Harnessing the Power of Multi-Source Data: an Exploration of Diversity and Similarity

by

Yang Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering-Systems)
in The University of Michigan
2016

Doctoral Committee:

Professor Mingyan Liu, Chair
Professor Alfred O. Hero III
Assistant Professor Grant Schoenebeck
Associate Professor Vijay Subramanian

© Yang Liu 2016
All Rights Reserved

To my Mom Yufeng Qi, and Dad Zhixian Liu.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to the people who made this dissertation possible. First, I would like to thank Professor Mingyan Liu with my deepest gratitude. I have been very fortunate to have Mingyan as my Ph.D advisor, who showed continuous support for my Ph.D career. Throughout my five years' study, Mingyan offered helpful and timely advices on both my research and career development. I am also very grateful for having Professor Alfred O. Hero III, Professor Vijay Subramanian and Professor Grant Schoenebeck served on my thesis committee, and I owe many thanks to them. They provided valuable comments and reviews for this dissertation.

My thanks also go to my collaborators, especially Professor Michael Bailey from UIUC and Professor Jing Deng from UNCG. The many insightful discussions with them have inspired multiple pieces of my work. I want to thank Professor Demosthenis Teneketzis for his excellent lectures. I have gained a lot of technical skills and understandings via attending his classes.

I thank my colleagues at EECS and friends at Ann Arbor, including but not limited to Qingsi, Ouyang, Jing, Armin, Parinaz etc. I very much enjoyed our collaborations, discussions and a whole lot other happy moments.

Most importantly I would like to thank my parents for their love, support and understanding, without which I will not be able to reach where I am. My special thanks goes to Tess Lou. We have been through thick and thin, and I could not thank her more for the encouragements.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Harnessing the Power of Multi-source Data: an Exploration of Diversity and Similarity

by

Yang Liu

Doctor of Philosophy

Chair: Mingyan Liu

This dissertation studies a sequence of problems concerning the collection and utilization of data from disparate sources, e.g., that arising in a crowd-sourcing system. It aims at developing learning methods to enhance the quality of decision-making and learning task performance by exploiting a multitude of diversity, similarity and interdependency inherent in a crowd-sourcing system and among disparate data sources.

We start our study with a family of problems on sequential *decision-making* combined with *data collection* in a crowd-sourcing system, where the goal is to improve the quality of data input or computational output, while reducing the cost in using such a system. In this context, the learning methods we develop are *closed-loop* and *online*, i.e., decisions made are functions of past data observations, present actions determine future observations, and the learning occurs as data inputs arrive. The similarity and disparity among different data sources help us in some cases to speed up the learning process (e.g., in a recommender system), and in some other cases to perform quality control over data input for which ground-truth may be non-existent or cannot be obtained directly (e.g., in a crowd-sourcing market using Amazon Mechanical Turks (AMTs)).

We then apply our algorithms to the processing of a large set of network malicious activity data collected from diverse sources, with a goal of uncovering interconnectedness/similarity between different network entities' malicious behaviors. Specifically, we apply our online prediction algorithm presented and analyzed in earlier parts of the dissertation to this data and show its effectiveness in predicting next-day maliciousness. Furthermore, we show that data-specific properties of this set of data allow us to map networks' behavioral similarity to similarity in their topological features. This in turn enables prediction even in the *absence* of measurement data.

CHAPTER I

Introduction

1.1 Background and Motivations

Machine learning techniques often rely on correctly labeled data for purposes such as building classifiers; this is particularly true for supervised discriminative learning. As shown in [68, 57], the quality of labels can significantly impact the quality of the trained classifier and in turn the system performance. Semi-supervised learning methods, e.g. [84, 12, 43] have been proposed to circumvent the need for labeled data or lower the requirement on the size of labeled data; nonetheless, many state-of-the-art machine learning systems such as those used for pattern recognition continue to rely heavily on supervised learning, which necessitates cleanly labeled data. At the same time, advances in instrumentation and miniaturization, combined with frameworks like participatory sensing, rush in enormous quantities of unlabeled data.

Against this backdrop, crowd-sourcing has emerged as a viable and often favored solution as evidenced by the popularity of the Amazon Mechanical Turk (AMT) system. Prime examples include a number of recent efforts on collecting large-scale labeled image datasets, such as ImageNet [15] and LabelMe [67]. The concept of crowd-sourcing has also been studied in contexts other than processing large amounts of unlabeled data, see e.g., user-generated map [25], opinion/information diffusion [24], and event monitoring [13] in large, decentralized systems.

Its many advantages notwithstanding, the biggest problem with crowd-sourcing is quality control, which is present in different types of crowd-sourcing systems, albeit of different nature. In the so-called *crowd-sourcing markets* where AMTs are used to perform computational tasks, as shown in several previous studies [31, 68], if labelers (e.g., AMTs) are not selected carefully the resulting labels can be very noisy, due to reasons such as varying degrees of competence, individual biases, and sometimes irresponsible behavior. At the same time, the cost in having a large amount of data labeled (payment to the labelers) is non-trivial. This makes it important to look into ways of improving the quality of the crowd-sourcing process and the quality of the results generated by the labelers. The main technical difficulty here is often the lack of verifiable ground-truth data, or ground-truth that could be obtained within reasonable amount of time; after all the whole point of a crowd-sourcing market is to try to obtain approximate ground-truth through the crowd.

In a second type of crowd-sourcing systems, often called *recommendation* systems, a similar quality control issue exist. Here recommendations (on movies, movie articles, vendors, restaurants and so on) made by disparate individuals are not only based on objective ground-truth but also subjective opinion and personal preference. For a user interested in taking others' recommendations into consideration when making her own decisions (e.g., which movies to see, which restaurants to visit and so on), the issue arises as to which recommenders' opinion should she value the most in her decision process so as to maximize her overall satisfaction.

More generally, whenever a dataset involves multiple sources – a source could be either the generator or the consumer of the data, this type of quality control issue become relevant due to objective and subjective measures of ground-truth (or the lack thereof). This motivates us to study how to design data processing and learning algorithms to address this challenge. We are particularly interested in the type of process scenarios that involve sequential *decision-making* combined with *data collection*, i.e., data is acquired sequentially in time and likely from multiple disparate sources, and past observations affects decisions

on data collection, which in turn determines future observations. Our goal is to improve the quality of data input and/or computational output, while reducing the cost in using such a system, by exploiting the diversity and similarity inherent in the multi-source system. Within this context, the learning methods we develop are *closed-loop* and *online*, i.e., decisions are functions of past data observations, present actions determine future observations, and learning occurs as data inputs arrive.

1.2 Outline of problems studied

This dissertation studies a sequence of problems concerning the collection and utilization of data from multiple sources, and aims at developing learning methods to enhance the quality of decision-making and learning task performance by exploiting a multitude of diversity, similarity and interdependency inherent among multiple data sources. Below we describe these problems in more detail.

1.2.1 Data quality control in a crowd-sourcing market

The first problem we consider is on labeler quality control/selection for crowd-sourcing system, whereby a user needs to assign a set of arriving tasks to a set of labelers whose qualities are unknown a priori. It is in the user's interest to estimate the labelers' quality over time so as to make more judicious task assignment decisions which lead to better labeling outcome. This problem in some sense can be cast as multi-armed bandit (MAB) problem. Within such a framework, the objective is to select the best of a set of choices (or "arms") by repeatedly sampling different choices (*exploration*) and their empirical quality is subsequently used to control how often a choice is used (*exploitation*). However, there are two distinct features that set our problem apart from the existing literature on bandit problems. Firstly, since the data is unlabeled to begin with and the labelers' quality is also unknown, a particular choice of labelers leads to unknown quality of their labeling outcome (mapped to the "reward" of selecting a choice in the MAB context). Whereas this reward

is assumed to be known instantaneously following a selection in the MAB problem, in our model this remains unknown and at best can only be estimated with a certain error probability. Secondly, to avoid having to deal with a combinatorial number of arms, it is desirable to learn and estimate each individual labeler's quality separately (as opposed to estimating the quality of different combinations of labelers). The optimal selection of labelers then depends on individual qualities as well as how the labeling outcome is computed using individual labels.

For this problem our goal is to design a good learning algorithm that estimates the labelers' quality as tasks are assigned and performed, which allows the user to over time learn the more effective combinations of labelers for arriving tasks.

1.2.2 Improving learning performance via crowd-sourced or multi-source data

We next consider two problems on improving learning performance via data collected through crowd-sourcing.

Crowd learning in a recommendation system: The first problem is set in a general recommendation system, e.g., Yelp, where users both give recommendations and seek recommendations to aid their decision-making abstracted to the form of making selections out of a set of options, e.g., restaurants, movies, etc. Left alone and without prior information, a user can only learn the best options by repeated exploration or sampling. This individual learning process may be aptly captured by the standard MAB model. With a recommendation system, however, the same user can tap into observations (or samples) made by others, but the challenge here is that user experiences are subjective, so it is a priori unclear how much value to associate with any given recommendation. The user's objective is to maximize her expected total reward (e.g., overall satisfaction from the sequence of options she chooses) over a certain time horizon, and this needs to be done through an online learning process, i.e., a sequence of exploration (sampling the return of each option)

and exploitation (selecting empirically good options) steps. Our goal is to design such a learning process that can effectively utilize "second-hand learning", i.e., by observing how others in the system act and what they recommend, in additional to "first-hand learning", i.e., direct sampling of options. This is accomplished by estimating pairwise differences among users, which are then used to "convert" other users' samples for one's own use.

Best crowd – a more general setting: The second problem is set in a more general machine learning context, whereby a user could train a classifier with her own data, but can also seek to improve her performance by requesting samples from other similar users at a cost. Similar as in the previous problem, in this case a user also needs to estimate her "similarity" with others in the system. A difference, however, is that now the user must also decide which is the best set of users to request data from as it incurs a cost. This gives rise to the notion of a *smarter* or best crowd that a user may identify and utilize.

1.2.3 Interdependency of network maliciousness: an application

We will also put results derived earlier in the application context of a particular set of data collected over the Internet that is of a multi-source nature. Our dataset centers on observations of malicious activities originated from different networks on the Internet. These are often symptoms of their security posture and policies adopted by them. In particular, the dynamics in such activities reveal rich information on the evolution of networks' underlying conditions and therefore are helpful in capturing behaviors in more consistent ways. At the same time, the interdependence of today's Internet also means that what we see from one network is inevitably related to others. This connection can provide insight into the conditions of not just a single network viewed in isolation, but multiple networks viewed together. This understanding in turn allows us to predict more accurately the security conditions or malicious activities of networks in general, and can lead to the design of better proactive risk aware policies for applications ranging from traffic engineering, peering and

routing.

We shall explore unique features in this dataset to highlight the effectiveness of one of our algorithms in making predictions, as well as to show that data-specific processing methods also arise within this contexts that can be exploited to offer additional insight.

1.3 Contributions

1.3.1 Data quality control in crowd-sourcing market [51]

For the labeler selection problem, we obtained the following set of results.

1. We designed an online learning algorithm to estimate the quality of labelers in a crowd-sourcing setting without ground-truth information but with mild assumptions on the quality of the crowd as a whole, and showed that it is able to learn the optimal set of labelers under both simple and weighted majority voting rules and attains no-regret performance guarantees (w.r.t. always using the optimal set of labelers).

2. We similarly provided regret bounds on the cost of this learning algorithm w.r.t. always using the optimal set of labelers.

3. We showed how our model and results can be extended to the case where the quality of a labeler may be task-type dependent, as well as a simple procedure to quickly detect and filter out "bad" (dishonest, malicious or incompetent) labelers to further enhance the quality of crowd-sourcing.

4. We established a lower bound on the learning regret for our online labeler selection problem.

5. Our validation included both simulation and the use of a real-world AMT dataset.

1.3.2 Crowd-Learning: improving online recommendation [50, 48]

For the online recommendation problem, we show that:

1. When complete information is shared (or being crowd-sourced), our crowd-learning algorithm results in up to a M-fold improvement to the regret bound under different problem settings.

2. When only partial information is available, the improvement of learning is tied to a weight/recommendation factor of how much we value others' opinion compared to her own. Interestingly we find in certain special case, simply integrating partial information allows the algorithm's performance bound to outperform its full information counterpart. To our best knowledge this is the first attempt to analyze online learning with crowdsourced data.

3. By applying our results to the movie ratings dataset MovieLens [39], we showed how our algorithms can be used as an online (causal) process to make real-time predictions/recommendations. Experiments show that in both online and offline settings our recommendation performance exceeds that of individual learning, and in the offline setting our results are also comparable with several existing offline solutions.

4. In order to further understand the difference between the notion of "crowd learn" and offline collaborative filtering methods, we adapted the idea of finding a crowd of similar users to existing matrix factorization based recommendation methods and show its performance.

1.3.3 Online prediction via multi-source data [52]

We extend the idea of crowd learning and study a more general and complicated online learning objective (training online classifiers) with crowd-sourced data, and we have the following set of results:

1. We first establish bounds on the expected learning error under ideal conditions, including that (1) the similarity information between data sources is known a priori, and (2) data from all sources are available for free.

2. We then relax assumption (1) and similarly establish the bounds on the error when such similarity information needs to be learned over time.

3. We then relax both (1) and (2) and design an efficient online learning algorithm that simultaneously makes decisions on requesting and combining data for the purpose of training the predictor, and learning the similarity among data sources. We again show that this algorithm achieves a guaranteed performance uniform in time (the best possible error rate plus additional learning error terms), and the additional cost with respect to the minimum cost required to achieve optimal learning rate diminishes in time. Moreover, the obtained bounds show clearly the trade-off between learning accuracy and the cost to obtain additional data. This provides useful information for system designers with different objectives.

1.3.4 Interdependency of network maliciousness [53, 49]

Our main contributions and findings for this application study are as follows:

1. We quantify the similarity relationship between two networks' dynamic malicious behavior. We show that (1) the online prediction algorithm we developed earlier combined with such quantitative understanding of similarity can achieve a much better prediction of future maliciousness; and (2) the crowd learning based collaborative filtering method with built-in similarity notions can enable more robust measurements through more accurate estimates of missing entries in the measurement data.

2. We then use statistical inference methods to evaluate the significance (or the degree of resemblance) of the set of spatial features in explaining the observed similarity in malicious behavior. We provide performance bounds on our inference model in the presence of noise and error inherent in the RBL data.

3. Of particular interest is a finding that within the set of spatial features, the topological distance between two networks is by far the most significant indicator of their

similarity in maliciousness.

4. We show the understanding in spatial similarity can enable prediction for a network without historical information which would otherwise be infeasible. The results of this particular application study shed lights on building cost-efficient Internet-scale measurement techniques.

1.4 Organizations

The remainder of this dissertation is organized as follows. We start in Chapter II with the data quality control problem for labeler selection in crowd-sourcing market. Then we proceed to discuss the crowd learning study on recommendation system with crowd-sourced ratings in Chapter III. We present the learning via multi-source data problem under a more generalized setting in Chapter IV. In our last part of the dissertation, we present the application study on security interdependency with a set of cyber security data collected from diverse sources in Chapter V. Chapter VI will conclude the dissertation, along with a discussion on future works. All proofs can be found in Appendices.

CHAPTER II

Quality control of crowd-sourcing systems

2.1 Introduction

Machine learning techniques often rely on correctly labeled data for purposes such as building classifiers; this is particularly true for supervised discriminative learning. As shown in [68, 57], the quality of labels can significantly impact the quality of the trained classifier and in turn the system performance. Semi-supervised learning methods, e.g. [84, 12, 43] have been proposed to circumvent the need for labeled data or lower the requirement on the size of labeled data; nonetheless, many state-of-the-art machine learning systems such as those used for pattern recognition continue to rely heavily on supervised learning, which necessitates cleanly labeled data. At the same time, frameworks like participatory sensing rush in enormous quantities of unlabeled data.

Against this backdrop, crowd-sourcing has emerged as a viable and often favored solution as evidenced by the popularity of the Amazon Mechanical Turk (AMT) system. Prime examples include a number of recent efforts on collecting large scale labeled image datasets, such as ImageNet [15] and LabelMe [67]. The concept of crowd-sourcing has also been studied in contexts other than processing large amounts of unlabeled data, see e.g., user-generated map [25], opinion diffusion [48], and event monitoring [13] in large, decentralized systems.

Its many advantages notwithstanding, the biggest problem with crowd-sourcing is qual-

ity control: as shown in several previous studies [31, 68], if labelers (e.g., AMTs) are not selected carefully the resulting labels can be very noisy, due to reasons such as varying degrees of competence, individual difference, and sometimes irresponsible behavior. At the same time, the cost for having large amount of data labeled (payment to the labelers) is non-trivial. This makes it important to look into ways of improving the quality of the crowd-sourcing process and the quality of the results generated by the labelers.

In this chapter we approach the labeler selection problem in an online learning framework, whereby the labeling quality of the labelers is estimated as tasks are assigned and performed, so that an algorithm over time learns to use the more effective combinations of labelers for arriving tasks. This problem in some sense can be cast as multi-armed bandit (MAB) problem, see e.g., [44, 5, 73]. Within such a framework, the objective is to select the best of a set of choices (or "arms") by repeatedly sampling different choices (referred to as *exploration*) and their empirical quality is subsequently used to control how often a choice is used (referred to as *exploitation*). However, there are two distinct features that set our problem apart from the existing literature in bandit problems. Firstly, since the data is unlabeled to begin with and the labelers' quality is also unknown, a particular choice of labelers leads to unknown quality of their labeling outcome (mapped to the "reward" of selecting a choice in the MAB context). Whereas this reward is assumed to be known instantaneously following a selection in the MAB problem, in our model this remains unknown and at best can only be estimated with a certain error probability. This poses significant technical challenge compared to a standard MAB problem. Secondly, to avoid having to deal with a combinatorial number of arms, it is desirable to learn and estimate each individual labeler's quality separately (as opposed to estimating the quality of different combinations of labelers). The optimal selection of labelers then depends on individual qualities as well as how the labeling outcome is computed using individual labels. In this study we will consider both a simple majority voting rule as well as a weighted majority voting rule and derive the respective optimal selection of labelers given their estimated quality.

Due to its online nature, our algorithm can be used in real time, processing tasks as they arrive. Our algorithm thus has the advantage of performing quality assessment and adapting to better labeler selections as tasks arrive. This is a desirable feature because generating and processing large datasets can incur significant cost and delay, so the ability to improve labeler selection on the fly (rather than waiting till the end) can result in substantial cost savings and improvement in processing quality. Below we review the literature most relevant to the study presented in this paper in addition to the MAB literature cited above.

Within the context of learning and differentiating labelers' expertise in crowd-sourcing systems, a number of studies have looked into offline algorithms. For instance, in [22], methods are proposed to eliminate irrelevant users from a set of user-generated dataset; in this case the elimination is done as post-processing to clean up the data since the data has already been labeled by the labelers (tasks have been performed). Another example is the family of matrix factorization or matrix completion based methods, see e.g., [83], where labeler selection is implicitly done through the numerical process of finding the best recommendation for a participant. Again this is done after the labeling has already been done for all data by all (or almost all) labelers. This type of approaches is more appropriate when used in a recommendation system where data and user-generated labels already exist in large quantities.

Recent studies [36, 35] have examined the fundamental trade-off between labeling accuracy and redundancy in task assignment in crowd-sourcing systems. In particular, it is shown in [36] that a labeling accuracy of $1-\varepsilon$ for each task can be achieved with a per-task assignment redundancy no more than $O(K/q \cdot \log(K/\varepsilon))$; thus more redundancy can be traded for more accurate outcome. In [36] the task assignment is done in a one-shot fashion (thus non-adaptive) rather than sequentially with each task arrival as considered in our paper, thus the result is more applicable to offline settings similar to those cited in the previous paragraph. In [35] an iterative algorithm is proposed for deciding tasks assignment and it is shown to outperform majority voting. Again the approach here is one-shot

where all questions are asked simultaneously and the allocation rule is non-adaptive.

Within online solutions, the concept of active learning has been quite intensively studied, where the labelers are guided to make the labeling process more efficient. Examples include [31], which uses a Bayesian framework to actively assign unlabeled data based on past observations on labeling outcomes, and [54], which uses a probabilistic model to estimate the labelers' expertise. However, most studies on active learning require either an oracle to verify the correctness of the finished tasks which in practice does not exist, or ground-truth feedback from indirect but relevant experiments (see e.g.,[31]). Similarly, existing work on using online learning for task assignment also typically assumes the availability of ground-truth (as in MAB problems). For instance, in [28] online learning is applied to sequential task assignment but ground-truth of the task performance is used to estimate the performer's quality. In [56], a Bayes update aided online solution was proposed to minimize the regret in a problem of disseminating news to a crowd of users. The performance of the developed algorithm was shown to be better than Thompson Sampling based solutions. However, again, for the setting considered in the above paper, ground-truth signals indicating whether the user likes or dis-likes pushed news are revealed immediately after each dissemination. In this sense, our results cannot be compared directly to those cited above.

Our work differs from the above as we do not require oracle or the availability of immediate ground-truth; we instead impose a mild assumption on the collective quality of the crowd (without which crowdsourcing would be useless and would not have existed), so an estimated or imperfect ground-truth can be inferred. Secondly, our framework allows us to obtain performance bounds on the proposed algorithm in the form of regret with respect to the optimal strategy that always uses the best set of labelers; this type of performance guarantee is lacking in most of the work cited above. Last but not least, our algorithm is very broadly applicable to a generic crowd-sourcing task assignment rather than being designed for specific type of tasks or data.

Our main contributions are summarized as follows.

1. We design an online learning algorithm to estimate the quality of labelers in a crowd-sourcing setting without ground-truth information but with mild assumptions on the quality of the crowd as a whole, and show that it is able to learn the optimal set of labelers under both simple and weighted majority voting rules and attains no-regret performance guarantees (w.r.t. always using the optimal set of labelers).

2. We similarly provide regret bounds on the cost of this learning algorithm w.r.t. always using the optimal set of labelers.

3. We show how our model and results can be extended to the case where the quality of a labeler may be task-type dependent, as well as a simple procedure to quickly detect and filter out "bad" (dishonest, malicious or incompetent) labelers to further enhance the quality of crowd-sourcing.

4. We establish a lower bound on the learning regret for our online labeler selection problem.

5. Our validation includes both simulation and the use of a real-world AMT dataset.

The remainder of this chapter is organized as follows. We formulate our problem in Section 2.2. In Sections 2.3 and 2.4 we introduce our learning algorithm along with regret analysis under a simple majority and weighted majority voting rule, respectively. We extend our model to account for the case where labelers' expertise may be task dependent in Section 2.5. Lower bound results on our learning algorithm is presented in Section 2.6 and we provide a matching and refined upper bound in Section 2.7. Numerical results are presented in Section 2.8. Section 2.9 concludes the chapter.

2.2 Problem formulation and preliminaries

2.2.1 The crowd-sourcing model

We begin by introducing the following major components of the crowd-sourcing system.

1. *User*. There is a single user with a sequence of tasks (unlabeled data) to be performed/labeled. Our proposed on-line learning algorithm is to be employed by the user in making labeler selections. Throughout our discussion the terms *task* and *unlabeled data* will be used interchangeably.

2. *Labeler*. There are a total of M labelers, each may be selected to perform a labeling task for a piece of unlabeled data. The set of labelers is denoted by $\mathscr{M} = \{1,2,...,M\}$. A labeler i produces the true label with probability p_i independent of the task, and independent of each other[1]; a more sophisticated task-dependent version is discussed in Section 2.5. This will also be referred to as the quality or accuracy of this labeler. We will assume no two labelers are exactly the same, i.e., $p_i \neq p_j, \forall i \neq j$ and $0 < p_i < 1, \forall i$. These quantities are unknown to the user *a priori*. We will also assume that the accuracy of the collection of labelers satisfies $\sum_{i=1}^{M} \frac{p_i}{M} > \frac{1}{2}$. The justification and implication of this assumption are discussed in more detail in Section 2.2.3.

Our learning system works in discrete time steps $t = 1,2,...,T$. At time t, a task $k \in \mathscr{K}$ arrives to be labeled, where $\mathscr{K}$ could be either a finite or infinite set. For simplicity of presentation, we will assume that a single task arrives at each time, and that the labeling outcome is binary: 1 or 0; however, both assumptions can be fairly easily relaxed[2]. For task k, the user selects a subset $S_t \subseteq \mathscr{M}$ to label it. The label generated by labeler $i \in S_t$ for data k at time t is denoted by $L_i(t)$.

[1] Such assumption of independency is made to simplify the analysis and presentation. In practice when labelers are correlated, machineries from correlated MAB literature can be borrowed. This merits a future study.

[2] Indeed in our experiment shown later in Section 2.8, our algorithm is applied to a non-binary multi-label case.

The set of labels $\{L_i(t)\}_{i\in S_t}$ generated by the selected labelers then need to be combined to produce a single label for the data; this is often referred to as the information aggregation phase. Since we have no prior knowledge on the labelers' accuracy, we will apply the simple majority voting rule over the set of labels; later we will also examine a more sophisticated weighted majority voting rule. Mathematically, the majority voting rule at time t leads to the following label output:

$$L^*(t) = \text{argmax}_{l\in\{0,1\}} \sum_{i\in S_t} 1\{L_i(t) = l\} \ , \tag{2.1}$$

with ties (i.e., $\sum_{i\in S_t} 1\{L_i(t) = 0\} = \sum_{i\in S_t} 1\{L_i(t) = 1\}$) broken randomly.

Denote by $\pi(S_t)$ the probability of correct labeling outcome following the simple majority rule above, and we have:

$$\begin{aligned}\pi(S_t) = &\underbrace{\sum_{S:S\subseteq S_t, |S|\geq \lceil \frac{|S_t|+1}{2} \rceil} \prod_{i\in S} p_i \cdot \prod_{j\in S_t\setminus S} (1-p_j)}_{\text{Majority wins}} \\ &+ \underbrace{\frac{\sum_{S:S\subseteq S_t, |S|=\frac{|S_t|}{2}} \prod_{i\in S} p_i \cdot \prod_{j\in S_t\setminus S}(1-p_j)}{2}}_{\text{Ties broken equally likely}} \ .\end{aligned} \tag{2.2}$$

Denote by c_i a normalized cost/payment per sample for labeler i and consider the following linear cost function

$$\mathscr{C}(S) = \sum_{i\in S} c_i, \ S \subseteq \mathscr{M} \ . \tag{2.3}$$

It should be noted that extension of our analysis to other forms of cost functions is feasible, though with more cumbersome notations. Denote

$$S^* = \text{argmax}_{S\subseteq\mathscr{M}} \pi(S) \ , \tag{2.4}$$

thus S^* is the optimal selection of labelers given each individual's accuracy. We also refer to $\pi(S)$ as the utility for selecting the set of labelers S and denote it equivalently as $U(S)$. $\mathscr{C}(S^*)$ will be referred to as the necessary cost per task. In most crowd-sourcing systems the main goal is to obtain high quality labels while the cost accrued is a secondary issue. For completeness, however, we will also analyze the tradeoff between the two. Therefore we shall adopt two objectives when designing efficient online algorithm: pick the best set of labelers while keep the unnecessary cost low.

In fact combinations of both labeling accuracy and costs that are appropriately defined can also serve as our utility function. For instance define $U(S) := \pi(S) - \mathscr{C}(S)$. The analysis is quite similar to what we present in this chapter and will not bother with repeating the details.

2.2.2 Offline optimal selection of labelers

Before addressing the learning problem, we will first take a look at how to efficiently derive the optimal selection S^* given accuracy probabilities $\{p_i\}_{i\in\mathscr{M}}$. This will be a crucial step repeatedly invoked by the learning procedure we develop next, to determine the set of labelers to use given a set of *estimated* accuracy probabilities.

The optimal selection is a function of the values $\{p_i\}_{i\in\mathscr{M}}$, the aggregation rule used to compute the final label, and the unit cost c_i. While there is a combinatorial number of possible selections, the next two results combined lead to a very simple and linear-complexity procedure in finding the optimal S^*.

Theorem II.1. *Under the simple majority vote rule, the optimal number of labelers $s^* = |S^*|$ must be an odd number.*

Theorem II.2. *The optimal set S^* is monotonic, i.e., if we have $i \in S^*$ and $j \notin S^*$ then we must have $p_i > p_j$.*

Proofs of the above two theorems can be found in the appendices. Based on above results given a set of accuracy probabilities, the optimal selection under the majority vote

rule consists of the top s^* (an odd number) labelers with the highest quality; we only need to compute s^*, which has a linear complexity of $O(M/2)$. A set that consists of the highest m labelers will be referred to as a *m-monotonic set*, and denoted as $S^m \subseteq \mathscr{M}$.

2.2.3 The lack of ground-truth

As mentioned, a key difference between our model and many other studies on crowdsourcing as well as the basic framework of MAB problems is that we lack ground-truth in our system; we elaborate on this below. In a standard MAB setting, when a player (the user in our scenario) selects a set of arms (labelers) to activate, she immediately finds out the rewards associated with those selected arms. This information allows the player to collect statistics on each arm (e.g., sample mean rewards) which is then used in her future selection decisions. In our scenario, the user sees the labels generated by each selected labeler, but does not know which ones are true. In this sense the user does not find out about her reward immediately after a decision; she can only do so probabilistically over a period of time through additional estimation devices. This constitutes the main conceptual and technical difference between our problem and the standard MAB problem.

