddits, 1.05% for unbanned subreddits). A large number of subreddits in both groups do not have any downvoted comments (502 and 473 for the banned an unbanned sample, respectively). The highest percentage of downvoted comments are also pretty similar (35.6% for banned subreddits, 32.23% for unbanned ones). This is somewhat surprising as we expected banned subreddits to have more downvoted comments compared to the unbanned sample.

6.3.2 Banned subreddit language

We were curious if banning was more likely to target English language subreddits or a broader set. Because we would like to utilize text analysis, understanding the distribution of languages was critical. In theory, Reddit provides the language of a subreddit as metadata. However, for a large number of banned subreddits the metadata for language is set to 'none.' To perform automated language identification we used the langid package [128] on comment text. While this largely worked, a few subreddits were harder to labels due to the scarcity of data (a few subreddits have all or almost all of their comments deleted) or the nature of the comments in the subreddit (e.g. URL only, area code only etc.). In general, we find 32 non-english subreddits including seven Spanish and six German subreddits. For 34 subreddits we were unable to determine the language. These 34 subreddits include the 13 subreddits which have all of their comments deleted or removed. The largest non-english ban was *Womad*, a subreddit named after a Korean radical feminist comment board. This subreddit was banned for 'proliferation of violent content'.

6.3.3 Subreddit ban times

Reddit does not provide a specific ban time. However, we can estimate it by the time of the last comment in that particular subreddit. As we use comments from 2010 to 2017, the subreddits that are banned in 2018 all have their last comment on 31st December 2017. To remedy this, we add first three months of 2018 Reddit comment data to estimate ban times of these subreddits. Controversial subreddits caught media attention as early as 2011 due to the subreddit *jailbait* (which was subsequently banned). The first subreddit ban in our dataset is in 2013, when *FIFA_CL* was banned for 'violating Reddit rules'. While three other subreddits were banned in 2013, all others in our dataset were banned in 2014 or later. Most interestingly, was that subreddits were not removed uniformly but rather sporadically over the years. Figure 6.4 shows counts per week for banned subreddits.

We observe similar (sporadic) patterns for daily banning. Figure 6.5 shows the frequency of subreddit bans per day from January 2010 to March 2018. Reddit banned a maximum of 56 subreddits in a single day on February 28, 2018. The top 5 largest subreddits banned on this day are *murdochmurdoch*, *snapfuck*, *CrackedSoftware*, *SocialMediaMarketing* and *AmazingTeens*. Four of these subreddits (except *CrackedSoftware*) were banned for either spam or being unmoderated. *CrackedSoftware* was banned due to copyright infringement.

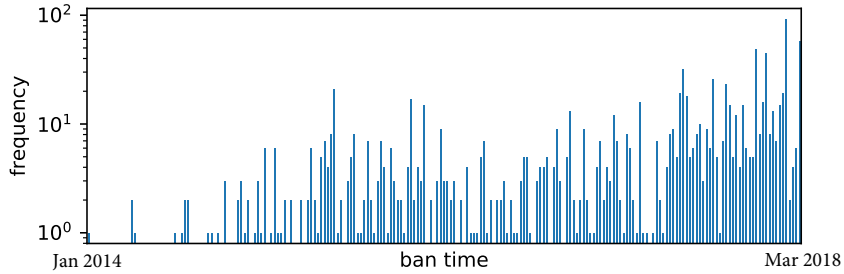

Figure 6.4: Subreddit ban time binned weekly from Jan 2014 to Mar 2018.

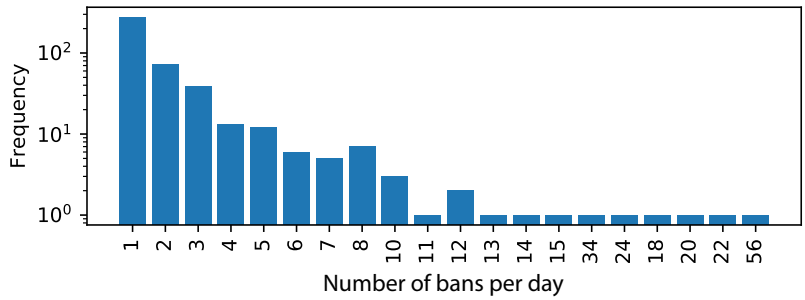

Figure 6.5: A bar chart showing the frequency of subreddit bans per day from January 2010 to March 2018.

Given this distribution, it is possible that Reddit banned subreddits in a reactive fashion. We can hypothesize that when a set of controversial subreddits were reported or caught media attention, Reddit reviewed these subreddits, banned them and subreddits similar to them. It might also be the case that Reddit follows some internal ban schedule or a combination of both. This implies that at least some subreddits which were banned within the span of a day would be very similar to each other. We verify this hypothesis by calculating textual and interaction similarity among all subreddits which are banned within one day from each other. We detail these similarity measures and features in the next section. We find that subreddits banned within a day from each other are indeed similar in terms of both textual and interaction similarity. Figure 6.1 shows average text similarity with banned subreddits banned within one day window of self-ban binned daily. As we have a very large number of bins, we only show this plot from June 2015 to March 2018, where most of the subreddits in our dataset were banned.

6.3.4 Banned subreddit active time

Just as we can observe the last comment time, we can also extract the first comment time and find the lifespan of banned subreddits. On average, a banned subreddit is active for a little over 2 years and 1 month (mean 761.7 days, median 589.5 days). The banned subreddit active for the longest time is *HealthProject* which was active for 2979 days. This subreddit was banned for spam on February 27, 2018. It is possible, even likely, that changing behavior over time may lead to banning. Seven subreddits were banned on the day they were created. On the other hand, 320 subreddits persisted for at least 1000 days. Figure 6.6 depicts a histogram showing active time of banned subreddits in our dataset binned weekly.

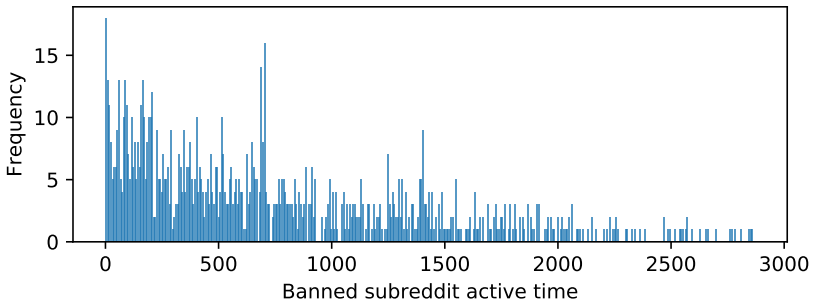

Figure 6.6: A histogram showing active time of banned subreddits in our dataset binned weekly.

6.4 Clustering Subreddits

In order to get a better sense of the types of bans, we utilize automated clustering. To support this, we make use of both textual and 'interaction' (i.e. network) features. Both can provide us with complementary signals [53].

6.4.1 Textual features

To better understand the content of banned subreddits we use a simple bag-of-words model. Specifically we weight terms using TF-IDF that treats all comments in a subreddit as part of a single 'document.' To account for the size differences in subreddits, we normalize term frequencies by the maximum term frequency for that particular subreddit.

In our analysis of text, we only take upvoted subreddit comments into consideration when extracting textual features. We believe the downvoted or neutral content

(with zero score) do not 'represent' the community and they are downvoted precisely due to this reason. Reddit automatically upvotes a users' own comment. We consider this vote as positive as the user as a part of the community views their own content positively. We can similarly use downvoted content to identify what content a community dislikes as a part of their identity. However, we have very little downvoted content compared to upvoted content. A lot of banned and unbanned subreddits do not have any downvoted content and a large number of banned subreddits have most if not all of their downvoted content deleted. For this reason, we do not use the downvoted content for textual feature generation.

6.4.2 Interaction features

Although Reddit does not have an explicit network structure, past work has revealed ways of inferring such structure [53] and specifically identifying community-on-community conflicts [51]. This work observed that the types of authors (norm-compliant or norm-violating in relation to a particular subreddit and others) were correlated with banning.

Briefly, the technique distinguishes between a user's norm-compliant and norm-violating behavior via community up/downvoting within the subreddit (complete details are presented in [51]). This follows a definition similar to Brunton's model of spam [28], where the community decides what is spam based on its own rules and regulations. A subreddit where a user has a significant number of comments (determined by a threshold) and is generally norm-compliant, is called a 'social home' of the user. Conversely, a subreddit where is a user has significantly commented but is generally norm-violating, is a 'antisocial home' of the user. A user with at least one social home and at least one antisocial home is called a 'controversial author'. By aggregating behaviors of these controversial authors we can identify community-to-community conflicts in Reddit. These conflicts are directional and weighted. If k authors have a social-home in subreddit A, and an anti-social home in subreddit B, we can denote a directed conflict from A to B. The raw author count is normalized by common authors in both subreddits to account for size difference in subreddits and we refer to this normalized weight as 'conflict intensity' from A to B. Note that, for each subreddit in a subreddit pair, there can be an outgoing conflict intensity and an incoming conflict intensity. For each subreddit, we can also find what percentage of subreddit users are positively perceived controversial authors. This measure is called 'con_author_percent' and it represents what fraction of a subreddit's user base is perceived negatively elsewhere in Reddit.

For some banned subreddits which were still active in 2016, both average outgoing conflict intensity (if a subreddit is in conflict with multiple other subreddits, this is represented by the average of outgoing conflict intensities) and con_author_percent measure rank very high. We deduce our interaction features for banned subreddits based on this observation.

6.4.2.1 Extracting interaction features

While finding controversial authors, we need to careful not to misidentify new users or users with very few comments as controversial authors. These users can have downvoted comments for violating a subreddit-specific rule they are not familiar with or due to brigading (i.e. random or targeted downvoting by people who are not part of the community). To avoid this type of misidentification, we need to set a minimum threshold of comments within a subreddit for a user. However, setting this threshold too high would miss genuine controversial authors. Based on the minimum size of banned subreddits in our dataset (100 comments), we set this threshold to be 5 comments.

Our extracted interaction features can be divided into two sets, one is based on the outgoing conflict intensity measure and the other is based on the con_author_percent measure. The first set of features identifies all outgoing conflicts a banned subreddit participated in. We use the number of outgoing conflicts the subreddit was involved in, average outgoing conflict intensity and maximum outgoing conflict intensity as features. For the second set of features, we use the con_author_percent measure and the average and the median number of subreddits where these authors were perceived negatively. For conflict intensity features, we made sure that we have at least two controversial authors per conflict edge. For controversial author features, we only considered subreddits which have more than one controversial authors.

one advantage of these features is that they are content-agnostic and can be extracted from any language subreddit without resorting to language-specific measures. However, not all banned (and unbanned) subreddits have interaction features. Among 1060 banned and 1060 unbanned subreddits, only a small set, 147 banned ones and 118 unbanned, have conflict intensity features. For controversial author features, 400 banned and 445 unbanned subreddits have these features.

No	Size	Top 5 largest subreddits	Reason for banning
1	503	*fatpeoplehate*, *fakeid*, *Incels*, *uncensorednews*, *european*	various reasons
2	72	*carinsurance*, *saggyballs*, *Raghunomics*, *starcitizens*, *Simsononline*	mostly spam and unmoderated
3	23	*pharmacynews*, *ViagraReviews*, *Pills*, *genericbrands*, *EDforum*	prohibited goods, drugs
4	7	*collegefootballs*, *nfl_stream*, *LiveStreamAllSports*, *nflstreamtoday*, *LiveNFLSuperBowl*	illegal streaming
5	7	*Clash_Of_Clans_Hack*, *nef_v_wex*, *LGH*, *poijthgfd*, *newcheatsonline*	hacks and cheats
6	7	*Escorts_Service*, *realEstateWebsites*, *RealEstateIndia*, *backpage_escorts*, *HotGirlsHot*	prohibited goods and services

Table 6.2: Banned subreddit clusters, their sizes, the top 5 largest banned subreddits in the cluster and common reasons for banning.

6.4.3 Measuring Similarity

Using the textual and interaction features, we can compute pairwise similarity for all banned subreddit pairs using cosine similarity. However, the textual features are fundamentally very different from the interaction features. For this reason, we compute two sets of pairwise similarities, one for text and the other for interaction. Interaction features are also divided into two subgroups. We calculate cosine similarity for each subgroup after normalization and take the average as the final interaction similarity for each pair.

To calculate similarities incorporating both textual and interaction features, we calculate the similarity between a subreddit pair A and B with both features the following way:

$$sim(A, B) = \alpha * text\_sim(A, B) + (1 - \alpha) * interaction\_sim(A, B)$$

where α ($0 <= \alpha <= 1$) is a parameter that determines the weight of text similarity. We choose the value of this parameter during clustering. For subreddit pairs where we do not have interaction features for one or both, we use only the text similarity (i.e. if $interaction\_sim(A, B) = 0$, then $\alpha = 1$.)

6.4.4 Generating Clusters

There are many applicable unsupervised clustering algorithms that we might employ for this analysis. We opt for DBSCAN [64], a density-based spatial clustering algorithm. We choose this algorithm specifically for the reason that it is robust to noise and it does not include every data point into some cluster if that degrades the quality of a cluster. DBSCAN is a density-based clustering method and hence takes the maximum distance between two data points in the same cluster as a parameter (ϵ). As we calculated pairwise similarities in the previous section, we need to convert it to pairwise distances to apply this algorithm.

We used cosine similarity for both text and interaction similarity, so both range

between 0 and 1. For subreddit pairs with interaction similarity, we use a linear combination of these two similarities, which also lies between 0 and 1. For subreddit pairs with only text similarity, the value already lies between 0 and 1. We use $(1-sim)$ as our distance measure where sim represents pairwise similarity.

We use the silhouette coefficient [164] to determine ϵ (the distance parameter in DBSCAN) and α (the weight parameter for text similarity). Silhouette coefficient is a measure based on the mean intra-cluster distance and the mean nearest-cluster distance for each point in the cluster. To choose ϵ, we vary α in ten steps between 0.1 to 1, and observe the highest silhouette coefficient. We choose α based on this coefficient, the number of subreddits not in any cluster (noise) and the number of clusters produced.

6.4.5 Results

We find that $\alpha = 0.5$ and $\epsilon = 0.66$ produces the best results for clustering. The largest cluster contains 503 subreddits, nearly half of the banned subreddits. This is not necessarily surprising as we did not expect, from our qualtititve observations, a uniform distribution of ban reasons. There are four other medium or small clusters. Surprisingly, however , there are 441 subreddits without a cluster (i.e. they are perceived as noise by DBSCAN). Table 6.2 show the banned subreddit clusters, their sizes, the top 5 largest banned subreddits in the cluster and the common reasons for banning. We find that while the largest cluster aggregates a few different types of banned subreddits (based on the labels we obtained from Reddit), the smaller clusters are usually very focused. The variation for the large cluster is somewhat unsurprising given the large number of Subreddits and the lack of any specific structure (e.g., hierarchical) in the reasons for banning.

When simultaneously clustering both banned and unbanned subreddits (the 2120, with same α and ϵ values), we notice a very similar distribution. The largest cluster roughly doubles in size to 926 subreddits, while retaining almost all of the original 503 but also many unbanned subreddits. For all other banned subreddit clusters, we see almost one-to-one matches (i.e. same clusters are present in both sets of clusters) with the exception cluster 6 in Table 6.2. This implies that these clusters are generally high quality and may be usable as classifiers to find related subreddits through the use of similarity values.

Subreddit	Comment count	Characteristics
Womad	44009	Korean radical feminist subreddit
FreeKarmas	15602	circumventing Reddit rules by giving free karma, very short comments
glassine	11821	location-based heroin review, comments are usually area codes
BabyFart	7196	bot controlled sub, only allowed comment is "BABY FART"
Test0324	5737	Italian subreddit
GOTporn	3840	pornographic video, image posting website with minimal very short comments
cheggrequests	3827	homework help on questions of Chegg.com
Ironsteel	3067	NBA and soccer based subreddit where all comments are urls
CorrieMckeagueNew	3066	a missing person hunt subreddit
fulltvshowsonanything	2933	illegal streaming website with minimal comments

Table 6.3: Top 10 largest subreddits in noise, their comment count after removing deleted comments and their characteristics

6.4.6 The 'Noise'

Of the 1060 banned subreddits in our analysis, 441 were identified as noise by DBSCAN. A subreddit which is identified as noise in this analysis means that it does not have high similarity with any of the clusters or other subreddits which are also identified as noise.

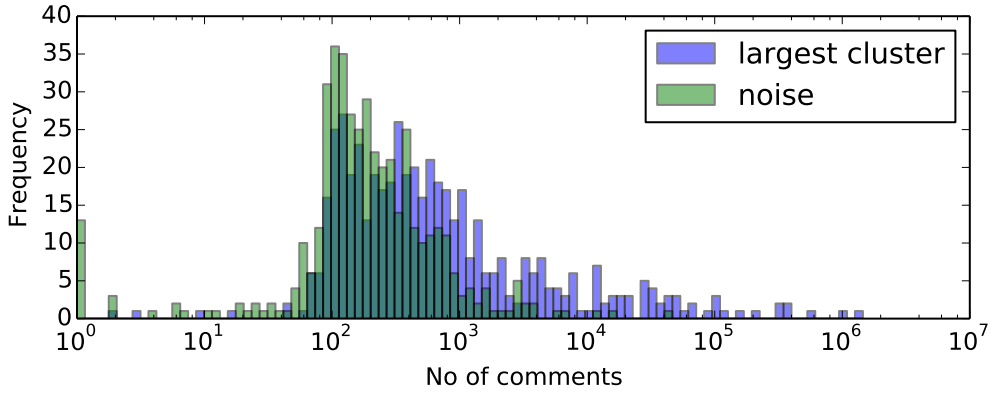

Figure 6.7: Largest cluster and noise subreddit comment count histograms with 100 bins. The x-axis is log-scaled.

To better understand these we further analyze this subset. Qualitatively, we find that some subreddits in this cluster have most of their comments deleted or removed. We calculated comment counts for all subreddits without the deleted and removed comments. This reveals that some subreddits are identified as noise due to scarcity of data. There are 13 subreddits where all of the comments are deleted and 100 subreddits in noise have less than 100 comments. In general, subreddits in noise have less data compared to subreddits in the largest cluster. Figure 6.7 shows the comment counts in noise against comment counts in the largest cluster. While

looking at medians, the largest cluster subreddits have 427 comments as median, where subreddits in noise have 179 comments as median. Out of 182 subreddits with more than 1000 comment after accounting for comment removal, only 28 belong in noise compared to 152 in the largest cluster. However this means there are still large banned subreddits with enough content which are classified as noise.

We manually check these subreddits to observe why they have very low similarity with other subreddits. We find that some of the largest subreddits in this list are in a language other than English, some have very specific esoteric rules for comments or very short minimal comments. Table 6.3 shows the top 10 largest subreddits in noise and their characteristics. In general, we find a large number of subreddits are picture or video based subreddits that post prohibited or violent content or posts them illegally. These subreddits do not interact with other subreddits much and have very short comments (e.g., thank you notes or exclamatory comments). There are also a large number of unmoderated and spam subreddits (75 spam and 44 unmoderated) which repeat the few spam messages a very large number of time. These subreddits do not have high text similarity with other subreddits for obvious reasons. However, we do have some esoteric subreddits like *glassine* and *Ironsteel* where the true intent of the subreddits are masked using clever commenting.

6.5 Prediction

Our prior analysis shows that some subreddits are simply too spare given our dataset (either due to comment deletion or the use of non-text content). Because the largest cluster is diverse and relatively rich in content, we focus on subreddits in this group for our prediction tasks. While we do not consider these in our prediction task, we believe that if we were able to 'catch' comments before they are deleted, our classifiers would work. Similarly, with additional data it may be possible to train different classifiers for banned subreddits in each different clusters as they may represent more range in the reason for banning.

We try to predict banned subreddits against unbanned ones in two ways. The first attempts to simply discriminate between banned and unbanned subreddits. As the textual and interaction features have high variance in our dataset (in that not all subreddits have interaction feature), we train two separate classifiers and combine their results.

Our second prediction experiment models a more realistic use scenario for moderators. Because moderators often concentrate their banning effort in time — and

on related content — we demonstrate the use of a banning-by-example ranking system. Upon being provided example subreddits to remove, our classifier can accurately identify other likely targets.