Given the lack of ground-truth, the crowdsourcing system is only useful if the average labeler is more or less trustworthy. For instance, if a majority of the labelers produce the wrong label most of the time, unbeknownst to the user, then the system is effectively useless, i.e., the user has no way to tell whether she could trust the outcome so she might as well abandon the system. It is therefore reasonable to have some trustworthiness assumption in place. Accordingly, we have assumed that $\bar{p} := \sum_{i=1}^{M} \frac{p_i}{M} > 1/2$, i.e., the average labeling quality is higher than 0.5; this is a common assumption in the crowd-sourcing literature (see e.g., [22]). Note that this is a fairly mild assumption: not all labelers need to have accuracy $p_i > 0.5$ or near 0.5; some labeler may have arbitrarily low quality (~ 0) as long as it is in the minority. When $\bar{p} := \sum_{i=1}^{M} \frac{p_i}{M} \leq 1/2$, which is often referred to as the case when majority people are wrong [63], it is possible to apply certain machine learn-

ing approach on the set of collected labels to determine whether evidence exists indicating the crowd is on the whole mis-leading. More advanced and formal reporting mechanisms have been developed to elicit the true answer, see e.g., the well known Bayesian Truth Serum (BTS) algorithm [62]. However, such mechanisms usually require reporting more information besides the label output. In this study we shall limit ourselves to the simpler case $\bar{p} > 0.5$ whereby only label output needs to be reported; similar problems for the case $\bar{p} < 0.5$ warrants a separate study.

Denote by X_i a binomial random variable with parameter p_i to model labeler i's outcome on a given task: $X_i = 1$ if her label is correct and 0 otherwise. Using Chernoff Hoeffding's inequality we have

$$\begin{aligned} & P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2) = 1 - P(\frac{\sum_{i=1}^{M} X_i}{M} \leq 1/2) \\ & = 1 - P(\frac{\sum_{i=1}^{M} X_i}{M} - \bar{p} \leq 1/2 - \bar{p}) \\ & \geq 1 - e^{-2M\cdot(\bar{p}-1/2)^2} . \end{aligned}$$

Define $a_{\min} := P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2)$; note this is the probability that a simple majority vote over the M labelers is correct. Therefore, if $\bar{p} > 1/2$ and further $M > \frac{\log 2}{2(\bar{p}-1/2)^2}$, then $1 - e^{-2M\cdot(\bar{p}-1/2)^2} > 1/2$, meaning a simple majority vote would be correct most of the time. Throughout the paper we will assume both these conditions are true. We will also have the following fact:

$$P(\frac{\sum_{i\in S^*} X_i}{|S^*|} > 1/2) \geq P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2),$$

where the inequality is due to the definition of the optimal set S^*.

2.3 Learning the Optimal Labeler Selection

In this section we present an online learning algorithm LS_OL that over time learns each labeler's accuracy, which it then uses to compute an estimated optimal set of labelers using the properties given in the previous section.

2.3.1 An online learning algorithm LS_OL

The algorithm consists of two types of time steps, exploration and exploitation, as is common to online learning. However, the exploration step design is complicated by the additional estimation due to the lack of ground truth revelation. Specifically, a set of tasks will be designated as "testers" and may be repeatedly assigned to the same labeler in order to obtain sufficient results used for estimating her label quality. This can be done in one of two ways depending on the nature of the tasks. For tasks like survey questions (with binary answers), a labeler may indeed be prompted to answer the same question (or equivalent variants with alternative wording) multiple times, usually not in succession, during the survey process. This is a common technique used by survey designers for quality control by testing whether a participant answers questions randomly or consistently, whether a participant is losing attention over time, and so on, see e.g., [64]. For tasks like labeling images, a labeler may be given identical images repeatedly or each time with added small iid noise.

With the above in mind, the algorithm conceptually proceeds as follows. A condition is checked to determine whether the algorithm should explore or exploit in a given time step. If it is to exploit, then the algorithm selects the best set of labelers based on current quality estimates to label the arriving task. If it is to explore, then the algorithm will either assign an old task (an existing tester) or the new arriving task (which then becomes a tester) to the set of labelers $\mathscr{M}$ depending on whether all existing testers have been labeled enough number of times. Because of the need to repeatedly assign an old task, some new tasks will not be immediately assigned (those arriving during an exploration step while an old task remains

under-sampled). These tasks will simply be given a random label (with error probability $1/2$) but their numbers are limited by the frequency of an exploration step ($\sim \log^2 T$), as we shall detail later.

Before proceeding to a more precise description of the algorithm, a few additional notions are in order. Denote the n-th label outcome (via majority vote over M labelers in exploration) for task k by $y_k(n)$. Denote by $y_k^*(N)$ the label obtained using majority rule over the N label outcomes $y_k(1), y_k(2), \cdots, y_k(N)$, and $1\{\cdot\}$ as the indicator function:

$$y_k^*(N) = \begin{cases} 1, & \frac{\sum_{n=1}^{N} 1\{y_k(n)=1\}}{N} > 0.5 \\ 0, & \text{otherwise} \end{cases} , \tag{2.5}$$

with ties broken randomly. It is this majority label after N tests on a tester task k that will be used to analyze different labeler's performance. As we show later in algorithm design, a tester task is always assigned to all labelers for labeling. Therefore these repeated outcomes $y_k(1), y_k(2), \cdots, y_k(N)$ are of the same statistical quality. We will additionally impose the assumption that these outcomes are also independent.

Denote by $E(t)$ the set of tasks assigned to the M labelers during explorations up to time t. For each task $k \in E(t)$ denote by $\hat{N}_k(t)$ the number of times k has been assigned. Consider the following random variable defined at each time t:

$$\mathscr{O}(t) = 1\{|E(t)| \leq D_1(t) \text{ or } \exists k \in E(t) \text{ s.t. } \hat{N}_k(t) \leq D_2(t)\} ,$$

where

$$D_1(t) = \frac{1}{(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2} \cdot \log t ,$$
$$D_2(t) = \frac{1}{(a_{\min} - 0.5)^2} \cdot \log t ,$$

and $n(S^m)$ is the number of all possible majority subsets (for example when $|S^m| = 5$, $n(S^m)$

is the number of all possible subset of size being at least 3) of S^m, ε a bounded constant, and α a positive constant such that $\alpha < \frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)}$. Note that $\mathscr{O}(t)$ captures the event whether an insufficient number of tasks have been assigned under exploration or whether any task has been assigned insufficient number of times in exploration.

Our online algorithm for labeler selection is formally stated as in Fig. 2.1. The above

---

Online Labeler Selection: LS_OL

1: Initialization at $t = 0$: Initialize the estimated accuracy $\{\tilde{p}_i\}_{i \in \mathscr{M}}$ to some value in $[0,1]$; denote the initialization task as k, set $E(t) = \{k\}$ and $\hat{N}_k(t) = 1$.

2: At time t a new task arrives: If $\mathscr{O}(t) = 1$, the algorithm explores.

2.1: If there is no task $k \in E(t)$ such that $\hat{N}_k(t) \leq D_2(t)$, then assign the new task to $\mathscr{M}$ and update $E(t)$ to include it and denote it by k; if there is such a task, randomly select one of them, denoted by k, to $\mathscr{M}$. $\hat{N}_k(t) := \hat{N}_k(t) + 1$; obtain the label $y_k(\hat{N}_k(t))$;

2.2: Update $y_k^*(\hat{N}_k(t))$ (using the alternate indicator function $I(\cdot)$):

$$y_k^*(\hat{N}_k(t)) = 1\{\frac{\sum_{\hat{t}=1}^{\hat{N}_k(t)} y_k(\hat{t})}{\hat{N}_k(t)} > 0.5\} \ .$$

2.3: Update labelers' accuracy estimate $\forall i \in \mathscr{M}$:

$$\tilde{p}_i = \frac{\sum_{k \in E(t), k \text{ arrives at time } \hat{t}} 1\{L_i(\hat{t}) = y_k^*(\hat{N}_k(t))\}}{|E(t)|} \ .$$

3: Else if $\mathscr{O}(t) = 0$, the algorithm exploits and computes:

$$S_t = \text{argmax}_m \tilde{U}(S^m) = \text{argmax}_{S \subseteq \mathscr{M}} \tilde{\pi}(S) \ ,$$

which is solved using the linear search property, but with the current estimates $\{\tilde{p}_i\}$ rather than the true quantities $\{p_i\}$, resulting in estimated utility $\tilde{U}()$ and $\tilde{\pi}()$. Assign the new task to those in S_t.

4: Set $t = t + 1$ and go to Step 2.

Figure 2.1: Description of LS_OL

algorithm can either go on indefinitely or terminate at some time T. As we show below the

performance bound on this algorithm holds uniformly in time so it does not matter when it terminates.

2.3.2 Main results

The standard metric for evaluating an online algorithm in the MAB literature is *regret*, the difference between the performance of an algorithm and that of a reference algorithm which often assumes foresight or hindsight. The most commonly used is the *weak regret* measure with respect to the best single-action policy assuming a priori knowledge of the underlying statistics. In our problem context, this means to compare our algorithm to the one that always uses the optimal selection S^*. It follows that this weak regret, up to time T, is given by

$$R(T) = T \cdot U(S^*) - E\left[\sum_{t=1}^{T} U(S_t)\right] ,$$
$$R_{\mathscr{C}}(T) = E\left[\sum_{t=1}^{T} \mathscr{C}(S_t)\right] - T \cdot \mathscr{C}(S^*) ,$$

where S_t is the selection made at time t by our algorithm; if t happens to be an exploration then $S_t = \mathscr{M}$. $R(T)$ captures the regret for the learning algorithm while $R_{\mathscr{C}}(T)$ is the one for cost. Define:

$$\Delta_{\max} = \max_{S \neq S^*} U(S^*) - U(S),\ \delta_{\max} = \max_{i \neq j} |p_i - p_j| ,$$
$$\Delta_{\min} = \min_{S \neq S^*} U(S^*) - U(S),\ \delta_{\min} = \min_{i \neq j} |p_i - p_j| .$$

ε is a constant such that $\varepsilon < \min\{\frac{\Delta_{\min}}{2}, \frac{\delta_{\min}}{2}\}$. For analysis we assume $U(S^i) \neq U(S^j)$ if $i \neq j$. Define the sequence $\{\beta_n\}$: $\beta_n = \sum_{t=1}^{\infty} \frac{1}{t^n}$. Our main theorem is stated as follows.

Theorem II.3. *The regrets can be bounded uniformly in time:*

$$R(T) \leq \frac{U(S^*)}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2} \cdot \log^2(T) + \Delta_{\max}(2 \sum_{\substack{m=1 \\ m\ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon}\beta_{2-z}) \ , \tag{2.6}$$

$$R_{\mathscr{C}}(T) \leq \frac{\sum_{i \notin S^*} c_i}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2} \cdot \log T + \frac{\sum_{i \in \mathscr{M}} c_i}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2} \cdot \log^2(T) + (M - |S^*|) \cdot (2 \sum_{\substack{m=1 \\ m\ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon}\beta_{2-z}) \ , \tag{2.7}$$

where $0 < z < 1$ *is a positive constant.*

First note that the regret is nearly logarithmic in T and therefore it has zero average regret as $T \to \infty$; such an algorithm is often referred to as a zero-regret algorithm. Secondly the regret bound is inversely related to the minimum accuracy of the crowd (through $a_{\min}$). This is to be expected: with higher accuracy (a larger $a_{\min}$) of the crowd, crowdsourcing generates ground-truth outputs with higher probability, and hence the learning process could be accelerated. Finally, the bound also depends on $\max_m m \cdot n(S^m)$ which is roughly on the order of $O(\frac{2^m\sqrt{m}}{\sqrt{2\pi}})$.

2.3.3 Regret analysis of LS_OL

We now outline key steps in the proof of the above theorem. This involves a sequence of lemmas; the proofs of most can be found in the appendix. There are a few that we omit for brevity; in those cases sketches are provided.

Step 1: We begin by noting that the regret consists of that arising from the exploration

phase and from the exploitation phase, denoted by $R_e(T)$ and $R_x(T)$, respectively:

$$R(T) = E[R_e(T)] + E[R_x(T)] \ .$$

The following result bounds the first element of the regret.

Lemma II.4. *The regret up to time T from the exploration phase can be bounded as follows:*

$$E[R_e(T)] \leq U(S^*) \cdot (D_1(T) \cdot D_2(T)) \ . \tag{2.8}$$

We see the regret depends on the exploration parameters as product. This is because for tasks arriving in exploration steps, we assign it at least $D_2(T)$ times to the labelers; each time when re-assignment occurs, a new arriving task is given a random label while under an optimal scheme each missed new task means a utility of $U(S^*)$.

Step 2: We now bound the regret arising from the exploitation phase as a function of the number of times the algorithm uses a sub-optimal selection when the ordering of the labelers is correct, and the number of times the estimates of the labelers' accuracy result in a wrong ordering. The proof of the lemma below is omitted as it is fairly straightforward.

Lemma II.5. *For the regret from exploitation we have:*

$$E[R_x(T)] \leq \Delta_{\max} \left(E[\sum_{t=1}^{T} (\mathscr{E}_1(t) + \mathscr{E}_2(t))] \right) \ . \tag{2.9}$$

Here $\mathscr{E}_1(t) = I_{S_t \neq S^*}$, conditioned on correct ordering of labelers, records whether the a sub-optimal section (other than S^*) was used at time t based on the current estimates $\{\tilde{p}_i\}$. $\mathscr{E}_2(t)$ records whether at time t the set $\mathscr{M}$ is sorted in the wrong order because of erroneous estimates $\{\tilde{p}_i\}$.

Step 3: We proceed to bound the two terms in (2.9) separately. In this part of the analysis we only consider those times t when the algorithm exploits.

Lemma II.6. *At time t we have:*

$$E[\mathscr{E}_1(t)] \leq \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \tag{2.10}$$

The idea behind the above lemma is to use a union bound over all possible events where the wrong set is chosen when the ordering of the labelers is correct according to their true accuracy.

Lemma II.7. *At time t we have:*

$$E[\mathscr{E}_2(t)] \leq M(\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \tag{2.11}$$

Step 4: Summing up all results and rearranging terms lead to the theorem. Specifically,

$$\begin{aligned} E[R_x(T)] &\leq \Delta_{\max} \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} 2 \sum_{t=1}^{T} m \cdot n(S^m) \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) + \Delta_{\max} \cdot M \cdot \sum_{t=1}^{T} (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \\ &\leq 2 \cdot \Delta_{\max} \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) \cdot \sum_{t=1}^{\infty} (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) + \Delta_{\max} \cdot M \cdot \sum_{t=1}^{\infty} (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \\ &= \Delta_{\max} (2 \cdot \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon} \beta_{2-z}) \ . \end{aligned}$$

Since $\beta_{2-z} < \infty$ for $z < 1$, we have bounded the exploitation regret by a constant. Summing over all terms in $E[R_e(T)]$ and $E[R_x(T)]$ we obtain the main theorem.

2.3.4 Cost analysis of LS_OL

We now analyze the cost regret. Following similar analysis we first note that it can be calculated separately for the exploration and exploitation steps.

For *exploration* steps we know the cost regret is bounded by

$$\sum_{i \notin S^*} c_i \cdot D_1(T) + \sum_{i \in \mathscr{M}} c_i \cdot D_1(T) \cdot (D_2(T) - 1)$$

where the second term is due to the fact for all costs associated with task re-assignments are treated as additional costs.

For *exploitation* steps the additional cost is upper-bounded by

$$(M - |S^*|) \cdot E[\sum_{t=1}^{T} (\mathscr{E}_1(t) + \mathscr{E}_2(t))] \ .$$

Based on previous results we know the cost regret $R_{\mathscr{C}}(T)$ will look similar to $R(T)$ with both terms bounded by either a log term or a constant. Plug in $D_1(T), D_2(T), E[\sum_{t=1}^{T}(\mathscr{E}_2(t)]$, $E[\sum_{t=1}^{T} \mathscr{E}_2(t)]$ we establish the regret for $R_{\mathscr{C}}(T)$ as claimed in our main result.

2.3.5 Discussion

We end this section with a discussion on how to relax a number of assumptions adopted in our analytical framework.

IID re-assignments

The first concerns the re-assignment of the same task (or iid copies of the same task) and the assumption that the labeling outcome each time is independent. In the case where iid copies are available, this assumption is justified. In the case when the exact same task must be re-assigned, enforcing a delay between successive re-assignments can make this assumption more realistic. Suppose the algorithm imposes a random delay τ_k, a positive random variable uniformly upper-bounded by $\tau_k \leq \tau_{\max}, \forall k$. Then following similar anal-

ysis we can show the upper bound for regret is at most $\tau_{\max}$ times larger, i.e., it can be bounded by $\tau_{\max} \cdot R(T)$, where $R(T)$ is as defined in Eqn. (2.6).

Prior knowledge of several constants

The second assumption concern the selection of constant ε by the algorithm and the analysis which requires knowledge on $\Delta_{\min}$ and $\delta_{\min}$. This assumption however can be removed by using a decreasing sequence ε_t. This is a standard technique that has been commonly used in the online learning literature, see e.g., [73]. Specifically, let

$$\varepsilon_t = \frac{1}{\log^{\eta}(t)}, \text{ for some } \eta > 0 \, .$$

Replace $\log(t)$ with $\log^{1+2\eta}(t)$ in $D_1(t)$ and $D_2(t)$ it can be shown that there exists T_0 s.t. $\varepsilon_{T_0} < \varepsilon$. Thus the regret associated with using an imperfect ε_t is bounded by $\sum_{t=1}^{T_0} \frac{2}{\log^{\eta} t} = C_{T_0}$, a constant.

Detecting bad/incompetent labelers

The last assumption we discuss concerns the quality of the set of labelers, assumed to satisfy the condition $\min\{a_{\min}, \bar{p}\} > 0.5$. Recall the bounds were derived based on this assumption and are indeed functions of $a_{\min}$. While in this discussion we will not seek to relax this assumption, below we describe a simple "vetting" procedure that can be easily incorporated into the LS_OL algorithm to quickly detect and filter out outliner labelers so that over the remaining labelers we can achieve higher values of $a_{\min}$ and $\bar{p}$, and consequently a better bound. The procedure keeps count of the number of times a labeler differs from the majority opinion during the exploration steps, then over time we can safely eliminate those with high counts.

The justification behind this procedure is as follows. Let random variable $Z_i(t)$ denote whether labeler i agrees with the majority vote in labeling a task in a given assignment in

exploration step t: $Z_i(t) = 1$ if they disagree and 0 otherwise. Then

$$\begin{aligned} P(Z_i(t) = 1) &= (1 - p_i) \cdot \pi(\mathscr{M}) + p_i \cdot (1 - \pi(\mathscr{M})) \\ &= \pi(\mathscr{M}) + p_i \cdot (1 - 2\pi(\mathscr{M})) \ , \end{aligned} \tag{2.12}$$

where recall $\pi(\mathscr{M})$ is the probability the majority vote is correct. Under the same assumption $a_{\min} > 1/2$ we have $\pi(\mathscr{M}) > 1/2$, and it follows that $P(Z_i(t) = 1)$ is decreasing in p_i, i.e., the more accurate a labeler is, the less likely she is going to disagree with the majority vote, as intuition would suggest. It further follows that for $p_i > p_j$ we have

$$E[Z_i(t) - Z_j(t)] = \varepsilon_{ij} < 0 \ .$$

Similarly, if we consider the disagreement counts over N assignments, $\sum_{t=1}^{N} Z_j(t)$, then for $p_i > p_j$ we have

$$\begin{aligned} &P(\sum_{t=1}^{N} Z_i(t) < \sum_{t=1}^{N} Z_j(t)) = P(\frac{\sum_{t=1}^{N}(Z_i(t) - Z_j(t))}{N} < 0) \\ &= P(\frac{\sum_{t=1}^{N}(Z_i(t) - Z_j(t))}{N} - \varepsilon_{ij} < -\varepsilon_{ij}) \\ &\geq 1 - e^{-2\varepsilon_{ij}^2 N} \ . \end{aligned} \tag{2.13}$$

That is, if the number of assignments N is on the order of $\sim \log T/\varepsilon_{ij}^2$, then the above probability approaches 1, which bounds the likelihood that labeler i (higher quality) will have fewer number of disagreements than labeler j. Therefore if we rank and order the labelers in decreasing order of their accumulated disagreement counts then the worst labeler is going to be at the top of the list with increasing probability (approach 1). If we eliminate the worst performer, then we improve $a_{\min}$ which leads to better bounds as shown in Eqn. (2.6) and Eqn. (2.7). Compared to the exploration steps detailed earlier where in order to differentiate labelers' expertise (by estimating p_i), $O(\log^2 T)$ assignments are, here we only

need $O(\log T)$ assignments, a much faster process. In practice, we could decide to remove the worst labeler when the probability of not making an error (per Eqn. (2.13)) exceeds a certain threshold.

2.4 Weighted Majority Voting and its regret

The crowd-sourced labeling performance could be further improved by employing more sophisticated majority voting mechanism. Specifically, under our online learning algorithm LS_OL, statistics over each labeler's expertise could be collected with significant confidence; this enables a weighted majority voting mechanism. In this section we analyze the regret of a similar learning algorithm using weighted majority voting.

2.4.1 Weighted Majority Voting

We start with defining the weights. At time t, after observing labels produced by the labelers, we can optimally (*a posteriori*) determine the mostly likely label of the task by solving the following:

$$\text{argmax}_{l\in\{0,1\}} P(L^*(t) = l | L_1(t), ..., L_M(t)) \ . \tag{2.14}$$

Suppose at time t the true label for task k is 1. Then we have,

$$\begin{aligned}
&P(L^*(t) = 1 | L_1(t), ..., L_M(t)) \\
&= \frac{P(L_1(t), ..., L_M(t), L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \\
&= \frac{P(L_1(t), ..., L_M(t) | L^*(t) = 1) \cdot P(L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \\
&= \frac{P(L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \cdot \prod_{i:L_i(t)=1} p_i \cdot \prod_{i:L_i(t)=0} (1 - p_i) \ .
\end{aligned}$$

And similarly we have

$$\begin{aligned}&P(L^*(t)=0|L_1(t),...,L_M(t))\\&=\frac{P(L^*(t)=0)}{P((L_1(t),...,L_M(t))}\cdot\prod_{i:L_i(t)=0}p_i\cdot\prod_{i:L_i(t)=1}(1-p_i)\ .\end{aligned}$$

Following standard hypothesis testing procedure and assuming equal priors $P(L^*(t)=1)=P(L^*(t)=0)$, a true label of 1 can be correctly produced if

$$\prod_{i:L_i(t)=1}p_i\cdot\prod_{i:L_i(t)=0}(1-p_i)>\prod_{i:L_i(t)=0}p_i\cdot\prod_{i:L_i(t)=1}(1-p_i)\ .$$

with ties broken randomly and equally likely. Take $\log(\cdot)$ on both sides and the above condition reduces to

$$\sum_{i:L_i(t)=1}\log\frac{p_i}{1-p_i}>\sum_{j:L_j(t)=0}\log\frac{p_j}{1-p_j}\ .$$

Indeed if $p_1=...=p_M$ the above reduces to $|\{i:L_i(t)=1\}|>|\{i:L_i(t)=0\}|$ which is exactly the simple majority voting. Under the weighted majority voting, each labeler i's decision is modulated by weight $\log\frac{p_i}{1-p_i}$. When $p_i>0.5$, the weight $\log\frac{p_i}{1-p_i}>0$, which may be viewed as an opinion that adds value; when $p_i<0.5$, the weight $\log\frac{p_i}{1-p_i}<0$, an opinion that actually hurts; when $p_i=0.5$ the weight is zero, an opinion that does not count as it mounts to a random guess. The above constitutes the weighted majority voting rule we shall use in a revised learning algorithm and the regret analysis that follow.

Before proceeding to the regret analysis, we again first characterize the optimal labeler set selection assuming known labelers' accuracy. In this case the odd-number selection property no longer holds, but thanks to the monotonicity of $\log\frac{p_i}{1-p_i}$ in p_i we have the same monotonicity property in the optimal set and a linear-complexity solution space.

Theorem II.8. *Under the weighted majority voting and assuming $p_i\geq 0.5,\forall i$, the optimal*

set S^* is monotonic, i.e., if we have $i \in S^*$ and $j \notin S^*$ then we must have $p_i > p_j$.

The assumption that all $p_i \geq 0.5$ is for simplicity in presentation without losing generality. This is because a labeler with $p_i < 0.5$ is equivalent to another with $p_i := 1 - p_i$ by flipping its label (assuming the average labeling quality is higher than 0.5).

2.4.2 Main results

We now analyze the performance of a similar learning algorithm using weighted majority voting. The algorithm LS_OL is modified as follows. Denote by

$$W(S) = \sum_{i \in S} \log \frac{p_i}{1 - p_i}, \ \forall S \subseteq \mathscr{M} \ , \tag{2.15}$$

and $\tilde{W}$ its estimated version when using estimated accuracies $\tilde{p}_i$. Denote by

$$\delta^W_{\min} = \min_{S \neq S', W(S) \neq W(S')} |W(S) - W(S')|$$

and let $\varepsilon < \delta^W_{\min}/2$. At time t (suppose at exploitation phase), the algorithm selects the estimated optimal set S_t. These labelers then return their labels that divide them into two subsets, say S (with one label) and its complement $S_t \backslash S$ (with the other label). If $\tilde{W}(S) \geq \tilde{W}(S_t \backslash S) + \varepsilon$, we will call S the majority set and take its label as the voting outcome. If $|\tilde{W}(S) - \tilde{W}(S_t \backslash S)| < \varepsilon$, we will call them equal sets and randomly select one of the labels as the voting outcome. Intuitively ε serves as a tolerance that helps to remove the error due to inaccurate estimations. In addition, the constant $D_1(t)$ is revised to the follow:

$$D_1(t) = (\frac{1}{\max_m \max\{4C \cdot m, m \cdot n(S^m)\}} - \alpha)^2 \cdot \varepsilon^2 \cdot \log t,$$

where C is a constant satisfying

$$C > \max_i \max\{\frac{1 + \varepsilon/4}{p_i}, \frac{1 - \varepsilon/4}{1 - p_i}, \frac{\varepsilon/4}{p_i}, \frac{\varepsilon/4}{1 - p_i}\} \ .$$

With above modifications in mind, we omit the detailed algorithm description for a concise presentation. We have the following theorem on the regret of this revised algorithm (again $R_{\mathscr{C}}(T)$ possesses a very similar format we omit its detail).

Theorem II.9. *The regret under weighted majority voting can be bounded uniformly in time:*

$$R(T) \leq \frac{U(S^*)}{(\frac{1}{\max_m \max\{4C\cdot m, m\cdot n(S^m)\}} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2} \log^2 T$$
$$+ \Delta_{\max}(2 \cdot \sum_{m=1}^{M} m \cdot n(S^m) + M + \frac{M^2}{2}) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon}\beta_{2-z}) \ .$$

Again the regret is on the order of $O(\log^2 T)$ in time. It has a larger constant compared to that under simple majority voting. However, the weighted majority voting has a better optimal solution, i.e., we are converging slightly slower to a however better target.

The proof of this theorem is omitted for brevity and because most of it parallels with the case of simple majority voting. There is however a main difference: under the weighted majority voting there is additional error in computing the weighted majority vote. Whereas under simple majority we simply find the majority set by counting the number of votes, under weighted majority the calculation of the majority set is dependent on the estimated weights $\log \frac{\tilde{p}_i}{1-\tilde{p}_i}$ which inherits errors in $\{\tilde{p}_i\}$. This additional error, in particular associated with bounding the error of getting

$$\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) < \varepsilon, \text{ when } W(\hat{S}) > W(S/\hat{S})$$

and

$$\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) \geq \varepsilon, \text{ when } W(\hat{S}) = W(S \backslash \hat{S})$$

for set $\hat{S} \subseteq S \subseteq \mathscr{M}$, could be separately bounded using similar methods as shown in the simple majority voting case (bounding estimation error with large enough number of sam-

ples) and can again be factored into the overall bound. This is summarized in the following lemma.

Lemma II.10. *At time t, for set $\hat{S} \subseteq S \subseteq \mathscr{M}$ and its complement $S/\hat{S}$, if $W(\hat{S}) > W(S/\hat{S})$, then $\forall t$ at exploitation phases $\forall\ 0 < z < 1$,*

$$P(\tilde{W}(\hat{S}) - \tilde{W}(S\backslash\hat{S}) < \varepsilon) \leq |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) .$$

Moreover, if $W(\hat{S}) = W(S\backslash\hat{S})$

$$P(|\tilde{W}(\hat{S}) - \tilde{W}(S\backslash\hat{S})| > \varepsilon) \leq |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) .$$

For rest of the proofs please refer to appendix.

2.5 Labelers with different types of tasks

We now discuss an extension where labelers' difference in their quality in labeling varies according to different types of data samples/tasks. For example, some are more proficient with labeling image data while some may be better at annotating audio data. In this case we can use contextual information to capture these differences, where a specific context refers to a different data/task type. There are two cases of interest from a technical point of view: when the space of all context information is finite, and when this space is infinite. We will denote a specific context by w and the set of all contexts as $\mathscr{W}$.

In the case of discrete context information, $|\mathscr{W}| < \infty$ and we can apply the same algorithm to learn, for each combination $\{i, w\}_{i \in \mathscr{M}, w \in \mathscr{W}}$, the pairwise labeler-context accuracy. This extension is rather straightforward except for a longer exploration phase. In fact, since exploration is needed for each labeler i under each possible context w, we may expect the

regret to be $|\mathscr{W}|$ times larger compared to the previous $R(T)$. This indeed can be more precisely established using the same methodology.

The case of continuous context information is more challenging, but can be dealt with using the technique introduced in [2] for bandit problems with a continuum of arms. The main idea is to divide the infinite context information space into a finite but increasing number of subsets. For instance, if we model the context information space as $\mathscr{W} = [0,1]$ then we can divide this unit interval into $v(t)$ sub-intervals:

$$[0, \frac{1}{v(t)}], ..., [\frac{v(t)-1}{v(t)}, 1] \ ,$$

with $v(t)$ being an increasing sequence w.r.t. t. Denote these intervals as $B_i(t)$, $i = 1, 2..., v(t)$, which become more and more fine-grained with increasing t and increasing $v(t)$.

Given these intervals the learning algorithm works as follows. At time t, for each interval $B_i(t)$ we compute the estimated optimal set of labelers by calculating the estimated utility of all subsets of labelers, and this is done over the entire interval $B_i(t)$ (contexts within $B_i(t)$ are viewed as a bundle). If at time t we have context $w_i \in B_i(t)$ then this estimated optimal set is used. The regret of this procedure consists of two parts. The first part is due to selecting a sub-optimal set of labelers for $B_i(t)$ (owing to incorrect estimates of the labelers' accuracy). This part of the regret is bounded by $O(1/t^2)$. The second part of the regret arises from the fact that even if we compute the correct optimal set for interval $B_i(t)$, it may not be optimal for the specific context $w_t \in B_i(t)$. However, when $B_i(t)$ becomes sufficiently small, and under a uniform Lipschitz condition we can bound this part of the regret as well.

Taken together, if we revise the condition for entering the exploration phase (constants $D_1(t)$ and $D_2(t)$) to grow on the order of $O(t^z \log t)$ instead of $\log t$, for some constant $0 < z < 1$, then the regret $R(T)$ in this case is on the order of $T^z \log T$; thus it remains

sub-linear and therefore has a zero average regret, but this is worse than the log bound we can obtain in other cases.

We omit all technical details since they are rather direct extensions combining our previously derived results with the literatures on continuous arms.

2.6 A lower bound on the regret

In this section we establish a lower bound on the regret of our online labeler selection problem.

2.6.1 $O(1)$ reassignment leads to unbounded regret

We first show that a constant number of re-assignments will lead to unbounded regret. Below we establish this by contradiction. Recall that we have used $D_2(t)$ to determine when reassignment is made, and in our algorithm we have used $D_2(t) = O(\log t)$. Suppose instead, $D_2(t)$ is given by T_0, a bounded constant. Let's consider one task with label $\theta \in \{0,1\}$. Denote the test outcomes by $x(1),...,x(T_0)$ (as given by the simple majority voting from all labelers). There are two hypotheses based on $x(\tau), \tau = 1,...,T_0$:

$$H_0 : \text{The label is 1, i.e., } \theta = 1.$$

$$H_1 : \text{The label is 0, i.e., } \theta = 0.$$

Denote by $I(\theta_1, \theta_0)$ the Kullback-Leibler (KL) divergence between two distributions $f_X(x;\theta = 1)$ and $f_X(x;\theta = 0)$, where $f_X(x;\theta)$ denotes the sample distribution with parameter θ, which in our case is the ground-truth label. Denote the vector $[1,...,t]$ by $[t]$. The next lemma is a well established result:

Lemma II.11 (Theorem 2.2, Tysabakov [75], 2009)**.** *The error probability P_e of the above*

hypothesis test up to time t is lower-bounded by

$$P_e \geq \frac{1}{2} \cdot e^{-I(P_{H_0}^{[t]}, P_{H_1}^{[t]})} .$$

Note that in our case

$$\begin{aligned}
&I(P_{H_0}^{[t]}, P_{H_1}^{[t]}) \\
&= E_{H_0}\left[\log \frac{f_X(x(1);\theta=1)}{f_X(x(1);\theta=0)} \cdot \frac{f_X(x(2);\theta=1)}{f_X(x(2);\theta=0)} \cdots \frac{f_X(x(T_0);\theta=1)}{f_X(x(T_0);\theta=0)}\right] \\
&= E_{H_0}\left[\sum_{t=1}^{T_0} \log \frac{f_X(x(1);\theta=1)}{f_X(x(1);\theta=0)}\right] \\
&= I(\theta_1, \theta_0) T_0 .
\end{aligned}$$

Since T_0 is bounded from above, $P_e > 0$, meaning that there is always a positive probability of making the wrong labeling decision. What this further means is that one can always find problem parameters whereby an algorithm will reach incorrect estimates of labelers' qualities which leads to "permanently" sub-optimal selection of labelers, resulting in unbounded regret. This is demonstrated using a counter example shown below.

Suppose we have three labelers with labeling qualities being $p_1 = \frac{1}{2} + \delta, p_2 = \frac{1}{2} + \varepsilon, p_3 = \frac{1}{2} + \varepsilon - o(1)$, where $o(1)$ is an arbitrarily small quantity, and δ, ε satisfies $0 < \varepsilon < \delta < \frac{1}{2}$. This setting satisfies the assumptions we made throughout the paper:

$$\begin{aligned}
&p_1 > p_2 > p_3. \\
&0 < p_i < 1, i = 1,2,3. \\
&\frac{p_1 + p_2 + p_3}{3} > 1/2.
\end{aligned}$$

Moreover the labeling accuracy using simple majority voting is as follows:

$$\pi(p_1,p_2,p_3) \geq \frac{1}{2}+\frac{\delta}{2}+\varepsilon-2\delta\varepsilon^2 > 1/2.$$

For this 3-labeler problem we have the following proposition.

Proposition II.12. *With $P_e > 0$, we can always find a δ such that in the above example, the regret of any online learning algorithm is on the order of $O(T)$.*

2.6.2 $D_2(t) > O(1)$

Now consider the case with $D_2(t) > O(1)$. Using Chernoff bound we know the following holds

$$\begin{aligned} P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} > 1/2) &= 1 - P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} \leq 1/2) \\ &= 1 - P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} - \bar{p} \leq 1/2 - \bar{p}) \\ &\geq 1 - e^{2(\bar{p}-1/2)^2 D_2(t)} \ . \end{aligned}$$

Notice we have used $\bar{p}$ to denote the average labeling accuracy, which is strictly larger than $1/2$. Thus $P_e \leq e^{2(\bar{p}-1/2)^2 D_2(t)} \to 0$. We have the following proposition.

Proposition II.13. *With $D_2(t) > O(1)$ we have*

$$R(T) \geq O(\frac{\log T \cdot D_2(t)}{I(p_1,p_2) - \frac{C_1}{(C_2+\delta)^2}\delta}) \ , \tag{2.16}$$

where $C_1, C_2 > 0$ are constants and

$$\delta = \frac{P_e}{1-P_e} \ .$$

Also from above results we see when $D_2(t) = O(1)$, it cannot be guaranteed that

$$I(p_1, p_2) - \frac{C_1}{(C_2 + \delta)^2}\delta > 0 \ ,$$

under which case the bound becomes meaningless. This is another implication why we need $D_2(t) > O(1)$.

2.7 A refined upper bound to match

We refine our algorithm and relax the requirement on setting $D_2(t) := O(\log t)$ to any $D_2(t) > O(1)$. We have the following results to match this lower bound. In particular we prove the following results.

Theorem II.14. *We can refine the upper bound of LS_OL to the following.*

$$R(T) \leq O(\log T D_2(T)) \ . \tag{2.17}$$

2.7.1 Tightness of $O(\log^2 T)$ for a type of policies

Note we previously had a $O(\log^2 T)$ upper bound. We show this bound is tight for a certain category of policies. We first define polynomially converging policy for our labeler selection problem.

Definition II.15. A polynomially converging policy is a policy that there exists a $z > 0$ such that the error in estimating ground-truth label at time t is decreasing polynomially

$$P_e \leq O(\frac{1}{t^z}) \ . \tag{2.18}$$

For polynomially converging policy, intuitively we need

$$e^{-2(\bar{p}-1/2)^2 D_2(t)} = O(\frac{1}{t^z}) \ ,$$

and again following Lemma II.11, we could successfully show that the number of reassignments needs to satisfy that $D_2(t) \geq \log t$. Then $O(\log t^2)$ is tight for polynomially decreasing policies.

2.8 Experiment Results

In this section we validate the proposed algorithms with a few examples using both simulation and real data.

2.8.1 Simulation study

Our first setup consists of $M = 5$ labelers, whose quality $\{p_i\}$ are randomly and uniformly generated to satisfy a preset $a_{\min}$ as follows: select $\{p_i\}$ randomly between $[a_{\min}, 1]$. Note that this is a simplification because not all $\{p_i\}$ need to be above $a_{\min}$ for the requirement to hold. An example of these are shown in Table 2.1 (for $a_{\min} = 0.6$) but remain unknown to the labelers. A task arrives at each time t. We assume a unit labeling cost

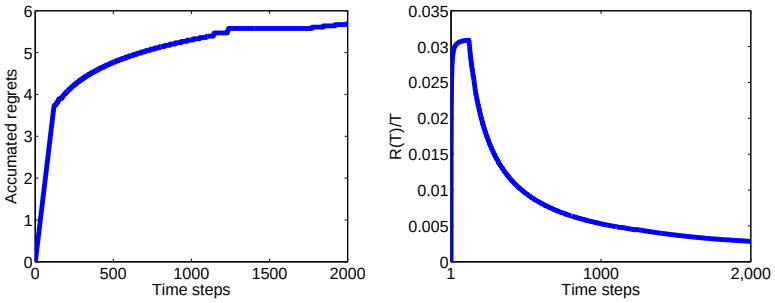
	L_1	L_2	L_3	L_4	L_5
p_i	0.763	0.781	0.625	0.783	0.727

Table 2.1: Sample of simulation setup

$c = 0.02$. The experiments are run for a period of $T = 2{,}000$ time units (2,000 tasks in total). The results shown below are the average over 100 runs. Denote by G_1, G_2 the *exploration constants* concerning the two constants (in $D_1(t)$ and $D_2(t)$) that control the exploration part of the learning. G_1, G_2 are set to be sufficiently large based on the other

parameters:

$$(G_1, G_2) = \left(\frac{1}{(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2}, \frac{1}{(a_{\min} - 0.5)^2}\right) .$$



(a) Accumulative regret (b) Average regret $R(T)/T$

Figure 2.2: Regret of the LS_OL algorithm.

We first show the accumulative and average regret under the simple majority voting rule in Fig. 2.2. From the set of figures we observe a logarithmic increase of accumulated regret and correspondingly a sub-linear decrease for its average quantity. The cost regret $R_{\mathscr{C}}(T)$ has a very similar trend as mentioned earlier (recall the regret terms of $R_{\mathscr{C}}(T)$ align well with the those in $R(T)$) and is thus not show here. We then compare the performance with labeler selection to the naive crowd-sourcing algorithm, by taking a simple majority vote over the whole set of labelers each time. This is plotted in Fig. 2.3 in terms of the average reward at each t. There is a clear performance improvement after an initialization period (where training happens).

In addition to the logarithmic growth, we are interested in knowing how the performance is affected by the inaccuracy of the crowd expertise. These results are shown in Fig. 2.4. We observe the effect of different choices of $a_{\min} = 0.6, 0.7, 0.8$. As expected, we see when $a_{\min}$ is small, the verification process of the labels takes more samples to become accurate. Therefore in the process more error is introduced in the estimation of the labelers' qualities, which results in slower convergence.

We next compare the performance between simple majority voting and weighted ma-

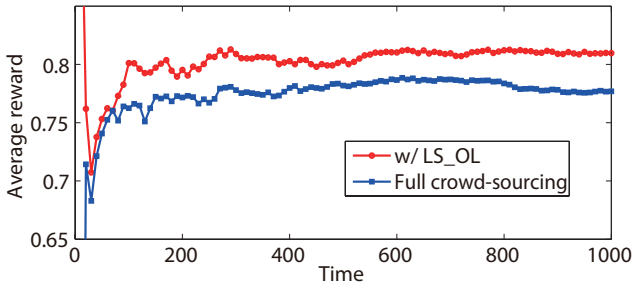


Figure 2.3: Performance comparison: labeler selection v.s. full crowd-sourcing (majority voting)

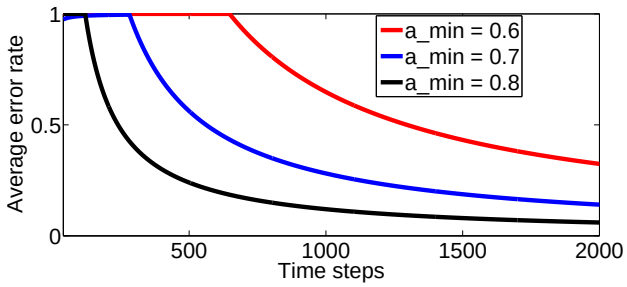


Figure 2.4: Effect of $a_{\min}$: higher $a_{\min}$ leads to much better performance.

jority voting (both with LS_OL). One example trace of accumulated reward comparison is shown in Fig. 2.5; the advantage of weighted majority voting can be seen clearly. We then repeat the set of experiments and average the results over 500 funs; the comparison is shown in Table 2.2 under different number of candidate labelers (all of their labeling qualities are uniformly generated).

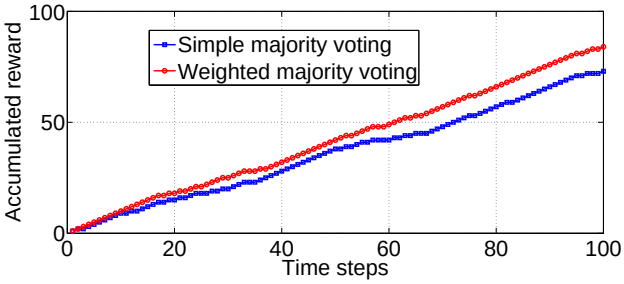


Figure 2.5: Comparing weighted and simple majority voting within LS_OL.

Average reward/ $M :=$	5	10	15	20
Full crowd-sourcing (majority voting)	0.5154	0.5686	0.7000	0.7997
Majority voting w/ LS_OL	0.8320	0.9186	0.9434	0.9820
Weighted majority voting w/ LS_OL	**0.8726**	**0.9393**	**0.9641**	**0.9890**

Table 2.2: Performance comparison. There is a clear gap between crowd-sourcing results with and without using LS_OL.

2.8.2 Study on a real AMT dataset

We also apply our algorithm to a dataset shared at [3]. This dataset contains 1,000 images each labeled by the same set of 5 AMTs. The labels are on the scale from 0 to 4 indicating how many scenes are seen from each image, such as filed, airport, animal, etc. A label of 0 implies no scene can be discerned. Besides the ratings from the AMTs, there is a second dataset from [3] summarizing keywords for scenes of each image. We also analyze this second dataset and count the number of unique descriptors for each image and use this count as the ground-truth or gold standard, to which the results from AMT are compared.

We start with showing the number of disagreements each AMT has with the group over the 1000 images. The total numbers of disagreement of the 5 AMTs are shown in Table 2.3, while Fig. 2.6 shows the cumulative disagreement over the set of images ordered by their numerical indices in the database. It is quite clear that AMT 5 shows significant and consistent disagreement with the rest. AMT 3 comes next while AMTs 1, 2, and 4 are clearly more in general agreement.

	AMT1	AMT2	AMT3	AMT4	AMT5
# of disagree	348	353	376	338	441

Table 2.3: Total number of disagreement each AMT has

The images are not in sequential order, as the original experiment was not done in an online fashion. To test our algorithm, we will continue to use their numerical indices to order them as if they arrived sequentially in time and feed them into our algorithm. By doing so we essentially test the performance of conducting this type of labeling tasks online whereby the administrator of the tasks can dynamically alter task assignments to obtain

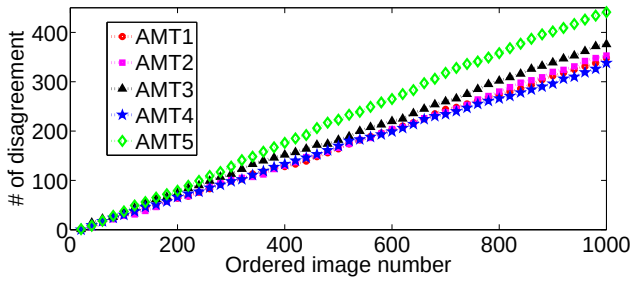


Figure 2.6: Cumulated number of disagreements.

better results. In this experiment we use LS_OL with majority voting and with the addition of the detection and filtering procedure discussed in Section 2.3.5, which is specified to eliminate the worst labeler after a certain number of steps such that the error in the rank ordering is less than 0.1. The algorithm otherwise runs as described earlier. Indeed we see this happen around step 90, as highlighted in Fig. 2.7 along with a comparison to using the full crowd-sourcing method with majority voting. The algorithm also eventually correctly estimates the best set to consist of AMTs 1, 2, and 4.

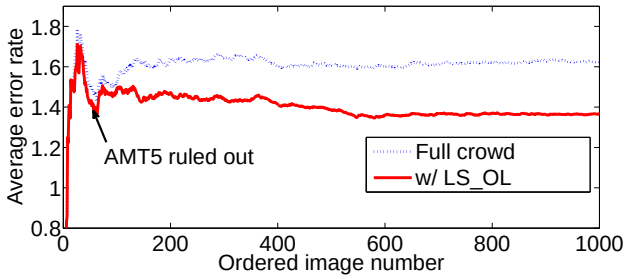


Figure 2.7: Performance comparison : an online view

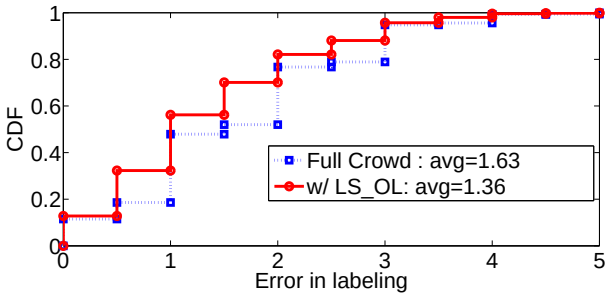


Figure 2.8: Performance comparison : a summery

All images' labeling error as compared to the ground truth at the end of this process

is shown as a CDF (error distribution over the images) in Fig. 2.8; note the errors are discrete due to the discrete labels. It is also worth noting that under our algorithm the cost is much lower because AMT 5 was quickly eliminated, while AMT 4 was only used very infrequently once the optimal set has been accurately inferred.

2.9 Concluding remarks

To our best knowledge, this is the first work formalizing and addressing the issue of labeler quality in an online fashion for the crowd-sourcing problem and proposing solutions with performance guarantee. We developed and analyzed an online learning algorithm that can differentiate between high and low quality labelers over time and select the best set for labeling tasks with $O(\log T \cdot D_2(T))$ regret uniform in time, where $D_2(T)$ is an arbitrary function with $D_2(T) > O(1)$. We also provided an order-matching lower bound. In addition, we showed how performance could be further improved by utilizing more sophisticated voting techniques. We discussed the applicability of our algorithm to more general cases when labelers' quality varies with contextually different tasks and discuss how to detect and remove malicious labelers when there is a lack of ground-truth. We validated our results via both synthetic and real world AMT data, alongside numerous observations and discussions.

CHAPTER III

Crowd-learn: Online recommendation system

3.1 Introduction

In this chapter we analyze the following learning problem in the context of crowd-sourcing such as in a recommendation system: M users each faces N options, such as those in restaurants, movies, etc. In a discrete time setting, at each step a user chooses K out of the N options, and receives randomly generated rewards, whose statistics depend on the options chosen as well as the user herself, but are unknown to the user. The objective of each user is to maximize her expected total reward (e.g., overall satisfaction of watching movies) over a certain time horizon through an online learning process, i.e., a sequence of exploration (sampling the return of each option) and exploitation (selecting empirically good options) steps. Taken separately, an individual user's learning process may be cast as a standard multi-armed bandit (MAB) problem which has been extensively studied, see e.g., [44, 4, 5, 73].

Our interest, however, is on how an individual's learning process may be affected by "second-hand learning", i.e., by observing how others in the group act and what they recommend. The challenge is that what is considered desirable options for one may be undesirable for another (think of restaurant choices: one Yelp user may favor large establishments with extensive menus while another may favor small out-of-the-way places), and this difference in preference is in general unknown *a priori*. Moreover, even when two users

happen to have the same preference (e.g., they agree one option is better than the other), they may differ in their absolute valuation of each individual option (again: two Yelp users may agree restaurant A is better than B, but one user may rate them 5 and 4 stars respectively, while the other 4 and 3 stars, respectively). Consequently if an individual wants to take others' actions into account in her own learning process, she would need to figure out whether their preferences are aligned. This raises the interesting question of whether learning from crowdsourcing is indeed beneficial to an individual, and if so what type of learning algorithm can effectively utilize the crowdsourced data in addition to one's direct observations. This is what we aim to address in this paper. A related subject is learning with side information, see e.g., [79, 55, 45]. In contrast, we do not require such statistical information; instead we examine how a user can estimate and learn from the crowdsourced data.

We will assume that users are heterogeneous in general, i.e., when choosing the same option they obtain rewards driven by different random processes with different mean values and possibly different distributions. We then consider two scenarios. (1) In the first case users crowd-source full information, meaning they disclose not only their choices but the reward outcomes of those choices and (2) when users exchange limited information, only their choices but not the rewards obtained. Performance is measured in *weak regret*, the difference between the user's total reward and the reward from a user-specific best single-action policy (i.e., always selecting the set of options generating the highest mean rewards for this user). We show that when complete information is shared, our crowd-learning algorithm results in upto a M-fold improvement to the regret bound under different problem settings. When only partial information is available, the improvement of learning is tied to a weight/recommendation factor of how much we value others' opinion compared to her own. Interestingly we find for certain special case, simply integrating partial information allows the algorithm's performance bound to outperform its full information counterpart. To our best knowledge, this is the first attempt to analyze online learning with crowd-

sourced data. By applying our results to the movie ratings dataset MovieLens [39], we show how our algorithms can be used as an online (causal) process to make real-time predictions/recommendations. Experiments show that in both online and offline settings our recommendation performance exceeds that of individual learning, and in the offline setting our results are also comparable with several existing offline solutions. In order to further understand the difference between the notion of "crowd learn" and offline collaborative filtering methods, we adapt the idea of finding a crowd of similar users to existing matrix factorization based recommendation methods and show its performance.

The remainder of the chapter is organized as follows. Section 3.2 presents the system model, and Sections 3.3 and 3.4 analyze the full and partial information scenarios, respectively. A generalization of the model to include additional context information is discussed in Section 3.5. Simulation results are given in Section 3.6. Experimental results using MovieLens data are presented in Section 3.7. For comparison with offline algorithms, a crowd learning adapted matrix factorization method is discussed in Sections 3.8. Section 3.9 concludes this chapter.

3.2 Problem formulation

Consider a network of M users indexed by the set $\mathscr{U} = \{1,2,...,M\}$ and a set of available options (also referred to as *arms* following the bandit problem literature), denoted by $\Omega = \{1,2,...,N\}$. The system works in discrete time indexed by $t = 1,2,\cdots$. At each time step a user can choose up to $1 \leq K \leq N$ options. For user i an option k generates an I-ID reward over time denoted by random variables $\{X_k^i(t)\}$, with a mean reward given by $\mu_k^i := \mathbb{E}[X_k^i]$. We will assume that $\mu_l^i \neq \mu_k^i$ for $l \neq k, \forall i \in \mathscr{U}$, i.e., different options present distinct values to a user. We further assume finite support over X_k^i, i.e., there is a finite positive constant $\bar{X}$ such that $X_k^i(\omega) < \bar{X}$, $\forall i,k$, where ω denotes an arbitrary realization. We will denote the set of top K options (in terms of mean rewards) for user i as N_K^i and its complement $\overline{N}_K^i$. Denote by $a^i(t)$ the set of choices made by user i at time t; the sequence

$\{a^i(t)\}_{t=1,2,...}$ constitutes user i's policy.

Following the classical regret learning framework, we will adopt the *weak regret* as a performance metric, which measures the gap between the total reward (up to some time T) of a given learning algorithm and the total reward of the best single-action policy given a priori knowledge on the average statistics, which in our case is the sum reward generated by the top K options for a user. This is formally given as follows for user i adopting policy a:

$$R^{i,a}(T) = T \cdot \sum_{k \in N_K^i} \mu_k^i - \mathbb{E}[\sum_{t=1}^{T} \sum_{k \in a^i(t)} X_k^i]. \tag{3.1}$$

The goal of an online learning algorithm is to minimize the above regret measure, whose time-average should ideally diminish, i.e., it is desirable to have $R^{i,a}(T) = o(T), \forall i$.

Existing online learning techniques can often attain this goal. For instance, using the celebrated UCB1 algorithm [5] a logarithmic regret uniform in time can be achieved with IID rewards; work in [73] showed that the same can be achieved with Markovian rewards using a renewal based algorithm RCA. Existing algorithms, however, act on each user i separately, i.e., user i only sees her own rewards X_j^is' sample path on which to base her decision. By contrast, in this study we investigate whether the regret performance can be improved by allowing a user i_1 access to observations (or decisions) made by another user i_2, i.e., by letting a user crowdsource data generated by other distinct users.

Specifically, we will consider two types of information shared /exchanged by the users. Under the first type, users disclose *full information*: they not only announce the decisions they make (the options they choose), but also the observations following the decisions, i.e., the actual rewards received from those options, such as perceived quality of a movie. Such announcements may be made at the end of each time step, or may be made periodically but at a lesser frequency. The second type of exchange is *partial information* where users disclose only part of decisions and/or observations. Specifically, we will assume in this

case the users only share their decision information, i.e., the set of choices they make, at the beginning of each time step, but withhold the actual observation/reward information following the decisions.

Different from full information case, where we do not further differentiate users' preference ordering over options, for the partial information scenario we model explicitly that users have different preference orderings over the N options. Specifically, in this case we will assume that the M users may be classified into G distinct groups, indexed by the set $\mathscr{G} = \{1, 2, ..., G\}$, with users within the same group (say group l) having a unique K-preferred set N_K^l; these preference sets (but not the group membership) are assumed to be public knowledge. Note that even with the same preference set, users may be further distinguished based on the actual ordering of these top K options. Our model essentially bundles these users into the same group, provided their top K choices or options are the same. This is because as a user is allowed K choices at a time, further distinguishing their preferences within these K options will not add to the performance of an algorithm.

3.3 Crowd-Learning, Full Information (CL-FULL)

In this case users disclose the rewards $X_k^i(t)$ for each option they chose at the end of a time step. The technical challenge here is that the statistics driving the rewards are not identical for all users even when using the same option. So information obtained from another user may need to be treated differently from one's own observations. In general the reward user i obtains from selecting option k can be characterized by $X_k^i = \mathscr{F}(X_k, \mathscr{N}_i, \mathscr{L}_i)$, where $\mathscr{F}(\cdot)$ is some arbitrary unknown function, X_k describes certain *intrinsic* or *objective* value of option k that is independent of the user (e.g., the rating given to a restaurant by AAA, and so on), $\mathscr{N}_i$ is a noise term, and $\mathscr{L}_i$ captures user-specific features that affect the *perceived* value of this option to user i (e.g., user i's taste or dietary restrictions which may affect her preference for different types of restaurants).

Next consider the relationship between users' observations. For simplicity and without

loss of generality, we will limit our attention to the following special case of user-specific valuations, where the rewards received by two users from the same option are given by a log-linear relationship:

$$X_k^i \stackrel{d}{=} (X_k^j)^{\delta_k^{i,j}} \ , \forall i \neq j \in \mathscr{U}, k \in \Omega, \tag{3.2}$$

where $\delta_k^{i,j}$ is a constant and unknown scaling factor also referred to as the *distortion* or *distortion factor* between two users. This relationship implies that one user's perception of a given option is statistically identical to another's to a constant power. Several common distributions, such as Log-normal, Pareto, and Weibull, satisfy such an assumption when used to model user observations. While certainly a simplification, this assumption does not impact the generality of our result. This is because in principle a user can assume any model to estimate a pairwise relationship. It can be shown that our analysis holds as long as such relationship is one-to-one in distribution. This concrete model leads to closed-form characterizations of the regret bounds, which help shed light on the effect of various problem parameters. It also proves to work well with the real dataset MovieLens. More discussions on a general relationship function can be found at the end of this section.