6.5.1 Classifying Using Text

We use a stochastic gradient descent classifier (SGD classifier) [193] to classify banned subreddits using textual features. An SGD classifier allows us to test a large variety of loss functions including the hinge, log, modified_huber, squared_hinge and perceptron losses. We trained an SGD classifier with all of these losses along with different types of regularization and regularization parameters (α). We trained our classifier with 10-fold cross-validation and found that our classifier performs best with 0.832 F_1-score with log-loss, elasticnet regularization (this regularization is a combination both l1 and l2 regularization) and regularization parameter $\alpha = 0.0001$. We use F_1-score as our evaluation metric as our dataset is unbalanced with 505 positive and 1060 negative samples.

6.5.2 Classifying Using Interaction

We use a random forest classifier with 20 trees to classify banned and unbanned subreddits with interaction features. As a large number subreddits do not have these features, we only train and test on samples with these features. We used ten-fold cross-validation like our previous classifier to train the model. We found that the best cross-validation F_1-score is 0.605, which is rather low by itself. However, as we only use six interaction features to obtain this result. This type of classification can be applied to datasets in any language with interaction features.

6.5.3 Combining Classifiers

We combine results from both our text and interaction classifiers to predict banned subreddits. First, we divide the data into a training set and a testing set with the testing set comprising of 10% of data divided in a stratified fashion. We train both our classifiers with previously found parameters. We do not combine our results using a simple majority voting of two classifiers for two reasons. First, we do not have interaction features for all samples. Second, the text classifier has higher performance accuracy (though we reliably can generate text features). Instead, we use the class prediction probabilities of both classifiers and take only the confident predictions.

As we observed better result using the text classifier and we have textual features for all samples, we use the text classifier to predict a sample's class if the classifier is at least 65% sure of its decision. On the other hand, if we have interaction features for that sample and the interaction classifier is very sure of its prediction (at least 80%), we use the interaction feature to predict the sample's class. We observe that the overall score varies depending on the train-test split. We ran the classification task 50 times with different splits and took the average score to account for these variations. We get 0.841 average F_1-score using both textual and interaction features and 0.839 average F_1-score using only textual features. Although this improvement is rather little, it is statistically significant with a significance level of 0.018. We believe the overall improvement is small due to most of our samples being English language subreddits and textual features can already predict the ban in most cases when we have interaction features.

6.5.4 Banning-by-example

We have established that the reasons behind subreddit bans are extremely varied and no single classifier can predict all types of bans. However, we also find that similar subreddit bans tend to cluster over time and we can use these banned subreddits of a type to identify subreddits misbehaving in a similar way. To test this, and demonstrate that a more targeted bank ranker can be built, we designed a banning-by-example ranker. We identified all days with at least 10 subreddit bans and used top n (varies from 1 to 10) largest banned subreddits on each day (example subreddits) to identify subreddits similar to them.

We use the pairwise similarity measure including both text and interaction similarity (as described in section 6.4.3) to determine similar subreddits. We re-estimate the parameter α (weight of text similarity) for this task. If we have more than one subreddit in the set of example subreddits, we take the average similarity with all the example subreddits as our similarity measure. For each day with at least 10 bans, we find top 10 similar subreddits from all of our banned and unbanned subreddits (not including the example subreddits themselves) and identify how many of them are banned (precision at k measure where $k = 10$). We calculate precision at 10 for each day and take the average as our final evaluation metric.

We find that we achieve the highest mean precision at 10 of 0.913 at $\alpha = 0.5$ and $n = 10$. In general, we observe that precision increases with n but the rate of growth slows down with higher values of n. Figure 6.8 shows variation of average precision at 10 vs the number of example subreddits (n) at different α values.

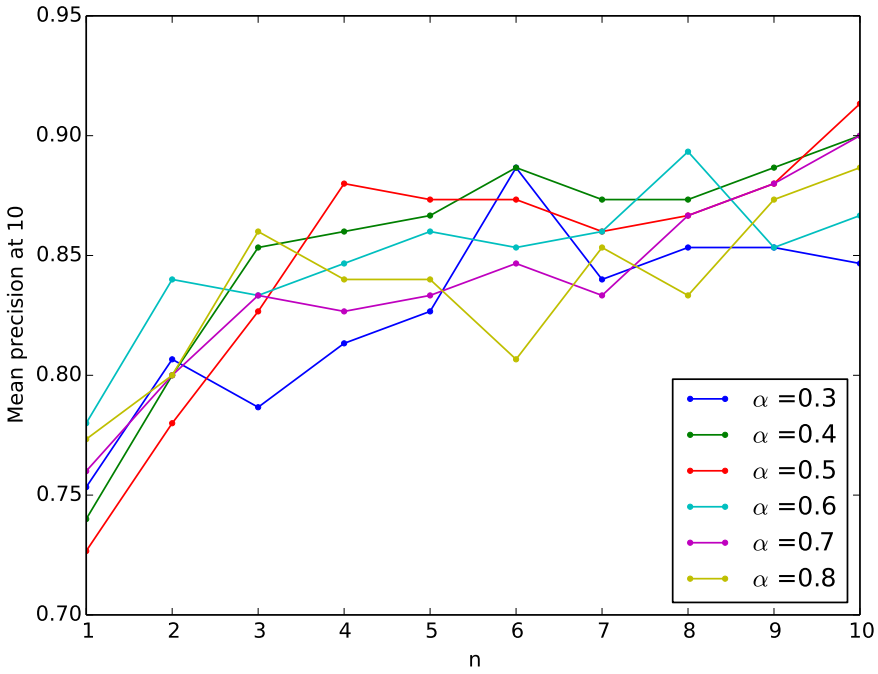

Figure 6.8: Average precision at 10 vs the number of example subreddits (n) at different α values.

We also test the banning-by-example methodology against 14,926 unbanned subreddits with at least 100 comments in January 2018. We exclude now defunct 'default' subreddits and non-english subreddits from this list. We only use text similarity as we do not have sufficient data to infer interaction features for the unbanned subreddits in just one month. We achieve the highest mean precision at 10 of 0.56 at $n = 10$ (for comparison, random guess achieves P@10 of 0.06). Figure 6.9 shows variation of average precision at 10 vs the number of example subreddits (n). This depicts a more realistic scenario for this approach and shows that there are many unbanned subreddits which are very similar to a lot of already banned subreddits. We observe that the unbanned subreddits which are similar to previously banned subreddits include a large number of pornographic subreddits (e.g., *wifesharing*, *snapchatnude*) and drugs-related subreddits (e.g., *drugsarebeautiful*, *Xanaxcartel*). We find that some of these subreddits are banned after our data collection (*snapchatnude*, *Xanaxcartel*), so they do not show up in our banned subreddit dataset. However, we note that support subreddits like *SuicideWatch* also show up in this list because of heavy use of suicide and death related terms in some banned subreddits. We believe that using the banning by example paradigm can provide human moderators with a filtered list which will

help improve community moderation.

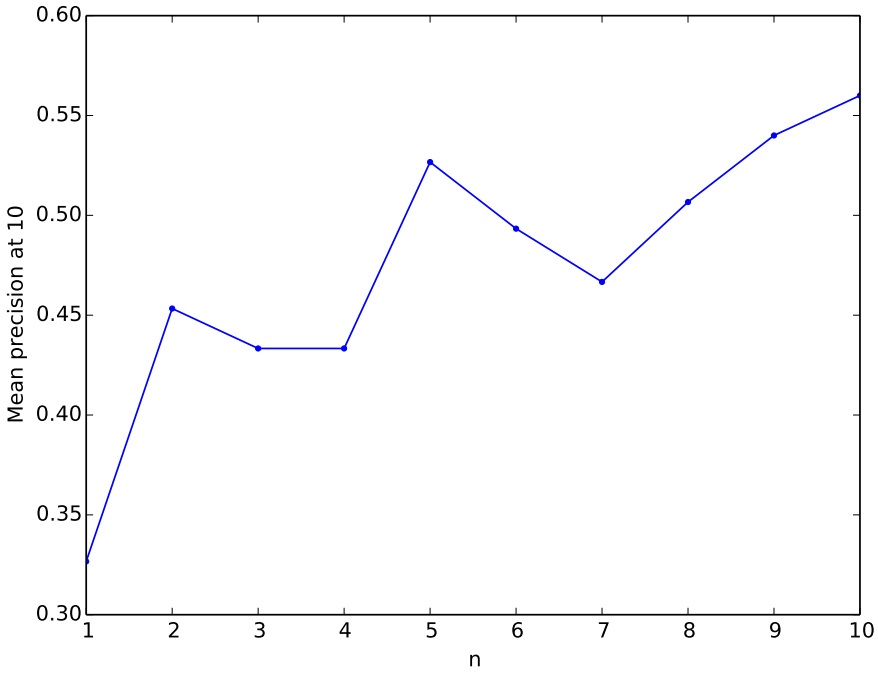

Figure 6.9: Average precision at 10 vs the number of example subreddits (n) using all unbanned subreddits from January 2018.

This banning-by-example approach can also be used in an agglomerative fashion to help identify banned subreddits from a specific category (given enough examples from that category). If we find a banned subreddit that does not fall into any category, we can start a new category with that particular subreddit and can find future similar subreddits and progressively add them as examples of that particular category.

6.5.5 Predictive Words

One outcome from our prediction task is that we are able to determine words that are indicative of banned subreddits by looking at their coefficients (a high value indicating importance). To get the top words predictive of banned subreddits, we identified top 1000 coefficients in the text classifier trained with all samples (i.e. all subreddits in the largest banned cluster and all unbanned samples) and evaluated the words. Theoretically, these words should give us an indication about the nature of content in the banned subreddits in the largest cluster.

We find that many of top-1000 words are violent, profane, threatening, indicative of causing bodily harm or death, anti-transgender, antisemitic, racist or pornographic

in nature. The nature of these words does indicate that toxic content is a prominent in the largest ban cluster (in contrast to other kinds of rule breaking).

6.6 Discussion and Limitations

While a corpus of banned subreddits is large, one limitation is that we do not have data for all known banned subreddits. Particularly those that have had all their content deleted. Though we find a diverse set of content in our corpus, there may still be a bias to the dataset in the data we do not have access to. While we do not know to what extent we are missing banned subreddits our hypothesis is that those comments that are deleted are likely more egregious norm or rule-violations and may be easier to detect. Conversely, we do find that our approach captures many banned subreddits which did not catch media attention and thus were invisible in the general public eye. We believe these smaller banned subreddits are very important in understanding the subreddit banning landscape. These may be harder to detect and are thus a more interesting moderation problem.

Although we show that banned subreddits can be predicted reasonably well given we have enough content, human moderators are still needed for more for subreddits such as *glassine* and *Ironsteel* where comments are coded. For example, all of the comments in *Ironsteel* are URLs which point to illegal streaming sites. These types of misbehavior are very difficult to analyze in an automated fashion. In our banning-by-example scenario we also need human moderators for identifying initial examples of banned subreddits of a specific type. In our work, we use Reddit's tendency to ban similar subreddits at once to our advantage. However, this is not applicable to all social media websites.

It is poosible to apply more sophisticated natural language techniques including phrase models, toxicity analysis to banned subreddit contents to improve our prediction results. Our analysis is currently limited to a limited set of unbanned subreddits. In a realistic use scenario, we anticipate needing to compare 'example' banned subreddits to a far larger corpora. Techniques such as embedding may allow our approach to scale. However, we note that we chose a simpler bag-of-words representation to support interpretability for this analysis. In addition to better representation, we believe more varied features will be able better distinguish between different ban reasons automatically.

6.7 Conclusion

In this analysis, we identified more than 1000 banned subreddits with their textual and interaction features. We believe this is the largest dataset on banned subreddits to date. We described both the reasons Reddit provides for banning as well as those derived through quantitative and qualitative analysis. We showed that subreddits removal is temporally clustered and those removed near each other (in time) are similar in other respects. Banned subreddits appear to fall into three main classes (internal, external, and meta) each with different properties. We demonstrated that it is possible to predict banned subreddits from content and interaction features. Finally, we showed how moderators could be supported through a banning-by-example classifier.

In the next chapter, we focus on differences of algorithms, specifically community detection algorithms and show how looking at differences along with similarities in algorithms can improve the problem of identifying communities.

CHAPTER VII

An Interactive Visualization Tool for Community Detection

7.1 Overview

In this chapter, we look into differences in algorithms to point out robustness (or the lack thereof) of certain outputs, the uncertainty of certain parts and in general improve overall results in conjunction with ensemble algorithms. We apply this idea on a set community detection algorithms to figure out if a given network has 'good' community structures, which nodes have weak community assignments and how to incorporate user knowledge using an interactive visualization tool CommunityDiff.

Community structure often provides useful insights about the underlying network. For example, communities in social network represent groups of people who share an interest, location, or other semantically meaningful attributes (e.g., were all 'friends in high school') [80, 179]. In biological and brain networks, communities capture functional groups [29, 91]. Many other non-network problems in machine learning and data mining are recast as networks so that the network analysis tools, including community detection, can be used. For example transforming text to graphs in NLP [139] or user-item recommender models [94] allows us to use community detection to solve problems as diverse as document summarization or movie recommendations.

In practice, there are many community detection algorithms with very different properties: some work better at large or small scales, many use specific optimization functions (e.g., modularity), others contain randomization, some work better with weighted networks, and so on. The output *partitions* (i.e. *communities* or *community labels*) can be significantly different even when the input is the same. Because the space of possible algorithms and parameters is vast, selecting a satisfactory approach is difficult (let alone optimal). Even for algorithm experts, the selection can be a

black art. For domain experts (e.g., social networks, biology, etc.) the choices can be overwhelming.

We are specifically interested in supporting domain experts who would like to partition a network into "communities" where labels are important. For example, the manual task of labeling blogs based on political party [3] or sub-communities in an ego-net [84]. Domain experts already have access to tools to support community detection. While they may be familiar with network analysis, the tools that are available are often generic and do not readily support community detection beyond executing the algorithms. Though we return to the issue of overlapping communities below, in many of these domains, "hard" partitions are desirable. This kind of community partitioning is one of the main functions of most network analysis packages used today (e.g., Pajek [16] and UCINET [25] in the social sciences; Cytoscape in biology [171]; and more general tools such as Gephi [15], GUESS [4], and NodeXL [82], etc.) or libraries such as *igraph* [46]. Each software or library contains multiple implementations and it is often difficult for an end-user to determine which is appropriate. Given the specific goal of "accurate" and well-understood partitions, the end-user would like to be provided with a viable automated starting point and a facility to understand differences and quickly "fix" labeling mistakes with as few interactions as possible.

One approach to dealing with too many choices is simply not pick any specific one and to combine algorithm output. This is one of the motivations behind *ensemble techniques* [130, 161]. Network scientists have devised ways to apply this technique to community detection [48, 113]. Ensembles can be treated as weighted combinations of different algorithms or parameters: each algorithm 'votes' on the output and their votes are combined in some way. The hope is that the ensemble will produce something closer to the "correct" answer. However, even with ensemble techniques achieving perfect clustering accuracy (according to some ground truth) is often impossible. This may be due to the presence of noisy or ambiguous data (missing edges, incorrect weights, extra edges, etc.) or the assumptions of community detection algorithms (e.g., that nodes in the same community are connected).

It is here where domain knowledge is critical to make corrections to algorithm output. Domain knowledge can include anything from knowing how many communities should exist or knowing which nodes should (or should not) be in the same community. In practice, however, the "completeness" of this knowledge can vary greatly. For example, the expert may only know that there are 5 communities but not what should be in them, or the expert may only be able to point at only subset of pairs of nodes that belong in the same community (e.g., experimental evidence in

a biological network may only provide clear functional groups for some of the data). Ideally, an analytical tool would allow the end-user to: determine that the "facts" they know are captured in the current partition (or not); make corrections to bring the partition in line with known facts; and make decisions on those cases where there is no background knowledge. The tool we describe here combines both sensemaking and data-wrangling. This is in the sense that the high level objective is to create 'clean' groupings (the wrangling part) which requires understanding the network in various ways (the sensemaking part). The particular activities vary depending on the prior knowledge of the end-user but both are entwined and both should be supported.

Distilling this analytical pipeline, we can abstractly view our workflow as being comprised of two main "phases": (1) finding the best initial partitioning of the data (the best starting communities) and then (2) making the necessary corrections (with or without the software's help). From the perspective of the interface, this requires supporting the end-users' decision making about whether a partition is "good" and then making decisions about corrections and refinements.

CommunityDiff (see Figure 7.1) was designed with this pipeline in mind. The tool supports interactive analysis of community structures by coupling a novel visual analytics environment and decision-making tools[1]. CommunityDiff supports two main task types. The first is rapidly labeling all nodes in the network when ground truth is known (e.g., quickly partitioning handwriting samples by digit type or classifying the end-user's own social network egonet). The second mode is when the knowledge is incomplete and the end-user must compare different partitions in making decisions. Both tasks are supported by a visualization of *ensemble spaces* — the output of individual algorithms as well as mixtures. Through this view, the end-user can rapidly compare different ensembles and quickly resolve on partitions that are close to their final objective. Additionally, we introduce a second set of tools that help the end-user judge between alternative labels for specific nodes. Finally, a novel active-learning framework ensures that end-user actions can be integrated into the machine learning process to constantly improve results. CommunityDiff 'signals' to the end-user which labels would lead to better classification (visually, by making certain nodes more salient). Any operation done by the end-user — ranging from moving nodes between groups, specifying the number of clusters, excluding nodes from group membership, and many others — is taken into account as a form of supervision. Decisions are supported by focusing on uncertainty of community labels. For example, nodes with uncertain labels are larger in size and there are explicit

[1] A brief video demonstrating the system is available at `https://youtu.be/KNdQqWXTT8w`

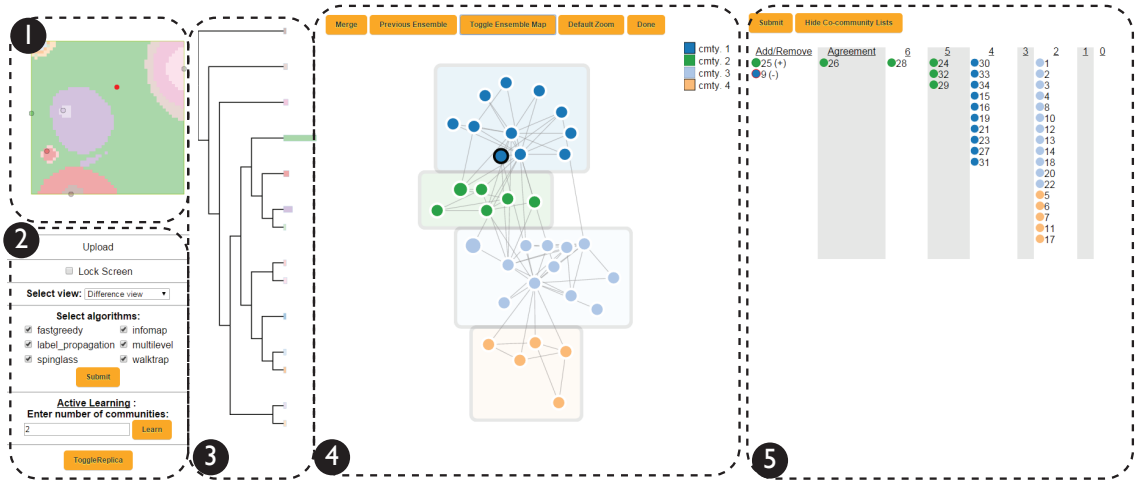

Figure 7.1: CommunityDiff: (1) The *Ensemble Space Heatmap* showing different algorithms as dots and different ensembles achieved using these algorithms, (2) A panel showing different features and controls for the *Ensemble Space Heatmap* and the active learning panel, (3) A dendrogram showing relations among and proportions of different ensemble outputs, (4) The network diagram, (5) The co-community lists. Note that this view was used in evaluation and some advanced features were hidden (e.g., downloading the network).

functions to show agreement of different algorithms for nodes in the same cluster.