Consider two users i and j, and option k. Denote by $\hat{r}_k^i(t)$ the sample mean of log-reward $\{\log X_k^i(t)\}_t$ (with its true mean denoted by $\hat{\mu}_k^i$) collected by i from option k. These quantities are not only available to user i, but also to all other users $j \in \mathscr{U} \setminus \{i\}$ due to the full information disclosure, and vice versa. User i then estimates the distortion between herself and user j by calculating the following

$$\tilde{\delta}_k^{i,j}(t) = \hat{r}_k^i(t) / \hat{r}_k^j(t) \ , \forall i \neq j \in \mathscr{U}, k \in \Omega \ . \tag{3.3}$$

With the above quantity we then make the following simple modification to the well-known UCB1 algorithm [5]. In the original UCB1 (or rather, a trivial multiple-play extension of it), user i's decision $a^i(t)$ at time t is entirely based on her own observations. Specifically,

denote by $n_k^i(t)$ the number of times user i has selected option k up to time t. The original UCB1 then selects option k at time t, if its index value given below is among the K highest:

$$\text{UCB1 index: } r_k^i(t) + \sqrt{\frac{2\log t}{n_k^i(t)}}\ .$$

Our modified algorithm takes this index as a baseline and makes the following changes: option k is selected at time t if its index value defined below is among the K highest:

$$\text{CL-FULL index:}$$
$$\frac{r_k^i(t)\cdot n_k^i(t) + \sum_{j\neq i}\Lambda^{i,j}(r_k^j(t))\cdot n_k^j(t)}{\sum_{j\in\mathscr{U}} n_k^j(t)} + \sqrt{\frac{2\log t}{\sum_{j\in\mathscr{U}} n_k^j(t)}},$$

where the operator $\Lambda^{i,j}(r_k^j(t))$ is defined as follows:

$$\Lambda^{i,j}(r_k^j(t)) = \frac{\sum_{s=1}^t (X_k^j(s))^{\tilde{\delta}_k^{i,j}(t)}\cdot 1\{k\in a^j(s)\}}{n_k^j(t)}\ .$$

We take $\delta_k^{i,i}=1$ by default, and denote $\delta_k^{i,*} := \max_{j\in\mathscr{U}}\delta_k^{i,j}$, and further denote $\delta^{i,*} := \max_{k\in\Omega}\delta_k^{i,*}$. To analyze the performance of this algorithm, we first prove the following result.

Lemma III.1. *Suppose the numbers of samples from option k for users i,j both exceed l. Define constant ε as follows:*

- *If $l = C_1 t$, let $\varepsilon := \frac{2\delta_k^{i,*}}{|\hat{\mu}_k^j| - 2\sqrt{\frac{\log t}{C_1 t}}}\cdot\sqrt{\frac{\log t}{C_1 t}}$,*
- *If $l = C_2\log t$, let $\varepsilon := \frac{2\delta_k^{i,*}}{\sqrt{C_2}|\hat{\mu}_k^j| - 2}$.*

Then for sufficiently large C_2 the estimation error on the distortion $\delta_k^{i,j}$ satisfies:

$$P(|\tilde{\delta}_k^{i,j} - \delta_k^{i,j}| > \varepsilon) \le \frac{4}{t^2}\ .$$

Denote $\bar{X}_k = \max_{i,\omega} X_k^i(\omega)$ and $\Delta_k^i := \mu_K^i - \mu_k^i$. We have the following series of results characterizing the performance of CL-FULL. They are organized based on the nature of $\delta_k^{i,j}$. We shall start with the simplest case when $\delta_k^{i,j}$ is option independent: $\delta_1^{i,j} = \delta_2^{i,j} = ... = \delta_K^{i,j}$, followed by k-dependent $\delta_k^{i,j}$s.

3.3.1 Option independent $\delta^{i,j}$

In this case, for user i to estimate $\delta^{i,j}$, she can simply choose the arm/option that has the largest number of collected samples (by users i and j) for the purpose of calculation. As we will show later, under mild conditions each user can always find an arm with $O(t)$ number of samples up to time t, with any of its peers. Based on results in Lemma III.1, with $O(t)$ number of samples for calculation, we can achieve an estimation error on the order of $O(\sqrt{\log t/t})$. We assume in this section for each pair of (i,j) we have $N_K^i \cap N_K^j \neq \emptyset$. This mild assumption is to make sure for any pair of users (i,j) we can find a common good option. We have the following theorem characterizing the performance of CL-FULL.

Theorem III.2. *Under CL-FULL, there exists a constant $C_1 > 0$ such that user i's weak regret is upper bounded by:*

$$R_{CL\text{-}FULL}^i(t) \leq \sum_{k \in \overline{N}_K^i} \lceil \frac{8\Delta_k^i}{M \cdot \left(\Delta_k^i - 2\bar{X}_k^2 \varepsilon_k(t)\right)^2} \log t \rceil + const. \,, \tag{3.4}$$

where

$$\varepsilon_k(t) = \frac{2\delta^{i,*}}{\min_{j \neq i} |\hat{\mu}_k^i| - 2\sqrt{\frac{\log t}{C_1 t}}} \cdot \sqrt{\frac{\log t}{C_1 t}} \,.$$

To compare with the original UCB1 algorithm which has a weak regret upper bounded

by

$$R^i_{\text{UCB1}}(t) \leq \sum_{k \in \overline{N}^i_K} \lceil \frac{8 \log t}{\Delta^i_k} \rceil + \text{const.}, \tag{3.5}$$

we note that in our result, the term $\frac{2\delta^{i,*}}{\min_j |\hat{\mu}^j_k| - 2\sqrt{\frac{\log t}{C_1 t}}} \cdot \sqrt{\frac{\log t}{C_1 t}} \to 0$ at approximately the rate of $O(\sqrt{\log t / t})$; if we ignore this term then our bound becomes

$$R^i_{\text{CL-FULL}}(t) \leq \sum_{k \in \overline{N}^i_K} \lceil \frac{8}{M \cdot \Delta^i_k} \log t \rceil + \text{const.} \, ,$$

which shows a M-fold performance improvement.

3.3.2 Option dependent $\delta^{i,j}_k$

The analysis below is similar as before, with a subtle difference in bounding the error ε in estimating $\delta^{i,j}_k$, which now needs to be done for each option k. Because of this we can no longer guarantee that there is always an $O(t)$ number of samples available for all k, and thus the estimation error is expected to be large. In fact when the number of explorations (number of samples in each option, as commonly defined in the MAB literature) chosen to be $L \log t$ we can only claim this amount of samples can be used to calculate each $\delta^{i,j}_k$. In particular, using result from Lemma 1, with probability at least $1 - \frac{4}{t^2}$ we have the following estimation error on each $\delta^{i,j}_k(t)$: $\varepsilon_k(t) = \frac{2\delta^{i,*}_k}{\sqrt{L}\hat{\mu}^j_k - 2}$. Note that this error does not decrease in t and depends on L. Therefore we expect larger learning error and regret terms. We have the following result.

Theorem III.3. *Under CL-FULL, user i's weak regret is upper bounded by,*

$$R^i_{\textit{CL-FULL}}(t) \leq \sum_{k \in \overline{N}^i_K} \lceil \max\{ \frac{(2\sqrt{\frac{2}{M}} + \frac{12\mu^i_k \mu^j_k \delta^{i,*}_k}{\min_{j \neq i} |\hat{\mu}^j_k|})^2}{\Delta^i_k}, \frac{8\Delta^i_k}{(\mu^i_k \cdot \min_{j \neq i} \mu^j_k)^2} \} \log t \rceil + \textit{const.} \, . \tag{3.6}$$

Compared to the previous bound in Eqn.(3.5) we see the two differ in several places.

First $\frac{8\Delta_k^i}{(\mu_k^i \cdot \min_{j\neq i}\mu_k^j)^2}$ is generally a smaller term compared to $\frac{8}{\Delta_k^i}$, as μ_k^i is generally larger compared to Δ_k^i. Secondly if we ignore the $\frac{12\mu_k^i\mu_k^j\delta_k^{i,*}}{\min_{j\neq i}|\hat{\mu}_k^j|)^2}$ term, the constant shown in $R^i_{\text{CL-FULL}}(t)$ becomes $\frac{8}{M}$, which is M-fold better than the UCB bound using individual learning, similar to what we have seen in Eqn.(3.4). This extra term corresponds to errors in estimating pair-wise discrepancies. From the above observations we see a clear trade-off between benefiting from crowdsourced data and additional estimation error incurred in order to put crowdsourced data to use.

3.3.2.1 A (slightly) refined bound for discrete X

When Xs are discrete random variables (which is often the case in a rating system), the regret bound proved in Eqn.(3.6) can be slightly refined. Denote each user's observation space by $\mathscr{X}_k^i$. The basic idea is after getting each estimate $\tilde{\delta}$ and using it to convert samples $(X_k^j(t))^{\tilde{\delta}_k^{i,j}}$, we assign a rating $x \in \mathscr{X}_k^i$ that is the closest to $(X_k^j(t))^{\tilde{\delta}_k^{i,j}}$. Formally,

$$X_k^i(t) = \text{argmin}_{x\in\mathscr{X}_k^i} \left|(X_k^j(t))^{\tilde{\delta}_k^{i,j}} - x\right| .$$

Then with the rest of CL-FULL stay the same, we have the following theorem.

Theorem III.4. *Under CL-FULL, user i's weak regret is upper bounded by,*

$$R^i_{CL\text{-}FULL}(t) \leq \sum_{k\in\overline{N}_K^i} \lceil \max\{\frac{8}{M\Delta_k^i}, (\frac{4\bar{X}_k^2\delta_k^{i,*} + 2\Delta_{\mathscr{X}_k^i}}{\Delta_{\mathscr{X}_k^i}\min_{j\neq i}|\hat{\mu}_k^j|})^2\Delta_k^i\}\log t\rceil + const., \tag{3.7}$$

where $\Delta_{\mathscr{X}_k^i} = \min_{x\neq y\in\mathscr{X}_k^i}|x-y|$.

We see that the term containing M is now separated from the rest of the terms compared to Eqn.(3.6); this is a slightly better bound because we removed the constant from this term, which leads to a smaller quantity. The other terms of both regret bounds are roughly on the same order.

3.3.3 A joint estimation of $\delta_k^{i,j}$: beyond pair-wise estimation

Note the extra learning error in the bound (3.6) is independent of M and does not decrease when the number of users increases; this is a potentially worrisome bottleneck. Below we examine ways to mitigate this.

We start by explaining why we do not have an M-fold scaling in this additional error term. Statistically speaking, since we use the converted samples to estimate the sample mean of each option, there are M-fold number of samples (though noisy) compared to individual learning. However, this is not true when learning the δs, as we need to use a user's own data to make such pairwise calculations (one-fold for each pair). This motivates us to consider using all samples to estimate δ simultaneously to achieve an M-fold speed-up.

Specifically, instead of pair-wise estimation, $\delta_k^{i,j}$ can be estimated as follows:

$$\tilde{\delta}_k^{i,j} = \frac{\hat{r}_k^i(t) + \sum_{l\neq i} \hat{r}_k^l(t)\tilde{\delta}_k^{i,l}}{\hat{r}_k^j(t) + \sum_{l\neq j} \hat{r}_k^l(t)\tilde{\delta}_k^{j,l}} \ .$$

That is, for each pair of users i, j, when estimating the $\delta_k^{i,j}$ for option k, we not only use the data from i, j, but also the converted data from other users $j \neq i$. This conversion and thus the estimate involves solving $\delta_k^{i,j}$s simultaneously. The above equation can be re-written as follows:

$$\tilde{\delta}_k^{i,j}\hat{r}_k^j(t) + \tilde{\delta}_k^{i,j}\sum_{l\neq j} \hat{r}_k^l(t)\tilde{\delta}_k^{j,l} = \hat{r}_k^i(t) + \sum_{l\neq i} \hat{r}_k^l(t)\tilde{\delta}_k^{i,l} \ ,$$

which is a quadratic equation of $\delta_k^{i,j}$s. Denote $\Phi_k := [\tilde{\delta}_k^{i,j}]_{i\neq j}$, we have the following Quadratic Matrix Equation (QME) whose solution leads to solutions for δ:

$$A + \Phi_k B + \sum_r C_r(\Phi_k)^T D_r(\Phi_k) F_r = 0 \ , \tag{3.8}$$

where $A, B, \{C_r, D_r, F_r\}$ are matrices with entries being functions of users' reward (log-reward) statistics. Suppose the true $\delta_k^{i,j}$ is the unique solution to the above QME when $\hat{r}_k^j(t)$ is replaced by the true mean $\hat{\mu}_k^j$, and denote it by Φ_k^0. Since $\hat{r}_k^j(t)$ is a noisy version of $\hat{\mu}_k^j$, what we want to show is under perturbation our estimated solution $\tilde{\delta}_k^{i,j}$ can be bounded. This small perturbation will add two perturbation terms (A_0, B_0) in the QME as follows:

$$A + A_0 + \Phi_k(B + B_0) + \sum_r C_r(\Phi_k)^T D_r(\Phi_k) F_r = 0 \ . \tag{3.9}$$

Suppose we have the following boundedness on the perturbation (which we prove to be the case in the appendix, see Proof for Theorem III.5):

$$|A_0(i,j)| \leq \varepsilon, |B_0(i,j)| \leq \varepsilon \ , \tag{3.10}$$

and when the solution to the QME can be bounded proportional to ε by a constant multiplier C_3 we have the following theorem:

Theorem III.5. *Under CL-FULL, user i's weak regret is upper bounded by*

$$R^i_{\textit{CL-FULL}}(t) \leq \sum_{k \in \overline{N}^i_K} \lceil \frac{(2\sqrt{2} + 8\bar{X}_k^3 \delta_k^{i,*} C_3)^2}{M\Delta_k^i} \log t \rceil + \textit{const.} \tag{3.11}$$

We see that indeed the factor M is now back in the regret bound.

3.3.4 Discussion and extensions

Note that the improvement we have demonstrated in this section are all in the bounds on the regret and not necessarily in the regret itself; the latter is examined later using numerical experiments. It can also be shown that similar result exists when the full information exchange occurs only at periodic intervals but not necessarily at the end of each time step so as to reduce potential communication cost. The proof is very similar to the one presented here and thus not repeated.

The analysis so far assumes a log-linear relationship between two users' reward distributions. This is obviously not generally true. Consider the following generalization, where a pair of users i, j's observations of any option k are given by

$$X_k^i \stackrel{d}{=} F(X_k^j) \ ,$$

where F is an arbitrary function. Then consider using Taylor expansion to make the following approximation:

$$X_k^i \stackrel{d}{\approx} \sum_{n=0}^{D} b_n \cdot (X_k^j)^n \ ,$$

where D is the degree of estimation. When $D = \infty$ the equality holds in distribution. It follows that as long as we can estimate the b_ns, we will similarly be able to convert a sample from one user to another. For instance, we could form the following system of equations in the case of $D = 2$:

$$\begin{aligned} E[X_k^i] &= \sum_{n=0}^{D} b_n E[\cdot (X_k^j)^n] \ , \\ E[(X_k^i)^2] &= E[(b_0 + b_1 X_k^j + b_2 (X_k^j)^2)^2] \ , \\ E[(X_k^i)^3] &= E[(b_0 + b_1 X_k^j + b_2 (X_k^j)^2)^3] \ . \end{aligned}$$

With above three equations we can potentially solve for b_0, b_1, b_2, where D clearly controls the trade-off between accuracy in converting observation data and the complexity of such estimation and conversion process. However, there is generally no closed form characterization of the solutions for b_ns. In this case even though a similar algorithm can be followed, its performance bound becomes harder to analyze.

3.4 Crowd-Learning, Partial Information (CL-PART)

3.4.1 Preference identification

In the previous section, we showed that by exchanging full information the crowd learning algorithm can improve the regret bound by a constant factor up to M. In many cases however, user information is not or cannot be exchanged in its entirety for a variety of reasons such as privacy or communication cost. In this section we consider one such cases where users only share their decisions/actions, but not their direct observations. We would like to examine a similar problem in this case, i.e., to what extent can a user tap into decisions made by other users to improve its own decision process, assuming other users act in self-interest, i.e., decisions are made to maximize their own satisfaction though their precise objective or utility functions need not be known.

The general idea is to use the shared decision to first obtain any user's frequency of selecting each option. These statistics are indicators of a user's preference, assuming the user is trying to maximize its own reward over time; thus decisions by another reveal preferences and may be exploited. In what follows we first propose a group classification procedure with performance guarantee to help a user distinguish similar users and then show how we can design online learning algorithms utilizing this information.

To differentiate users' preferences, consider the following sample frequency based group identification procedure. Each user keeps the same set of statistics $n_k^i(t)$ as before: the number of times user i is seen using option k. Users then estimate each other's preference by ordering the statistics: at time t user i's preference is estimated to be the set $\tilde{N}_K^i(t)$, which contains options k whose frequency $n_k^i(t)$ is among the K highest of all i's frequencies. User i is then put in a preference group with whose (known) preferred set $N_K^l(t)$ is the closest in distance. Specifically, assign user i to group $g^i(t)$ if:

$$g^i(t) = \text{argmax}_{l\in\mathscr{G}} D^{i,l}(t) := \text{argmax}_{l\in\mathscr{G}} |\tilde{N}_K^i(t) \cap N_K^l| \ ,$$

with ties broken randomly. The performance of the above classification step will be analyzed later within the context of a learning algorithm.

3.4.2 Algorithm and performance

After differentiation, denote by $\mathscr{U}_i$ the set of similar users for each user i and denote $M_i = |\mathscr{U}_i|$. There are two options a user can choose to implement an algorithm: (i) to use information from only those identified as having the same preferences, i.e., users in set $\mathscr{U}_i$, and (ii) to use all users' information.

We start with the easier case (i). Denote by $n_{k,i}(t)$ the total number of times option k has been selected by the crowd $\mathscr{U}_i$ up to time t, i.e., $n_{k,i}(t) := \sum_{i\in\mathscr{U}_i} n_k^i(t)$. Then define $\beta_k^i(t) := \frac{n_{k,i}(t)}{\sum_{l\in\Omega} n_{l,i}(t)}$ to denote the frequency at which option k is used by the group up to time t. This will be referred to as the group recommendation.

Then we rewrite the frequency parameter to take advantage of crowd learning in the following way. Based on $\beta_k^i(t)$ we first order the options in descending order; their rank denoted by $\mathbf{rank}_k$. Denote by $\overline{B}_K(t)$ the non-top K options based on the frequency estimate:

$$\overline{B}_K(t) := \{k \in \Omega : \mathbf{rank}_k > K\} .$$

Define user specific frequency estimate for $j \in \mathscr{U}^i$ as

$$\beta_k^i(j;t) = \frac{n_k^j(t)}{t} .$$

Now we redefine the frequency parameters as follows

$$\tilde{\beta}_k^i(t) := \min_{j\in\mathscr{U}_i} \beta_k^i(j;t),\ k \in \overline{B}_K(t) . \tag{3.12}$$

For $k \notin \overline{B}_K(t)$ we have

$$\tilde{\beta}_k^i(t) := \frac{1 - \sum_{k' \in \overline{B}_K(t)} \tilde{\beta}_{k'}^i(t)}{K} .$$

Note the above definition is a valid probability measure: $\forall j \in \mathscr{U}$

$$\sum_{k' \in \overline{B}_K(t)} \tilde{\beta}_{k'}^i(t) \leq \sum_{k' \in \overline{B}_K(t)} \beta_k^i(j;t) \leq 1 ,$$

and it follows that $\sum_{k \in \Omega} \tilde{\beta}_k^i(t) = 1, \forall t$. With these definitions and notations, we construct the following algorithm CL-PART(I), by biasing toward potentially good options as indicated by the group. Under the CL-PART(I) algorithm, option k is selected at time t if its index value defined below is among the K highest:

$$\text{CL-PART(I) index: } r_k^i(t) + \alpha(t)\tilde{\beta}_k^i(t)\sqrt{\frac{\log t}{n_k^i(t)}} + \sqrt{\frac{2\log t}{n_k^i(t)}} ,$$

where $\alpha(t) \in [0, \sqrt{2})$ is a weighting factor over the group recommendation capturing how much user i is valuing recommendation from the group. The upper bound of $\sqrt{2}$ is due to technical reasons seen in the proof.

A few remarks are in order. (1) In the above index expression, the middle bias term serves as a recommendation: a larger group frequency $\tilde{\beta}_k^i(t)$ means a better recommendation. But its effect diminishes as $n_k^i(t)$ increases. This reflects the notion that over time a user becomes increasingly more confident in her own observations and relies less and less on the group recommendation. (2) Secondly, the weight $\alpha(t)$ captures how much the user values the group recommendation compared to her own observations: a small value puts a small weight on the group information which may be under-utilizing this information, while a large α may be placing too much confidence in others.

To prove the performance of CL-PART(I), first we have the following result on the user classification process.

Lemma III.6. *Under CL-PART(I), for user $i \in \mathscr{U}$ belonging to group $r \in \mathscr{G}$ the probability of mis-classification at time t is bounded as follows:*

$$P(g^i(t) \neq r) \leq \frac{2C}{t^2},\ \forall i \in \mathscr{U}, t > 0. \tag{3.13}$$

for some positive constant $C > 0$.

We then have the following results on characterizing the regret performance of CL-PART(I).

Theorem III.7. *Under CL-PART(I), let $\varepsilon(t) := \sqrt{\frac{\log t}{t}}$ and set*

$$\alpha(t) := \sqrt{2}(1-\gamma) - \sqrt{N} \cdot \varepsilon(t)$$

at each time t, where $0 < \gamma \leq 1$ is an arbitrary small number. We have user i's weak regret is upper bounded by

$$R^i_{\textit{CL-PART(I)}}(t) \leq \sum_{k \in \overline{N}^i_K} \lceil \frac{4\left(\sqrt{2} - \frac{\alpha(t)(1-\frac{1}{M_i})}{K}\right)^2}{\Delta^i_k} \log t \rceil + \textit{const.}$$

γ is to make sure we will not violate the technical assumption $\alpha(t) < \sqrt{2}$ for any t. γ is tunable and can be made arbitrarily small.

Notice we have

$$\begin{aligned}\alpha(t)(1-\frac{1}{M_i}) &= \frac{\sqrt{2}(1-\gamma) - \sqrt{N} \cdot \sqrt{\frac{\log t}{t}}}{K} \cdot (1 - \sqrt{\frac{\log t}{t}}) \\ &\xrightarrow[t]{} \frac{\sqrt{2}(1-\frac{1}{M_i})}{K}.\end{aligned}$$

So the bound converges to

$$\sum_{k \in \overline{N}_K^i} \lceil \frac{8(1-\frac{1-\frac{1}{M_i}}{K})}{\Delta_k^i} \log t \rceil + \text{const.}$$

Interestingly, we see that when $K = 1$, i.e., when each user selects only one option at a time, the regret reduces to $\frac{8}{M_i} + \text{const.}$, which gives us M_i-fold performance improvement. Recall that with full information the regret bounds do not always converge to this quantity; this result thus implies that in some cases restricting to partial information may be more beneficial than trying to use full information. Intuitively, this could happen when the additional error incurred in learning pairwise differences exceeds its benefits. When $K = 2$ the constant factor reduces from 8 to 4 (with M_i large), a roughly 50% improvement.

3.4.3 Leveraging more information

We next consider leveraging more user information beyond the set $\mathscr{U}_i$. The motivation is that by adding more samples we may potentially improve the algorithm's performance, especially when the population of similar users is relatively small compared to the total population. Restricting our attention to $\mathscr{U}_i$ previously ensures that users have the same top-K choices. Removing this restriction means that we may have $N_K^i \neq N_K^j$; however, it is possible there exists option $k \in \overline{N}_K^i \cap \overline{N}_K^j$. In this sense, j's sample frequency estimation of k can again be utilized by i, and vice versa. Accordingly, we propose the following modification to the algorithm given in the previous subsection.

We largely re-use the index construction as detailed in CL-PART(I), but with the following difference: we discount choices made by users believed to belong to a different group, so as not to be overly influenced by choices made by users with different preferences. Specifically, user i assigns the following weight to option k:

$$\beta_k^i(t) = \frac{\sum_{j \in \mathscr{U}} (n_k^j(t))^{\omega^{i,j}}}{\sum_{k \in \Omega} \sum_{j \in \mathscr{U}} (n_k^j(t))^{\omega^{i,j}}}\ , \tag{3.14}$$

where weights $\omega^{i,j} = 1$ if i estimates user j to be in the same group as itself, and $\omega^{i,j} < 1$ otherwise; $\omega^{i,j}$ can also be chosen as a function of the difference between different preference groups. This modified index leads to algorithm CL-PART(II).

We define the following set for each $k \in \overline{N}_K^i$ we have

$$\mathscr{U}_k^i = \{j \in \mathscr{U} : k \in \overline{N}_K^j\} . \tag{3.15}$$

And denote $M_k^i = |\mathscr{U}_k^i|$. That is, M_k^i records the number of users who have k falls out of their top choices. Notice $\mathscr{U}^i \subseteq \mathscr{U}_k^i$ and thus $M_k^i \geq M$. We can then similarly prove the following result.

Theorem III.8. *Under CL-PART(II), at each time t, user i's weak regret is upper bounded by*

$$R^i_{\textit{CL-PART(II)}}(t) \leq \sum_{k \in \overline{N}_K^i} \lceil \frac{4\left(\sqrt{2} - \frac{\alpha(t)(1-\frac{1}{M_k^i})}{K}\right)^2}{\Delta_k^i} \log t \rceil + \textit{const.}$$

Note that due to the fact $M_k^i \geq M$ we have achieved a strictly better bound compared to the one in Theorem III.7.

3.5 The case with contextual information

In this section we discuss a generalization of the previous model by incorporating additional contextual information, which is often used to capture a continuum of choices. The contextual information often captures the fact that although users may have overall preferences towards choices, they can be further distinguished by the *context* under which a choice is made, e.g., preference for a restaurant may be influenced by different menu items. Specifically, assume at each time t an extra piece of contextual information $Y(t) \in \mathscr{Y}$

arrives, and a user's reward is context-dependent. When $|\mathscr{Y}|$ is finite, we in effect have $N \times |\mathscr{Y}|$ choices, each denoted by (i,y). All our previous results hold over this new expanded set of choices.

Now consider the case $|\mathscr{Y}|$ is infinite, e.g., $\mathscr{Y} = [0,1]$. The extension of our earlier results in this case is also feasible by following existing literature on continuum arms, see e.g., [2, 74], which consists of (1) assuming a Lipschitz condition (with L being the multiplication constant and θ being the power one) on the reward $X_k^i(t,y)$, i.e., expected rewards are similar when contexts are close, and (2) quantizing the context interval $[0,1]$ into an increasing number of sub-intervals, each treated as a bundle with an average reward over the sub-interval.

Following this procedure, we let the number of sub-intervals at time t be $p(t) = t^{-z/2}$ with constant $0 < z < 1$, and define $\Delta^* = \max_{i,k} \Delta_k^i$. We then have a set of theorems that parallel their finite-arm counterparts; the full detail is omitted for brevity except for the following example (proof is also omitted as it followed standard technique and our previous analysis).

Theorem III.9. *The weak regret of user i under contextual CL-FULL is upper bounded by*

$$R^i_{CL\text{-}FULL}(t) \leq \sum_{k \in \overline{N}^i_K} \Delta_k^i \cdot \lceil \frac{8 \cdot t^z \cdot \log t}{M} \rceil + (2\bar{X}^2 + 5\Delta^*) \lceil \frac{t^{1-z/2}}{1-z/2} \rceil + 2L \lceil \frac{t^{1-z\theta/2}}{1-z\theta/2} \rceil + const.$$

The one under the original UCB1 has a similar format, but without the M-fold speed-up.

3.6 Numerical Experiment

3.6.1 Simulation setup

In this section we evaluate and validate the performance of our crowd-learning algorithms using simulated data. In our simulation we have ten users with five options; each

user targets the top three options at each time, i.e., $M = 10, K = 3, N = 5$. Furthermore, for each option the reward is given by a truncated exponentially distributed random variable (bounded). The distortion factor between each pair of users for each option is generated according to a Gaussian random variable with mean 1 and variance 1. We use "crowd regret" to denote the sum of regrets from all users. Throughout the evaluation section, the regret results are averaged over multiple sample realizations for a certain parameter setting.

3.6.2 Simulation results

We start with performance evaluation for CL-FULL. From Figure 3.1 we see with full information exchange the crowd learning algorithm significantly outperforms individual learning. Moreover, its performance is comparable to a centralized scheme (denoted as UCB Centralized in the figure), whereby the M users are centrally controlled and coordinated in their learning using UCB1, allowing simultaneous selection of the same options by multiple users, and each receiving MK samples at each time step without distortion.

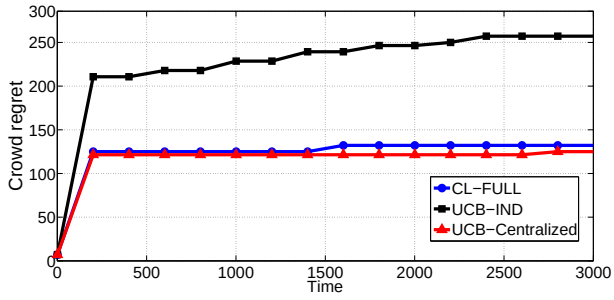


Figure 3.1: Comp. between CL-FULL, UCB-IND, UCB Centralized

We next consider CL-PART. The mis-classification rate of the algorithm is given in Figure 3.2, which is shown to decay nicely. The performance of CL-PART is compared to UCB-IND under different parameter γ in Figure 3.3. We see that CL-PART consistently outperforms UCB-IND. A smaller γ gives better performance, which confirms our analytical results. Also, as we noted in our analysis and see here, a smaller K results in less regret.

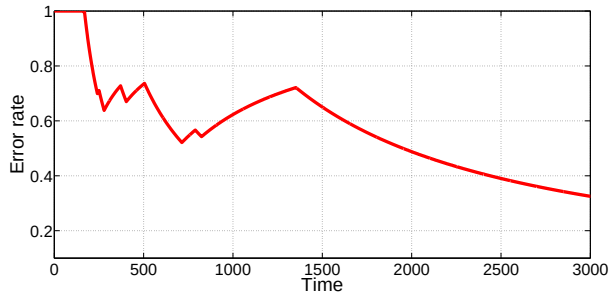


Figure 3.2: Conver. of differentiating users from different groups.

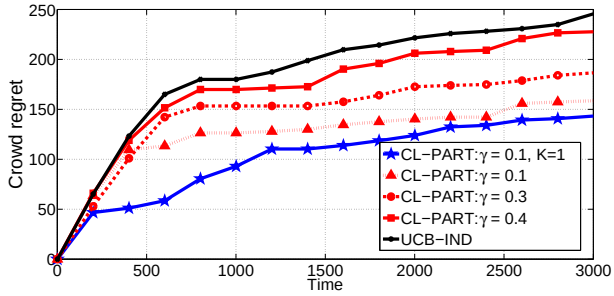


Figure 3.3: Performance of CL-PART, with different γ parameters.

Finally, we compare CL-PART(I) and CL-PART(II). From results in Figure 3.4 we observe that with more appropriately designed algorithm and leveraging more data, a (much) better performance can be achieved.

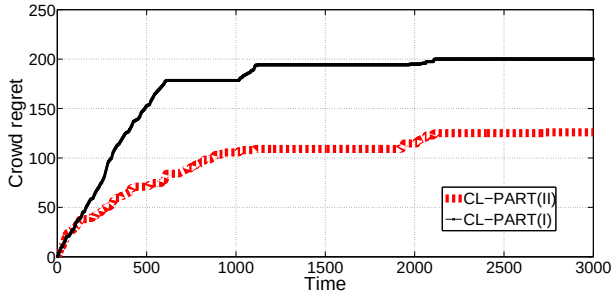


Figure 3.4: Comparison between CL-PART(I) and CL-PART(II).

3.7 An Empirical Study Using MovieLens

We now apply the idea of crowd-learning to the MovieLens data [39] collected via a movie recommendation system. We will use MovieLens-1M dataset for this experiment

(this is one of the three sets available; the others are 100K and 10M in size). It contains around one million rating records provided by 6040 users on 3952 movies (from 18 categories/genres), dated from April 25, 2000 to February 28, 2003. Each rating is on a scale of 1-5; throughout the rest of this section we will also refer to a rating record as a *review*. In general each user contributes to multiple reviews, with a significant portion of users providing a large amount, e.g., about 70% have more than 50 reviews. This makes the dataset quite suitable for testing our algorithm as we try to estimate groups or crowds of "similar" users.

Our goal is to use the MovieLens data to verify whether our crowd-learning algorithm can help us predict how a user is going to rate movies in the future given this and other users' reviews in the past. The basic idea in applying learning is that based on such a prediction a user would then choose (more often) to watch a movie that she thinks she is going to like rather than dislike, thereby maximizing her satisfaction over time. Obviously this decision aspect of the learning algorithm cannot be verified using this dataset as we have no information on the users' actual decision process. Our experiment thus only focuses on the estimation/prediction aspect of the learning algorithm. Put in a different way, this would also allow us to show whether our algorithm can provide better recommendations for a particular user for a particular movie. Within this context we further aim to highlight the potential uses of our algorithm in an online fashion as more data samples become available.

3.7.1 Experiment design

To perform this experiment we must explain the algorithm in the context of this dataset. We first note that in this case time is no longer discrete, as reviews arrive as an arbitrary arrival process in continuous time. Therefore the discrete time steps used in the algorithm is replaced by a clock driven by this arrival process, i.e., one tick for each arrival or each batch of arrivals. Mathematically, this means to assign integer time indices $t = 1, 2, \cdots, T$ as the discretized time stamps for the reviews, up to some maximum T. Under this assignment

multiple reviews may have the same time stamp, and there is at least one review at each time.

Secondly, the reviews tend to arrive in clusters upon new movie releases, and it may be hard to find any review for a movie that was released some time ago. This means that if we are to treat each movie as a separate option (or arm) then these options are not simultaneously available at all times as movies come and go, and along with them their corresponding reviews. We thus bundle these movies into categories so that over a long period of time there are always reviews available for a genre. Adopting the classification given in [39], we will bundle movies by their genres, e.g., Action, Adventure, Comedy, etc., resulting in 18 categories/genres; each is regarded as an option/arm for our algorithm. A review for any movie within a genre is treated as a sample of the corresponding option. In doing so, we have for each user a continuing sequence of arriving samples for each option, and our algorithm (the estimation part) can then be applied in a straightforward way. We do lose the finer distinction between movies of the same genre and thus strictly speaking our prediction result is for the whole genre; this prediction is then used as a proxy for a specific movie within that genre. As we shall soon see our prediction performance even with this coarse grained bundling is competitive with existing methods. Since each review contains the actual rating, not just preference ordering, we will mainly use CL-FULL for testing. We nevertheless will also test CL-PART.

The prediction performance is measured by the error and squared error, both averaged over the number of samples in the data. Specifically, adopting the same notation used in our analysis, $a^i(t)$ denotes the set of movies user i reviewed at time t (under the discretized time), and $X_k^i(t)$ denotes her rating for movie k for $k \in a^i(t)$; also let $\hat{X}_k^i(t^-)$ denote the prediction/estimation of user i's rating for movie k given all reviews that have arrived before t, i.e., up to and including $t-1$, using the learning algorithm. The average error (referred

as Average Error) over a horizon T is given by:

$$\mathscr{E}_A(T) = \frac{\sum_{t=1}^{T} \sum_{(i,k):k\in a^i(t)} |\hat{X}_k^i(t^-) - X_k^i(t)|}{\sum_{t=1}^{T} \sum_{(i,k):k\in a^i(t)} 1}, \tag{3.16}$$

where the denominator is the total number of reviews received by time t. Similarly the average root-squared error (referred as RMSE) over a horizon T is given by:

$$\mathscr{E}_S(T) = \sqrt{\frac{\sum_{t=1}^{T} \sum_{(i,k):k\in a^i(t)} |\hat{X}_k^i(t^-) - X_k^i(t)|^2}{\sum_{t=1}^{T} \sum_{(i,k):k\in a^i(t)} 1}}. \tag{3.17}$$

The estimate $\hat{X}_k^i(t^-)$ is obtained following the learning algorithm with the modification mentioned earlier and summarized below. Note that since the algorithm works at the category/genre level, we have $\hat{X}_k^i(t^-) = \hat{X}_l^i(t^-)$ for all k, l in the same movie category.

Algorithm 1 Online Movie Recommendation

1: *loop*:
2: At time t, do the following simultaneously for each user i s.t. $a^i(t) \neq \emptyset$:
    1: given $X_k^i(t)$, $k \in a^i(t)$, update i's preference ranking over arms (categories of movies);
    2: update user i's *similarity group* (group identification): users that share the same set of top K preferred options as i;
    3: estimate the distortion between i and those in her similarity group (CL-FULL) ;
    4: update user i's rating for each option by including estimates from those in her similarity group weighted by the estimated distortion as given in the original algorithm.
3: At the end of time step t, we obtain $\hat{X}_k^i((t+1)^-)$ for all i, k.
4: **goto** *loop*.

The error given in Eqns (3.16)-(3.17) will be used in two ways in our experiments. In the first, online setting, we will examine it as a function of T so it reflects the true *prediction* error because the error is measured between future values and estimates based on all past values. In the second, offline setting, we compute these errors by interpreting $\hat{X}_k^i(t^-)$ as i's estimated rating for movie k given all reviews provided at all times other than t (both before

and after), i.e., $X_k^i(t)$ is treated as the only missing entry in the data to be estimated. This latter case is more aptly referred to as *estimation* than prediction because the processing is non-causal, but it appears that the term prediction has been adopted to describe this type of process in various contexts, see e.g., [41].

3.7.2 Online prediction result

We start by plotting $\mathscr{E}_A(T)$ as a function of T; this is shown in Figure 3.5 where we have used $K = 3$, i.e., a user's top 3 preferences (unordered) determine her similar group. The data suggests that a user rarely reviews more than 6 different categories of movies, thus the choice of $K = 3$ is a reasonable trade-off between too restrictive a definition of similarity group (a large K and a very small similarity group) and an overly liberal one (a small K and a very large similarity group). As expected we see a downward trend as the prediction becomes more accurate with more past samples and crowd learning clearly outperforms individual learning. However, this trend is not exactly monotonic primarily because the arrivals of new movies which can give rise to increased error within the corresponding category.

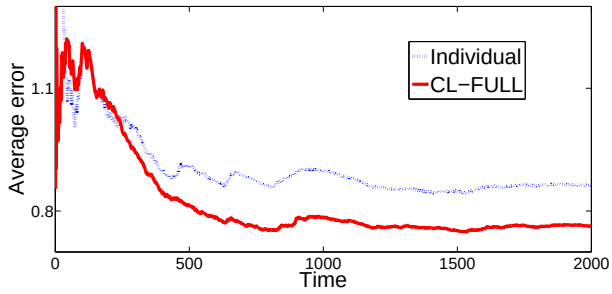


Figure 3.5: Convergence of regret

We next compare how crowd learning compares with individual learning; in the latter case the errors are calculated according to Eqns. (3.16)-(3.17) by computing the estimate $\hat{X}_k^i(t^-)$ using only user i's own past reviews. To eliminate the fluctuation in error at the beginning of the learning, we will use the first half of the data for training and measure errors

Cate. k	1	2	3	4	5	6
Indv.	0.4853	0.6343	0.6620	0.6868	0.7134	0.7278
CL-FULL ($K=3$)	0.4618	0.5173	0.5356	0.6371	0.6551	0.6529
CL-FULL ($K=6$)	0.4826	0.6249	0.6543	0.6776	0.7084	0.7263
Indv.	0.5376	0.7367	0.8251	0.6551	0.7503	0.7574
CL-FULL ($K=3$)	0.6002	0.7195	0.7997	0.7933	0.8648	0.8389
CL-FULL ($K=6$)	0.7071	0.8438	0.8773	0.9276	0.9685	0.9997

Table 3.1: Average error (top 3 rows) and average square root error (bottom 3)

only over the second half, i.e., we separate the dataset to (Training, Test) by $(50\%, 50\%)$ based on their time of arrival.

Furthermore, to see more clearly the effect of the parameter choice K, we compare the error performance by categories. Specifically, we measure the prediction error of each user for movies in her most preferred category and average this over all users. This gives us the error, for example, for Category $k = 1$ shown in Tables 3.1; errors for the other categories are similarly computed up to $k = 6$. For each user the preference ordering of categories is determined using her average rating for each category over the entire data trace. Due to our choice of $K = 3$, we see a clear degradation in the average error performance under the crowd learning algorithm when we go from $k = 3$ to $k = 4$, although the latter still outperforms individual learning. This is to be expected because in making prediction for one's top 3 categories we have the advantage of using the similarity group for help; this may not apply to the next 3 categories as members of the similarity group may not share the same preference over the next 3 categories. This is also seen under the root-squared error measure, where crowd-learning underperforms individual learning for $k \geq 4$. For a complete discussion we also added the performance when choosing $K = 6$. Clearly the big prediction gap between the top 3 and latter categories goes away. We however observe the performance under setting $K = 6$ is generally worse compared with the case of setting $K = 3$ implying the fact that $K = 6$ does not accurately align users by similarity.

3.7.3 Validating partial information algorithm

We next show the convergence of group recommendation factors β using only partial information. To do so we ignore the detailed ratings and use only the review counts that user i has provided for category k, and track the convergence of the estimated frequency $\tilde{\beta}_k^i(t)$. Note that the decision to provide a review does not necessarily imply preference for movies. So the number of reviews is used here as a proxy; as we see below it proves to be a useful one. Accordingly, here similarity is determined by one's rating frequency of each category. From Theorem 3 we know that ideally the frequency for visiting the top choices, $\beta_i^* := \sum_{k \in N_K^i} \tilde{\beta}_k^i(t)$, should converges faster (smaller variance) when combining decision information from similar users. We thus test the average and variance of β_i^* for all users over time. K is set to 3. In Figure 3.6, we see clearly that by adding crowd recommendation, better convergence of β^* is achieved (through reduction in variance indicated by the error bars).

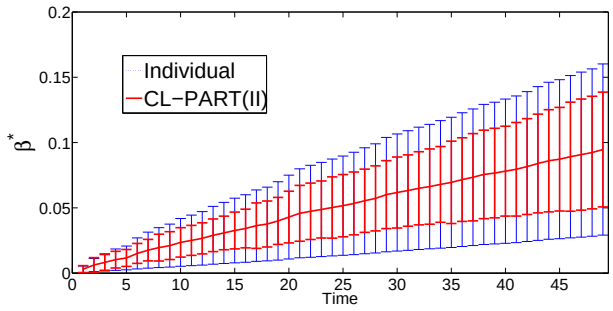


Figure 3.6: Algorithm performance and comparison.

3.7.4 Offline estimation result

We end this section by showing that our algorithms can also be used in an offline setting, to estimate the value of missing entries in the dataset as part of post-processing. The result then becomes comparable to several existing solutions. In particular, we compare our estimation results with that of the following three methods, among the best we have seen in the literature and all of which are essentially based on matrix factorization techniques, see

for example [9]. (1) SoCo [47], which is a social network and contextual information aided recommendation system. A random decision tree is adopted to partition the original user-item-rating matrix (user-movie-rating matrix in the context of the MovieLens-1M dataset) so that items with similar contexts are grouped. (2) RPMF [83], which uses what is known as the Random Partition Matrix Factorization method, a contextual collaborative filtering model based on a tree structure constructed by using random partition technique. (3) MF, the basic matrix factorization techniques utilizing the user-item matrix.

We would like to emphasize that unlike our crowd-learning algorithm, these methods are *not* designed to be executed online, as matrix factorization relies on the existence of the whole matrix (with a few missing entries) which contain all past, present and future samples. For comparison, we also add an "Individual" algorithm in the mix which resembles the individual learning algorithm examined in the preceding subsection but in offline estimation form, i.e., a user's rating is estimated based on all her other (past and future) ratings.

Algorithm	Crowd	Ind.	SoCo	RPMF	MF
Avg. Error	**0.6880**	0.8145	0.7066	0.7223	0.7596
RMSE	0.9054	1.0279	**0.8722**	0.8956	0.9666

Table 3.2: Comparison with matrix factorization methods.

The comparison is presented in Table 3.2: the SoCo results are quoted from [47] and RPMF, MF results are from [83] (all on the same MovieLens dataset), respectively. We see clearly that our crowd-learning algorithm has the best average error performance, while its RMSE underperforms SoCo and RPMF, though comparable. This is because the crowd-learning algorithm uses sample mean measures to guide its learning and has an objective of improving sample mean estimates rather than the mean squared error, so there is a slight mis-match. This is also due to the quantization noise introduced by the bundling of movies mentioned earlier.

It is worth noting that while our crowd learning algorithm does not rely on exogenous

social or contextual information as in SoCo or an explicit tree structure as in both SoCo and RPMF, our learning and estimation of similarity groups implicitly introduces a type of social connectivity among users that achieves similar goals. Furthermore, our estimation of the differences among members of the same similarity group adds one more layer of qualification of peer users when their inputs are combined; these distortion factors make the selection of similarity groups more robust and error tolerant. All these factors contribute to its superior error performance.

Nevertheless the performance difference in the offline setting is not significant. The reasons are as follows. Firstly, offline solutions already utilize the concept of finding similar users/rating in an implicit way. Secondly, often times the offline solutions work better for low rank rating matrices. The gists of our crowd learning based methods are to differentiate similar users from dissimilar ones and thus reduce the rank of subsequent, separated sub-rating matrix. However, the rank of MovieLens data is already low due to its sparse structure, which is the reason we did not see a more pronounced improvement.

3.8 Offline Crowd-Learning

It is indeed remarkable that our crowd-learning algorithms manage to achieve comparable results with the set of offline algorithms, particularly considering the fact that we have significant quantization error built in the algorithm when we bundled the movies into genres. This motivates us to further investigate whether this improvement comes from the idea of "crowd learning" (hopefully) or due to other differences in the algorithm design. To answer this question, we combine the idea of finding a crowd of similar users with classical collaborative filtering and demonstrate this with matrix factorization (MF) [42]. The MF based recommendation methods are rooted in finding and utilizing the hidden interleave embedded in users' ratings. To exploit the idea of "similar crowd" in a more explicit way, we propose a waypoint based collaborative filtering method, the basic idea being to associate similar users to the same group; the hope is that we could potentially reduce the rank

of each of them, thus more similar ratings.

Algorithm 2 Waypoints based matrix factorization (`W-MF`)

1: **Input**: A training rating matrix $\mathscr{M} \in R^{M \times n}$.
2: *Similarity calculation*: For each pair of users (i, j), calculate their similarity $S_{i,j}$ based on the rating samples.
3: *Sampling*: Sample $O(\log M)$ center nodes uniformly. Name these nodes as Waypoint nodes and denote them as $\{w_i\}_{i=1}^{O(\log M)}$.
4: *Association*: For other users $j \notin \{w_i\}_{i=1}^{O(\log M)}$ we associate j to Waypoint w_k that is most similar to itself.

$$w_k = \text{argmax}_{k \in \{w_i\}_{i=1}^{O(\log M)}} S_{j,k} \ .$$

5: *Local MF*: All together we form $O(\log M)$ groups. Form a rating matrix for each group and execute MF method for each one of them.

Denote each user i's rating as $\mathbf{X}^i \in R^n$. It should be noted there are many different ways of calculating the similarity measure $S_{i,j}$, for instance the cosine similarity:

$$S_{i,j} = \frac{\mathbf{X}^{i,T} \cdot \mathbf{X}^j}{||\mathbf{X}^i|| \cdot ||\mathbf{X}^j||} \ . \tag{3.18}$$

The performance of the above algorithm can be found in the Appendix.

3.8.1 Experiments

We apply the above algorithm and apply it to the Movielens data. Firstly we observe for some formed groups, users are not particularly similar to each other. We thus require a group $\mathscr{C}_k$ to satisfy $\frac{\sum_{i \neq j, i,j \in \mathscr{C}_k} S_{i,j}}{|\mathscr{C}_k|^2 - |\mathscr{C}_k|} > \tau$, i.e., we would like the average intra-group similarity to be larger than some threshold. We shall only keep such groups, that is for these groups, we only use data from inside the group to carry out MF based prediction. For groups failing this condition, all data will be used. We have obtained the following experiments results.

Algorithm	MF	W-MF
Avg. Error	0.7596	0.7575
RMSE	0.9666	0.9658

We do not observe a clear performance improvement. The main reasons are as follows. The essence of our crowd learning based methods is to differentiate similar from dissimilar users so as to reduce the rank of the rating matrix. However, the rank of Movielen data is already low due to its sparse structure. We will further validate this conjecture using a set of high rank data in Chapter V.

3.9 Concluding remarks

In this chapter we considered a crowd-learning problem in the context of online recommendation systems with crowdsourced data, and analyzed two cases, where users share full or partial information. We constructed UCB1-like index algorithms and derived bounds on their weak regret. These bounds generally see a multi-fold improvement due to crowdsourcing, i.e., the exploitation of both first-hand and second-hand learning. We demonstrated the power of crowd learning using simulations as well as numerical experiments over MovieLens 1M dataset. We showed that our algorithm can achieve an improvement in movie recommendation quality in an online manner. We also compared our method with several offline algorithms and further proposed an offline algorithm using explicitly the idea of crowd learning.

CHAPTER IV

Finding One's Best Crowd: Online Prediction By Exploiting Source Similarity

4.1 Introduction

The ability to learn (classify or predict) accurately with sequentially arriving data has many applications. Examples include predicting future values on a prediction market, weather forecasting, TV ratings, and ad placement by observing user behavior. The subject of learning in such contexts has been extensively studied. Past literature is heavily focused on learning by treating each source or object's historical data separately, see e.g., [44, 55, 45] for single source multi-armed bandit problems for learning the best options of returned rewards, [38] for a support vector machine based forecasting for financial time series data, [26] for a model predicting spammers using a network's past statistics, and [32] for forecasting stock price index, among other.

More recent development has increasingly been focused on improving learning through integrating data from multiple sources with similar statistics, see e.g., [27] for wind power prediction using both temporal and spatial information. The idea of increasing sample spaces by exploiting similarity proves to be helpful especially when the data arrives slowly, e.g., weather reports generated a few times per day. This idea naturally arises when different data sources are physically correlated, e.g., wind turbines on the same farm, or

environmental monitoring sensors located within close proximity. However, it also fits well in the emerging context of crowdsourcing, where different sources (e.g., Amazon Mechanical Turks) contribute to a common data collection objective (e.g., labeling a set of images), and exploiting multiple data sources can improve the quality of crowdsourced data. For instance the idea of aggregating selectively data from a crowd to make prediction more accurate is empirically demonstrated and referred to as finding a "smaller but smarter crowd" in [20, 23].

In this chapter we seek to make the notion of a "smarter" crowd quantitatively precise and develop methods to systematically identify and utilize this crowd. Specifically, we consider a problem involving K (potentially-)disparate data sources, each may be associated with a user. A given user can use its own data to achieve a certain learning (prediction, classification) objective but is interested in improving its performance by tapping into other data sources, and can request data from other sources at a cost. Accordingly, decisions need to be made judiciously on which sources of data should be used so as to optimize its learning accuracy. This implies two challenges: (1) we need to be able to measure the similarity/disparity between two sources in order to differentiate which sources are more useful toward the learning objective, and (2) we need to be able to determine the best set of sources given the measured similarity. Prior work most relevant to the present study is [14], where the problem of combining static IID data sources is analyzed. There are however a number of key differences: 1) in [14] the similarity information is assumed known a priori and the cost of obtaining data is not considered. 2) The results in [14] are established pre-collected IID data, while we focus on an online learning setting with Markovian data sources. In addition, the methodology we employ in this chapter is quite different from [14] which draws mainly from VC theory [78], while our study is based on both VC theory and the multi-armed bandit (MAB) literature [5].

We will start by establishing bounds on the expected learning error under ideal conditions, including that (1) the similarity information between data sources is known a priori,

and (2) data from all sources are available for free. We then relax assumption (1) and similarly establish the bounds on the error when such similarity information needs to be learned over time. We then relax both (1) and (2) and design an efficient online learning algorithm that simultaneously makes decisions on requesting and combining data for the purpose of training the predictor, and learning the similarity among data sources. We again show that this algorithm achieves a guaranteed performance uniform in time, and the additional cost with respect to the minimum cost required to achieve optimal learning rate diminishes in time. Moreover, the obtained bounds show clearly the trade-off between learning accuracy and the cost to obtain additional data. This provides useful information for system designers with different objectives. To our best knowledge this is the first study on online learning by exploiting source similarity with provable performance guarantees.

The rest of the chapter is organized as follows. We first formulate our problem in Section 4.2, and derive the error bounds under ideal conditions in Section 4.3. We present the results when similarity information needs to be learnt in Section 4.4. We then elaborate a cost-efficient online algorithm and analyze its performance in Section 4.5. Section 4.6 concludes this chapter.

4.2 Problem Formulation

4.2.1 Learning with multiple data sources

Consider K sources of data each associated with a unique user, which we also refer to as the whole crowd of sources. We index them by $\mathscr{D} = \{1,2,\cdots,K\}$. The sources need not be governed by identical probability distributions. Data samples arrive in discrete time to each user; the sample arriving at time t for user i is denoted by $z_i(t) = (x_i(t), y_i(t))$, $t = 1,2,\cdots$, with $x_i(t)$ denoting the features and $y_i(t)$ denoting the labels. At each time t, $x_i(t)$ is revealed first followed by a prediction on $y_i(t)$ made by the user, after which $y_i(t)$ is revealed and $z_i(t)$ is added to the training set. For simplicity of exposition, we

will assume $x_i(t)$ to be a scalar; however our analysis easily extends to more complex forms of data, including batch arrivals. The objective of each user is to train a classifier to predict $y_i(t)$ using collected past data, and after prediction at time t, $y_i(t)$ will be revealed and can be used for training in the future steps. As a special case, when the target is to predict for future, $y_i(t)$ can be taken as $x_i(t+1)$. For analytical tractability we will further assume that the data arrival processes $\{x_i(t)\}_t$, $\forall i$, are mutually independently (but not necessarily identical), and each is given by a first order[1] finite-state positive recurrent Markov chain, with the corresponding transition probability matrix denoted by P^i on the state space $\mathscr{X}^i$ ($|\mathscr{X}^i| < \infty$). Denote by $P^i_{x,y}$ the transition probability from state x to y under P^i, and by π^i its stationary distribution on $\mathscr{X}^i$. For simplicity we will assume that $\mathscr{X}^1 = \mathscr{X}^2 = ... = \mathscr{X}^K = \mathscr{X}$, though this assumption can be easily relaxed, albeit with more cumbersome notation. The motivation for such modeling choice is by observing that for many applications the sequentially arriving data does not follow IID distribution as has been studied in the literature; consider e.g., weather conditions. Suppose labels $y_i(t) \in \mathscr{Y}^i$ and again for simplicity let us assume $\mathscr{Y}^1 = \mathscr{Y}^2 = ... = \mathscr{Y}^K = \mathscr{Y}$, and $|\mathscr{Y}| < \infty$. Denote $y^* := \max_{y \in \mathscr{Y}} |y|$.

For the classification job, a straightforward approach would be for each user i to build a classifier/predictor by using past observations of its own data up to time t: $\{z_i(1), ..., z_i(t)\}$. Denote the classifier by f_i for user i, and a loss function $\mathscr{L}$ to measure the classification error. For instance $\mathscr{L}$ can be taken as the squared loss function $\mathscr{L}(f_i, z_i(t)) = [y_i(t) - f_i(x_i(t))]^2$. With the definition of loss function, the classification task for a user is to find the classifier that best fits its past observations:

$$f_i(t) = \text{argmin}_{f \in \mathscr{F}} \sum_{n=1}^{t} \mathscr{L}(f, z_i(n)) \,, \tag{4.1}$$

where we have used $\mathscr{F}$ to denote the set of all models of classifier (hypothesis space). For example, $\mathscr{F}$ could contain the classical linear regression models.

[1] A high order extension is also straightforward.

The idea we seek to explore in this chapter is to construct the classifier f_i by utilizing similarity embedded among data sources, i.e., we ask whether f_i should be a function of all sources' past data and not just i's own, and if so how should such a classifier be constructed. Specifically, if we collect data from a set Ω_k of sources and use them as if they were from a single source, then the best classifier is given by

$$f_{\Omega_k}(t) = \text{argmin}_{f \in \mathscr{F}} \sum_{j \in \Omega_k} \sum_{n=1}^{t} \mathscr{L}(f, z_j(n)) \ . \tag{4.2}$$

It was shown in [14] that the expected error of the above classifier is bounded by a function of certain source similarity measures; the higher the similarity the lower the error bound.

Our interest is in constructing the best classifier for any given user i by utilizing other data sources. To do so we will need to measure the similarity or discrepancy between sources and to judiciously use data from the right set of sources. We will accomplish this by decomposing the problem into two sub-problems, the first is to use a similarity measure to determine a preferred set Ω_k^* to use, and the second is to construct the classifier using data from this set.

4.2.2 Pair-wise similarity between data sources

We first introduce the notion of cross-classification error, which is the expected loss when using classifier f_j (trained using source j's data) on user i's data and can be formally defined as $r_i(f_j) = E_i[\mathscr{L}(f_j, z_i)]$ where the expectation is with respect to user i's source data distribution. In principle, this could be used to measure the degree of similarity between two data sources i and j. However, this definition is not easy to work with as it involves a classifier that is only implicitly given in (4.1). Instead, we introduce a notion of similarity between two data sources i and j, that satisfies the following two conditions: (1) it can be obtained from the statistics of two respective data sources, and (2) it satisfies the following

bound:

$$r_i(f_j) \leq \beta_1(1 - S_{i,j}) + \beta_2 \ , \tag{4.3}$$

where $\beta_1, \beta_2 \geq 0$ are normalization constants and $0 \leq S_{i,j} \leq 1$ denotes the similarity measure; the higher this value the more similar two sources. The relationship captured in Eqn. (4.3) between the error function and similarity can also take on alternate forms; we adopt this simple linear relationship for simplicity of exposition. The following example shows the existence of such a measure.

Suppose for each user i, corresponding to each state/feature $x \in \mathscr{X}$, labels $y \in \mathscr{Y}$ is generated according a probability measure Q^i_x and denote each probability as $Q^i_{x,y}$ and $\sum_{y \in \mathscr{Y}} Q^i_{x,y} = 1$. Consider the following example. Take $\mathscr{L}$ as the squared loss and $S_{i,j}$ as:

$$S_{i,j} = 1 - \max_{x \in \mathscr{X}, y \in \mathscr{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2 \ . \tag{4.4}$$

Then we can show[2] that, by setting $\beta_1 := 2\sum_{y \in \mathscr{Y}} y^2$ and $\beta_2 := 2\sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y} - y)^2$, i.e. two times the intrinsic classification error with user i's own (perfect) data, which is independent with other sources j, the choice of $S_{i,j}$ satisfies both conditions. We note that the choice of such an S is not unique. For example, we could also take $S_{i,j}$ to be

$$S_{i,j} = 1 - \sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2 \ ,$$

while setting $\beta_1 := 2(y^*)^2$. Later we will argue that an S that leads to a tighter bound can help achieve a better performance in classification. As it shall become clearer later when such similarity information needs to be estimated, the trade-off between selecting a tighter and looser similarity measure comes from the fact that tighter similarity may incur more

[2]Please refer to supplementary materials.

learning error as it requires the evaluation of more terms.