We demonstrate, through both automated and lab experiments, that CommunityDiff allows for rapidly constructed, high confidence, and accurate community labels. Currently, we only focus on disjoint community detection rather than the overlapping variant as overlapping community detection algorithms are generally poor performers. In fact, discrete community detection often performs better than custom overlapping-community algorithms for overlapping community detection benchmarks [158]. For this reason, use of overlapping community detection algorithms is far less common compared to their hard-partition counterparts. We find that the most common social network analysis tools focus on hard partitions. While abstractly CommunityDiff can model overlapping communities (e.g., the overlap of two communities A and B can be modeled as a third community C that is the intersection of the two) and hierarchical communities (e.g., a community can be further partitioned into sub-communities), the current visualizations of CommunityDiff are not optimized for these variants.

The community-detection task we describe is an instance of a broader set of analytical pipelines. In many other Data Mining and Machine Learning tasks there are

numerous alternative implementations (clustering algorithms, classification systems, etc.). With any of these, the domain expert must compare different models or solutions and make corrections in order to arrive at a satisfactory answer. In addition to demonstrating the effectiveness of CommunityDiff for the particular community-detection problem, we capture both design guidelines and visualization techniques that can be directly applied to other Interactive Machine Learning (IML) systems.

In this project, we contribute CommunityDiff, an interactive community detection analysis tool that allows end-users to easily compare, combine, and modify communities generated by multiple algorithms. We demonstrate how the idea of "ensemble-spaces" and specific heatmap-based representations can allow the end-user to identify common communities, where there is disagreement between algorithms, and to easily correct errors in labels. Along with other visual elements (e.g., conventional node-link diagrams, dendrograms and a novel co-community list), CommunityDiff also integrates active learning to visually suggest those nodes that should be labeled and which readily adapts the community structures based on end-user feedback. CommunityDiff is intended to support a common task for interactive machine learning systems, where the human must interpret the output of the algorithm and make corrections.

The idea of ensemble-spaces, heatmap-based representations and highlighting "hard-to-label" nodes based on differences and disagreements of community detection algorithms. A paper on which this chapter is based on is published at [50].

7.2 Related Work

Related work to CommunityDiff falls broadly into two categories: interactive features to support humans as part of the machine learning 'use' pipeline; and the algorithm side that produces better results based on human intervention.

7.2.1 Interactive Machine Learning

There are many visual techniques to analyze the output of machine learning and data mining algorithms. Simple scatter plots and histograms (for understanding data distributions) or precision/recall, ROC curves, or confusion matrices (for evaluating output) are a common representation available to developers in most platforms [20, 56, 138, 183]. To improve the designer's ability to debug different pipeline components a number of specific visualization techniques have been developed to provide an enhanced view at different outputs. Examples include visual model com-

parison [150, 172] and model interpretation [19, 33, 118, 142, 180, 194]. These range from generic [180, 194] to model and data specific (e.g., graphs [172] or text [142]), to algorithm-specific (e.g., Bayesian text classification [19] and SVMs [33]). Often these focus on static representations or basic comparisons (e.g., ROC or precision/recall) at the end of the analysis pipeline.

A good example of general visual model analysis is LoVis [194], which compares different linear models on any kind of data visually to identify local patterns in the data. In graph specific visual model comparison, Sharara et al. [172] introduced G-Pare, a visual analytics tools which compares output of different machine learning algorithms on networks and provides both a global view of algorithm outputs and difference of algorithms on specific subsets of nodes. This is similar in intent to our ability CommunityDiff's functions, comparing outputs of different community detection algorithm on the whole graph or a specific subset of nodes, though in our case we emphasize the use of ensembles and not simply comparisons. For algorithm-specific models, Becker et al. [19] provides a visualization of naive Bayes classifier outputs, specifically focusing on probabilistic importance of each feature and importance of each feature in a specific example. However, these systems often focus solely on comparison and not the interactive corrections.

Recognizing that by adding interactivity, visualizations could also support better analytical pipelines, researchers have turned to Interactive Machine Learning (IML) [61, 65, 155]. Through interaction with visualizations and other means of "dialog" (e.g., questions to the end-user), IML systems can obtain new training examples for classifications [67, 68], refine features [26, 135], identify trade-offs the designer is willing to make [97], modify individual algorithm behavior (e.g., Hidden Markov Models [49] or classifiers [12], and create effective ensembles [98, 177]. Such solutions are often an extension to visual workflow designs such as the Wekinator [67] extension to Weka [183] or plugins for Orange [56] or KNIME [20].

IML systems often focus on improving training data. For example CueFlik by Fogarty et al. [68] allows end-users in an image search engine to create their own rules for image search and improve upon searched image ranking in a personalized active learning framework. Other IML system, such SmartStripes [135], focus on interactive feature selection that allows user to explore dependency between different feature and entity subsets. Ensemble classifier algorithms can also interactively benefit from human input as demonstrated by Kapoor et al. [98] (with the aid of misclassification cost visualization for a multi-class ensemble classifier). These approaches are often singular in focus and reinforce traditional 'black box' models that

do not support the different modalities under which the end-user can operate (labeling, exploring, comparing, etc.). Our contention is that even simple analytical pipelines (e.g., generating community labels or clustering) require multiple modes of operation and multiple steps. To be effective, systems need to support different types of analytical work which requires varied, but integrated, visualization components. Through CommunityDiff we attempt to provide such a solution for domain-experts by building support for different steps through different visual elements.

A final area of related work is in intelligent data wrangling [79, 95]. Generating and correcting community labels on a graph can be viewed as a form of data wrangling. The end-user is 'fixing' and validating some element of the data, in this case labels. Though data wrangling work is most often focused on more standard columnar data (e.g., [79]) work in the field has moved to incorporate visualizations, active learning, and mixed-initiative frameworks. In this way, CommunityDiff can be situated in this broad space.

7.2.1.1 The Human Side of ML

The IML community has developed a number of design guidelines for eliciting feedback from human participants [9]. Horvitz [92] provided some key guidelines (for example, considering uncertainty about and user's goal and employing 'dialog' to resolve key uncertainties) for designing mixed-initiative user interfaces that allows efficient communication between humans and the user interfaces. Our goal is to leverage these lessons as a starting point for our interactions. For example, Cakmak et al. [32] found that individuals providing feedback dislike acting as oracles for active learners (e.g., answering a constant stream of tedious questions) and Stumpf et al. [175] found that end-users wanted variety in the feedback they provided (e.g., modifying features, weights, etc.). Others have focused on the necessary level of interpretability in representations [107]. In part inspired by this work, we devise a set of guidelines specifically intended to guide the construction of effective visual analytics solutions.

7.2.2 Ensembles and Community Detection

CommunityDiff functions, in part, by allowing end-users to build ensembles of community-detection algorithm outputs as these often represent a "better" solution. Lancichinetti et al. [113] showed that use of consensus clustering (a kind of ensemble technique that uses voting) can boost accuracy and stability of a community detection algorithm. In this method, a single algorithm is run over the input network a number

of times. The edge-weights of this network are modified using the results of different runs. The original algorithm is run over the modified network again. This works well in case of fast, but unstable, algorithms such as Fastgreedy [147]. Dahlin et al. [48] used a node-based fusion of communities algorithm which applies hierarchical clustering of nodes with a special linkage rule. Burgess et al. [31] create ensembles by generating variations of the same graph using randomized link-prediction and treating each run of the community detection algorithm as input to an ensemble.

Our idea of creating an ensemble space for community detection is very much inspired by the system described by Grimmer and King [78]. Their system focused on generating ensembles of multi-dimensional clustering algorithms. A metric space of these approaches allowed them to identify "holes" in this space and to generate ensembles that proved to be effective in finding the correct classification. We expand on this visualization to integrate additional "heat-maps" that capture differences between algorithms, key community-structure metrics (e.g., modularity). This ensemble visualization is used in combination with a number of other visual and algorithmic elements to support the end-user's labeling and partitioning tasks.

7.2.3 Active learning

Active learning (AL) has seen a number of advancements in recent years from the algorithmic perspective [168] though it is rare for the focus to be on graph structures. More critically, very little attention has been paid to the interplay between visualization and active learning. From the algorithmic perspective a notable exception is Leng et al. [120] who used a label-propagation based approach to incorporate active learning in community detection. Macskassy et al. [131] used graph features for active learning in networked datasets which can be viewed as a form of community detection in networked data. Within the HCI context, Amershi et al. [10] demonstrated the use of an active learning scheme for grouping individuals by using node properties such as 'age' or 'place of birth' (rather than network structure). The Apolo system [37] utilized active learning to help end-users collect and label related papers. The task was framed around an iterative process of collecting and labeling a small set of papers which could then focus on the "expansion" of this set.

7.3 The Trouble With Community Detection

Before describing the design of CommunityDiff in detail it is worth considering why interactive tools might offer an advantage when performing community detec-

tion. To understand why, we consider *how* community detection algorithms work, and therefore, fail. Community detection is dependent on structure and metadata (on nodes or edges). If edges were only to exist between nodes that were in the same community, or nodes contained explicit metadata that identified which community they belong to (e.g., a node identified as 'group 1'), the solution would be obvious. Because networks are never this well structured, community detection algorithms attempt to find communities by optimizing on some metric (e.g., modularity or being part of a k-core [179]). Because networks are noisy and community detection algorithms can at best utilize an approximation of 'optimality' (based on their metrics), there can be no guarantee of a 'correct' output.

This should be unsurprising as community detection is a form of clustering. Specific studies on networks have shown that a "perfect and general" community detection algorithm is likely out of reach either because the networks are noisy [31] (i.e. sampled nodes and edges do not reflect reality) or the algorithms make assumptions about structure [60] (e.g., overly focusing on metrics such as modularity) or metadata [151] (e.g., that metadata reflects ground truth). The consequence of these assumptions is that fully-automated community detection, even in simple networks (e.g., [190]), can fail to produce the 'correct' community assignments.

Even when we tolerate imperfect partitioning (e.g., the down-stream use of the communities will still work) being able to evaluate different algorithms is extremely useful. Different algorithms can have systematic biases that need to be understood (e.g., restrictive models of communities that produce too many partitions with very few nodes). Thus, human feedback and interaction is a crucial part of generating high quality, believable, and usable partitions.

Even those developing community detection algorithms may find it difficult to evaluate the algorithms due to scarcity of ground truth partitions. Algorithm designers often rely on synthesized data or external node metadata in lieu of ground truth community assignment [189]. However, recent research suggests that metadata partitions in the network may not align with community assignments suggested by network structure [93, 151]. Thus, it may be hard to fully characterize a community detection algorithm's performance. Because of this, even experts using community detection algorithms may utilize multiple algorithms (or ensembles) to better understand the quality of the partition. The intuition is that if multiple algorithms arrive at the same result, the researcher can infer that the resulting partition is stable. On the other hand if each algorithm produces widely varying results from each out, one gets a sense of instability.

Despite the various ways community detection algorithms can fail, the manifestation of those failures are somewhat consistent. Broadly, algorithms will produce a different distribution of errors, but we observe the following three categories:

1. **Split community** — A single community split into two (or more) sub-communities. Each sub-community is "valid" (in that there is no incorrectly placed node) but a more valid labeling has all nodes in the same community. Often, variants of the Louvain [24] and InfoMap [162] community detection algorithms generate these kinds of errors more than other algorithms due to 'field-of-view' limits [60].

2. **Single-node misclassification** — this occurs when a single node is assigned to the wrong community. This occurs most often for "bridge" nodes, where the node has strong connections to two (or more) communities and is mislabeled in the end. These mistakes tend to be less frequent than the split communities but potentially critical as such nodes often have high centrality scores, making them 'critical' to the network. Many algorithms 'fail' in this way on networks such as Zachary's Karate Club [190].

3. **Merged community** — Communities can be mistakenly merged (often due to strong connections between them). This results in large communities and overall less number of communities. Label propagation, for example, tends to merge different communities to a single one. Algorithms that rely on modularity [146] may also run into resolution limits that are unable to pull out smaller sub-communities from larger ones [70].

Figure 7.2 depicts the three kinds of errors shown in community detection. It is possible to think of split and merged community errors as being composed of multiple single-node misclassifications. That is, one could resolve these errors using many single-node relabeling (i.e. moves). If errors were completely random, our end-users would need to perform single-node corrections, one at a time, to build the correct structures. However, this is not realistic in the context of community detection (and most other clustering problems). Split communities are a very common occurrence (merged communities, slightly less so). Thus we would like to ensure that our end-user can quickly correct these errors without moving one node at a time. These three errors are ones that a human agent would either need to correct or understand (i.e. mentally model).

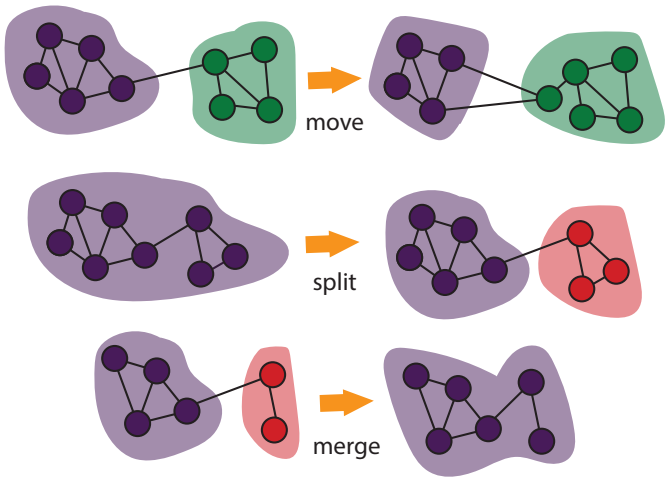

Figure 7.2: Different kinds of errors in community detection. (a)Single-node misclassification, (b)Merged community, (c)Split community. On the right hand side, the incorrect community assignment is shown and on the left hand side the corrected version through a specific action (shown in figure) is presented.

7.3.1 The Analytical Pipeline

Recall that the end-user is using two 'pipelines' in our scenario: a *sensemaking* activity that helps the end-user model the space (i.e. possible communities) and a *data wrangling* activity by which the end-user corrects the algorithmic output (or manually produces labels). These are clearly not independent of each other, and one may view sensemaking as including wrangling decisions in organizing data and conversely wrangling decisions can be aided by analysis. Depending on the state of the network and background knowledge of the end-user, they may focus more on one task or another. For example, if they know all the community labels, this is largely a wrangling task. The less complete the knowledge, the more the end-user must rely on sensemaking techniques to make or validate labeling decisions. CommunityDiff is designed to help with both tasks.

Wrangling work, which in some sense is a more limited task, is supported through automation. Active learning, mixed-initiatve interactions, supervision, and other human-in-the-loop paradigms can take advantage of human action to accelerate the wrangling process [95]. Wrangling is naturally 'easier' when the end-user has a perfect mental model of the ground truth. At worst, the end-user could label each node — one at a time. Automation speeds this up. However, with incomplete knowledge, the end-user may need guidance to help decide between different labels which is where

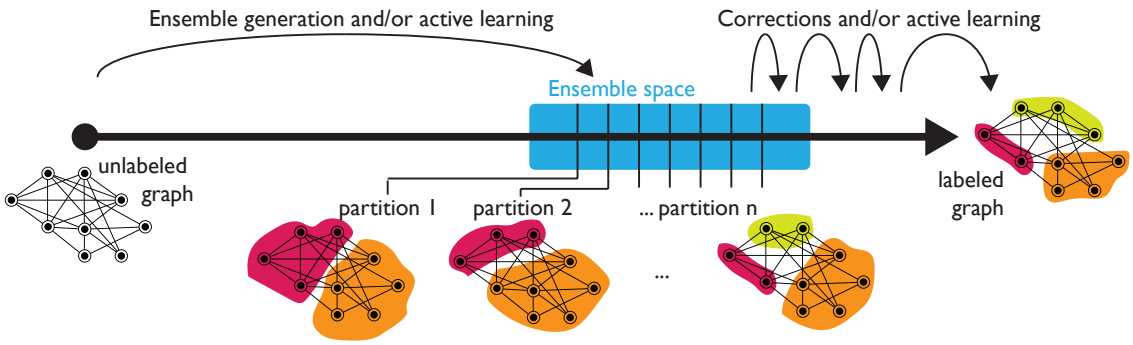

Figure 7.3: Exploration process from an unlabeled graph to the final partition. The end-user moves from the unlabeled graph to a reasonable starting point generated by the initial execution of the partitioning algorithms and ensembles, followed by corrections through the use of direct manipulation and active learning.

the broader notion of sensemaking comes into play.

Work with data often requires iterative loops of sensemaking, where the analyst engages in cycles of data collection, organization, hypothesis testing/analysis, back to data collection, and so on [152] (eventually terminating on a 'satisfying' answer). For a community detection task, we can view this process as: the analyst starts with an unlabeled graph and through a sequence of decisions and 'wrangling' actions produces a graph where nodes are annotated with a community label. It is this framing that drives our design. The many algorithmic and design decisions we made are motivated by a need to support this progression. In the context of community detection, the sensemaking and analysis process might be aided by high-level information (e.g., how many communities were found? how do the community sizes range?) and low-level information (e.g., is this pair of nodes in the same community?). In some cases, the domain-expert can not evaluate the answers to these questions (e.g., they may not know if the two nodes should be in the same community or may only have a sense that the number of communities range from 3 to 5). Here, information about algorithm or output confidence — which in our case can be derived by using multiple algorithms — can guide a decision.

We graphically depict (a partially linearized) view of this idea in Figure 7.3. As we argue below, CommunityDiff is intended to capture viable analytics start states by producing a solution space (the ensemble space) and presenting information that help progress the analyst in making decisions. By providing 'views' into this space, the end-user may pick solutions that are *close* to the final target (both an act of

sensemaking and data-wrangling). Our goal is to make these good solutions highly salient and the differences between solutions obvious. From this point, the analyst can make smaller decisions to allow them to reach the target labeling. When the analyst has a clear model of ground truth such decisions should not require significant manual work (e.g., clicking and dragging individual nodes). When the answer is less clear, CommunityDiff is designed to provide a high level overview of options and a fast mechanism for acting on these options (e.g., through the Co-Community View). Furthermore, CommunityDiff constantly uses the human feedback to improve the classification of nodes that still need to be labeled (aiding the data-wrangling activity).

7.4 The CommunityDiff Design

Having described the ways in which algorithms can fail, and the mechanisms by which a domain scientist can navigate this space, we can describe the key features of CommunityDiff. We also briefly illustrate its use with an example and offer a set of design guidelines that informed our specific implementation.

Figure 7.1 depicts a typical configuration of CommunityDiff. Upon loading up a new network, Blocks 1-4 are visible. Block 5 is initially hidden from view and is activated to support 'fine' manipulation of the network structure. The network is shown in Block 4 (referred as the network diagram), with an initial community assignment of nodes, which CommunityDiff interprets as the best community assignment for the network without any input from the user (further elaborated in next paragraph).

In the standard node-link view, each community is shown inside a bounding box for easy identification of the communities. The nodes and the bounding boxes of different communities are colored differently. CommunityDiff also highlights the nodes (by making them larger in size) for which the system is less confident of the community assignment. This allows the end-user to focus on these specific nodes (elaborated in 7.5.3.3). This interface allows the end-user to correct specific failures in the community detection output (e.g., moving specific nodes, splitting and merging communities). However, this may be costly if there are many errors or difficult if the end-user does not have a mental model of the ground truth. To address both problems, CommunityDiff offers alternative community detection 'outputs.'