Without loss of generality, for the remainder of our discussion we will focus on user 1. We will also denote $s_i := \min\{S_{1,i}, S_{i,1}\}, \forall i$. While the definition given in (4.4) is symmetric in i and j such that $S_{1,i} = S_{i,1}$, this needs not be true in general under alternate definitions of similarity. Note that $s_1 = 1$. We will then relabel the users in decreasing order of their similarity to user 1: $1 = s_1 \geq s_2 .... \geq s_K \geq 0$.

4.3 Solution with Complete Information

As mentioned earlier, the problem of finding the best set of data sources to use and that of finding the best classifier given this set are inherently coupled and strictly speaking need to be jointly optimized, resulting in significant challenges. The approach we take in this chapter is as follows. We will first derive an upper bound on the error of the classifier given in (4.2) when applied to user 1, for a set of k independent Markov sources; this bound is shown to be a function of k and their similarity with user 1. This bound is then optimized to obtain the best set. Below we derive this upper bound assuming (1) the similarity information is known and (2) data is free, i.e., at time t all past and present samples from all sources are available to user 1.

4.3.1 Upper bounding the learning error

First notice we have the following convergence results for positive recurrent Markov Chain we consider in this chapter [66],

$$||\tilde{\pi}^i(t) - \pi^i||_{\text{TV}} \leq C_{\text{MC}} \cdot (\lambda_2^i)^t \ ,$$

where C_{MC} is some positive constant, $\tilde{\pi}_x^i(t)$ is the empirical distribution of state x for data source i's Markov chain upto time t for user i and π_x^i denotes its stationary distribution, and $0 < \lambda_2^i < 1$ is the second largest eigenvalue which specifies the mixing speed of the process.

The total variation distance $||p-q||_{\text{TV}}$ between two probability measures p and q that are defined on $\mathscr{X}$ is defined as follows

$$||p-q||_{\text{TV}} := \max_{S\in 2^{\mathscr{X}}} |\sum_{x\in S}(p(x)-q(x))| \,. \tag{4.5}$$

Denote $\rho_{k(t)}(t) := \max \mathscr{L} \cdot C_{\text{MC}} \frac{\sum_{i\in k(t)}(\lambda_2^i)^t}{|k(t)|}$, where $\max \mathscr{L}$ is the maximum value attained by the loss function. Throughout the chapter we denote $[k] := \{1,2,...,k\}$ as the ordered and continuous set up to k, and $k(t)$ for any other un-ordered set invoked at time t and use $|k(t)|$ to denote its size. For squared loss function we have the following results:

Theorem IV.1. *At time t, with probability at least* $1-O(\frac{1}{t^2})$ *the error of a classifier* $f_{[k]}(t)$ *constructed using data from k sources of similarity* $s_i, i \in k(t)$ *can be bounded as*

$$\begin{aligned} r_1(f_{k(t)}(t)) \leq & \underbrace{4\min_{f\in\mathscr{F}} r_1^{\text{IID}}(f) + 6\beta_2}_{\text{Term 1}} + \underbrace{6\beta_1 \frac{\sum_{i\in k(t)}(1-s_i)}{|k(t)|}}_{\text{Term 2}} \\ & + \underbrace{\rho_{k(t)}(t)}_{\text{Term 3}} + \underbrace{8y^*(2\sqrt{2d}+y^*)\sqrt{\frac{\log|k(t)|t}{|k(t)|t}}}_{\text{Term 4}} \,, \end{aligned} \tag{4.6}$$

where d is the VC dimension for $\mathscr{F}$*, and* $r_1^{\text{IID}}(f)$ *is the expected prediction error when the data are generated according to an IID process.*

Denote the upper bound for $r_1(\cdot)$ in Eqn. (4.6) with set $k(t)$ of data sources (after ordering based on their similarity with user 1) at time t by $\mathscr{U}_{k(t)}(t)$. The results may be viewed as an extension to the previous one from [14] where static and IID data sources were considered. This upper bound can serve as a good guide for the selection of such a set and in particular the best choice of $|k(t)|$ given estimated values of s_i's [3]. Note that Terms 1 is independent of this selection and it is a function of the baseline error of the classification problem, Term 2 is due to the integration of disparate data sources, Term 3 comes from the

[3] We will show finding such optimal set can be done by a linear search later.

mixing time of a Markov source, and Term 4 arises from imperfect estimation and decision using a finite number of samples ($|k(t)|t$ samples up to time t).

Below we first point out the key steps in the proof that differ from that in [14] (full proof is in the supplementary materials), and then highlight the properties of this bound.

4.3.1.1 Main steps in the proof

Our analysis starts with connecting Markovian data sources to IID sources so that the classical VC theory [78] and corresponding results can apply. The idea is rather simple: by the ergodicity assumption on the arrival process, the estimation error converges to that of IID data sources as shown in [1]. In particular, we can bound the difference in error when applying a predictor $f \in \mathscr{F}$ to a Markovian vs. an IID source (with distribution being the same as the steady state distribution of the Markov chain) at time t, constructed with available data as follows:

$$
\begin{aligned}
&|r_i(f(t)) - r_i^{\mathrm{IID}}(f(t))| \\
&= |\sum_{x\in\mathscr{X}} \tilde{\pi}_x^i E_{y\sim\mathscr{Y}}[\mathscr{L}(f(t),(x,y))] - \sum_{x\in\mathscr{X}} \pi_x^i E_{y\sim\mathscr{Y}}[\mathscr{L}(f(t),(x,y))]| \\
&\leq \max\mathscr{L}\cdot C_{\mathrm{MC}}(\lambda_2^i)^t \ .
\end{aligned}
$$

We impose α-triangle inequality on the error function $\forall i,j,k$, of the corresponding data sources $r_i(f_j) \leq \alpha\cdot[r_i(f_k)+r_k(f_j)]$, where $\alpha \geq 1$ is a constant. When $\mathscr{L}$ is the squared loss function, we have $\alpha = 2$, following Jensen's inequality. Then $\forall f$

$$
\frac{r_1(f)}{k} \leq \frac{\alpha\cdot[r_1(f_i)+r_i(f)]}{k} \ .
$$

Sum over all $i \in k(t)$ we have

$$
r_1(f) \leq \frac{\alpha\beta_1}{|k(t)|}\cdot\sum_{i\in k(t)}(1-s_i) + \alpha\beta_2 + \alpha\cdot\bar{r}_{k(t)}(f) \ ,
$$

where $\bar{r}_{k(t)}(f) = \frac{\sum_{i \in k(t)} r_i(f)}{|k(t)|}$ is the average regret by applying f onto the $|k(t)|$ data sources. Due to the bias of mixing time for Markovian sources we have the following fact :

$$\bar{r}_{k(t)}(f) \leq \bar{r}^{\text{IID}}_{k(t)}(f) + \rho_{k(t)}(t).$$

The rest of the proof focuses on bounding $\bar{r}^{\text{IID}}_{k(t)}(f)$, i.e., the expected prediction error on IID data sources, which is similar in spirit to that presented in [14].

4.3.1.2 Properties of the error bound

The upper bound $\mathscr{U}_{k(t)}(t)$ has the following useful properties.

Proposition IV.2. *For sources ordered in decreasing similarity $s_1 \geq s_2 \cdots$, $\frac{\sum_{i=1}^{k} s_i}{k}$ is non-increasing in k.*

This is straightforward to see by noting that

$$\frac{\sum_{i=1}^{k+1} s_i}{k+1} - \frac{\sum_{i=1}^{k} s_i}{k} = -\sum_{i=1}^{k} \frac{s_i}{k(k+1)} + \frac{s_{k+1}}{k+1} = \sum_{i=1}^{k} \frac{s_{k+1} - s_i}{k(k+1)} \leq 0 \,.$$

Terms 3 and 4 both decrease in time. While Term 4 converges at the order of $O(1/\sqrt{t})$, Term 3 generally converges with geometric rate, which is much faster than Term 4 and can be ignored for now. We then know because of the use of multiple sources, Term 4 decreases $|k(t)|$ times faster, leading to a better bound. This shows how the use of multiple sources fundamentally changes the behavior of the error bound.

The upper bound also suggests that the optimal selection is always to choose those with the highest similarity, which leads to a linear search for the optimal number k. Based on above discussions, the trade-off comes from the fact a larger k returns a smaller average similarity term $\sum_{i=1}^{k} s_i/k$ (and thus a larger $\sum_{i=1}^{k}(1-s_i)/k$), while with more data we have a faster convergence of Term 4. Define the optimal set of sources at time t as the one minimizing the bound $\mathscr{U}_{k(t)}(t)$, and denote it by $k^*(t)$. We then have the following fact,

Proposition IV.3. *When $\{s_i\}_{i\in\mathscr{D}}$ is known, $\exists\ t_o$, such that $\forall t \geq t_o$, if $i \in k^*(t)$ then $i \in k^*(n), \forall t_o \leq n \leq t$.*

This implies that if a data source is similar enough to be included at t, then it would have been included in previous time steps as well except for a constant number of times. This also motivates us to observe a threshold or phase transitioning phenomenal in selecting each user's best crowd. This result is also crucial in proving Theorem IV.6 where it helps establish bounded number of missed sampling for an optimal data source in an adaptive algorithm.

Proposition IV.4. *A set of tighter similarity measures S returns better worst case performance.*

Consider two such similarity measures s' and s with $s'_i \geq s_i$ (with at least one strict inequality). Suppose at any time t and optimal set of crowd for s is $k(t)$, then simply by selecting $k(t)$ for s' we achieve a better worst case performance (a smaller $\sum_{i=1}^{k}(1-s_i)/k$ in upper bound).

4.4 Overhead of Learning Similarity

As we show in the previous section, once the optimal set of data sources is determined, the classification/prediction performance is bounded. However in a real crowdsourcing system, neither of the two assumptions may be valid. In this section we relax the first assumption and consider a more realistic setting where the similarity information remains unknown *a priori* and can only be learned through shared data. In this regards we need to estimate the similarity information $\{s_i\}_{i\neq 1}$ while making decision of which set of data sources to use.

The learning process works in the following way. At step t, we first estimate similarity

$\tilde{s}_i$ according to the following:

$$\tilde{s}_i = 1 - \max_{x\in\mathscr{X},y\in\mathscr{Y}} |\tilde{Q}^i_{x,y}(t) - \tilde{Q}^1_{x,y}(t)|^2 \ ,$$

where $\tilde{Q}^i_{x,y}(t) := \frac{n_{i,x\to y}(t)}{n_{i,x}(t)}$ are the estimated transition probability matrices with $n_{i,x}(t)$ denoting the number of times user i is sampled to be in state $x \in \mathscr{X}$ up to time t and $n_{i,x\to y}(t)$ denoting the number of observed samples from data source i being in (x,y). Different from the previous Section, now since $\{s_i\}_{i\neq 1}$ is unknown, in order to select data sources, the estimate of the upper bound $\mathscr{U}_{k(t)}(t)$ becomes a function of $\{\tilde{s}_i\}$: $\mathscr{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$, which is obtained by simply substituting all s terms in $\mathscr{U}_{k(t)}(t)$ with $\tilde{s}$. Denote the terms that are being affected by choosing set $k(t)$ in $\mathscr{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ as follows:

$$\tilde{\mathscr{U}}^{tr}_{k(t)}(t) = 6\beta_1 \frac{\sum_{i\in k(t)}(1-\tilde{s}_i)}{k} + 8y^*(2\sqrt{2d}+y^*)\sqrt{\frac{\log|k(t)|t}{|k(t)|t}} \ .$$

Note we are omitting $\rho_{k(t)}(t)$ as it is on a much smaller order and will not affect our results order-wise.

Then the learning algorithm first orders all data sources according to $\{\tilde{s}_i\}$, and then choses $\tilde{k}^*(t)$ by a linear search such that

$$\tilde{k}^*(t) = \arg\max_{[k],1\leq k\leq K} \tilde{\mathscr{U}}^{tr}_{[k]}(t) \ .$$

We have the following results.

Theorem IV.5. *At time t, with probability at least* $1 - O(\frac{1}{t^2})$ *the error of trained classifier* $f_{\tilde{k}^*(t)}(t)$ *using* $\tilde{k}^*(t)$ *data sources can be bounded as follows*

$$r_1(f_{\tilde{k}^*(t)}(t)) \leq \mathscr{U}_{k^*(t)}(t) + O(\sqrt{\frac{\log t}{t}}) \ . \tag{4.7}$$

Clearly from above results we see there is an extra $O(\sqrt{\log t/t})$ term capturing the loss

of learning the similarity information.

4.5 A Cost-efficient Algorithm

Now we relax the second restriction on data acquisition. In reality data acquisition from other sources are costly. In our study, we explicitly model this aspect whereby at each time step a user may request data from another user at a unit cost of c. This modeling choice not only reflects reality, but also allows us to examine the trade-off between a user's desire to to keep its overall cost low while keeping its prediction performance high. We present a cost-efficient algorithm with performance guarantee. As one may expect, with less data the prediction accuracy will degrade. But the number of unnecessary data will also be bounded from above.

4.5.1 A cost-efficient online algorithm

Denote by $n_i(t)$ the number of collected samples from source i up to time t and $N_{k(t)}(t) = \sum_{i\in k(t)} n_i(t)$. Notice in this section $n_i(t) \neq t$ in general. Denote $D(t) := O(t^z)$; z will be referred to as the exploration constant satisfying $0 < z < 1$. Later we will show how z controls the trade-off between data acquisition and classification accuracy. Again denote by $n_{i,x}(t)$ the number of times user i is sampled to be in state $x \in \mathscr{X}$ up to time t and construct the following set at each time t :

$$\mathscr{O}(t) = \{i : i \in \mathscr{D}, \exists\, x \in \mathscr{X}, n_{i,x}(t) < D(t)\}\ .$$

We name the algorithm as `K-Learning`, which consists mainly of the following two steps (run by user 1):

Exploration: At time t, if any data source has a state x that has been observed (from requested data) for less than $D(t)$ times, i.e., if $\mathscr{O}(t)$ is non-empty, then the algorithm enters an exploration phase and collects data from *all* sources $k_2(t) = \mathscr{D}$ and predicts via its own

Algorithm 3 `K-Learning`

Initialization:
Set $t = 1$ and similarity $\{\tilde{s}_i(1)\}_{i\in\mathscr{D}}$ to some value in $[0,1]$; $n_{i,x}(t) = 1$ for all i and x.
loop:
Calculate $\mathscr{O}(t)$.
if $\mathscr{O}(t) \neq \emptyset$ **then**
    Explores, sets $k_1(t) = \{1\}, k_2(t) = \mathscr{D}$.
else
    Exploits, orders data sources according to $\{\tilde{s}_i(t)\}_{i\in\mathscr{D}}$ and computes $k_1(t)$ that minimizes $\tilde{\mathscr{U}}^{tr}_{k_1(t)}(t)$, which is solved using the linear search property, and the current estimates $\{\tilde{s}_i(t)\}_{i\in\mathscr{D}}$. Set $k_2(t)$ as $k_2(t) := \text{argmax}_{k'(t)\subseteq D}\{|k'(t)| : \tilde{\mathscr{U}}^{tr}_{k'(t)}(t) \in [\tilde{\mathscr{U}}^{tr}_{k_1(t)}(t) - \sqrt{\frac{\log t}{t^z}}, \tilde{\mathscr{U}}^{tr}_{k_1(t)}(t) + \sqrt{\frac{\log t}{t^z}}]\}$.
end if
Construct classifier $f_{k_1(t)}$ using data collected from sources in $k_1(t)$. Request data from $k_2(t)$.
$t := t+1$ and update $\{n_{i,x}(t)\}_{i,x}$, $\{\tilde{s}_i(t)\}_{i\in\mathscr{D}}$ using collected samples.
goto *loop*.

data $k_1(t) = \{1\}$. The prediction at exploration phase is *conservative* since without enough sampling user 1 cannot be confident in calculating its optimal set of similar sources, in which case the user would rather limit itself to its own data.

Exploitation: If $\mathscr{O}(t)$ is empty at time t then the algorithm enters an exploitation phase, whereby it first estimates similarity measures of all sources. For our analysis we will use the same definition given earlier: $\tilde{s}_i(t) = 1 - \max_{x\in\mathscr{X},y\in\mathscr{Y}} |\tilde{Q}^i_{x,y}(t) - \tilde{Q}^1_{x,y}(t)|^2$. The algorithm then calculates $k_1(t)$ using the estimated bound $\tilde{\mathscr{U}}^{tr}_{k_1(t)}(t)$, and uses data from this set $k_1(t)$ of sources for training the classifier, while requesting data from set $k_2(t)$, where $k_2(t)$ is set to be:

$$k_2(t) := \text{argmax}_{k'(t)\subseteq D}\{|k'(t)| : \tilde{\mathscr{U}}^{tr}_{k'(t)}(t) \in [\tilde{\mathscr{U}}^{tr}_{k_1(t)}(t) - \sqrt{\frac{\log t}{t^z}}, \tilde{\mathscr{U}}^{tr}_{k_1(t)}(t) + \sqrt{\frac{\log t}{t^z}}]\} .$$

Notice when calculating $k_2(t)$ we set a tolerance region (due to imperfect estimation of $\tilde{\mathscr{U}}^{tr}_{k_1(t)}(t)$) so that a sample data from an optimal data source will not be missed with high

probability.

4.5.2 Performance of `K-Learning`

There are three types of error in the learning performance: (1) Error due to exploration, in which case the error comes from conservative training due to no enough sampling. Due to technical difficulties, we approximate the error (compared to the performance with optimal classifier) by the worst case performance loss, that is the performance difference in upper bounds. (2) Prediction error associated with incorrect computation of $k_1(t)$ (i.e., $k_1(t) \neq k^*(t)$) in exploitation due to imperfect estimates on $\{s_i\}_{i\neq 1}$. (3) Prediction error from sub-sampling effects. This is because even though under the case that $k_1(t) = k^*(t)$, i.e., $k^*(t)$ is correctly identified, due to incomplete sampling, $\exists\, i > 1, n_i(t) < t$, $\hat{\mathscr{U}}_{k_1(t)} \neq \mathscr{U}_{k_1(t)}$, where $\hat{\mathscr{U}}_{k_1(t)}$ is the upper bound for the classification error with collected data: this can be similarly derived following the proof of Theorem IV.1 and results in [14]:

$$\hat{\mathscr{U}}_{k(t)}(t) = 4 \min_{f\in\mathscr{F}} r_1^{\widetilde{\mathrm{IID}}}(f) + 6\beta_2 + 6\beta_1 \frac{\sum_{i\in k(t)} n_i(t)(1-s_i)}{N_{k(t)}(t)} + \tilde{\rho}_{k(t)}(t) + 8y^*(2\sqrt{2d} + y^*) \cdot \sqrt{\frac{\log N_{k(t)}(t)}{N_{k(t)}(t)}}\,,$$

where

$$\tilde{\rho}_{k(t)} := \max \mathscr{L} \cdot C_{\mathrm{MC}} \frac{\sum_{i\in k(t)} (\lambda_2^i)^{n_i(t)}}{|k(t)|}\,,$$

and $\min_{f\in\mathscr{F}} r_1^{\widetilde{\mathrm{IID}}}(f)$ is error rate over a biased data distribution due to incomplete sampling, compared to the target IID distribution. We emphasize that the difference between $\mathscr{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ and $\hat{\mathscr{U}}_{k(t)}(t)$: $\mathscr{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ is the estimation of upper bound $\mathscr{U}_{k(t)}(t)$ with estimated similarity information $\tilde{s}$, while $\hat{\mathscr{U}}_{k(t)}(t)$ bounds actual error of the learning task at each step. In $\mathscr{U}_{k(t)}(t)$ and $\mathscr{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$, full samples are assumed to have been collected for each data source in $k(t)$, i.e., $n_i(t) = t$. However this is not true for $\hat{\mathscr{U}}_{k(t)}(t)$, except for $n_1(t)$ the data source for user 1 itself. Also due to dis-continuous sampling for

Markovian data, the sampled data distribution is biased which results in $\min_{f\in\mathscr{F}} \widetilde{r_1^{\text{IID}}}(f)$. The main gist of bounding this discrepancy is that due to Proposition IV.3 we are able to bound the missed samples for a data source appearing in the optimal set.

A subtle difference between the results in this section and the previous one is the performance of the classifier trained during an *exploration* phase is simply the one using user 1's own data, which is bounded away from the optimal performance bound (via data sources $k^*(t)$). Denote the worse case performance loss (difference in performance upper bound) in exploration phases upto time t as $R_e(t)$, that is

$$R_e(t) = \sum_{n=1}^{t} 1\{O(n) \neq \emptyset\} \cdot |\mathscr{U}_{[1]}(t) - \mathscr{U}_{k^*(t)}(t)| \ . \tag{4.8}$$

This is a quantity we are interested in determining for exploration phases. For exploitation phases, we evaluate the prediction/classification performance as the ones with classifier $f_{k_1(t)}(t)$.

Theorem IV.6. *At time t,*

- *The number of exploration phases is bounded as follows,*

$$E[\sum_{n=1}^{t} 1\{O(n) \neq \emptyset\}] \leq O(t^z) \ .$$

  Further the per round performance loss due to exploration phases $\frac{E[R_e(t)]}{t}$ is bounded as follows: with probability being at least $1 - O(e^{-Ct^z})$ where $C > 0$ is a constant,

$$\frac{E[R_e(t)]}{t} \leq O(\sqrt{z \cdot \log t} \cdot t^{z/2-1}) \ .$$

- *If t is an exploitation phase, with probability being at least $1 - O(\frac{1}{t^2})$ we bound the*

average prediction error for classifier $f_{k_1(t)}(t)$ with data sources $k_1(t)$ as follows,

$$r_1(f_{k_1(t)}(t)) \leq \mathscr{U}_{k^*(t)}(t) + O(\sqrt{\frac{\log t}{t^z}}) + O(\log t \cdot t^{-2/3}) \ .$$

Note on the bound:

- $O(\sqrt{z \cdot \log t} \cdot t^{z/2-1})$ is the average error invoked by exploration. This term is diminishing with t, that is the average amount of exploration error is converging to 0. $O(\sqrt{\log t / t^z})$ is the learning error incurred in exploitation phases, which is in analogy to the $O(\sqrt{\log t / t})$ term as shown in the bound proved in Theorem IV.5. $O(\log t \cdot t^{-2/3})$ is also incurred in exploitation phases. This is a unique error term associated with sub-sampling of Markovian data: due to (1) missed sampling and (2) discontinuous sampling.

- It should be noted that the prediction error term $O(\sqrt{\log t / t^z})$ decrease with z for $0 < z < 1$. That is with a higher z, a tighter bound can be achieved. With $z \to 1$ (number of samples cannot go beyond t at time t), we can show the prediction error term converges to $O(\sqrt{\log t / t})$, which is consistent with the results we reported in last section. Also it worths pointing out $O(\log t \cdot t^{-2/3})$ is generally on a smaller order compared to $O(\sqrt{z \cdot \log t} \cdot t^{z/2-1})$ and $O(\sqrt{\log t / t^z})$: simply set z to be $z > 2/3$.

- This observation also sheds lights on establishing the tightness of this bound for z close to 1, as $O(\sqrt{\log t / t})$ is the uniform convergence bound as proved in statistical learning theory [78].

4.5.3 Cost analysis

To capture the effectiveness of cost saving, we define the following difference in cost:

$$\text{Cost measure} : R_c(t) = c \sum_{n=1}^{t} \sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \ .$$

$R_c(t)$ will be referred to as the cost measure, which quantifies the amount of data requests from non-optimal data sources. We have the following main results.

Theorem IV.7. *At time t, we have* $E[R_c(t)] \leq O(ct^z)$.

Notes on the bound:

- First of all note that $E[R_c(t)] = o(t)$ when $t < 1$ and thus $E[R_c(t)]/t \to 0$ as $t \to \infty$. This demonstrates the cost saving property of our algorithm as the average number of redundant data request is converging to 0.

- Clearly z controls the trade-offs between prediction accuracy $r_1(f_{k_1(t)}(t))$ and data acquisition cost regret $E[R_c(t)]$. A higher z leads to a more frequent sampling scheme and thus higher cost regret, while with a small z the sampling is conservative which leads to higher prediction error.

4.6 Concluding remarks

In this chapter we considered a problem of finding best crowd for a user to enhance its online prediction performance with disparate sources of sequentially arriving data. In particular we proposed and analyzed an online algorithm to help users adaptively distinguish between similar and dis-similar data sources and aggregate appropriately selected external sources of arriving data for the purpose of training the predictor. Meanwhile our algorithm helps avoid requesting redundant data from sources that are helpless (or even harmful) and thus saves cost.

CHAPTER V

Enhancing Multi-source Measurement Using Similarity and Inference: A Case Study of Network Security Interdependence

5.1 Introduction

In this chapter we will put some of the results derived earlier in the application context of a particular set of data collected over the Internet that is of a multi-source nature. We will highlight the utility of some of our algorithms in this context. We will also demonstrate that often times data-specific processing methods arise within specific application contexts that can be exploited to offer additional insight.

Our dataset centers on observations of malicious activities originated from different networks on the Internet. These are often symptoms of their security posture and policies adopted by them. In particular, the dynamics in such activities reveal rich information on the evolution of networks' underlying conditions and therefore are helpful in capturing behaviors in more consistent ways. At the same time, the interdependence of today's Internet also means that what we see from one network is inevitably related to others. This connection can provide insight into the conditions of not just a single network viewed in isolation, but multiple networks viewed together. This understanding in turn allows us to predict more accurately the security conditions or malicious activities of networks in general, and

can lead to the design of better proactive risk aware policies for applications ranging from traffic engineering, peering and routing.

In this chapter we study this connectedness through the following: (1) The notion of similarity as have been defined and studied in previous chapters; in this application context similarity is a quantitative measure of the extent to which the dynamic evolutions of malicious activities from two networks are alike. (2) We shall map this behavioral similarity to their similarity in certain spatial features, including their relative proximity to each other. We then seek to understand how such similarity can help us in uncovering or predicting the maliciousness of a network.

In this study we utilize a set of commonly used IP-address based host reputation blacklists (RBLs) collected over the past year and a half as representations of observed maliciousness. These RBLs broadly cover three categories of malicious activities: spam, phishing, and active scanning. We measure a network's maliciousness from the behavior observed from the vantage points of these RBLs. In doing so, an emphasis is placed on capturing the *dynamic* behavior of malicious activities, rather than an average over time. Specifically, the maliciousness of a network is determined by its presence on these RBLs (either collectively or by different malicious activity types) as a temporal process (e.g., the amount of IP addresses within the network that show up on these RBLs at any given time). The similarity between two networks' maliciousness can then be measured by correlating the two corresponding temporal signals. Due to our emphasis on measuring maliciousness as a time-varying process, the resulting similarity between two networks not only captures the similarity in the magnitude of their presence on these RBLs, but also more importantly captures any *synchrony* in their behavior. This is a distinct aspect that sets our study apart from existing literature, which largely focuses on measuring the level of malicious activities as a time average (see e.g., [82]). To avoid suffering from the issue of dynamic IP allocation, we aggregate the RBL data to BGP routed prefix level to have a relatively more steady description, that is we study this maliciousness on network-level.

To measure the relative proximity of two networks, we consider a variety of spatial characteristics (precise definitions are given in subsequent sections). The first is AS membership: two prefixes belonging to the same AS are considered close. The second is AS type: two prefixes that belong to ASes of the same (business) type are considered close. The third is AS connectivity: the physical distance between two prefixes as measured by the number of hops (direct connections) between the ASes they each belong to. The fourth is country/geographical affiliation: two prefixes residing in the same country are considered close. These spatial characteristics are obviously non-exhaustive, but constitute a fairly comprehensive set of features. We are going to use statistical inference to determine key metrics (out from above) that impact our similarity measure. This basic technique has been used for analysis over a networks' spatial features [7]. An interesting branch of graph analysis is community detection. [58] proposes a spectral clustering algorithm along with theoretical analysis and performance guarantees and [18] analyzed and adopted spectral clustering for community detection in large scale networks. In addition, [81] proposed an evolutionary clustering technique to track the dynamics of community change. Causal inference over latent variables is discussed in [71] and most relevantly, [60] proposed a multi-layer model for analyzing inference problems. These results are going to offer us a rather novel perspective in understanding large scale temporal relationships in various malicious activities.

The contributions of this study lie in two distinct areas. The first is specific to the application of cyber security. In this context we make interesting observations on how network-level malicious activities, a sign of the corresponding networks' security posture, are related to each other, and how this understanding helps us gain insight into the interdependence of network security. Of particular interest we found within the set of spatial features the topological distance between two networks is by far the most significant indicator of their similarity in maliciousness, i.e., the closer two networks the more likely they have similar security posture or in other words, there exist "good" and "bad" neighborhoods

Type	Blacklist Name
Spam	CBL[10], SBL[70], SpamCop[69], WPBL[80], UCEPOTECT[76]
Phishing/Malware	SURBL[72], Phish Tank[61], hpHosts[30]
Active attack	Darknet scanner list, Dshield[17], OpenBL[59]

Table 5.1: The RBL datasets

on the Internet.