CommunityDiff initially takes six different community detection algorithm and their combinations to identify possible communities. The algorithms can be dynamically and selectively enabled and disabled and CommunityDiff is architected to

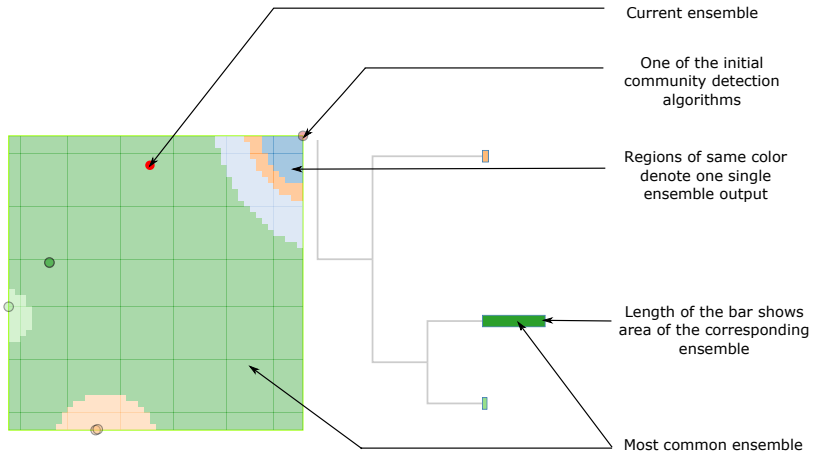

Figure 7.4: Elaboration of different components of the ensemble heatmap and the dendrogram. Initial community detection algorithms are shown as multi-colored dots (except the red dot) in the heatmap. Each differently colored region in the heatmap represents a different ensemble output. The red dot corresponds to the current ensemble output shown in the network diagram. The output of different algorithms can be viewed by clicking corresponding multi-colored dots. The dendrogram shows relationship among different ensembles. The interactive bars in the dendrogram correspond to the ensemble which is colored the same in the heatmap and the length corresponds to the area of that region. Output of the corresponding ensemble algorithm can be viewed by either clicking the corresponding region in the heatmap or the corresponding bar in the dendrogram.

support the addition of new algorithms. The user can choose any subset of the six algorithms (with the constraint that there must be at least two algorithms for the ensemble operations). For purpose of this system, we chose six very well known community detection algorithms as a starting point. We compare and contrast these algorithms and their different combinations (the ensembles) using an abstract metric space where each algorithm is represented by a dot and each point in this space represent an unique ensemble algorithm based on the point's distance from the 'dots' (shown in Figure 7.4). The distance between two algorithms (dots) is proportional to how different their output is. We call the visualization of this space the *ensemble space heatmap* or simply *ensemble heatmap* (explained in detail in Section 7.5.2). Output of each of the original algorithms and each ensemble can be viewed by clicking on the point. The community assignment is dynamically changed accordingly in the network diagram panel with minimum number of community reassignments from the previous figures. Nodes which changed communities are highlighted temporarily for easy identification. However, not all ensemble algorithms produce different output. In fact, we see very few unique outputs compared to the vast number (2500) of ensembles calculated. Thus, in the ensemble map, ensemble algorithms that produce the same output are colored the same, i.e. the ensemble map is partitioned by different colored regions where each region produce a different output. The largest region in the ensemble map produce the most stable community assignment and this assignment is initially reflected in the network diagram.

While the heatmap view makes salient broad areas of stability and the relationship between algorithms, it is not the most effective view for certain tasks. For example, these ensemble map regions can have arbitrary shape and it is not always easy to tell which is the largest region if there are multiple large regions. Due to their shape, it is difficult to rank other large regions. Moreover, from the ensemble map, we have no idea how different the outputs of these different regions are. For this purpose, we provide a dendrogram (shown in Figure 7.1(3) and Figure 7.4) based on the similarity of ensemble outputs. The dendrogram also provided bars to show the size of different ensemble regions so it is easy to identify and rank large regions. Bars are colored according the color of the region in the ensemble map and highlights the corresponding region when hovered on. This is further detailed in Section 7.5.3.2.

CommunityDiff also provides the user with a choice of different algorithms, multiple views of the ensemble space (explained in Section 7.5.3.1) and active learning with specified number of communities with the option of manually labelled nodes (elaborated in Section 7.5.4.2) in the panel shown in Figure 7.1(2).

The user can bring the node-specific co-community lists (shown in Figure 7.1(5)) by shift-clicking a particular node in the network diagram (the node with thicker borderlines in Figure 7.1(4)). These lists show how many original algorithms put all the other nodes in the highlighted nodes community. This gives a sense of how robust a particular community in the network is. These list also allow the user to iteratively add/remove nodes from the highlighted nodes community to achieve their desired partition. The user can multiple add/remove constraints at once. It is to be noted that these lists are different for each node and this panel is not initially visible. The user can hide this panels by clicking the 'Hide Co-community Lists' button.

To help with navigation, CommunityDiff supports additional functions such as zooming, toggling node labels, hiding particular communities etc. which is further detailed in Section 7.5.3.5. Overall, CommunityDiff provides an end-user with an easy way to compare and contrast different algorithms and ensembles, to incorporate own knowledge about the network/communities without intricate knowledge about the algorithms and have a sense of how good or bad the overall partitioning of the network or a particular community is.

7.4.1 Example Interaction

Let's follow a political scientist, Alice, as she looks at a small social network she has collected by mining messages on a specific issue between politicians on Twitter. She knows there are a two to four main groups developing different legislation on the issue, but not necessarily which politician has committed to each group (she does know which party they belong to, and some past interactions). Her goal is to generate groupings of politicians that are likely to push for the different legislation. CommunityDiff can help, by giving Alice a workable starting point, the ability to correct errors produced by the algorithms (broadly, wrangling), and information to help decide on labels when there is disagreement or uncertainty (broadly, sensemaking).

Upon loading the network Alice sees 4 blocks in the interface (Figure 7.1(1-4)) as co-community lists are initially hidden from view. CommunityDiff automatically executes a number of community finding algorithms and generates the ensemble space view in the heatmap (see Figure 7.1(1)). The heatmap represents a view of the six algorithms as well as the ensembles of those algorithms (each dot is a base algorithm and each color is a unique ensemble). In this view Alice can quickly identify the most stable ensemble (the large green surface, here we refer ensemble algorithm outputs as ensemble or ensemble output). This is reinforced by the dendrogram view (Figure 7.1(3)) which not only shows that this particular ensemble is common, but it is also

similar to two other very stable ensembles. Upon clicking on a point in this ensemble inside the heatmap view, the network map (Figure 7.1(4)) dynamically changes to show the current partitioning. Alice can click on other points in the ensemble space to compare different partitions. CommunityDiff makes the differences salient by briefly highlighting nodes that have 'moved' (i.e. changed community assignment or moved from one community to another after performing certain actions) thus supporting visual comparison between algorithms or ensembles. Alice can quickly test different partitions of her network to identify a good starting point for further analysis (the one that is closest to what she knows is true about this network).

Alice can further configure the ensemble space by adding or removing algorithms in the control panel (Figure 7.1(2)) or change the heatmap view. This panel also allows her to force CommunityDiff to provide feedback to the learning system. Alice can constrain CommunityDiff by either setting a specific number of clusters or using direct manipulation to move nodes between communities. For example, Alice may know that the two communities are aligning with the two main political parties so she may set the number of communities to two. These changes become constraints for the machine learning system and will be adhered to by the newly proposed partitioning.

The underlying active learning system can focus Alice on nodes that seem to be unstable. These are made more salient by size. Alice clicks on one of these nodes (politician 29) to see the *Co-Community View* (Figure 7.5). A number of columns show how often different algorithms agree on which other politician should be in the same community as politician 26. For example, four of the algorithms agree that politician 32 belongs in the same community, but only 3 of the 6 believe that politician 9 should be. Given her background experience, Alice quickly selects politician 32 (which was in the high-agreement column, i.e. majority of algorithms agree that this node should be in the community of the highlighted node) to force it to be in the same community as politician 29. Alice knows that politician 9 can not possibly belong in the same community as he is a core member of the opposing party. She indicates to CommunityDiff that it is a negative example (Figure 7.5). Upon submitting her changes, CommunityDiff propagates these changes and constraints. Once satisfied with any particular grouping, Alice can hide it form the network diagram and focus only on the remaining groups. Alternatively, she could manually label the unstable node and asked the classifier to get to a new partitioning.

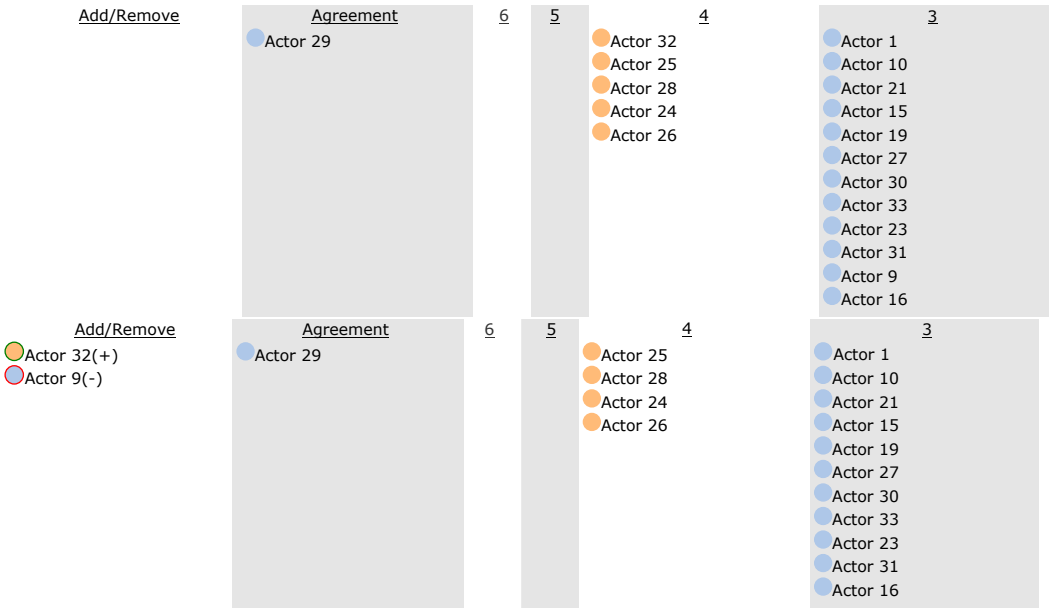

Figure 7.5: A partial view of the co-community lists before (figure on top) and after (figure on bottom) Alice's decisions. In both figures, the 'Agreement' column shows actor 29, whose co-community lists Alice is working on. Initially, actor 32 is on column '4', which means 4 algorithms agree it should be in the community of actor 29. On the other hand actor 9 is on column '3'. After Alice decided to add actor 32 to actor 29's community and remove actor 9 from the said community, these nodes move to 'Add/Remove' column.

7.4.2 Design Guidelines

Before describing our implementation we define a set of design guidelines that motivated our decisions. These were produced based on the analysis pipeline model described above.

1. ***Respect the Mental Model*** — *The system should seek to quickly bring the system state in line with end-users background knowledge.* The decision of whether the result is 'in line' requires end-user feedback. Thus, whenever possible, the end-user should be able to quickly and accurately judge the current results relative to their mental-model [123]. For example, Alice, our political scientist, can quickly reject partitions that produce more than the two to four communities she was expecting.

2. ***Respect the Unknown*** — *The system should help the end-user resolve gaps in their mental model* [123]. When an end user cannot answer a question such as, 'does node a belong in community C or D,' the system should provide

evidence to support this decision. Options that are more likely to be correct should be made more salient (e.g., large bars in the dendrogram or areas in the heatmap to indicate stable partitioning). The system should also make obvious where feedback from the end-user would be useful. In CommunityDiff, we implement node stability (explained in 7.5.3.3 in detail), which gives the user a sense of how certain the system is about the node's community assignment. Visually, unstable nodes are made larger in size. The larger the node, the more unstable the node is. This also gives the user a sense of which nodes to focus on. Moreover, using the co-community lists, the user can explicitly analyze a node and its current community assignment with exact knowledge of how many algorithms agree on this assignments, which nodes are more likely to be in the same cluster as a current node etc. Alice, for example, would be able to quickly see how likely a politician is to vote with one group or another given their past connections to both. At a community level, co-community lists also tell the user if there is smaller coherent cluster in a large community or if there is strong connection between two communities. In general, via the ensemble heatmap and dendrogram, CommunityDiff tells the user which results are likely and how likely they are.

3. ***Respect Decisions*** — *Assume that an end-user will not likely reverse a decision. Once a decision is made, it should be persistent, i.e. it should not be necessary to make it again.* [92, 173] If the end-user indicates that there are only 3 communities or that nodes A and B should always be in the same community but C and D should never be in the same group, this should be respected by the system. CommunityDiff iteratively learns from the user's previous decisions and maintains all previous constraints whenever a new constraint is added. Once Alice decides that there are three communities or that politician 32 belongs with 29, new executions of community detection should not break this (and should, in fact, leverage the information to produce better output).

4. ***Assume Greedy Progression*** — *Actions that create the biggest change should be available and be explicit.* Through a combination of direct manipulation on 'batches' of nodes or communities and targeted active learning, the system should help the end-user minimize the number of steps needed to complete the labeling. CommunityDiff provided a number of ways to minimize user effort and ease including merging multiple communities, active learning with or without labelled nodes, fixing number of communities, processing multiple con-

straints at once via the co-community lists. Alice, for example, can focus her data-wrangling effort on the largest legislative group before moving onto the next one.

5. ***Acknowledge Outliers*** — *The interface should provide a way for capturing outliers.* We follow the "NetViz Nirvana" criteria for network visualization by Dunne et al. [59] (specifically, the fourth point: "Clusters and outliers are identifiable") . While we assume that most nodes will have a strong affinity to one community or another, occasionally a node will fit into multiple groups or to none. When this determination is made algorithmically, these outliers should be obvious to the end-user. Node size (determined by node stability) clearly points out outlier nodes. Using co-community lists these nodes can be further analyzed.

7.5 System Details

Having established the high level goals and interface, we focus on the specific implementation and details of the visualizations.

7.5.1 System and Interface Architecture

CommunityDiff is composed of a back-end which processes the graph data (implemented in Python), and a Web-based front-end (implemented using D3 and Javascript). Recall that the front-end consists of four major panels: an *Ensemble Panel* (consisting of a heatmap projection dendrogram in, panels 1 and 3, respectively, in Figure 7.1), a *Control Panel* (Figure 7.1(2)), a *Network Visualization Panel* (Figure 7.1(4)), and a *Co-Community Panel* (Figure 7.1(5)). The panels are roughly placed in an ordered way (from left to right) to match the likely workflow: identifying a 'good starting point,' validating the automated label choices, and then refining labels. In all the views, the decisions of the end-user (selection of algorithms, movement of nodes between groups, labeling groups, etc.) directly influence the learning algorithm.

The **Ensemble Panel** contains two visualizations. The first is the **ensemble heatmap** projection (see Figure 7.6 and Figure 7.6) which is generated by calculating a similarity matrix (M) for each of the community detection algorithms. A cell in this matrix, $M_{alg1,alg2}$ quantifies how similar two partitions of the network are. Using Multidimensional Scaling (MDS), the similarity matrix is projected into two

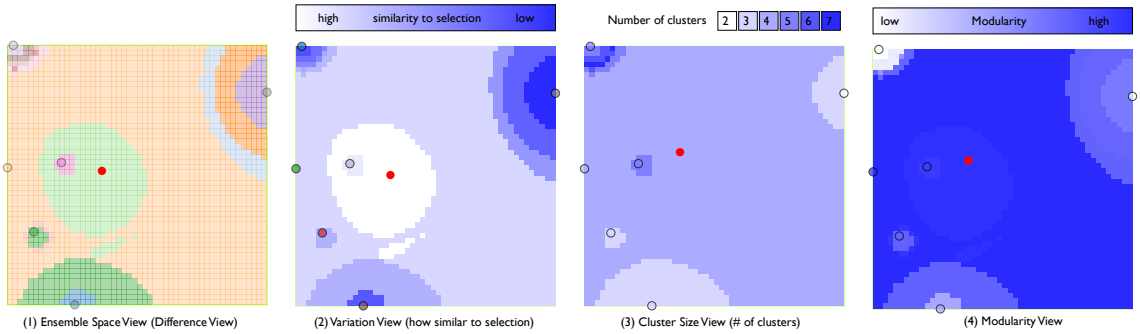

Figure 7.6: Four different views for the ensemble space heatmap. (a) Difference view: each differently colored region represent a unique ensemble output, (b) Variation view: shows how similar (white or light blue) or different (darker shades of blue) different ensemble outputs are compared to the current ensemble (red dot), (c) Number of clusters view: shows how many clusters are there in each ensemble output and (d) Modularity view: shows the relative modularity scores of different ensemble outputs.

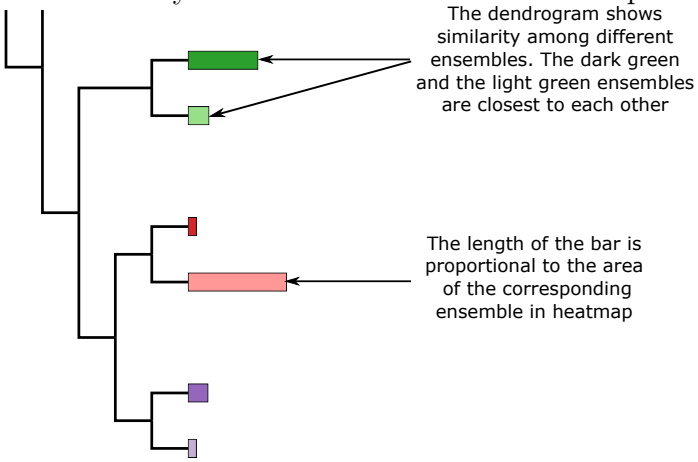

Figure 7.7: A small portion of the Dendrogram visualization. The dendrogram shows hierarchical similarity between different ensemble outputs. For example the dark violet and the light violet ensembles are more similar compared to the pink and the dark violet ensembles. The length of the bar corresponding to an ensemble represents the area of the region covered by the corresponding ensemble in the ensemble heatmap.

dimensions. Thus, algorithms generating similar partitions appear close together in this projection (represented as colored dots). From this projection, an ensemble is built for each discretized cell in the heatmap (we describe this metric space in more detail below). Because different ensembles may still produce the same partitioning, there are far fewer ensembles then there are cells, leading to the contours that can be seen in Figure 7.6. This view is interactive and can control many of the other panels (e.g., clicking on the point for the 'fastgreedy' algorithm will load that partition into the network view as will clicking on any other point in the space). The colors in the heatmap can also be set to encode different features of the space including the distribution of cluster sizes.

The second Ensemble Panel visualization is the **dendrogram view** (Figure 7.1(3) and Figure 7.7). The view shows the relationship between ensembles based on hierarchical clustering of their similarities. The length of the colored bar provides another view of the prevalence of each ensemble (proportional to the area in the heatmap view). As with the heatmap, the dendrogram is interactive (e.g., click to change ensembles or hover to highlights the point in the heatmap). This view enables the end-user to find related ensembles or vastly dissimilar ones and are worth exploring. Broadly, the ensemble panel allows the end-user to identify a partition of the graph that is near to their mental model and to support the exploration of alternatives when a classification is unknown.

A **Control Panel** (Figure 7.1(2)) allows the end-user to upload new files, add or remove different algorithms, and control different active learning properties. On occasion, it is helpful to compare two heatmap encodings (e.g., one capturing cluster size and the other capturing a partition 'score' such as modularity). The control panel allows the creation of a duplicate view for this purpose.

The **Network Visualization Panel** (Figure 7.1(3)) is a standard node-link diagram which displays the current 'working' partition. Layout is done through a standard force-directed algorithm. In the current implementation nodes are assigned colors based on which community they belong to. Interactive features include: display of labels, "collapsing" of communities to hide them (used when the end-user is satisfied with the nodes assigned to that community). Collapsing of the nodes has the benefit that completed 'work' can be removed from view to allow the end-user to focus on the rest of the graph. Communities are enclosed in a rectangle to double-encode the community structure. Through this view, the end-user can make corrections such as moving nodes between communities, merging communities, or creating custom labels. The network visualization, in part, is designed to support the identification of

'outliers' (i.e. nodes that might be unstable).

The end-user may also enable a **Co-community View** (Figure 7.1(5) and Figure 7.8) by clicking on a specific node. This view allows the end-user to find which communities a node could belong to as well as rapidly making corrections and engaging the active learning system. For example, in Figure 7.8, the end-user has selected "Actor 3" (this node appears in the agreement column). Initially, all other nodes appear on the right in one of the six columns. Column 6 includes all nodes that have been found to be in the same community as Actor 3 by all six algorithms (e.g., all six algorithms agree that Actor 14 should be in the same community as Actor 3). No nodes appear in columns 5, 4 and 3 as there is no consensus for nodes at this level. Column 2 contains a number of nodes that two of the algorithms would connect to Actor 3. Column 0 are all nodes that should *not* be in the same community as Actor 3 (by consensus of all 6 algorithms). The end-user can click on nodes to either explicitly indicate they belong with Actor 3 or explicitly should not be (these appear in the leftmost column with +'s or -'s next to them). Here the end-user indicated that both Actor 4 and 28 should be in the same community as 3. Once the end-user is done, clicking on a submit button causes active learning to relabel the network based on these new constraints. This view is also explicitly designed to allow the end-user to make decisions when they don't know where nodes belong but also to rapidly correct algorithmic mistakes.

7.5.2 Algorithmic Details

Below we describe specific details in generating ensembles and metric spaces that are visualized in the interface. Much of the computation is done at the server level as the calculation of communities and ensembles can be parallelized.

Algorithmic Inputs: For the CommunityDiff prototype, we chose six popular community detection algorithms: FastGreedy [147, 42], InfoMap [163], Label Propagation [159], Multilevel [136], Spinglass [160] and Walktrap [153]. These algorithms are some of the most popular community detection algorithms in use today and capture a broad range of objective functions and underlying techniques. Note again, that nothing prevents the addition of new algorithms.

Fastgreedy, as its name suggests, greedily merges communities iteratively by maximizing modularity, a measure of "modular strength" of a network. Modularity, Q is captured as:

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v, c_w)$$

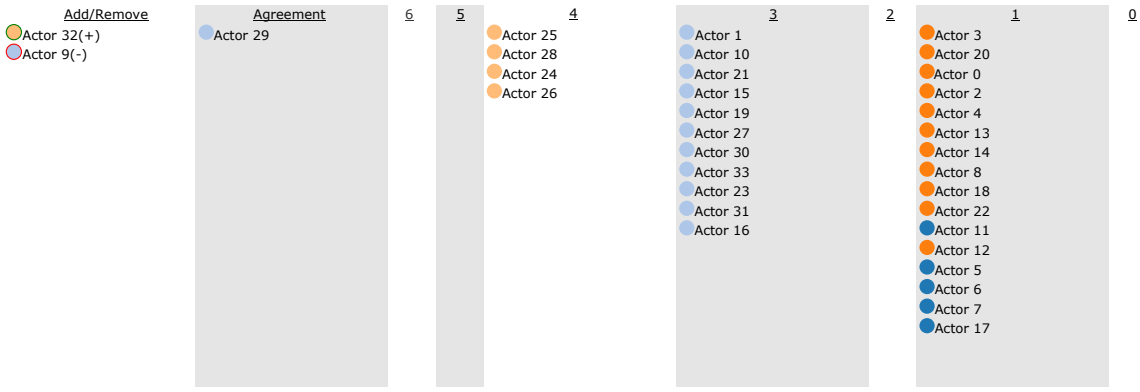

Figure 7.8: The Co-Community Lists: The node in the 'Agreement' column shows whose co-community lists are being shown. For the columns '0' to '6', column j shows that j of the initial community detection algorithms agree that the nodes shown in column j should be in the community of the node in 'Agreement' column. The 'Add/Remove' column shows which nodes the user thinks should be added to the community of the node in 'Agreement' column (depicted with '+') and which nodes should be removed (depicted with '-').

Where v and w and two nodes, k_i is the degree of node i, and c_i is the community label for node i, m is the total number of edges in the graph, A is the adjacency matrix representation of the graph (i.e. $A_{vw} > 0$ if an edge exists between v and w), and δ is the Kronecker delta — an indicator for testing if the communities are equal. The intuition for this function is that we are testing the number of edges within a community versus the number expected with random assignment. Stated differently, a strong community contains more edges between its members than expected by chance.

InfoMap, on the other hand, follows a very different approach, and aims to provide the shortest description length of a random walker trajectory. The description length is measured by the expected number of bits per vertex to encode the random walk path. This algorithm uses the minimum description length principle in information theory and follows the idea that a random walk within a community is likely to stay within the same community as the number of intra-community edges are higher compared to the number of inter-community edges.

The remaining four algorithms have additional variability and we refer the interested reader to the source publication. While the algorithms tend to agree on high-confidence communities, in practice, they generate enough variety in identified communities that they are a good mechanism for building ensemble spaces. All six

algorithms can operate over weighted networks (a common requirement in network analysis).

7.5.2.1 A Metric Space for Ensembles

In order to define a 'space' for our ensembles, it is necessary to define a suitable metric to represent relative distances between different partitions of the graph (i.e. the outputs of two or more community detection algorithms). Ideally, the distance metric will be bound within some fixed range, preferably between zero and one. We are equally satisfied with a dissimilarity metric as one that measures similarity (we can simply subtract the value from 1). As a specific example, if two partitions agree on every node pairing (i.e. they are in the same community or are not) the metric should return zero. On the other hand, if the two algorithms disagree on every pair we would expect to find the distance to be one (normalized).

The most popular metric for this purpose in the community detection literature is Normalized Mutual Information (NMI) [179]. Let X and Y be two arrays which denote the community partitions determined by two different community detection algorithms on the same network. The NMI for two partitions X and Y is calculated by determining the entropy for the two partitions (i.e. $H(X)$ and $H(Y)$) and mutual information, $I(X;Y)$ as:

$$H(X) = -\sum_{x \in X} p(x) log(p(x))$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log(\frac{p(x,y)}{p_1(x)p_2(y)})$$

The normalized for $(NMI(X;Y))$ is then calculated as:

$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

Roughly, we are capturing a partition as a probability distribution of a node falling into a community. The entropy reflect the information contained in this distribution and the mutual information is the shared information between the partitions. NMI lies between zero and one, and can compare two algorithms which produces different number of communities unlike some other metrics. The inverse, $(1 - NMI)$, can be used as a suitable distance metric as it satisfies all the properties discussed above.

Given a graph G and a set of community detection algorithms $CD = \{cd_1, cd_2, ..., cd_n\}$, let op_i correspond to the set of clusters obtained by running cd_i over G. We calculate the distance of op_i for each of the outputs produced by the different community clustering methods, so that for each method, we have an n-dimensional distance vector where n is the number of different community detection algorithms.

7.5.2.2 Multidimensional Scaling (MDS)

For visualizing the relationship between the outputs of the six algorithms we would like to project them into a 2D plane where their distances in this plane is proportional to their (dis)similarity. We have opted to use MDS [45] for this purpose as our qualitative experience is that it is effective in this context (fast and accurate). Other dimensionality reduction techniques (e.g., PCA, t-SNE, etc.) can also be used. By projecting the algorithms into a 2D plane we are now able to create ensembles that combine the algorithms by using euclidean distance to generate a set of ensemble weights (i.e. how much each algorithm's 'vote' counts towards the ensemble).

7.5.2.3 Creating Ensembles

Having a suitable representation of the metric space (the MDS projection) enables us to automatically calculate an ensemble for each point in this space. Our objective is that any selected point in the MDS projection will have an associated ensemble (i.e. partition) that is similar to our original algorithms in proportion to the distance to those algorithms. For example, note the red dot in the leftmost panel of Figure 7.6. We would like the ensemble at this point to be very similar to the algorithm immediately to its left (the light purple dot) and very dissimilar from the algorithm at the bottom of the space (in blue).

There are many possible ways to combine different community detection algorithm to form an ensemble algorithm. Ideally, CommunityDiff would allow the end-user to select different ensemble algorithms. However, we have currently implemented one based on a variant of the 'co-community graphs' method, which also takes the original graph structure into account (leveraging our prior work [31]). An edge between two nodes in a co-community graph indicate that those two nodes belong in the same community. The absence of such an edge indicates that the two nodes are not in the same community. Within the context of ensembles, the weight on an edge between any two nodes is determined by the number of algorithms that put them in the same community (e.g., if 3 of 6 algorithms place nodes A and B together the edge between them has a weight of .5). With one algorithm, the co-community graph will contain a number of disconnected components (each component capturing one community). However, a co-community graphs that is generated by multiple algorithms will often contain only one connected component which will be extremely dense. The intuition is that with many algorithms, at least one algorithm will put each pair in the same community. To mitigate this problem we only retain co-community edges that were

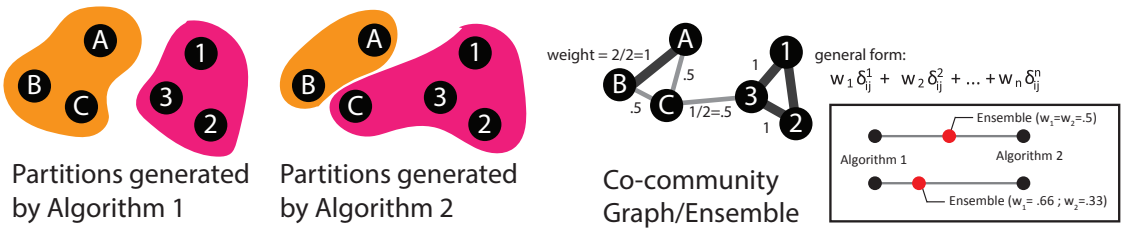

Figure 7.9: A graphical example of ensemble generation through the co-community graphs. Algorithms 1 and 2 generate two unique partitions: {{A,B,C},{1,2,3}} and {{A,B},{1,2,3,C}}. The co-community graph (right) has an edge between each pair of nodes that were in the same community (in any of the partitions). The weight of the edge is proportional to the number of times the nodes appear together among the algorithms (e.g., both algorithms 1 and 2 agree that nodes A and B should be together, hence a weight of 1; however, they disagree on node C, once placing it with A and B and once with 1/2/3). Each algorithm is thus a vote. When generating the ensemble, this vote is weighted by the position of the ensemble relative to the algorithms in the ensemble metric space (MDS). The inset shows two different weighting schemes based on ensemble placement (the weights must sum to 1; δ_{ij}^{n} is the Kronecker delta set to 1 if nodes i and j are in the same community for according to paritioning algorithm n, 0 otherwise).

also an edge in the original graph. The output is a graph that is no more dense than the original input graph but has co-community weights on the edges.

Briefly, to find the co-community graph for a given ensemble point (x, y in our MDS projection) we apply the following algorithm: given a graph G and a set of community detection algorithms $CD = \{cd_1, cd_2, ..., cd_n\}$, where c_i has weight w_i, first run the algorithms over G. Let w_i be the distance of the ensemble 'point' (x, y) in the metric space to algorithm i (recall that each algorithm is placed at some point in the MDS projection). Intuitively, if the ensemble point is close to i's position, then w_i is high (this is determined by the inverse of the Euclidean distance). For each edge e joining vertices v_i and v_j in the network, initialize weight of e as zero and if v_i and v_j are put in the same cluster by cd_i, then increase the edge weight by w_i. The total sum of weights for all community clustering algorithms is normalized to 1. For an edge e joining vertices v_i and v_j, if very few algorithms put v_i and v_j in the same cluster or the algorithms that put v_i and v_j in the same cluster have low weight (we do not want the ensemble algorithm output to be similar to result of these algorithms), e has a low weight (see Figure 7.9 for a graphical illustration). We prune these edges

by removing edges below a certain threshold to improve the ensemble output. A low threshold would not improve the results by much and a high threshold decomposes the network into too many connected components. As we normalize the clustering algorithm weights, the edge weights are restricted to range between 0 (no algorithms put the vertices joined by the edge in the same cluster) to 1 (all algorithms put the vertices in the same cluster). In practice, a threshold value of 0.33 does not over-split the graph and improves the clustering results. Note that, for different networks, different threshold values may improve the results further, but in general, our chosen threshold value works on a wide variety of networks.

Because the co-community graph with multiple algorithms does not properly disconnect components we run one final community detection algorithm on the co-community graph to split it into communities. Our experience is that InfoMap works most effectively for the kind of data produced by co-communities. Other algorithms, like Louvain or Walktrap, can also be used for the final run. In principal, any community detection algorithm can be chosen as long as the algorithm performs reasonably on the original graph.

As we described above, not every ensemble will be unique. Different combinations of algorithms can still produce the exact same output. In CommunityDiff, each *unique* ensemble receives an identifier. In most cases, the same ensemble will be appear contiguously in the ensemble space resulting in the contour plots seen in the figure.

From the perspective of implementation, though it would be possible to generate an ensemble for every pixel in the 2D-space this is somewhat impractical due to computational costs. It is also unlikely that ensembles vary dramatically at those scales (a shift of one pixel does not cause weights to change enough to impact the final ensemble). For CommunityDiff, we perform a grid-search on a discretized grid. Specifically, we assume a 300×300 space and use a non-overlapping grid of size 5×5. Ensembles are only computed once per grid.

In the control panel the end-user can eliminate certain algorithms from consideration in the ensemble space. Conversely, they can use the current ensemble (manually generated or through active learning) as a new "anchor" point in the metric space. This partition is treated the same way as the output of the other algorithms when calculating the map (a new dot is added to the heatmap for this ensemble).

7.5.3 Visualization and Interface Elements

We have selected a number of visualizations and interactive elements to support the end-user. We have opted for largely conventional forms (e.g., trees, heatmaps, node-link diagrams) as these are familiar and effective for the task.

7.5.3.1 Ensemble Space Heatmaps

Once created, our ensemble space heatmaps can be used for a number of tasks related to the different design objectives we defined. We propose four different kinds of views of the metric space to help the user choose a suitable algorithm or ensemble. These different views align with our first design guideline as they allow fast and accurate selection of a desired algorithm or ensemble. When clicking on the heatmap (in any configuration) the network diagram is dynamically changed (this clicked on point becomes the 'working' ensemble).

Difference view (Figure 7.6(1)): This is the most general static view of the metric space. Each unique ensemble has a unique (orthogonal) color. Though rare, it is interesting to note that two separate (i.e. disconnected) regions in the metric space may produce the same clustering (visible because they have the same color). The goal of this view is to help the end-user identify the 'jumps' in the initial labeling phase (see the first big 'jump' in Figure 7.3).

Variation view (Figure 7.6(2)): This view is the only dynamically generated heatmap. When the end-user clicks on a specific ensemble point, all other grid points are colored by comparing their associated ensemble to the current selection (using the NMI metric described above). In the figure, the end-user clicked to place the red dot (the selected ensemble). The white region surrounding that point has a distance of zero (meaning high similarity). The further one moves from that point the more dissimilar the ensembles get. This allows the end-user to identify other, subtly-different ensembles, but also ones that are radically different. This view supports our design guidelines in helping resolve knowledge gaps and identifying outliers.

Cluster Size View (Figure 7.6(3)): Recalling the 'mental model guideline,' domain experts often have a sense of the number of clusters they should expect to see. This view plots the number of clusters detected by the ensemble at each of the grid point. The end-user can readily choose ensembles or algorithms that have the same number of clusters as the goal or close to the number of clusters of the goal.

Modularity view (Figure 7.6(4)): There are a number of measures of community 'quality.' Modularity is one that is often applied. This view depicts the modularity

score for each ensemble cell in the space. This allows easy choosing of high modularity ensembles or confirming that the chosen ensemble produces high modularity output. In the future, we would like to add additional such metrics. As with the variation view, access to known metrics (i.e. modularity) supports decision making by the end-user.

The goal of all these views is to support exploration within the ensemble space to find a good starting point for additional labeling. This corresponds to allowing the user to see, and decide between, possible starting points in the blue 'ensemble space' in Figure 7.3.

7.5.3.2 Dendrogram

In the construction of the heatmap representation of the ensembles we noticed that it was often difficult to understand how common different ensembles were (in particular those that were not large or had odd shapes). It was also difficult for end-users to understand the relationship between all the different ensembles in the heatmap. While algorithms were situated in the metric space so that they could be compared, the contour 'blobs' for the ensembles were not so easy to interpret (i.e. how does one easily measure the distance between two irregularly shaped contours in the metric space?). To better support these comparisons we generated a dendrogram (see Figure 7.7). The view is a depiction of a standard agglomerative hierarchical clustering (using the NMI based distance metric). The size of the bar at the end of each leaf indicates how prevalent that ensemble is. The heatmap and dendrogram views are linked so that brushing over one highlights the other. Clicking on one changes the ensemble in selection in the other.

As with the heatmap, the dendrogram was intended to support the rapid exploration of alternative ensembles that could act as starting points for further labeling. As a second benefit both allowed the end-user to understand the stability of different ensembles in making decisions. The end-user could act based on their knowledge (e.g., number of clusters) but receive useful information to guide their exploration when their mental model was incomplete.

7.5.3.3 Node-Link Diagram

The network diagram view (see Figures 7.1(3) and 7.11) is a standard implementation using a force layout [103]. To incorporate community groupings we add additional constraints such that all nodes from a community are grouped together

inside a rectangular bounding box. Each bounding box (along with the nodes inside them) is colored differently. We would like to ensure that there is color stability for the communities between ensembles. More explicitly, we would not want to disrupt the end-user by randomly assigning colors every time a new ensemble was chosen (most communities are stable between these ensembles and a large, random color change would make a small change appear big).

We applied the following algorithm to achieve this stability. For each node we determine a stability score: In a graph, G, for each edge, e_{ij}, connecting two vertices v_i and v_j, we weight the edge by the number of community detection algorithms among the initial six that put v_i and v_j in the same cluster. For each node v_i, we assign a stability score k_i, where k_i is the weight of the maximum weight edge connected to v_i. Put another way, we ask: if edges were deleted based on a threshold from low to high, when would the node become disconnected from all others? In this case, the most stable nodes have a stability score of six as we are using six community clustering methods. Each node is assigned a color initially. A community takes on the color of the most stable node in the community (ties are broken by lexicographic ordering). However, even with the consistent colors, it is often difficult for the end-user to track changes when they switch between ensembles. To better support mental model preservation, nodes that change color during a different community assignment are briefly highlighted by changing their stroke width (i.e. outline).

Calculating the stability score has an additional benefit that stability is inversely proportional to ambiguity. By resizing the node based on this measure we can guide the end-user to nodes that would help the active learning infrastructure. The end-user can also make corrections in this view by selecting multiple communities for merging or clicking on nodes to reassign them to alternative communities. Communities can also be hidden from view to eliminate distraction.

7.5.3.4 Co-Community View

While dealing with real-world networks, in most cases the user only hsa partial knowledge about communities in the network. In majority of cases, this knowledge comes in the terms of which nodes belong together and which nodes do not. Co-community lists present an easy way to incorporate these knowledge. These lists can be viewed by shift clicking any node in the network diagram. The first list is the *Add/Remove* list which is initially empty. The next list the shows the node clicked on the network diagram, which is hereby referred as the focus node. The next seven lists are generated based on how many community detection algorithms put the other

nodes in the same community as the focus node. So, a list marked as *5*, contains the nodes that are put in the same community as the focus node by five community detection algorithms. Note that, the number of these lists varies according to the number of community detection algorithms chosen by the user. Within each list, the nodes are sorted by the shortest distance from the focus node.

As described before, the Co-community View (Figure 7.8) allows the end-user to make rapid comparisons and corrections. It is intended to be used during the final phase of labeling when many small corrections must be made. The view supports the design guidelines of respecting the unknown (i.e. uncertainties) and the mental model. By focusing on one node at a time (this is often a central node to a community or a bridge that connects two communities), the end-user can quickly isolate both mistakes and "trends." For example, if many nodes of the same community appear in the 6 column (the high agreement) the end-user can (a) see this due to color coding, and (b) select all these nodes by double clicking to add them definitively to the focus node's community.

There are clearly scaling concerns with the co-community view (long lists of nodes are hard to maintain). However, we have experimentally found (see below) that true positives often appear in the columns 5 or 6 (high positive agreement) whereas true negatives (the bulk) are in columns 1 or 0 (high negative agreement) and can safely be disregarded.

Co-community view allows processing of each individual node compared to the focus node. We use ranked list to give idea about the 'goodness' of the cluster of the focus node. If the nodes in the lists with high positive agreement (column 5 or 6) are of same color as the focus node, then the cluster is stable. On the other hand, if these lists contain nodes with different colors then the focus node has strong connection to multiple communities or there is a split community mistake. We do not use any kind of aggregated charts because that restricts manipulation of individual nodes. In the agreement lists, nodes are sorted according to shortest distance from the focus node as distant nodes are more unlikely to be in the same community as the focus node.

7.5.3.5 Other Controls

CommunityDiff has a number of other basic features to support the analyst. These include undo operations to move back to previous ensemble states, file upload (loading in a new network) and downloads (retrieving the community labels). CommunityDiff operates on GML graphs, a standard graph descriptor language. Brief descriptions for these controls are provided below.

1. **Merge:** As discussed before, a common problem of many community partitions is over-segmentation, where a single community is divided into multiple smaller communities. We provide a simple remedy of this problem — a merge functionality. A user can click on the legend boxes in the network diagram to select corresponding community or click on a selected legend to deselect them. Community bounding boxes in the network diagram are highlighted when hovering over corresponding legends and vice versa. Clicking the *Merge* button merges selected communities.