Beyond the application, our study also contributes to the understanding of collaborative filtering methods in non-classical settings, and highlights how data-specific features and domain knowledge can greatly enhance our ability in data processing. Although our evaluations in this chapter are limited to the specific set of measurement data, we believe the notion of similarity and the explicit use of similarity in filtering more generally applicable, especially when traditional collaborative filtering methods fail.

In the remainder of this chapter, our datasets are detailed in Section 5.2 and the behavioral similarity study in Section 5.3. We present the inference method, its performance analysis and experimental results in Section 5.4. Section 5.5 concludes this chapter.

5.2 The Dataset and Preliminaries

5.2.1 RBL

Our analysis is performed over 11 IP address-based reputation blacklists over the period January 2013-June 2014. The sampling rate is once per day (i.e., the list content is refreshed on a daily basis). Table 5.1 summarizes these lists and the types of malicious activities they target: spam, phishing/malware, and active scanning. All combined this dataset includes more than 164 million unique IP addresses.

5.2.2 Internet geographical datasets

In addition to the RBLs, this study also employs the following sets of data on spatial and proximity features of networks which are all extracted from BGP global routing table

snapshots and other public information sources:

1. *AS membership*, which reveals which prefix belongs to which AS.

2. *AS types* which associates a given AS with one of four types based on their broad overall role in the Internet eco-system: Enterprise Customers (ECs), Small Transit Providers (STPs), Large Transit Providers (LTSs), and Content/ Access/ Hosting Providers (CAHPs); this is done following methods proposed in [16].

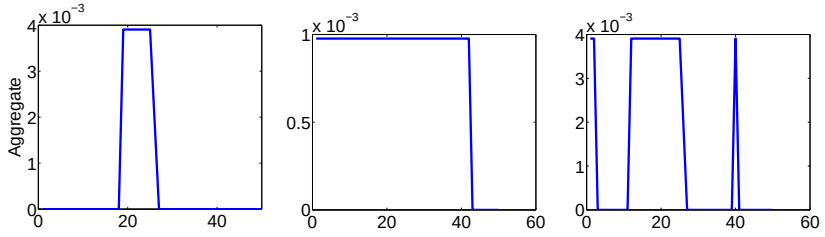
3. *Country affiliation*, which associates a given prefix with the country in which the prefix's owner resides.

4. *AS distance (Hop)*, which gives the shortest hop count between a pair of ASes. Specifically, for prefixes within the same AS, their AS distance is considered to be 0. If they belong to ASes that can directly communicate (neighbors based on Internet routing and peering relationship), their AS distance is considered to be 1. Using the same adjacency information, we calculate the shortest path between any pair of prefixes; its length is then taken to be their AS distance.

5.2.3 Data aggregation

The RBL dataset is at the individual IP address level, while our analysis is done at an aggregate prefix level. We aggregated IP addresses to the BGP routed prefixes that are collected by all vantage points of Route Views [77] and RIPE [65] projects. The prefix level aggregation can be done over a single blacklist or over some combined form of multiple blacklists (e.g., those belonging to the same type of malicious activities). In this study we will use two versions of this combination: (1) Combine all 11 lists in a union fashion (i.e., an IP is listed on the union list on a given day as long as it shows up on at least one of the individual blacklists on that day). This list will be referred to as the *complete union* list. (2) Combine all lists within the same malicious activity type, which leads to three separate union lists, referred to as the Spam, Phishing, and Scanning lists, respectively.

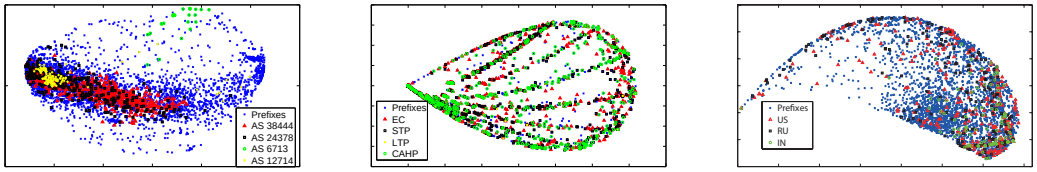
We then aggregate at the prefix level over a given union list to obtain a discrete-time *aggregate signal* for each prefix i; these are denoted by $r_i^U(t)$, $r_i^{sp}(t)$, $r_i^{ph}(t)$, $r_i^{sc}(t)$, $t = 0,1,2,\cdots$, for signals obtained from the complete union, spam, phishing and scanning lists, respectively; example signals are shown in Figure 5.1. There are two types of aggregate one can define: the normalized version and the un-normalized version. For the normalized version, $r_i^*(t)$ is given by the fraction of the total number of IPs on the $*$-list on day t that belong to prefix i over the total number of addresses within prefix i; here we use $*$ to denote any of the union lists. In the un-normalized version $r_i^*(t)$ is simply defined as the total number of IPs on the $*$-list on day t that belong to prefix i. These two are equivalent when prefixes are of the same size, but the normalized version prevents a larger prefix from overwhelming a small one during comparison when we examine prefixes of difference sizes. Throughout this study we will use the normalized version. In all, our RBL dataset represents over 400,000 unique prefixes, and therefore 400,000 time series data.



(a) Prefix 1 (b) Prefix 2 (c) Prefix 3

Figure 5.1: Examples of the aggregate signal

For simplicity in presentation, we will treat the RBLs as error-free in introducing our methodology; this is obviously not true in reality and furthermore there is often no reliable information on the error rates of these lists. To remedy this in Section 5.4.3 we present a sensitivity analysis that provides performance bounds on our method as functions of potential errors in the dataset.



(a) Spam : AS (b) Phishing : Type (c) Scan : Country

Figure 5.2: Similarity graphs for each of the three malicious types. Top 5,000 prefixes of Years 2013-2014.

5.3 Measuring similarity in maliciousness

In this section we quantify similarity between two aggregate signals $r_i^*(t)$ and $r_j^*(t)$ for two prefixes i, j. We also refer to this similarity measure as *behavioral similarity*.

Let $\mathbf{r}_i^*$ and $\mathbf{r}_j^*$ be the vector forms of $r_i^*(t)$ and $r_j^*(t)$, $t = 1, \cdots, \tau$, respectively, for some horizon τ. There are various ways to measure the *temporal similarity* between these two vectors, e.g., by using the Gaussian kernel which measures the distance between two vectors, see for e.g. [6]. Our subsequent methodology does not rely on the specific choice of this similarity measure, although it may impact the numerical conclusions.

5.3.1 Anatomy of similarity graphs and topological interpretation

For concreteness in this study we first visualize the interconnectedness in networks' malicious activities. For demonstration we adopt the following, highly intuitive correlation measure to quantify similarity:

$$S_{i,j}^* = \frac{2(\mathbf{r}_i^*)^T \cdot \mathbf{r}_j^*}{|\mathbf{r}_i^*|^2 + |\mathbf{r}_j^*|^2}, \ \forall i \neq j \ , \tag{5.1}$$

where T in the superscript denotes transpose. There are many other ways of defining above similarity measure, for example the cosine similarity:

$$S_{i,j}^{*,c} = \frac{(\mathbf{r}_i^*)^T \cdot \mathbf{r}_j^*}{|\mathbf{r}_i^*| \cdot |\mathbf{r}_j^*|}, \ \forall i \neq j \ . \tag{5.2}$$

Its meaning is straightforward: it captures how similar in shape these two vectors are. Given N prefixes, the collection of N^2 similarity values can be represented in a *similarity matrix* $S^* = [S^*_{i,j}]$. It is also straight-forward to see that $0 \leq S^*_{i,j} \leq 1$; the higher the value the more similar the two prefixes. To convey what this similarity measure can tell us, we note that S^* may be interpreted as weighted adjacency matrices of an underlying similarity graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$: V is the set of N prefixes, $\mathscr{E}$ the set of weighted edges, with weights $S^*_{i,j}$ representing the closeness, i.e., similarity between two connected prefixes $i, j \in \mathscr{V}$. In this part of study, the signal window for calculating similarity is the entire horizon of RBL data, i.e., a 18-month signal spanning January 2013 upto June 2014.

For each prefix i, its aggregate signal $r^*_i(t)$ has a temporal mean. Over the 18-month period, a majority of the prefixes have very low presence on these RBLs (over 80% of prefixes have $< 2\%$ of their IPs listed on average), while a small number ($< 1\%$) has a very high number of malicious IPs. For this as well as computational reasons, throughout our analysis we will limit our attention to the top 5,000 most significant (or malicious by this average measure) prefixes given a particular union list. Note that different union lists result in different sets of top 5,000 most malicious prefixes, e.g., those heaviest in scanning activities may not be the same as those in spamming, and so on. This sub-sampling strategy makes computations feasible and does not impact the presentation of the methodology.

We shall try to do so on a 2D plane, wherein each point represents a prefix, and the pairwise Euclidean distance between every two points is approximately inversely proportional to their edge weights. Thus the closer two points are, the more similar the corresponding prefixes in their aggregate signals. Three such graphs are shown in Figure 5.2, using similarity matrices S^{sp}, S^{ph} and S^{sc} (calculated using Eqn. (5.1)), each generated by a union list for one of the malicious types, respectively, using one-month data from 2013; the main observations remain largely the same from month to month.

In inspecting Figures 5.2a to 5.2c, we note that these similarity matrices indeed capture something quite interesting, especially in comparing the differences between the three ma-

licious types. There is obvious clustering in all cases; the difference in the nature of the clustering is, however, striking. The spam data shows a prominent single cluster, though it appears as a "belt", rather than a "ball" – it shows a type of "continuity" in similarity (i.e., successive neighbors are very close to each other but they collectively form a very large neighborhood). This type of continuity similarly exists in the scanning graph, though it appears that for scanning almost all prefixes belong to this single cluster, whereas in the case of spam there is a significant number of prefixes that lie outside the cluster. This continuity is most strikingly seen in the phishing data, where the points form a clear set of curves/lines. The most direct explanation for such formation is that a sequence of prefixes share similar aggregate signals but with a progressive phase shift (or time delay).

We below offer a brief interpretation of the above observations. It is generally understood that spam activities are organized into a tiered system where workers obtain a spam workload from a higher level proxy agent while at the same time optimizing the bot structure by finding nearby nodes [34]. Therefore, we can expect that spam activities are organized as distinct campaigns. This results in IP addresses of worker bots being listed in various RBLs in a synchronized manner where nearby prefixes (most likely within the same AS) would demonstrate a higher degree of similarity. This behavior stands out in Figure 5.2a where we see a high degree of clustering due to each of the four highlighted ASes. For phishing and malware spread, one of the most interesting phenomena commonly observed is fast-flux [29], whereby a single malicious domain is mapped to a constantly changing IP address. This leads to a single malicious event propagating through different prefixes over time with the result that these prefixes exhibit high similarity in their dynamic behavior. In addition, Figures 5.2b shows that phishing activity is highly dominated by C-AHP ASes. For the scanning graph, we observe that distributed SSH scanning has rapidly gained popularity as a mechanism to side-step simple rate based blocking counter measures [33]. This is characterized by the use of a large number of un-related distributed IP addresses to make only a few scanning attempts each at a time. In general the IP address-

es chosen are unlikely to be from a single or closely related set of prefixes so as to avoid drawing attention to this activity.

5.3.2 Using similarity to enhance measurement

5.3.2.1 Using similarity to enhance temporal prediction (`K-Learning`)

As we have shown in earlier Chapters, knowing such similarity can improve learning performance towards different goals. We now apply the online prediction technique we studied in Chapter IV to our data and show indeed with similarity information we can improve the day-ahead prediction of network maliciousness. We begin by describing how to make temporal prediction on a network's maliciousness using its historical information. Specifically, we will model each prefix's aggregate signal as a discrete-time Markov chain[1]. The transition probability matrix of this Markov chain can be trained using past data $r_i^*(t)$ with a state space of R_i and over a training window of size T as follows:

$$p^*_{x,y} = \frac{n_{x,y}(T)}{\sum_{w \in R_i} n_{x,w}(T)}, \ \forall\, x, y \in R_i \tag{5.3}$$

where $p^*_{x,y}$ is the estimated probability of transition from state x to state y, and $n_{x,y}(T)$ the number of transitions in the signal from x to y observed within this window. This can be shown to be the optimal posterior estimate [21]. Subsequently we can predict the state of the signal at time t using conditional expectation, given its state at time $t-1$:

$$\hat{r}_i^{\text{temp}}(t) = \sum_{w \in R_i} p^*_{r_i(t-1),w} \cdot w \ . \tag{5.4}$$

This will be referred to as the temporal prediction. While the above is shown over one time step (i.e., day ahead in the aggregate signal), multi-step prediction can be easily obtained using multi-step transition probabilities computed using $p^*_{x,y}$, though the prediction

[1]The aggregate signal $r_i^*(t)$ takes on a finite number of values as the number of IP addresses in a prefix is finite.

accuracy generally decays over longer periods.

We then implement (K-Learning) based temporal prediction as detailed in Chapter IV, with taking label $y_i(t)$ as the next day feature $x_i(t+1)$, and similarity information as calculated above. We measure the prediction performance by the following error function,

$$e_i = |\hat{r}_i(t) - r_i^*(t)|^2. \tag{5.5}$$

Figure 5.3 compares the difference between using only temporal prediction (5.4) and the (K-Learning), both for one day-ahead prediction. This is shown for all prefixes as a CDF over the prediction error. We see quite clearly the improvement by using similarity information. For instance, we can increase the number of predictions at $< 5\%$ accuracy by about 20%.

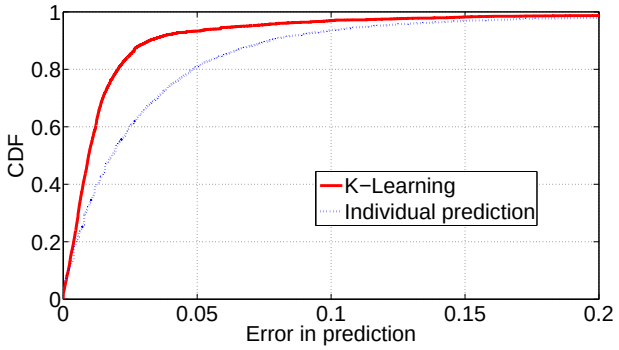


Figure 5.3: Prediction performance comparison.

5.3.2.2 Experiment results with (W-MF)

As we promised at the end of Chapter III, we are going to validate (W-MF) with the measurement data we have, which is high rank. The experiments is done for 2014. We use 2013 data to calculate network similarity. To validate our algorithm, we randomly omit certain entries in our database (to emulate missing measurements) and keep $O(\log N \cdot T)$ number of samples, as commonly done in matrix completion studies. To serve as a baseline, we adopt the SVT algorithm as introduced in [8]. We sub-sample 100 waypoints. The

experiments are done for the union list data. First we show in Figure 5.4, the separated sub-matrices enjoy a rank reduction for a majority of the cases. Simulation results for predicting missing measurement data are shown in Figure 5.5. We see that our similarity aided matrix completion algorithm achieves more than four times performance improvement. Similar results hold when we separate the list to different type of malicious activities.

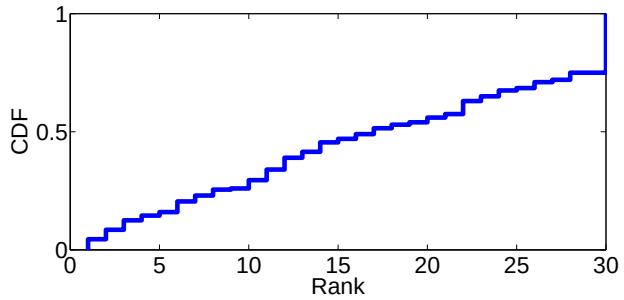


Figure 5.4: CDF of ranks of separated sub-matrices.

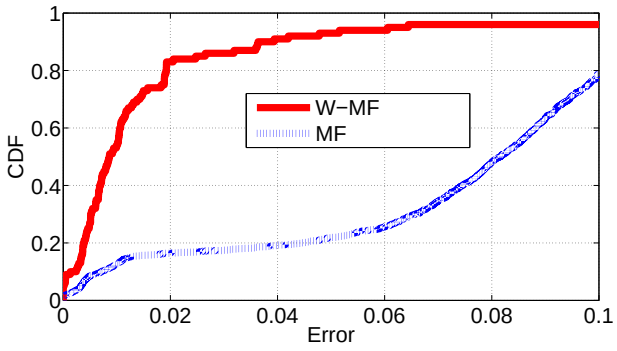


Figure 5.5: CDF of data filling errors. Separating networks. Average Error (AE) = 0.079 for direct matrix completion. AE = 0.0187 for (W-MF). Performance is more than 4 times better.

5.4 From Maliciousness to Topological Similarity

In this section we present an application specific study of similarity, and we will explore whether the similarity shown in the previous section is correlated with the set of spatial and proximity features (*topological similarity*) outlined earlier. Empirically such correlation appears to exist, as indicated in Figure 5.2. For instance, we see in Figure 5.2a that those

belonging to a few largest ASes (among all ASes represented by this set of prefixes) also belong to the same cluster. Below we assess this correlation more quantitatively using statistical inference.

5.4.1 An inference model over graphs

The correlation relationship that we seek to quantify is illustrated in Figure 5.6 (left figure). On the left hand side (LHS) we have a set of latent (or relational) variables or graphs; these are given by similarity matrices derived from known relationships between the networks. In our case, these are derived from information on AS membership, AS type, AS hop count, and country affiliation. These latent relational similarity matrices were derived among N prefixes. All of these are $N \times N$ matrices being symmetric about the diagonal. (1) A_1 denotes the AS-membership similarity matrix. Entry $A_1(i,j) = 1$ if prefixes i and j belong to the same AS, and $A_1(i,j) = 0$ otherwise. (2) A_2 denotes the AS-type similarity matrix. Entry $A_2(i,j) = 1$ if prefixes i and j belong to ASes of the same type (or belong to the same AS), and $A_2(i,j) = 0$ otherwise. (3) A_3 denotes the AS-distance similarity matrix. Entry $A_3(i,j) = \xi^{h(i,j)}$, where $0 < \xi < 1$ is a scaling factor to make this matrix consistent with others, and $h(i,j)$ denotes prefixes i,j's AS distance or hop count defined earlier. (4) A_4 denotes the country-affiliation similarity matrix. Entry $A_4(i,j) = 1$ if prefixes i and j are owned by entities (companies, organizations, etc.) residing in the same country, and $A_4(i,j) = 0$ otherwise.

On the right hand side (RHS) we have a set of observable variables/graphs given by the similarity matrices derived from the aggregate maliciousness signals. We shall continue to use the similarity matrix S given by simple correlation defined earlier, also known as dice similarity; we will also consider the cosine similarity (denote by S^c in Figure 5.6).

The edges shown in the figure are directed; however, this does not mean, nor do we claim any causal relationship between the two sides. The directionality merely suggests that one side is *latent*, though known, factors that may act in known or unknown ways,

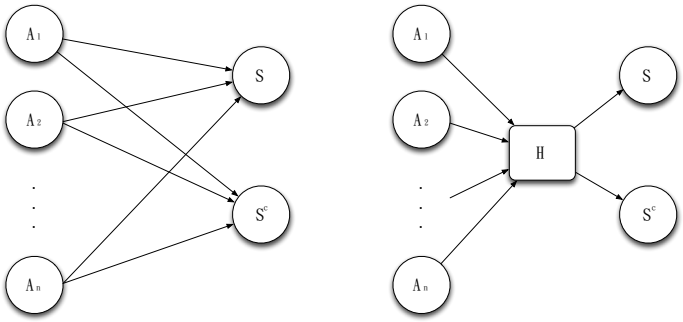


Figure 5.6: On the left, we attempt to establish the explanatory power of factors on the LHS of observations shown on the RHS. The degrees of significance are given by inferred edge weights. On the right, a multi-layer graphical inference model. A connector H (hidden matrix) is introduced to simply the analysis.

while the other side is *active* (i.e., the manifestation seen in actual behavior). Our goal is to infer, given the two sets of similarity matrices, the edge weights on all directed edges. There are different interpretations for these edge weights. First, the weights convey the relative strengths of the correlation between latent variables and observed variables, e.g., which relationship factor best explains, or resembles the most, the similarity seen in the actual data. However, if causal relationship between the two sides could be established via separate means, then these edge weights also indicate to what degree a latent variable contributes to/cause the observation. Our goal in this study is to quantify the degree of correlation, and not the determination of a causal relationship.

5.4.2 A multi-layer graphical inference model

We consider a *multi-layer inference model*, shown on the right side of Figure 5.6. Here we have added a "hidden" similarity matrix H as an intermediate step between the two sides. The idea is to separately establish correlation between the latent matrices and this hidden H, and between H and the observed matrices. While this is certainly a simplification, the hidden matrix H is not necessarily a mathematical construct solely for the purpose of simplicity; it can have real physical interpretation as well. Take for instance the case of spam. The observed spam activity of a given network is ultimately determined by how

many hosts in that network were taken over as proxies by a spam campaign. The number of such hosts may be viewed as determined by the likelihood a network is targeted by the spam campaign as well as its vulnerability in falling victim when targeted. The similarities in these probabilities may be regarded as the hidden H matrix. In other words, the similarity observed on the RHS can be ultimately attributed to the similarity in this probability similarity matrix H. This reasoning also applies to the other types of malicious activities. Therefore, we may correlate the set of spatial features with the observed matrices through H.

The introduction of H allows us to solve the problem using the following decomposition-matching procedure without having to assume priors commonly used in the literature (see e.g., [60]):

1. The LHS problem: the inference of H using the latent variable matrices $\{A_i\} \to H$.

2. The RHS problem: the inference of H using the observed matrices $S, S^c \to H$. For notational convenience S, S^c will also be collectively written as $\{S_j\}$ in the discussion below.

In both cases H is unknown. To get around this, we will estimate H as a linear combination of the A_i's and of the S_j's, respectively, and then solve both problems simultaneously. Specifically, we shall estimate H from A_i's as $\sum_i \alpha_i A_i, \sum_i \alpha_i = 1$, and from S_j's as $\sum_j \beta_j S_j, \sum_j \beta_j = 1$, respectively, and then solve the two sets of edge weights by minimizing the difference between the two inferred versions. Physically this means to estimate H as its projection onto the spaces spanned by $\{A_i\}$ and $\{S_j\}$, respectively. Although this is clearly an approximation, under certain assumptions the linear model can indeed be shown to be the optimal fit; more details are given in the Appendix.

The above inference problem is formally given by the following optimization problem

(P_BI):

$$\begin{aligned} \min_{\alpha,\beta} \quad & \mathscr{L}(H_\alpha, H_\beta) \\ \text{s.t.} \quad & H_\alpha = \textstyle\sum_i \alpha_i A_i,\ H_\beta = \sum_j \beta_j S_j\ , \\ & \textstyle\sum_i \alpha_i = 1, \sum_j \beta_j = 1\ . \end{aligned}$$

where $\mathscr{L}$ is a loss function measuring the discrepancy between H_α and H_β, the inference on both sides. We will take $\mathscr{L}$ to be the normalized Frobenius norm, $\mathscr{L}(H_\alpha, H_\beta) = ||H_\alpha - H_\beta||_F$; for an arbitrary matrix $M = [M_{ij}]$ this norm is defined as $||M||_F := \sqrt{\sum_{i,j} M_{i,j}^2}/N$.

Note that the constraints are all linear in α, β, and the Frobenius norm is convex as an objective function. Furthermore, $H_\alpha - H_\beta$ is a linear operation over α, β. We thus conclude that **(P_BI)** is convex and can be solved efficiently.

5.4.3 Sensitivity analysis

In this section we derive a number of performance bounds concerning the sensitivity of our inference approach to potential error/noise in the raw data used, as well as to a key parameter choice in the method (the observation window).

Sensitivity to errors in the RBL data

Our RBL dataset is obtained from third parties; as such we do not have reliable means of assessing their quality (i.e., false positive and miss detection rates) and have so far assumed they are error-free. It is however extremely important to understand how potential errors in the data may affect our method. In what follows we characterize the performance of our inference algorithm in the presence of such errors.

Assume that both sides of the inference model, A_is and S_js, contain errors (the latter due to errors in the RBL data). Specifically, we consider two types of errors, one referred to as the *bounded additive error* and the other *flip error*. Under the first type of error, each entry of a similarity matrix is subject to an additive and bounded perturbation, which need not be independent from one entry to another. This type of error is mostly applicable to S^* and $S^{q,*}$ on the RHS of the inference model since their entries are continuous in value. For

A_is with binary entries, the second type of error is more applicable, where values flip from 0 to 1 or 1 to 0 with a certain probability.

Consider first the case where bounded additive errors are present in both A_is and S_js; these noisy versions are denoted as $\tilde{A}_i$ and $\tilde{S}_j$. Denote the optimal solution to the problem **(P_BI)** solved using the error-free A_i and S_j as $(\{\alpha_i^o\},\{\beta_j^o\})$ and the corresponding objective function value $\mathscr{L}$; denote their noisy counterparts (i.e., solved using $\tilde{A}_i$ and $\tilde{S}_j$) as $(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})$ and $\tilde{\mathscr{L}}$, respectively. We have the following results.

Theorem V.1. *If $\forall i,j$ and for any pair (k,l) we have*

$$|\tilde{A}_i(k,l)-A_i(k,l)|\leq\delta,\ |\tilde{S}_j(k,l)-S_j(k,l)|\leq\delta\ ,$$

i.e., the per-element errors are uniformly bounded by some δ, then the noisy solution $(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_i^o\})$ satisfies

$$||(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_i^o\})-(\{\alpha_i^o\},\{\beta_j^o\})||_2\leq\sqrt{\frac{2\delta}{C}}\ , \tag{5.6}$$

with positive constant $C>0$.

In other words, the error in the inference result induced by the error in the data can be precisely bounded. The proof of this result can be found in the Appendix. Key ingredients of the proof are (1) the triangle inequality for the Frobenius norm, (2) the sub-gradient inequality for the Frobenius norm due to its convexity.

Next consider the flip errors, and assume

$$\tilde{A}_i(k,l)=\begin{cases} A_i(k,l), & \text{w.p. } 1-\varepsilon \\ 1-A_i(k,l), & \text{w.p. } \varepsilon\ , \end{cases} \tag{5.7}$$

and that the flips are independent of each other. We have the following results that relate the flip probability to the Frobenius norm.

	α :[AS, Type, Ctry., Hop]	β :[Raw, Quan.]	$\|\|\cdot\|\|_F$
Truth	[0.40, 0.10, -, 0.50]	[0.8,0.2]	0.1986
Solution	[0.45, 0.05, -, 0.50]	[0.7,0.3]	0.2118
Additive ($\delta = 0.05$)	[0.45, 0.15, -, 0.40]	[0.7,0.3]	0.1697
Flip ($\varepsilon = 0.05$)	[0.45, 0.05, -, 0.50]	[0.7,0.3]	0.2167

Table 5.2: Simulation studies with error-free solutions, bounded additive and flip errors.

Lemma V.2. *With probability at least* $1 - 2e^{-2\delta^2 \cdot \frac{N^2-N}{2}}$

$$||\tilde{A}_i - A_i||_F \leq \sqrt{2 \cdot (\varepsilon + \delta)}\ , \forall i. \tag{5.8}$$

With the above results, we have effectively converted the flip errors to bounded additive errors, which then allows the same analysis used in Theorem V.1 to go through, resulting in the following theorem; the proof is omitted for brevity.

Theorem V.3. *Denote the number of feature matrices* A_i *as* n_A *and assume only flip errors are present in* $\{A_i\}$. *Then with probability at least* $1 - 2n_A \cdot e^{-2\delta^2 \cdot \frac{N^2-N}{2}}$, *the noisy solution* $(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\})$ *satisfies*

$$||(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\}) - (\{\alpha_i^o\}, \{\beta_j^o\})||_2 \leq \sqrt{\frac{2\sqrt{2 \cdot (\varepsilon + \delta)}}{C}}\ . \tag{5.9}$$

Note that the two types of error can be easily combined to get a bound on when both types exist at the same time, as shown below.

Theorem V.4. *When both* bounded additive error *and* flip error *exist, with probability at least* $1 - 2n_A \cdot e^{-2\delta^2 \cdot \frac{N^2-N}{2}}$, *the noisy solution* $(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\})$ *satisfies*

$$||(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\}) - (\{\alpha_i^o\}, \{\beta_j^o\})||_2 \leq \sqrt{\frac{2 \cdot \max\{\sqrt{2 \cdot (\varepsilon + \delta)}, \delta\}}{C}}\ .$$

In Table 5.2 we repeat the same simulation performed earlier but with bounded additive and flip errors, respectively.

Effect of finite observation window

We next characterize the effect of using a finite-length time series to construct S^* and $S^{q,*}$ (in all our experiments the window size used is a month, which is a natural separation.). We will do so by assuming the raw time series data $r_i^*(t)$ can be modeled as an ergodic Markov chain with transition probability matrix P_i on the state space $\mathscr{R}_i$, for otherwise the variation in the time series itself over time can be arbitrary. Denote by R_i, a random variable, the state of this Markov chain in steady state. Denote the number of observations for each Markov Chain as τ. Our results are formally stated in the following theorem.