2. **Hide/show communities:** Often a user wants focus on a particular set of communities in a network. By shift clicking on the legend boxes in the network diagram, a user can hide a community or show a hidden community in both the network diagram and the co-community lists. This helps focus on the communities the user is currently working on.

3. **Download:** We also provide a way to download community assignments if a desirable partition is achieved using a *Download* button. It downloads a text file containing node ids and their corresponding community assignment. If a partition is generated using an ensemble, we provide a way to download weights for different community detection algorithms using a *Download weights* button.

4. **Find ensemble:** We provide a fast and easy way to highlight an ensemble region nearest to the community assignment (based on NMI score) shown in the networks diagram panel, using the *Find ensemble* button. If there is an exact match, then the selected region is highlighted in green, otherwise in yellow.

5. **Undo:** Because CommunityDiff can constantly recompute communities based on interaction, it is possible that an action will lead to a non-desirable partitioning. We provide the end-user with the option to return to the previous state. This includes changing ensemble or a community detection algorithm, active learning, labeling a node, merging two or more communities and changing community assignment using the co-community lists.

In addition to the commands above, CommunityDiff also allows the end-user to explicitly rename communities, upload new files, or download the output of the analysis. Visualization specific interactions include zooming and panning, viewing and hiding node labels, and hiding/showing different visualization panels.

7.5.4 Human-in-the-Loop Machine Learning

Feedback from the end-user — by means of interaction — allows CommunityDiff to make dynamic alterations to the partitions in anticipation of the end-user's actions. This again is designed to eliminate unnecessary labeling steps.

7.5.4.1 Co-Community View Processing

Within the context of the co-community view, once nodes are selected for either inclusion ("must link" to the target node) or exclusion ("most not link"), CommunityDiff propagates these decisions to help partition the rest of the graph (as per our guidelines, manual decisions — those made by explicit selections — are respected and retained). In the simplest case, if our target is node A and we manually indicate that node B should be in the same community and A and B are adjacent (i.e. connected by an edge) or B is adjacent to at least one node in A's community, B is placed in A's community. If B is not adjacent to A's community, we merge the communities of A and B (there can be multiple ways to tackle this problem, we opt for the simplest solution of merging the two communities as very little research is done in tackling this particular problem). When nodes are explicitly indicated to not be part of the target node's community (say node C), C's label will be assigned to the next most likely community that it is not "banned" from. If no such community is found, we create a new community for each connected component in the subgraph spanned by these nodes.

7.5.4.2 Active Learning

In the traditional sense, active learning asks the user to label specific data points and performs semi-supervised learning iteratively. In CommunityDiff, we do not explicitly ask users to label certain nodes, but we do highlight unstable nodes (larger in size) which could be manually labelled and provide useful information to the learner (a blend of mixed-initiative and active learning). Addressing this "question" will trigger the algorithm to re-partition the graph. However, any form of supervision (e.g., merging, splitting, etc.) will be taken into account.

A key background task for CommunityDiff is an adaptive classification component that can produce better communities given end-user behavior. CommunityDiff uses both constraints on the number of clusters and community constraints in generating new partitions. For the purpose of speed and efficiency (recall, this is one of our design goals) only nodes that can make a difference to the active learning

algorithm should be manually labeled. The uncertainty (and conversely, stability) metric we use was described above as a side-effect of community color generation.

When a learning step is engaged, CommunityDiff applies the following procedure: Assume k is the desired number of communities (indicated in the control panel). Clearly, the classifier can not produce more or less than the number of community labels then have been manually created. If there are k_1 manually labeled community labels, and $k_1 > k$, then the value k is adjusted upwards to k_1. The node labels of the labeled nodes are chosen from the community assignments of the network shown in the network diagram panel. First it is decided which labels should be selected. If there are k_1 labels among the manually labeled nodes, we choose these k_1 nodes. If $k_1 < k$, we choose the other $k - k_1$ labels using community sizes. We choose the labels of the largest $k - k_1$ communities not including the k_1 labels already selected. Apart from the manually labeled nodes, only the most stable nodes of the communities having the selected labels, are chosen as labeled nodes. All the other node labels are determined by a semi-supervised classifier by Zhu et al. [195]. A brief description of which is provided in the next section.

We choose the graph structure for the semi-supervised classifier based on the output of the community detection algorithms where we retain the original graph structure but each edge is weighted according to how many algorithms put the nodes connecting that edge in the same community. This is similar to choosing the co-community graph for a specific ensemble except the fact that each algorithm is weighted equally and there is no removal of edges based on a threshold.

We can think of shifting a node v from *community A* to *community B* as a form of soft assignment. We are only assuming that v has the same community assignment as the most stable nodes in *community B*, that means after the active learning, unstable nodes in *community B* may have a different community assignment than v.

7.5.4.3 Classifier description

In CommunityDiff we have used a modified version of the semi-supervised classifier by Zhu et al. [195]. This classifier employs Gaussian random fields and harmonic functions to predict node labels given only network structure.

Given a graph $G = (V, E)$ with $l+u$ vertices, we assume we have l labeled instances $L = \{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled instances $U = \{x_{l+1}, ..., x_{l+u}\}$. w_{ij} denotes the weight of the edge between i and j. If there exists no edge between i and j, w_{ij} is zero. We also assume the number of classes is C.

The aim of this classifier is to compute a function $f : V \to \mathbb{R}$ over G to label the

unlabeled nodes using f. Ideally, unlabeled points near each other should have the same labels (due, in part, to homophily [22]). f can be rendered as a Gaussian field with energy function,

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2 \tag{7.1}$$

The minimum energy function is harmonic in nature. That is, for labeled data, the value given by f matches the labels exactly and for unlabeled data, $\Delta f = 0$, where Δ is the combinatorial Laplacian defined as $\Delta = D - W$, where D is the diagonal matrix with the non-zero diagonal entries $d_i = \sum_j w_{ij}$ and $W = [w_{ij}]$ is the weight matrix.

However, the labels for the unlabeled data can be calculated using only matrix operations. The Laplacian matrix can be partitioned as follows,

$$\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix} \tag{7.2}$$

and f can be defined as a $C \times (l + u)$ matrix, such that,

$$f = \begin{bmatrix} f_l \\ f_u \end{bmatrix} \tag{7.3}$$

where, $f_l[i][j] = 1$, if x_i is labeled as j. The other entries of f_l are zero.

$f_u[i][j]$ denotes the probability that x_{l+i} has label j. This can be computed as,

$$f_u = -\Delta_{uu}^{-1}\Delta_{ul} f_l \tag{7.4}$$

For each unlabeled point the label with highest probability is chosen.

7.6 Evaluation

To evaluate CommunityDiff, we would like to demonstrate that individual components (e.g. the ensemble generation and co-community lists) are effective as well as demonstrating the usability of the overall system. Our hypothesis is that an ensemble will perform better compared to any individual algorithm and that the best ensemble is often the "largest" one. Similarly, we need to show that the co-community lists provide an idea about the 'goodness' of a cluster along with batch processing of selected nodes. We can verify both of these components via automated experiments where ground truth knowledge is available or can readily be inferred. To test the

end-to-end system we performed a controlled lab study.

It is worth noting that a key challenge of evaluating a new community detection algorithm or software is the lack of network data with known community assignments (ground truth). The handful of 'toy' datasets often used for this task are often too small to be useful (e.g., Zachary's Karate Club Network [190] or the Football Network [76]). Synthetic datasets such as those generated by the LFR benchmark [114] often create advantages for community detection algorithms that vanish with real networks [31]. To create datasets for automated evaluation, we apply a common technique which is to find unlabeled networks and create a ground truth label based on some feature of the nodes that is not structural (i.e. does not depend on edge configuration). While this is imperfect, it is nonetheless common practice for large-scale evaluation [151].

7.6.1 Ensemble Evaluation

In the DBLP co-authorship network [189], 846,082 nodes represent authors and the 2,783,165 edges are co-authorship relations. Authors are placed in communities based on their frequent publication venues. Authors who publish in the same conferences or journals thus form a community. As our focus in this work is on disjoint communities we utilized the DBLP data to construct a ground-truth label based on the *most* frequent publication venue (i.e. an author who publishes 10 times in venue X and once in Y will be labeled X). Using the DBLP data (`http://dblp.uni-trier.de/xml/`) we created our own DBLP co-authorship network with ground truth labels created as described above. As journals tended to be the longest running 'venues' (and had clear continuity) we only utilized articles published in journals for this analysis (yielding 1530 "communities").

To simulate a realistic use case for this data (e.g., a user organizing a sub-field for a review article) we generated random sub-graphs from the larger network. A node in the larger graph was chosen at random and a subgraph spanned by its 2-step neighborhood was collected. In this subgraph we ensure that each ground truth community was connected as all community detection algorithms make this assumption (i.e. there should be a path between each node in a community that does not require moving through another community). Additionally, we removed communities with only 1 or 2 nodes from each network and networks with a single community. We generated 100 such networks as a test dataset. The sizes of these networks vary from 50 to 150 nodes. We also made sure there is at least 2 communities in the network and disregard networks with 15 and more communities as in these cases the communities

are very small (3-10 nodes).

Each of the 100 networks was analyzed using the CommunityDiff backend. We were specifically interested in the partitions generated by each algorithm and a few key (i.e. "common") ensembles. To determine the accuracy of these partitions, we compared each to the ground truth labeling using the NMI metric. In three of the networks, all algorithms produced the exact same output (which was the ground truth). As our goal is to study those situations where ensembles can be used, we exclude these three networks from further evaluation. Recall our general hypothesis that ensemble methods better than base algorithms. Figure 7.10 clearly shows that the best ensemble outperforms all the other algorithms as it has the highest average NMI score with ground truth and performs best in 88 cases out of 97. One of the reasons for using the heatmap ensemble projection was the belief that an ensemble that is stable across most of the space (i.e. the largest and most salient) would be the best starting point for labeling. To test whether this was valid, we tested the most common ensemble in our ensemble space. We find that in 25 out of 97 cases, the most common ensemble performs better than any base algorithm. This is followed by spinglass which performs the best in 22 cases and InfoMap which is best for 18 cases. However, the mean NMI for InfoMap is much lower and more variable. In those situations when the most common ensemble is not the best partitioning, the best ensemble is found within the top-5 most common ensembles.

7.6.2 Co-Community Evaluation

To determine whether the Co-community View would correctly emphasize nodes, we performed a second experiment to test whether nodes listed in the top columns of the Co-community View were likely to be good additions to a selected node's community. Specifically, given a "target" node, we would like to know if those nodes displayed in high-agreement columns (i.e. the nodes for which all or all but one algorithm agree that they should belong to the same community as the target node) are likely to be in the same community as the target. To verify this, we chose 10 random networks from the 100 sampled DBLP networks and for each network chose 10 random nodes and studied their co-community lists manually. For 3 of the chosen networks we find that the different algorithms give widely different results and for most of the 10 randomly clicked nodes (for all 3 networks) the high-agreement columns (those with 5 to 6 algorithms agreeing on a community) are empty. We disregard these networks for this study. For the 7 remaining networks, We observed that in 59 out of 70 instances, all nodes listed under list 6 and 5 (i.e. were agreed upon by most

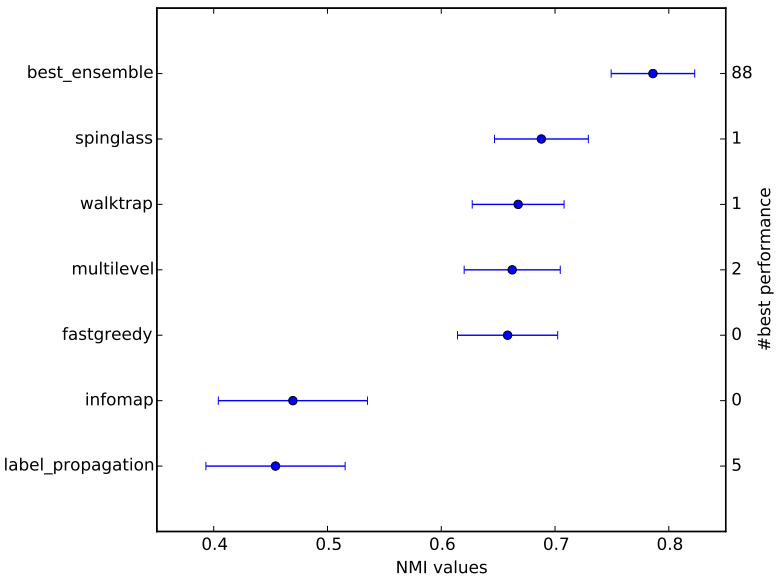

Figure 7.10: Comparison between average NMI values with ground truth of different community detection algorithms and number of times an algorithm performs the best. The figure shows that the best ensemble output has higher average NMI score and performs the best among all algorithm in 88 out of 97 cases.

algorithms) were in the ground truth community of the target node. We manually validated that there were multiple nodes in these lists in all 70 cases.

7.6.3 User Study

In order to reliably test CommunityDiff with end-users we needed to identify a network dataset where ground truth was available but where highly specific domain expertise was not necessary so that we could have a larger participant pool. To ensure high agreement we would also prefer a network where community labels are not subjective. For this purpose, we generated our own network for evaluating our system. We created a network by utilizing the handwritten digits from MNIST [119] (these are the digits 0-9 written by humans and used in OCR challenges). We cast this as a network problem by creating an artificial network where digits are the nodes and two nodes are linked by similarity between the two digit images. This approach is roughly equivalent to standard clustering where points near each other in the n-dimensional space are considered for inclusion in the same cluster. Though this may appear artificial as the data itself is not structured as a network, this transformation for classification and clustering purposes is common in text clustering (e.g, [63, 139])

or recommender systems (e.g., [94]) among other domains (where edges have the meaning "is-similar-to"). Regardless, it allows us to generate a network that satisfies our goal of reduced domain expertise and high agreement.

To create the network, we randomly sampled 30 images each of the digits 0,2,4,6 and 7 (these have high inter-rater reliability in ground-truth labels). We calculate pairwise similarity between images using a Gaussian kernel and only retain the top 5% values as edges. We took the largest connected component of this graph (117 nodes and 554 edges) as the test dataset. When displaying these nodes in CommunityDiff, we included a small image of the handwriting sample on top of the usual circle. Thus, we dot explicitly provide the ground truth and allow the users to decide on their own which nodes should belong together based on the images. Figure 7.11 shows a screenshot of the MNIST network in CommunityDiff.

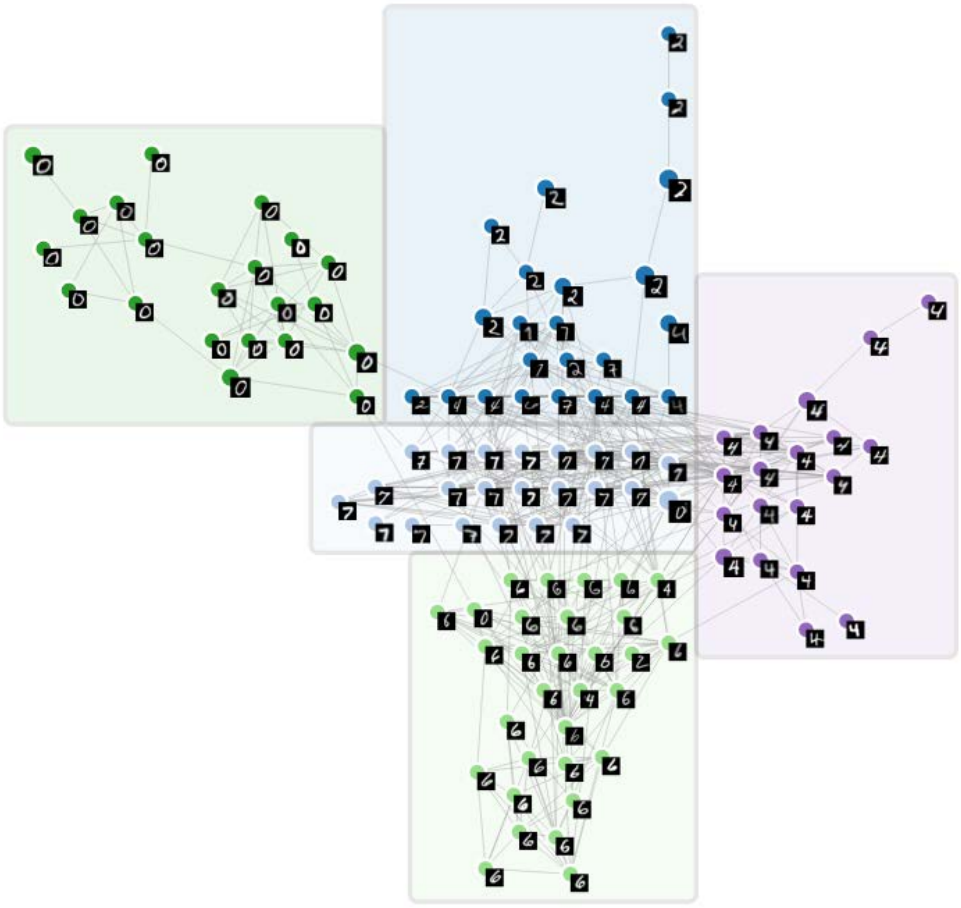

Figure 7.11: A screenshot showing MNIST handwritten digits network in CommunityDiff. This screenshot does not reflect the ground truth partition of the network as all communities except communities containing '0' and '4' has multiple different digits as nodes.

Our subjects included 12 graduate students at a large University (11 with experience in using network datasets including online social networks, linguistic networks like word networks and email communication networks) from 3 different departments namely, computer science (5 students), school of information (6 students) and mechanical department(1 student). All participants were given basic training with CommunityDiff (to familiarize them with different components and commands of the system) and handed a cheat sheet to remind them of commands. Subjects were then asked to achieve optimal partitioning of the handwritten digit network as quickly as possible. The study was conducted using a laptop computer (this computer is only used for display and browser, the code was running from a Linux server) in an office setting.

Each participant was able to achieve the correct clustering (5 clusters, each cluster contains images of a single digit) within 10 minutes in their first attempt with the fastest one achieving it in 3 minutes. We logged all atomic operations (i.e. "moves") made by the subjects. The number of operations needed to get to the final output varied from 5 to 24.

As we do not have a competing system to compare, we compare performance of different users who used different features of CommunityDiff to reach ground truth to validate the usefulness of different features. Four of the participants got to the final clustering output using only merging and manual labelling from the network diagram but on average (mean) they performed 20.25 operations which is much higher than participants using at least one other feature of CommunityDiff (on average 14.63 moves). Similarly, users who made use of the ensemble map made 13.86 moves on average (7 users) whereas those who did not, made 20.2 moves on average (5 users). We believe this demonstrates that the use of different features results in more efficient solutions. However, additional testing with more subjects and with features explicitly disabled would allow us to better isolate the key features that improve efficiency but we leave this to future work.

Anecdotally, we also found that users perform much better in their second attempt. Two participants volunteered to perform the task a second time and they produced the final output within 5 and 7 operations (though part of this may be because they were familiar with the data). These attempts were not included in the study but they establish that with experience, users may learn more efficient solutions.

In a post study survey users indicated that there was a benefit to the different views of the heatmap and liked the active learning and flexibility of the tool. Some users felt the initial interface was overly complex and some actions were not intuitive enough.

Some of these critical comments are: "Complexity needs to reduce, esp. the initial interface. Maybe put an 'advanced' section." , "Make the selection/moving gesture more intuitive, such as allow drag-and-drop." This is good advice for future iterations of the system. As a response to what they liked most using CommunityDiff, subject comments included: "Different views/metric to decide a partition", "Active learning (it allows the user to adaptively choose number of clusters)" and "Many different ways to manipulate nodes and communities."

7.7 Case study

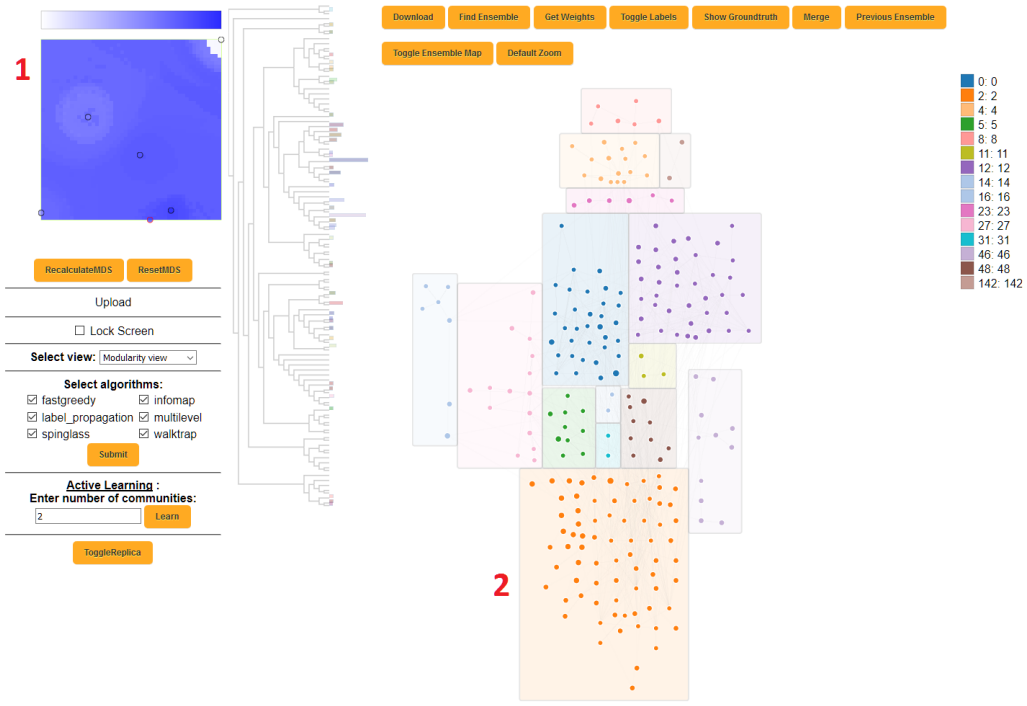

Figure 7.12: A screenshot showing 2016 Reddit co-conflict network in CommunityDiff. 1 denotes the modularity view which shows the selected algorithm, multilevel produces a high modularity output. 2 denotes the political discussion community where there are a lot of large nodes depicting poor community structure.

We analyze the co-conflict network (discussed in chapter V) and its communities using CommunityDiff. Figure 7.12 shows the multilevel output of the network. Looking at the difference view and the dendrogram we can conclude that the community structure is not very robust as different algorithms disagree with each other a lot. The modularity view reveals that multilevel and fastgreedy algorithms perform

the best. We observe that node sizes in the network diagram (especially in the political discussion community) vary a lot which means that the algorithms are uncertain about community assignment for a lot of nodes. We also observe many very small communities for some ensemble outputs. Based on the number of clusters and the modularity score, we chose the multi-level algorithm for our experiment.

Focusing on specific communities, we observe that some communities have topically cohesive structures. The co-community lists of the subreddit *Firearms* (Figure 7.13) reveal that all algorithms put *gunpolitics* and *shitguncontrollerssay* in the same community as *Firearms* and 5 out of 6 algorithms put *Austin* in the same community. This points to a robust community structure among these pro-gun subreddits.

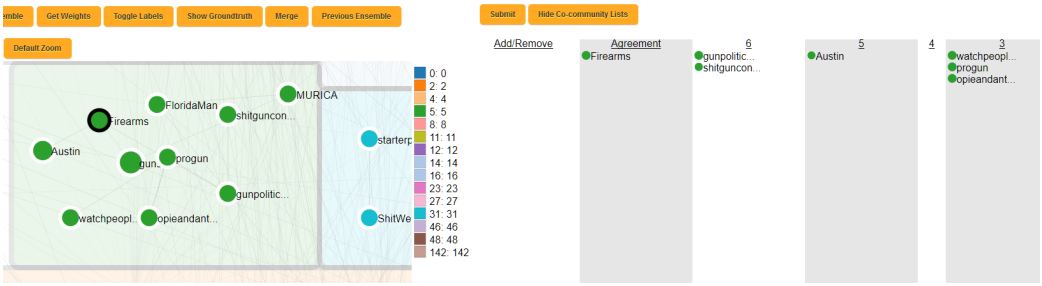

Figure 7.13: The *Firearms* community in the co-conflict network.

7.8 Discussion

We believe that CommunityDiff effectively supports the analytical process of community detection. The system provides explicit mechanisms to "jump" to a reasonable set of labels/partitions. This jump, however, is done with consideration of alternatives. By allowing for rapid comparison of different algorithmic outputs and ensembles we believe that end-users can gain more confidence in their choice and are less likely to be biased. A second set of interfaces (the Network and Co-community View) provide a mechanism for quickly making decisions and corrections to complete the task. Underlying the whole process is a classification sub-system that leverages the end-users manipulations to continuously improve the partitioning. By subtly guiding the end-user by varying visual salience, CommunityDiff can guide the end-user to focus their attention on those decisions that can make those most difference. Taken together these components solve different work 'modes' for this task. Though they can function independently, they are designed to work well within this multi-step process.

The workflow described for community-detection is one example of many that require decision making among many alternative algorithms. While ensemble spaces can be applied in any situation where a metric is available for comparing algorithm output, the broader design of CommunityDiff may also be portable to other problems. Our hope is to continue to develop the idea of CommunityDiff and to test it in other contexts.

The current prototype of CommunityDiff does have some limitations. The most evident one is that it is most effective for smaller networks as response time of each interaction goes up with network size. Visual representation of very large networks is a hard problem in general. However, in the case of CommunityDiff, a more critical issue is that the algorithms we implement are computationally intensive (e.g., creating 1000s of ensembles). We have not experimentally identified the 'breaking point' for the tool but have found that with graph sizes of 300 or fewer nodes (an 'ego-net' sized network), CommunityDiff can pre-compute all ensembles in under a minute. With networks of 1500 nodes, the lag is more pronounced but once pre-computation completes the graph can be loaded and used at interactive speeds. It is worth noting that many real world networks sizes fall into this range (social networks with average 150 individuals, protein-protein interaction network with average 1300 nodes [188]).

Disabling some of the high-cost community detection algorithms like spinglass (something one would do regardless of the visualization tool) allows use of larger graphs. Interestingly, a majority of computationally intensive operations can be parallelized. We currently support parallelization with respect to 10 threads in precomputation of the ensemble space heatmaps which provides better response time for initial processing of a graph. It is worth noting that, the ensemble technique and the classifier can be used on much larger graphs if we do not care about the response time or the visualization aspect.

Similarly, CommunityDiff is limited on how many communities it can handle efficiently. We have found that the interface becomes increasingly cluttered if we have more than 50 communities. However, as the number of communities in a real world network are usually far less compared to number of nodes, we usually see the scalability issues with respect to graph size pose more of a problem compared to scalability with respect to number of communities. For example, a social network with 150 nodes has around 4 communities whereas protein-protein interaction networks with about 1300 nodes has about 40 communities [188] (these are handled well by CommunityDiff). To further address the scaling problem, we have begun to design algorithms that adaptively fill in the ensemble space by proceeding from coarse to

finer grid sizes dynamically.

Visually, CommunityDiff is constrained by limitations of force-directed node-link layouts, a modification of which we use in our network diagram. For networks with more than 1500 nodes, the node-link diagram becomes too cluttered to be useful. A similar problem is present with co-community lists as with number of nodes these lists can become very long. However, an user is likely to work with only a few communities at once, so we provide a 'Hide communities' button to hide communities and all corresponding nodes from the network diagram and the co-community lists to help the user focus on the task at hand with an uncluttered interface.

Though we believe the node-link diagram is an effective (and familiar) tool for visualization there are significant developments in group-structure visualization that may yield better results in the future [181]. This may be more critical if CommunityDiff begins to support other community structures. Currently, CommunityDiff works on undirected input graphs and generates only hard partitions (rather than overlapping communities). As discussed above, overlapping community detection algorithms are still inferior compared to the disjoint algorithms and unpopular in many domains. Though this limits the tasks for which CommunityDiff is currently suitable for, we believe it can be adapted for additional inputs and outputs. For example, if we just use overlapping community detection algorithms as base algorithms of CommunityDiff, ensemble space heatmaps and co-community lists would retain most of their functionality. However, we would need efficient visual representation of overlapping communities in a network and a classifier that could work with overlapping communities (in active learning) to make the whole pipeline work. Effective visualization of networks with overlapping communities is a hard problem in general, but solutions do exist for this task (e.g., [7]). Similar modifications may be necessary to the Co-Community lists with additional commands specifying how many communities a node should belong to. Finding a network classifier resulting in overlapping communities is a harder task. However, the semi-supervised classifier used in CommunityDiff generates probabilistic community assignments which could be used for assigning multiple communities to a node. The co-community lists are an area for possible improvement. The lists function well in our examples, but because the underlying data is networks and sets, alternative encodings may be more effective [8].

In the future, we also want to further evaluate the system by using multiple real-world networks and varied domains for the purpose of ecological validity. We intend to deploy CommunityDiff for others to use and hope to collect information on how it functions in real-world contexts. This type of evaluation would also help determine

if certain features are more useful for certain kinds of networks or certain types of users and would give us a sense of performance of different algorithms on specific kinds of networks.

We believe that the idea of using differences of algorithms to effectively capture the analytical process of community detection can also be applied to similar real-world problems with no ground-truth and varying outputs (depending on algorithm used) including outlier detection and clustering.

7.9 Conclusion

CommunityDiff is a novel tool supporting end-users in the construction of community-labeled networks. Though there are many network analysis tools and packages, one of the most common features — the efficient production of accurate community labels — is often lacking. The noisiness of networks, variable performance of community finding algorithms, and incompleteness of knowledge all factor into the challenging nature of this problem.

With CommunityDiff, we demonstrated a set of features that address many of these concerns. We showed how an ensemble space could be created from competing algorithms that often out-performs single algorithms. Visualization of this space allows the end-user to make easy comparisons between different options. Focused visualizations further allow the end-user to view "votes" in a co-community graph, and allows the end-user to rapidly make decisions and correct system mistakes. As the user makes direct manipulation actions, CommunityDiff improves the partitions thus boosting labeling efficiency. By 'nudging' the user, through visual cues, CommunityDiff can suggest high-value feedback for the end-user to concentrate on. The overall system thus satisfies the needs of end-users by combining algorithms and visualizations. Though CommunityDiff focuses on community detection, we believe that many of the techniques and guidelines described here — including the idea of differences of algorithms — are portable to other data mining problems where end-user interactivity is valued and/or necessary.

CHAPTER VIII

Conclusion and Future Work

8.1 Summary

By focusing on differences between different online communities, via contrasting different similarity measures or user behavior within these communities, we gained insights on community relations, hierarchies and conflicts. We also show this idea can be applied to algorithms to improve resulting ensemble algorithms, to better understand specific results and to isolating specific instances for human labeling. We applied our idea to understand subreddit interactions and conflicts. Although our analysis of community-to-community relations is focused on Reddit, this idea is applicable to other social network platforms with community structure as well (e.g. Facebook pages, Twitter hashtag communities, news forums). We perform much of our community-to-community analysis via inferring a set of graphs with the same set of nodes, in our case subreddits, but different sets of edges based on different measures. These node-aligned networks reveal the multidimensional nature of relationships among online communities and help us gain a deeper understanding of these relationships.

When contrasting author overlap and textual similarity between pairs of subreddits, we not only uncover interesting relationships between pairs like topic and community fragmentation, communities at ideological war and hierarchical community structure, we also discover the identities of these communities themselves. Based on outgoing and incoming links, we can identify general and special focus subreddits, mainstream and marginalized online populace. We also find interesting misaligned communities which share a lot of similarities due to being part of a larger group, but the subgroups are different enough to maintain there own identities (e.g. different sports subreddits related to basketball, soccer etc. form a misaligned community which represents the overall sports subreddit type). This chapter gives a general

pipeline for analyzing any social media where there is an idea about online communities who interact with each other. The pipeline is not restricted to only author overlap and textual similarity and can be extended to any number of different similarity measures provided the data is available. Uncovering these relationships are useful for understanding how online communities interact, recommending similar communities, identifying marginalized groups automatically and identifying differing viewpoints on the same topic.

Contrasting user behavior in subreddits identifies inter-group conflicts. This also enforces the notion that some users behave according to their (online) surroundings i.e. which subreddit they are currently posting. Instead of specific case studies, we extract the complete conflict map in Reddit using 2016 comment data and find out which subreddit pairs are in conflict with each other during that time. We identify some general properties of these conflicts like their high reciprocity. We observe that several banned subreddits rank very high in a couple of subreddit-level features that we use to characterize conflicts. We find that these features can be used to implicate communal misbehavior in a more comprehensive study.

To study banned subreddit features and the efficacy of the above-mentioned interaction features, we first collect a dataset of 1060 subreddits along with their content. These subreddits were active in some period from 2010 to 2017 and have at least 100 comments. This is the largest dataset of its kind to date. Through both quantitative and qualitative analyses we find subreddit ban reasons are extremely varied, though they fall into one of three main categories, internal, external and meta reasons. We find that subreddit bans are clustered in time and by reason. We cluster and predict these subreddits using both textual and interaction features and implement a banning-by-example schema to identify banned subreddits of a particular type by using other subreddits of that type as examples. This schema is helpful for community moderators to study misbehaving communities in any social media or forum websites.

Finally, we apply the idea of extracting insights from differences to a set of community detection algorithms where the result can vary widely depending upon input or the algorithm without any knowledge of the best output. In these cases, we create a visualization tool, CommunityDiff to show how different these algorithms are and use different weighted combinations of these algorithms to identify the most stable output. We also identify cases where human input is warranted by looking into nodes where the algorithms do not agree about the community assignment. We created a system to incorporate user knowledge (number of communities, specific groupings) to achieve a desirable solution.

8.2 Extensions

Our general pipeline for comparing and contrasting different similarity measures among different entities is applicable to various different entities including online communities, user behavior within a community and even algorithms. The types of similarity measures are also not limited to author overlap and textual similarity. We can use any similarity measure depending on the entities in our pipeline. Another important aspect is that we can use the same similarity measure at different times to identify temporal patterns. For example, we can look at author overlap among subreddits in 2016 and 2017 to identify changes in common authors. This idea can be applied to other similarity measures including textual similarity as well. We provide an expanded pipeline which can be useful for many network analysis problems.

Aggregating user behavior over different communities to identify conflicts can be applied to any social media or news forums with overlapping community structure. This includes Twitter follower groups, Facebook pages, news forum commenter communities etc. We can use user metadata or graph alignment algorithms to infer communities and use our pipelines on social media where user communities are not explicit. An important aspect of identifying conflicts using up/downvotes is that the procedure is content-agnostic, which means that we do not need to rely on natural language processing to detect these conflicts. This is very useful for detecting conflicts where the content is in a language other than English. In absence of measures like up and downvotes, we can use content dependent techniques like toxicity analysis to determine behavior.

As part of these projects, we collected a banned subreddit dataset with over 1000 banned subreddits. This dataset is largest of its kind to date and can be very useful for studying community-wide misbehavior and hate-speech. We provide a banning-by-example schema which is immediately useful for moderators who want to identify specific types of misbehavior or misbehaving communities. The usefulness of this approach is not limited to Reddit and all social media platforms with an inherent or derived community structure can benefit from it.

In general, looking at differences among outputs of different algorithms on the same input helps us determine which output is more stable and identify cases for human labeling. This is not limited to community detection algorithms and can be applied to any set of algorithms to improve ensemble output and active labeling for interactive machine learning.

Finding similarities between entities and algorithm is widely used but our aim

was to demonstrate that studying differences can be similarly beneficial as well. We hope that this dissertation will help guide research by introducing both methods and analyses about studying differences in online communities, algorithms and foster interest in studying differences otherwise.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211 – 230, 2003.

[2] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ACM.

[3] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ACM.

[4] Eytan Adar. Guess: A language and interface for graph exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 791–800, New York, NY, USA, 2006. ACM.

[5] B. Thomas Adler, Luca De Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West. Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II*, CICLing'11, pages 277–288, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] Rodrigo Aldecoa and Ignacio Marn. Exploring the limits of community detection strategies in complex networks. *CoRR*, abs/1306.4149, 2013.

[7] Basak Alper, Nathalie Henry Riche, Gonzalo Ramos, and Mary Czerwinski. Design study of linesets, a novel set visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2259–2267, 2011.

[8] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges. In R. Borgo, R. Maciejewski, and I. Viola, editors, *EuroVis - STARs*. The Eurographics Association, 2014.

[9] Saleema Amershi, Maya Cakmak, W Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105, 2014.

[10] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 21–30, New York, NY, USA, 2012. ACM.

[11] Nazanin Andalibi, Oliver L. Haimson, Munmun De Choudhury, and Andrea Forte. Understanding social media disclosures of sexual abuse through the lenses of support seeking and anonymity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3906–3918, New York, NY, USA, 2016. ACM.

[12] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 392–396, New York, NY, USA, 1999. ACM.

[13] Shrey Bagroy, Ponnurangam Kumaraguru, and Munmun De Choudhury. A social media based index of mental well-being in college campuses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1634–1646, New York, NY, USA, 2017. ACM.

[14] Sairam Balani and Munmun De Choudhury. Detecting and characterizing mental health related self-disclosure in social media. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 1373–1378, New York, NY, USA, 2015. ACM.

[15] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[16] Vladimir Batagelj and Andrej Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):47–57, 1998.

[17] Jason Baumgartner. Reddit data, 2017.

[18] M. Bazzi, M. Porter, S. Williams, M. McDonald, D. Fenn, and S. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, 14(1):1–41, 2016.

[19] Barry Becker, Ron Kohavi, and Dan Sommerfield. Visualizing the simple baysian classifier. In Usama Fayyad, Georges G. Grinstein, and Andreas Wierse, editors, *Information Visualization in Data Mining and Knowledge Discovery*, pages 237–249. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[20] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. *KNIME: The Konstanz information miner*. Springer, 2008.

[21] Prantik Bhattacharyya, Ankush Garg, and Shyhtsun Felix Wu. Analysis of user keyword similarity in online social networks. *Social Network Analysis and Mining*, 1(3):143–158, 2011.

[22] P.M. Blau. *Inequality and Heterogeneity: A Primitive Theory of Social Structure*. MACMILLAN Company, 1977.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[25] Stephen P Borgatti, Martin G Everett, and Linton C Freeman. Ucinet for windows: Software for social network analysis, 2002.

[26] Sebastian Bremm, Tatiana von Landesberger, Jürgen Bernard, and Tobias Schreck. Assisted descriptor selection based on visual comparative data analysis. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'11, pages 891–900, Chichester, UK, 2011. The Eurographs Association, John Wiley and Sons, Ltd.

[27] Ivan Brugere, Brian Gallagher, and Tanya Y. Berger-Wolf. Network structure inference, a survey: Motivations, methods, and applications. *ACM Comput. Surv.*, 51(2):24:1–24:39, April 2018.

[28] Finn Brunton. Constitutive interference: Spam and online communities. *Representations*, 117(1):30–58, 2012.

[29] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[30] Horst Bunke, Peter J. Dickinson, Miro Kraetzl, and Walter D. Wallis. *A Graph-Theoretic Approach to Enterprise Network Dynamics (Progress in Computer Science and Applied Logic (PCS))*. Birkhauser, 2006.

[31] Matthew Burgess, Eytan Adar, and Michael Cafarella. Link-prediction enhanced consensus clustering for complex networks. *PLoS ONE*, 11(5):1–23, 05 2016.

[32] Maya Cakmak, Crystal Chao, and Andrea L Thomaz. Designing interactions for robot active learners. *Autonomous Mental Development, IEEE Transactions on*, 2(2):108–118, 2010.

[33] Doina Caragea, Dianne Cook, and Vasant G. Honavar. Gaining insights into support vector machine pattern classifiers using projection-based tour methods. In *Proceedings of the Seventh ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '01, pages 251–256, New York, NY, USA, 2001. ACM.

[34] Alissa Centivany and Bobby Glushko. "popcorn tastes good": Participatory policymaking and reddit's. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1126–1137, New York, NY, USA, 2016. ACM.

[35] Eshwar Chandrasekharan, Umashanthi Pavalanathan, Anirudh Srinivasan, Adam Glynn, Jacob Eisenstein, and Eric Gilbert. You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech. *Proc. ACM Hum.-Comput. Interact.*, 1(CSCW):31:1–31:22, December 2017.

[36] Eshwar Chandrasekharan, Mattia Samory, Anirudh Srinivasan, and Eric Gilbert. The bag of communities: Identifying abusive behavior online with preexisting internet data. In *CHI'17*, CHI '17, pages 3175–3187, New York, NY, USA, 2017. ACM.

[37] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 167–176, New York, NY, USA, 2011. ACM.

[38] Jilin Chen, Gary Hsieh, Jalal U. Mahmud, and Jeffrey Nichols. Understanding individuals' personal values from social media word use. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work, Social Computing*, CSCW '14, pages 405–414, New York, NY, USA, 2014. ACM.

[39] Justin Cheng, Michael S. Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Anyone can become a troll: Causes of trolling behavior in online discussions. *CoRR*, abs/1702.01119, 2017.

[40] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Antisocial behavior in online discussion communities. In *International AAAI Conference on Web and Social Media*, 2015.

[41] Thomas Chesney, Iain Coyne, Brian Logan, and Neil Madden. Griefing in virtual worlds: causes, casualties and coping strategies. *Information Systems Journal*, 19(6):525–548, 2009.

[42] Aaron Clauset, M. E. J. Newman, , and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.

[43] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Political polarization on twitter. In *Proc. 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[44] R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977.

[45] Trevor F Cox and Michael AA Cox. *Multidimensional scaling*. CRC Press, 2000.

[46] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9, 2006.

[47] Tiago Oliveira Cunha, Ingmar Weber, Hamed Haddadi, and Gisele L. Pappa. The effect of social feedback in a reddit weight loss community. *CoRR*, abs/1602.07936, 2016.

[48] Johan Dahlin and Pontus Svenson. Ensemble approaches for improving community detection methods. *CoRR*, abs/1309.0242, 2013.

[49] Jianyong Dai and Jianlin Cheng. Hmmeditor: a visual editing tool for profile hidden markov model. *BMC genomics*, 9(Suppl 1):S8, 2008.

[50] Srayan Datta and Eytan Adar. Communitydiff: Visualizing community clustering algorithms. *TKDD*, 12(1):11:1–11:34, 2018.

[51] Srayan Datta and Eytan Adar. Extracting inter-community conflicts in reddit. In *International AAAI Conference on Web and Social Media*, 2019.

[52] Srayan Datta and Eytan Adar. Identifying, analyzing and predicting banned subreddits. *Unpublished Manuscript*, 2019.

[53] Srayan Datta, Chanda Phelan, and Eytan Adar. Identifying misaligned inter-group links and communities. *Proc. ACM Hum.-Comput. Interact.*, 1(CSCW):37:1–37:23, December 2017.

[54] Munmun De Choudhury, Emre Kiciman, Mark Dredze, Glen Coppersmith, and Mrinal Kumar. Discovering shifts to suicidal ideation from mental health content in social media. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2098–2110, New York, NY, USA, 2016. ACM.

[55] Munmun De Choudhury, Sanket S. Sharma, Tomaz Logar, Wouter Eekhout, and René Clausen Nielsen. Gender and cross-cultural differences in social media disclosures of mental illness. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 353–369, New York, NY, USA, 2017. ACM.

[56] Janez Demšar, Blaž Zupan, Gregor Leban, and Tomaz Curk. Orange: From experimental machine learning to interactive data mining. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy,*

September 20-24, 2004. Proceedings, pages 537–539. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[57] Karthik Dinakar, Roi Reichart, and Henry Lieberman. Modeling the detection of textual cyberbullying. 2011.

[58] Judith S Donath. Identity and deception in the virtual community. *Communities in cyberspace*, 1996:29–59, 1999.

[59] C. Dunne and Ben Shneiderman. Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts. *University of Maryland, HCIL Tech Report HCIL-2009-13*, 2009/// 2009.

[60] Scott Emmons, Stephen Kobourov, Mike Gallant, and Katy Brner. Analysis of network clustering algorithms and cluster quality metrics at scale. *PLoS ONE*, 11(7):1–18, 07 2016.

[61] Alex Endert, M Shahriar Hossain, Naren Ramakrishnan, Chris North, Patrick Fiaux, and Christopher Andrews. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems*, 43(3):411–435, 2014.

[62] D. Eppstein, M. S. Paterson, and F. F. Yao. On nearest-neighbor graphs. *Discrete & Computational Geometry*, 17(3):263–282, 1997.

[63] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004.

[64] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[65] Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, New York, NY, USA, 2003. ACM.

[66] Ethan Fast and Eric Horvitz. Identifying dogmatism in social media: Signals and models. *CoRR*, abs/1609.00425, 2016.

[67] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 147–156, New York, NY, USA, 2011. ACM.

[68] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: Interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 29–38, New York, NY, USA, 2008. ACM.

[69] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75 – 174, 2010.

[70] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[71] Deen Freelon, Marc Lynch, and Sean Aday. Online fragmentation in wartime. *The ANNALS of the American Academy of Political and Social Science*, 659(1):166–179, 2015.

[72] Devin Gaffney and J Nathan Matias. Caveat emptor, computational social science: Large-scale missing data in a widely-published reddit corpus. *arXiv preprint arXiv:1803.05046*, 2018.

[73] Patxi Galán-García, José Gaviria de la Puerta, Carlos Laorden, Igor Santos, and Pablo García Bringas. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. In *SOCO-CISIS-ICEUTE*, 2013.

[74] Eric Gilbert. Widespread underprovision on reddit. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 803–808, New York, NY, USA, 2013. ACM.

[75] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220, New York, NY, USA, 2009. ACM.

[76] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[77] Wael H. Gomaa and Aly A. Fahmy. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013. Full text available.

[78] Justin Grimmer and Gary King. General purpose computer-assisted clustering and conceptualization. *Proceedings of the National Academy of Sciences*, 108(7):2643–2650, 2011.

[79] Philip J. Guo, Sean Kandel, Joseph M. Hellerstein, and Jeffrey Heer. Proactive wrangling: Mixed-initiative end-user programming of data transformation scripts. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 65–74, New York, NY, USA, 2011. ACM.

[80] Mohammad Hamdaqa, Ladan Tahvildari, Neil LaChapelle, and Brian Campbell. Cultural scene detection using reverse louvain optimization. *Sci. Comput. Program.*, 95(P1):44–72, December 2014.

[81] W. L. Hamilton, J. Zhang, C. Danescu-Niculescu-Mizil, D. Jurafsky, and J. Leskovec. Loyalty in Online Communities. *ArXiv e-prints*, March 2017.

[82] Derek Hansen, Ben Shneiderman, and Marc A Smith. *Analyzing social media networks with NodeXL: Insights from a connected world*. Morgan Kaufmann, 2010.

[83] Claire Hardaker. Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions. *Journal of Politeness Research*, 6(2):215–242, 2010.

[84] Jeffrey Heer and Danah Boyd. Vizster: Visualizing online social networks. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 5–, Washington, DC, USA, 2005. IEEE Computer Society.

[85] Mark Heimann, Wei Lee, Shengjie Pan, Kuan-Yu Chen, and Danai Koutra. Hashalign: Hash-based alignment of multiple graphs. In Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and Lida Rashidi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 726–739, Cham, 2018. Springer International Publishing.

[86] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 117–126, New York, NY, USA, 2018. ACM.

[87] Monika Henzinger. Finding near-duplicate web pages: A large-scale evaluation of algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 284–291, New York, NY, USA, 2006. ACM.

[88] Jack Hessel, Alexandra Schofield, Lillian Lee, and David M. Mimno. What do vegans do in their spare time? latent interest detection in multi-community networks. *CoRR*, abs/1511.03371, 2015.

[89] Jack Hessel, Chenhao Tan, and Lillian Lee. Science, askscience, and badscience: On the coexistence of highly related communities. *CoRR*, abs/1612.07487, 2016.

[90] Heath Hohwald, Manuel Cebrian, Arturo Canales, Rubén Lara, and Nuria Oliver. Inferring unobservable inter-community links in large social networks. In *2009 International Conference on Computational Science and Engineering*, pages 375–380. IEEE, 2009.

[91] Petter Holme, Mikael Huss, and Hawoong Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538, 2003.

[92] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 159–166, New York, NY, USA, 1999. ACM.

[93] Darko Hric, Richard K. Darst, and Santo Fortunato. Community detection in networks: Structural communities versus ground truth. *Phys. Rev. E*, 90:062805, Dec 2014.

[94] Mohsen Jamali and Martin Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 397–406, New York, NY, USA, 2009. ACM.

[95] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visuatizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, October 2011.

[96] Ta-Chu Kao and Mason A Porter. Layer communities in multiplex networks. *Journal of Statistical Physics*, pages 1–17, 2017.

[97] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1343–1352, New York, NY, USA, 2010. ACM.

[98] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Learning to learn: Algorithmic inspirations from human problem solving. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1571–1577. AAAI Press, 2012.

[99] Ramakanth Kavuluru, María Ramos-Morales, Tara Holaday, Amanda G. Williams, Laura Haye, and Julie Cerel. Classification of helpful comments on online suicide watch forums. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '16, pages 32–40, New York, NY, USA, 2016. ACM.

[100] Charles Kiene, Andrés Monroy-Hernández, and Benjamin Mako Hill. Surviving an "eternal september": How an online community managed a surge of newcomers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1152–1156, New York, NY, USA, 2016. ACM.

[101] Mikko Kivel, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[102] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[103] Stephen G. Kobourov. Spring embedders and force directed graph drawing algorithms, 2012. cite arxiv:1201.3011Comment: 23 pages, 8 figures.

[104] Danai Koutra, Tai-You Ke, U. Kang, Duen Horng Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'11, pages 245–260, Berlin, Heidelberg, 2011. Springer-Verlag.

[105] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. Deltacon: Principled massive-graph similarity function with attribution. *ACM Trans. Knowl. Discov. Data*, 10(3):28:1–28:43, February 2016.

[106] Joseph B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, February 1956.

[107] Todd Kulesza, Simone Stumpf, Weng-Keen Wong, Margaret M Burnett, Stephen Perona, Andrew Ko, and Ian Oberst. Why-oriented end-user debugging of naive bayes text classification. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 1(1):2, 2011.

[108] Srijan Kumar, Justin Cheng, Jure Leskovec, and V.S. Subrahmanian. An army of me: Sockpuppets in online discussion communities. In *WWW'17*, WWW '17, pages 857–866, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[109] Srijan Kumar, William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *WWW'18*, WWW '18, 2018.

[110] Srijan Kumar, Francesca Spezzano, and V. S. Subrahmanian. VEWS: A wikipedia vandal early warning system. *CoRR*, abs/1507.01272, 2015.

[111] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: Mining a social network with negative edges. In *WWW'09*, WWW '09, pages 741–750, New York, NY, USA, 2009. ACM.

[112] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. arXiv e-print 0908.1062, August 2009. Physical Review E 80, 056117 (2009).

[113] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *CoRR*, abs/1203.6093, 2012.

[114] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78(4):046110, October 2008.

[115] Alex Leavitt. "this is a throwaway account": Temporary technical identities and perceptions of anonymity in a massive online community. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work, Social Computing*, CSCW '15, pages 317–327, New York, NY, USA, 2015. ACM.

[116] Alex Leavitt and Joshua A. Clark. Upvoting hurricane sandy: Event-based news production processes on a social news site. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1495–1504, New York, NY, USA, 2014. ACM.

[117] Alex Leavitt and John J. Robinson. The role of information visibility in network gatekeeping: Information aggregation on reddit during crisis events. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 1246–1261, New York, NY, USA, 2017. ACM.

[118] Gregor Leban, Blaž Zupan, Gaj Vidmar, and Ivan Bratko. Vizrank: Data visualization guided by machine learning. *Data Mining and Knowledge Discovery*, 13(2):119–136, 2006.

[119] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[120] Mingwei Leng, Yukai Yao, Jianjun Cheng, Weiming Lv, and Xiaoyun Chen. Active Semi-supervised Community Detection Algorithm with Label Propagation. In *Database Systems for Advanced Applications*, volume 7826 of *Lecture Notes in Computer Science*, pages 324–338. Springer Berlin Heidelberg, 2013.

[121] Zhe Liu and Ingmar Weber. *Is Twitter a Public Sphere for Online Conflicts? A Cross-Ideological and Cross-Hierarchical Look*, pages 336–347. Springer International Publishing, Cham, 2014.

[122] Zhe Liu and Ingmar Weber. Predicting ideological friends and foes in twitter conflicts. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 575–576, New York, NY, USA, 2014. ACM.

[123] Zhicheng Liu and John T. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Trans. Vis. Comput. Graph.*, 16(6):999–1008, 2010.

[124] Avishay Livne, Matthews P. Simmons, W. Abraham Gong, Eytan Adar, and Lada A. Adamic. The party is over here: structure and content in the 2010 election. In *ICWSM*. Association for the Advancement of Artificial Intelligence, 2011.

[125] Kiel Long, John Vines, Selina Sutton, Phillip Brooker, Tom Feltwell, Ben Kirman, Julie Barnett, and Shaun Lawson. "could you define that in bot terms"?: Requesting, creating and using bots on reddit. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3488–3500, New York, NY, USA, 2017. ACM.

[126] Gilad Lotan. Israel, Gaza, War & Data, 2014.

[127] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A Statistical Mechanics and its Applications*, 390:1150–1170, March 2011.

[128] Marco Lui and Timothy Baldwin. Cross-domain feature selection for language identification. In *In Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 553–561, 2011.

[129] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[130] Richard Maclin and David W. Opitz. Popular ensemble methods: An empirical study. *CoRR*, abs/1106.0257, 2011.

[131] Sofus A. Macskassy. Using Graph-based Metrics with Empirical Risk Minimization to Speed Up Active Learning on Networked Data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 597–606, New York, NY, USA, 2009. ACM.

[132] Trevor Martin. community2vec: Vector representations of online communities encode semantic relationships. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 27–31. Association for Computational Linguistics, 2017.

[133] Trevor Martin. Dissecting trumps most rabid online following, 2017.

[134] J. Nathan Matias. Going dark: Social factors in collective action against platform operators in the reddit blackout. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1138–1151, New York, NY, USA, 2016. ACM.

[135] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 111–120, Oct 2011.

[136] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Generalized Louvain method for community detection in large networks. In *ISDA*, pages 88–93. IEEE, 2011.

[137] Emily Merritt. *An analysis of the discourse of Internet trolling: A case study of Reddit. com*. PhD thesis, 2012.

[138] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 935–940, New York, NY, USA, 2006. ACM.

[139] Rada F. Mihalcea and Dragomir R. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[140] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[141] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[142] David Mimno and Moontae Lee. Low-dimensional embeddings for interpretable anchor-based topic inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1319–1328, Doha, Qatar, October 2014. Association for Computational Linguistics.

[143] B. L. Monroe, M. P. Colaresi, and K. M. Quinn. Fightin'Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict. *Political Analysis*, 16(4):372, 2008.

[144] Lev Muchnik, Sinan Aral, and Sean J. Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.

[145] M E Newman. Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103(23):8577–8582, June 2006.

[146] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[147] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, September 2003.

[148] Randal S. Olson and Zachary P. Neal. Navigating the massive world of reddit: Using backbone networks to map user interests in social media. *CoRR*, abs/1312.3387, 2013.

[149] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications*, 1(1):19–30, 2010.

[150] Kayur Patel, Steven M. Drucker, James Fogarty, Ashish Kapoor, and Desney S. Tan. Using multiple models to understand data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1723–1728. AAAI Press, 2011.

[151] Leto Peel, Daniel B. Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks, 2016.

[152] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4, 2005.

[153] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.

[154] Kyle Porter. Analyzing the darknetmarkets subreddit for evolutions of tools and trends using lda topic modeling. *Digital Investigation*, 26:S87–S97, 2018.

[155] Reid Porter, James Theiler, and Don Hush. Interactive machine learning in data exploitation. *Computing in Science & Engineering*, 15(5):12–20, 2013.

[156] Liza Potts and Angela Harrison. Interfaces as rhetorical constructions: Reddit and 4chan during the boston marathon bombings. In *Proceedings of the 31st ACM International Conference on Design of Communication*, SIGDOC '13, pages 143–150, New York, NY, USA, 2013. ACM.

[157] Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 225–236, New York, NY, USA, 2014. ACM.

[158] Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 225–236, New York, NY, USA, 2014. ACM.

[159] Usha N. Raghavan, Reka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks, September 2007.

[160] Joerg Reichardt and Stefan Bornholdt. Statistical Mechanics of Community Detection. *Physical Review E*, 74:016110, 2006.

[161] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010.

[162] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[163] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[164] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[165] T. Safavi, C. Sripada, and D. Koutra. Scalable hashing-based network discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 405–414, Nov 2017.

[166] Klaus Peter Schneider, MA Susanne StrubelBurgdorf, and Felix Metzger. Impoliteness in cmc: Differences in trolling of men's rights activists vs. trolling of feminists. 2013.

[167] M. ngeles Serrano, Marin Bogu, and Alessandro Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488, 2009.

[168] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[169] Pnina Shachaf and Noriko Hara. Beyond vandalism: Wikipedia trolls. *Journal of Information Science*, 36(3):357–370, 2010.

[170] Moshen Shahriari and Mahdi Jalili. Ranking nodes in signed social networks. *Social Network Analysis and Mining*, 4(1):172, Jan 2014.

[171] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[172] Hossam Sharara, Awalin Sopan, Galileo Namata, Lise Getoor, and Lisa Singh. G-pare: A visual analytic tool for comparative analysis of uncertain graphs. In *IEEE VAST*, pages 61–70. IEEE, 2011.

[173] Michael Shilman, Desney S. Tan, and Patrice Simard. Cuetip: A mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 323–332, New York, NY, USA, 2006. ACM.

[174] M. Steyvers and T. Griffiths. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. Laurence Erlbaum, 2007.

[175] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *Int. J. Hum.-Comput. Stud.*, 67(8):639–662, August 2009.

[176] John Suler. The online disinhibition effect. 7:321–6, 07 2004.

[177] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1283–1292, New York, NY, USA, 2009. ACM.

[178] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 613–624, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[179] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[180] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[181] Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. The State of the Art in Visualizing Group Structures in Graphs. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015.

[182] Greg Wadley, Wally Smith, Bernd Ploderer, Jon Pearce, Sarah Webber, Mark Whooley, and Ron Borland. What people talk about when they talk about quitting. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, OzCHI '14, pages 388–391, New York, NY, USA, 2014. ACM.

[183] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H. Witten. Interactive machine learning: Letting users build classifiers. *Int. J. Hum.-Comput. Stud.*, 56(3):281–292, March 2002.

[184] Eugene J Webb, Donald Thomas Campbell, Richard D Schwartz, and Lee Sechrest. *Unobtrusive measures: Nonreactive research in the social sciences*, volume 111. Rand McNally Chicago, 1966.

[185] Zhongyu Wei12, Yang Liu, and Yi Li. Is this post persuasive? ranking argumentative comments in the online forum. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 195, 2016.

[186] James D. Wilson, John Palowitch, Shankar Bhamidi, and Andrew B. Nobel. Community extraction in multilayer networks with heterogeneous community structure. *CoRR*, abs/1610.06511, 2016.

[187] Zhaoming Wu, Charu C. Aggarwal, and Jimeng Sun. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 447–456, New York, NY, USA, 2016. ACM.

[188] Jaewon Yang and Jure Leskovec. Overlapping communities explain core–periphery organization of networks. *Proceedings of the IEEE*, 102(12):1892–1902, 2014.

[189] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

[190] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[191] J. Zhang, W. L. Hamilton, C. Danescu-Niculescu-Mizil, D. Jurafsky, and J. Leskovec. Community Identity and User Engagement in a Multi-Community Landscape. *ArXiv e-prints*, May 2017.

[192] J. Zhang and P. S. Yu. Multiple anonymized social networks alignment. In *2015 IEEE International Conference on Data Mining*, pages 599–608, Nov 2015.

[193] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML 2004: Proceedings of the twenty-first international conference on machine learning. omnipress*, pages 919–926, 2004.

[194] Kaiyu Zhao, Matthew O. Ward, Elke A. Rundensteiner, and Huong N. Higgins. Lovis: Local pattern visualization for model refinement. *Comput. Graph. Forum*, 33(3):331–340, 2014.

[195] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.