Theorem V.5. *With high probability (being at least* $1 - O(e^{-\tau\varepsilon^2})$*), the noisy solution* $(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\})$ *satisfies*

$$||(\{\tilde{\alpha}_i^o\}, \{\tilde{\beta}_i^o\}) - (\{\alpha_i^o\}, \{\beta_j^o\})||_2 \leq \sqrt{\frac{2\max\{f_1(\varepsilon), f_2(\varepsilon)\}}{C}},$$

for small enough $\varepsilon > 0$ *and* $f_1(\varepsilon), f_2(\varepsilon)$ *are both bounded, continuous, and monotonically increasing function in* ε *for* ε *small enough, and* $f_1(0) = f_2(0) = 0$*.*

5.4.4 Inference on the RBL dataset

We now apply the multi-layer graphical inference model to the RBL dataset. We first do so for each month of the measurement period using the aggregate signal obtained for that month. A subset of the results (April-Nov 2013) are shown in Table 5.3; the rest is omitted for brevity while noting that those results are very consistent with what's shown here.

There are a number of interesting observations from Table 5.3. First of all, AS membership, AS type and country affiliation similarity are very minor compared to the strength of the distance similarity as an indicator. It appears that the first three spatial features are subsumed in the distance feature, or in other words, the distance information successfully encodes the other features such that it becomes a near-sufficient descriptor for similarity

Month	α : [AS, Type, Hop, Ctry.]	β :[Dice, Cosine]	$\|\|\cdot\|\|_F$
Apr.	[0.0, 0.1, 0.8, 0.1]	[0.8, 0.2]	0.3612
May.	[0.0, 0.1, 0.8, 0.1]	[0.9, 0.1]	0.3604
Jun.	[0.0, 0.1, 0.8, 0.1]	[0.9, 0.1]	0.3443
Jul.	[0.0, 0.1, 0.8, 0.1]	[0.8, 0.2]	0.3270
Aug.	[0.0, 0.1, 0.8, 0.1]	[0.7, 0.3]	0.3344
Sep.	[0.0, 0.1, 0.8, 0.1]	[0.7, 0.3]	0.3485
Oct..	[0.1, 0.1, 0.8, 0.1]	[1.0, 0.0]	0.4019
Nov.	[0.0, 0.1, 0.8, 0.1]	[0.6, 0.4]	0.3702

Table 5.3: Inference result. ξ is set to be 0.85.

Maliciousness	α :[AS, Type, Hop, Ctry.]	β :[Dice, Cosine]	$\|\|\cdot\|\|_F$
Spam	[0.3, 0.1, 0.5, 0.1]	[0.5, 0.5]	0.2506
Phishing	[0.1, 0.1, 0.7, 0.1]	[0.3, 0.7]	0.2479
Scan	[0.5, 0.1, 0.3, 0.1]	[0.9, 0.1]	0.2080

Table 5.4: Inference result along each malicious type separately.

in malicious activities. There are a number of reasons for why this may be true. Topological distance naturally contains AS membership information: those prefixes in the same AS are considered most similar both in AS membership as well as in AS-distance. For the same reason these prefixes with the closest AS-distance will also bear the same AS type. Topologically close prefixes are also more likely to reside in the same country, so any geo-political and macro-economic information embedded in the country affiliation is also contained in the distance information. It is, therefore, not entirely unexpected that AS-distance should be a relevant factor in assessing similarity in maliciousness. It is, however, quite surprising how significant this factor is, so much so that one could arguably ignore the other factors in describing the observed similarity in maliciousness.

We next repeat the same inference analysis along different malicious activity types, i.e., by using S^{sp}, S^{ph}, S^{sc}, respectively, as the similarity matrices on the RHS of the inference model. This is shown for the month of October 2013 (middle point in our database) in Table 5.4. As before, the AS-distance remains a dominant indicator that dwarfs the other spatial features for spam and phishing; however, the scan similarity depends more on AS membership. This is likely due to the fact that a scanning campaign typically involves acquiring/compromising a large number of hosts to use for scanning and these hosts

are often from the same networks; this results in networks within the same AS exhibiting similar behavior. By contrast, a phishing campaign involves a much smaller number of compromised hosts, but they move from network to network as the campaign tries to evade takedown; this likely results in neighboring networks exhibiting similar behavior. Also of note is when we view each malicious types separately, cosine similarity plays a more significant role in phishing and spam signals, which suggests more synchrony (or coordination botsniffer) among activities of these two types. This is because the cosine similarity captures similarity in shape rather than magnitude. For instance consider two vectors $\mathbf{r}_i^*$ and $\mathbf{r}_j^*$ such that $\mathbf{r}_j^* = n \cdot \mathbf{r}_i^*$ (i.e., j is n times that of i component-wise). We would then have $S_{i,j}^* = 2(\mathbf{r}_i^*)^T \cdot n\mathbf{r}_i^* / (|\mathbf{r}_i^*|^2 + |n \cdot \mathbf{r}_i^*|^2) = (2n)/(n^2+1)$, which approaches 0 as n becomes large. By contrast, the similarity between $\mathbf{r}_i^*$ and $\mathbf{r}_j^*$ would be preserved by cosine similarity.

One other important observation is that the norm error is much lower in this case, suggesting a highly accurate linear model in explaining the observed data. This translates into very high accuracy in using spatial features to estimate the similarity in malicious behaviors along each type.

5.4.5 Estimating maliciousness in the absence of measurement data

Consider the problem where we possess no information about a prefix's maliciousness, but only know its spatial relationship with some other prefixes whose aggregated signals are known. The missing data may be caused by a variety of measurement errors. Below we show how we can estimate the missing information using only topological similarity and malicious activity data of a number of known entities.

Recall that as shown earlier, the similarity in maliciousness is most strongly correlated with topological distance. Figure 5.7 shows how similarity measures are distributed for prefixes within n hops of each other. Accordingly, we define a prefix i's neighborhood N_i as the set of prefixes within two hop, i.e., $\forall j$ such that $h(i,j) \leq 2$, and these prefixes'

maliciousness signals are well measured. The choice of 2-hop neighbors is based on observations from Figure 5.7, which shows a clear drop in similarity beyond hop 2. We next define the spatial similarity matrix A as follows, using the inference results obtained from Table 5.3:

$$A = 0 \cdot A_1 + 0.1 \cdot A_2 + 0.8 \cdot A_3 + 0.1 \cdot A_4 \ . \tag{5.10}$$

We then calculate for each $j \in N_i$ a weight ω_j using $A_{i,j}$:

$$\omega_j = \frac{e^{A_{i,j}/\sigma^2}}{\sum_{k \in N_i} e^{A_{i,k}/\sigma^2}} \ .$$

The final estimate for an unknown prefix i's signal is given by $\hat{r}_i(t) = \sum_{j \in N_i} \omega_j \cdot r_j(t)$.

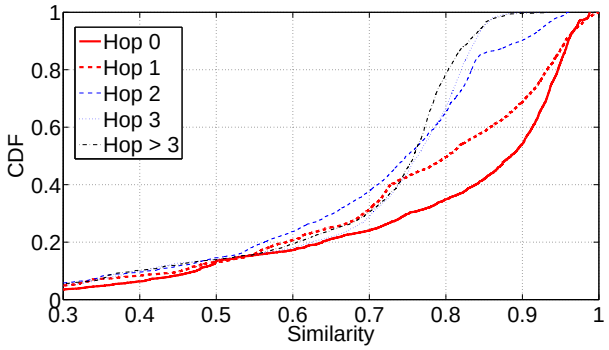


Figure 5.7: Similarity distribution w.r.t. AS hops

Note that we obtained the edges weights using data from 2013. We will use the above model to estimate malicious activities in 2014 and compare it to the ground truth $r_i(t)$. This is done on a monthly basis. For each month in 2014, we randomly select a set of prefixes and remove their RBL data, and produce an estimate for them using data from their "neighbors". Generally we observe that about half of the prefixes have less than 10% estimation error, defined as $\frac{\sum_{t=1}^{d} |r_i(t) - \hat{r}_i(t)|}{d}$, where d is the number of days for the corresponding month, while about 80% of all prefixes have less than 20% estimation error. We present the results in Figure 5.8. Also shown in Figure 5.8 are results obtained by performing the same type

of estimation for each malicious types by considering different types of aggregate signals. For each type of maliciousness the similarity matrix A is calculated using the inference result corresponding to that type as given in Table 5.4. Since the inference are more accurate when we consider each malicious type separately, improvement in estimation quality is to be expected. It is nonetheless remarkable how accurate the estimation is based purely on spatial features as shown in Figure 5.8, especially in the case of scan.

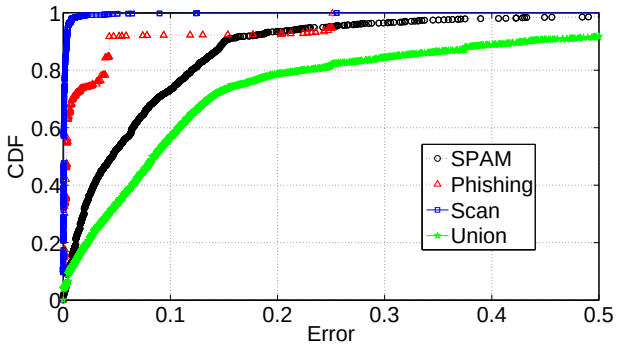


Figure 5.8: Estimation results for 2014 using topological similarity for different categories of maliciousness, as well as a union of them.

5.5 Concluding remarks

We presented a study on analyzing the relationships between dynamic malicious activities of different network entities, and their correlation with a number of spatial relationships, including their AS membership, AS type association, topological distances, as well as country affiliations and we demonstrate the effectiveness of the results by applying them to enhance prediction of networks' future maliciousness. By adopting a simple similarity measure we were able to capture hidden features in network malicious activities previously unseen when only time-averaged quantities are used. Usefulness of such information is demonstrated concretely via showing how to combine it with existing method to enhance prediction results and measurement efficiency. We also concluded, by using statistical inference, that within the set of spatial features considered, the topological distance between

two networks is by far the most significant indicator of their similarity in maliciousness. We then showed how these results can be utilized by providing prediction where it would otherwise have been impossible.

CHAPTER VI

Conclusion

In this dissertation we set out to rigorously study how diversity and similarity hidden in disparate data sources (e.g., crowd-sourcing system) affect performances of a machine learning system w.r.t. different objectives. Our general goal is to demonstrate methods of quantifying and learning such information, and show in what ways they can help process multi-source data within several broadly applicable problem settings. And we show with appropriately designed algorithm, we can harness the power of such information to different degrees. Besides a number of analytical results, we also delve into a set of real world Internet measurement data to firstly reveal similarities among real Internet entities and then present several applications utilizing such understanding.

The dissertation, as a whole, can be viewed as initial steps towards rigorously framing the discussion of uncovering connectedness in a big data system. Big data system, instead of simply implying a large volume of data, can potentially provide more opportunities due to its complex data structures (that is with data being connected in certain way). This dissertation provides several examples on how this hidden interleave may benefit various learning objectives dealing with data systems with potentially multiple and diverse sources, as well as challenges for uncovering them.

6.1 Future work

One interesting direction of extending the current dissertation is to utilize the results generated from our learning methods to design better crowd-sourcing market and platform. Despite its popularity and usability, most crowd-sourcing markets suffer from data quality issues, which most of times due to irresponsible behaviors of workers. Chapter II of this dissertation provides one solution from task assigner's point of view. We however hypothesis it would be more efficient if the crowd-sourcing platforms can implement smarter mechanisms, rather than leaving this to each individual task assigner. For instance with an online learning process, crowd workers' inserted efforts can be consistently monitored and learned (through the contributed data); therefore based on such results, better bonus or punishment schemes can be put into use for market design.

Another relevant study is to look into privacy preserving learning methods for crowd-sourcing related topics. Throughout the current dissertation we have assumed the cleanness of crowd-sourced data, that is crowd workers contribute their data voluntarily without manipulation. However this assumption may not hold completely, especially when the data itself is sensitive (e.g., medical record data etc.) Then a very interesting question is when users are malicious or privacy concerned, how to discern such maliciousness and incentivize data report with truthfulness, or in another words, how to develop efficient learning algorithm with potentially falsified data input.

On the application side, we would like to seek a better understanding of the similarity in maliciousness and its implication for cyber security events. Similarity information is believed to be useful for semi-supervised learning when large amount of labels are missing, which is the case for cyber security research as security events are largely under-reported. We plan to leverage similarity information for uncovering such under-reported incidents and combine it with semi-supervised learning algorithms to better predict malicious activities.

APPENDICES

APPENDIX A

Proofs for Chapter II

A.1 Proof of Theorem II.1

We prove by contradiction. Suppose m is even and we order all selected users by their labeling capability in descending order $p_1 \geq ... \geq p_m$. We now prove

$$\pi(p_1, ..., p_{m-1}) \geq \pi(p_1, ..., p_m) \tag{A.1}$$

Consider the following. By adding p_m, the gain for $\pi(p_1, ..., p_{m-1})$ is $\frac{p_m \cdot P(T_1)}{2}$, where

$$T_1 = \{\#\text{ of correct labels} = \#\text{ of wrong labels} - 1\}.$$

That is, only when the number of correct labels is exactly the same as the number of wrong labels less one does adding a m-th correct label change the outcome of the majority vote (in this case there is a tie so the label changes with probability $1/2$); when the former number is smaller or bigger, adding one more vote does not change the results. On the other hand, the loss is $\frac{(1-p_m) \cdot P(T_2)}{2}$, where

$$T_2 = \{\#\text{ of correct labels} = \#\text{ of wrong labels} + 1\}.$$

We now compare $p_m \cdot P(T_1)$ and $(1-p_m)\cdot P(T_2)$. Within the set T_2, each event is of the form where some labeler i gives the correct label while the rest are half correct and half wrong. Denote this event by ω_i and by ω_{-i} the event that the rest of the labels (given by those other than i) are half right and half wrong. Note for each ω_i there is a corresponding event $\hat{\omega}_i \in T_1$ where i gives the wrong labels while the rest are half correct and half wrong. Since $p_i \geq p_m$ we have

$$(1-p_m)\cdot\ p_i \cdot P(\omega_{-i}) \geq p_m \cdot (1-p_i)\cdot P(\omega_{-i})\ . \tag{A.2}$$

At the same time, $P(\omega_i) = p_i \cdot P(\omega_{-i}), P(\hat{\omega}_i) = (1-p_i)\cdot P(\omega_{-i})$, i.e., $(1-p_m)\cdot P(\omega_i) \geq p_m \cdot P(\hat{\omega}_i)$. This is true for all ω_i. Therefore we have

$$\begin{aligned}(1-p_m)\cdot P(T_2) &= (1-p_m)\cdot P(\cup_i\ \omega_i)\\ &= \sum_{\omega_i}(1-p_m)\cdot P(\omega_i) \geq \sum_{\hat{\omega}_i} p_m \cdot P(\hat{\omega}_i) = p_m \cdot P(T_1)\ .\end{aligned} \tag{A.3}$$

Therefore we have proved $\pi(p_1,...,p_{m-1}) \geq \pi(p_1,...,p_m)$. Moreover

$$U(\{1,2,...,m-1\}) - U(\{1,2,...,m\}) = \pi(p_1,...,p_{m-1}) - \pi(p_1,...,p_m) > 0. \tag{A.4}$$

Therefore a selection of an even number of labelers can always be improved by removing the least accurate labeler, resulting in an odd number of labelers in the selection.

A.2 Proof of Theorem II.2

Consider a m-set S. Suppose there is a $i \notin S$ and a $j \in S$ such that $p_i > p_j$. Then the probability of making a correct annotation is given by

$$P_S(\#\text{ of correct labels} > \#\text{ of wrong labels}) = p_j \cdot P(T_1) + (1-p_j)\cdot P(T_2)\ ,$$

where

$$T_1 = \{\#\text{ correct label} > \#\text{ wrong label} - 1 \text{ in } S\backslash j\}\ , \tag{A.5}$$

$$T_2 = \{\#\text{ correct label} > \#\text{ wrong label} + 1 \text{ in } S\backslash j\}\ . \tag{A.6}$$

Now replace j with i and denote $\hat{S} = S\backslash j \cup \{i\}$ we have

$$P_{\hat{S}}(\#\text{ of correct labels} > \#\text{ of wrong labels}) = p_i \cdot P(T_1) + (1 - p_i) \cdot P(T_2)\ .$$

It follows that

$$P_{\hat{S}} - P_S = (p_i - p_j) \cdot (P(T_1) - P(T_2))\ . \tag{A.7}$$

If an event $\omega \in T_2$ we must also have $\omega \in T_1$, thus we have $T_2 \subset T_1$; therefore $P(T_1) - P(T_2) > 0$, and we conclude that $P_{\hat{S}} - P_S > 0$, completing the proof.

A.3 Proof of Lemma II.4

Denote by $n(T)$ the number of times an exploration phase has been activated up to time T. Since for labeler i there is at most $D_1(T) \cdot D_2(T)$ number of exploration phases, we have

$$\begin{aligned} n(T) &= \sum_{i=1}^{T} 1\{\text{at least one task in exploration phase at } t\} \\ &\leq \sum_{k=1}^{D_1(t)} \sum_{i=1}^{T} 1\{\text{ task } k \text{ in reassignment phase at } t\} \leq D_1(T) \cdot D_2(T)\ , \end{aligned}$$

where the first inequality comes from union bound. Then

$$E[R_e(T)] \leq U(S^*) \cdot n(T) = U(S^*) \cdot (D_1(T) \cdot D_2(T))\ .$$

A.4 Proof of Lemma II.6

Firstly notice via union bound we have the following bound at any time t:

$$E[\mathscr{E}_1(t)] \leq \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \ . \tag{A.8}$$

Now consider each term in the above summation $P(\tilde{U}(S^m) \geq \tilde{U}(S^*))$. We will use the following fact to bound it.

Lemma A.1. *The probability of using a sub-optimal selection S^m is bounded as follows,*

$$\begin{aligned} P(\tilde{U}(S^m) \geq & \tilde{U}(S^*)) \leq P(\tilde{U}(S^m) > U(S^m) + \varepsilon) \\ & + P(\tilde{U}(S^*) < U(S^*) - \varepsilon) \ , \end{aligned} \tag{A.9}$$

and for $S \in \{S^m, S^*\}$ we have

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) \leq n(S) \cdot \sum_{i \in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \ . \tag{A.10}$$

We shall now use the above lemma; its own proof is given later in this appendix.

Consider each term $P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|})$ in the lemma

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \\ & = \underbrace{P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|} \mid \frac{\sum_{k:k \in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha \cdot \varepsilon}{t^z})}_{\textbf{Term 1}} \cdot P(\frac{\sum_{k:k \in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha \cdot \varepsilon}{t^z}) \\ & + P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|} \mid \frac{\sum_{k:k \in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z}) \cdot \underbrace{P(\frac{\sum_{k:k \in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z})}_{\textbf{Term 2}} \ , \end{aligned} \tag{A.11}$$

where $0 < z < 1$ is a constant. This is different from the classical learning problem in the sense we need to deal with extra errors associated with imperfect feedbacks. The first term takes care of the event when the sum of error is lower than certain threshold while the second term captures the other case.

For **Term 1** the conditional probability is bounded as follows:

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|} \mid \frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha\cdot\varepsilon}{t^z}) \\ & \leq P(|\tilde{p}_i - p_i| > (\frac{1}{n(S)\cdot|S|} - \frac{\alpha}{t^z})\cdot\varepsilon) \\ & \leq 2\cdot e^{-2((\frac{1}{n(S)\cdot|S|}-\frac{\alpha}{t^z})\cdot\varepsilon)^2\cdot D_1(t)} \leq \frac{2}{t^2}\ , \end{aligned} \tag{A.12}$$

since $D_1(t) = \frac{1}{(\frac{1}{n(S)\cdot|S|}-\alpha)^2\cdot\varepsilon^2}\cdot\log t$. Consider **Term 2**,

$$P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha\cdot\varepsilon}{t^z}) \leq \frac{\frac{E[\sum_{k:k\in E(t)} 1\{y_k^*=0\}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}} = \frac{\frac{\sum_{k:k\in E(t)} E[1\{y_k^*=0\}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}}\ , \tag{A.13}$$

by the Markov inequality. Note more strict bound could be obtained via other bounding techniques. Consider each term in the summation

$$E[1\{y_k^* = 0\}] = P(y_k^* = 0) = P(\sum_{n=1}^{\hat{N}_k(t)} 1\{y_k(n)\} > 0.5\cdot\hat{N}_k(t)) \leq e^{-2(a_{\min}-0.5)^2\cdot\hat{N}_k(t)} \leq \frac{1}{t^2}\ ,$$

where $\hat{N}_k(t)$ is the number of feedbacks received for task k upto time t; the inequality is due to the fact that $\hat{N}_k(t) \geq D_2(t) \geq 1/(a_{\min}-0.5)^2\log t$. This means that for each labeler, it has performed on at least $D_1(T)$ tasks, and each task must have at least $D_2(T)$ testing results available.

Consequently we have

$$P(\frac{\sum_{k\in E(t)} I\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha\cdot\varepsilon}{t^z}) \leq \frac{1/t^2}{\alpha\cdot\varepsilon/t^z} = \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}\ .$$

The other two terms in the summation are bounded by 1 since they are probability measures. Summing up, we have

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) \leq n(S) \cdot |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \ . \tag{A.14}$$

Summing over S^m, m odd completes the proof.

A.5 Proof of Lemma II.7

We have the following fact:

$$E[\mathscr{E}_2(t)] \leq P(\cup_{i \in \mathscr{M}} |\tilde{p}_i - p_i| > \varepsilon) \leq \sum_{i \in \mathscr{M}} P(|\tilde{p}_i - p_i| > \varepsilon) \ .$$

This is because if $|\tilde{p}_i - p_i| \leq \varepsilon, \forall i$ then we must have for $p_i > p_j$,

$$\tilde{p}_i - \tilde{p}_j \geq p_i - \varepsilon - p_j - \varepsilon > 0,$$

which means there is no error in ordering. Similarly as above we have

$$P(|\tilde{p}_i - p_i| > \varepsilon) \leq \frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}} \ . \tag{A.15}$$

Therefore,

$$E[\mathscr{E}_2(t)] \leq M(\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \ . \tag{A.16}$$

A.6 Proof of Lemma A.1

We first bound the inequality in Eqn.(A.9). To see why this inequality is true, consider the following fact

$$\{\omega : \tilde{U}(S^m) \geq \tilde{U}(S^*)\} \subseteq \{\omega : \tilde{U}(S^m) > U(S^m) + \varepsilon\} \cup \{\omega : \tilde{U}(S^*) < U(S^*) - \varepsilon\} \ , \quad \text{(A.17)}$$

since if $\tilde{U}(S^m) \leq U(S^m) + \varepsilon,\ \tilde{U}(S^*) \geq U(S^*) - \varepsilon$, we then have

$$\tilde{U}(S^m) - \tilde{U}(S^*) \leq U(S^m) + \varepsilon - U(S^*) + \varepsilon \leq -\Delta_{\min} + 2\varepsilon < -\Delta_{\min} + \Delta_{\min} = 0 \ ,$$

which contradicts the fact that $\tilde{U}(S^m) \geq \tilde{U}(S^*)$. Thus

$$P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \leq P(\tilde{U}(S^m) > U(S^m) + \varepsilon) \ ,$$

by the union bound. The bounding effort then reduces to bounding each of above probabilities. Note that for any set S, plug in $U(S)$ we have

$$|\tilde{U}(S) - U(S)| = |\sum_{S' : S' \subseteq S,\ |S'| \geq \lceil \frac{|S|}{2} \rceil} (\prod_{i \in S'} \tilde{p}_i \prod_{S \setminus S'} (1 - \tilde{p}_j)) - \prod_{i \in S'} p_i \prod_{S \setminus S'} (1 - p_j))| \ . \quad \text{(A.18)}$$

Therefore

$$\begin{aligned}
&P(|\tilde{U}(S) - U(S)| > \varepsilon) \\
&= P(\sum_{S' : S' \subseteq S,\ |S'| \geq \lceil \frac{|S|}{2} \rceil} |\prod_{i \in S'} \tilde{p}_i \prod_{S / S'} (1 - \tilde{p}_j) - \prod_{i \in S'} p_i \prod_{S \setminus S'} (1 - p_j)| > \varepsilon) \\
&\leq \sum_{S' : S' \subseteq S,\ |S'| \geq \lceil \frac{|S|}{2} \rceil} P(|\prod_{i \in S'} \tilde{p}_i \prod_{S \setminus S'} (1 - \tilde{p}_j) - \prod_{i \in S'} p_i \prod_{S \setminus S'} (1 - p_j)| > \frac{\varepsilon}{n(S)}) \ ,
\end{aligned}$$

where the last inequality comes from the union bound. We further use the following results (which can be proved separately but the proof of omitted): For $k \geq 1$ and two sequence

$\{l_i\}_{i=1}^m$ and $\{q_i\}_{i=1}^m$ and $0 \le l_i, q_i \le 1, \forall i = 1, ..., k.$, we have

$$|\prod_{i=1}^{m} l_i - \prod_{j=1}^{m} q_j| \le \sum_{i=1}^{m} |l_i - q_i| \ . \tag{A.19}$$

Using this result, we have

$$\begin{aligned}
&|\prod_{i\in S'} \tilde{p}_i \prod_{j\in S\setminus S'} (1-\tilde{p}_j) - \prod_{i\in S'} p_i \prod_{j\in S\setminus S'} (1-p_j)| \\
&\le \sum_{i\in S'} |\tilde{p}_i - p_i| + \sum_{j\in S\setminus S'} |(1-\tilde{p}_j) - (1-p_j)| = \sum_{i\in S} |\tilde{p}_i - p_i| \ .
\end{aligned}$$

Therefore using the union bound we have

$$P(|\prod_{i\in S'} \tilde{p}_i \prod_{S\setminus S'} (1-\tilde{p}_j) - \prod_{i\in S'} p_i \prod_{S\setminus S'} (1-p_j)| > \frac{\varepsilon}{n(S)}) \le \sum_{i\in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|}) \ ,$$

and therefore sum up all above we have,

$$\begin{aligned}
P(|\tilde{U}(S) - U(S)| > \varepsilon) &\le \sum_{S':S'\subseteq S,\ |S'|\ge\lceil\frac{|S|}{2}\rceil} \sum_{i\in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|}) \\
&= n(S)\cdot\sum_{i\in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|}) \ .
\end{aligned} \tag{A.20}$$

A.7 Proof of Theorem II.8

We prove by contradiction. Suppose there exists a pair (i, j), $i \in S, j \notin S$ such that $p_i < p_j$. We discuss the following cases. First of all as we already noted we have $\log \frac{p_i}{1-p_i} < \log \frac{p_j}{1-p_j}$. Consider the following fact the probability for correct labeling is given as,

$$P_c = p_i \cdot P(T_1) + (1-p_i)\cdot P(T_2) + \frac{P(T_3)}{2} \ , \tag{A.21}$$

where

$$T_1 = \{\sum_{u\in S_c} \log\frac{p_u}{1-p_u} > \sum_{e\in S_w} \log\frac{p_e}{1-p_e} - \log\frac{p_i}{1-p_i}\} ,$$
$$T_2 = \{\sum_{u\in S_c} \log\frac{p_u}{1-p_u} > \sum_{e\in S_w} \log\frac{p_e}{1-p_e} + \log\frac{p_i}{1-p_i}\} ,$$

and

$$T_3 = \{\text{A tie happens.}\} \tag{A.22}$$

where $S_c \neq S_w$ and $S_c \cup S_w = \mathscr{M} - \{i\}$, indicating the set of correct and wrong labelers respectively. Essentially the first two events correspond to cases when there is a majority group (including and excluding i respectively) and T_3 corresponds to the case when there is a tie.

Changing p_i to p_j since

$$P(T_1^j) \geq P(T_1),\ P(T_2) \geq P(T_2^j) ,$$

if $p_i > 0.5$, where $T_q^j, q \in \{1,2\}$ correspond to $T_q, q \in \{1,2\}$ by replacing i with j. And we have

$$\begin{aligned} &p_j \cdot P(T_1^j) + (1-p_j)\cdot P(T_2^j) - p_i \cdot P(T_1) - (1-p_i)\cdot P(T_2) \\ &\geq (p_j - p_i)\cdot (P(T_1) - P(T_2)) \geq 0 . \end{aligned}$$

For T_3. Consider the case p_i is in S_c. Then changing p_i to p_j will break the equilibrium and the probability of a correct output will become

$$p_j \cdot P(S_c)\cdot P(S_w) > p_i \cdot P(S_c)\cdot P(S_w) = \frac{P(T_3)}{2} , \tag{A.23}$$

where $P(S_c), P(S_w)$ correspond to the probability from the correct and wrong labelers, i.e.,

$$P(S_c) = \sum_{u \in S_c} p_u,\ P(S_w) = \sum_{e \in S_w} (1 - p_e)\ , \tag{A.24}$$

and the last inequality comes from the fact that at the equal case the probabilities of the label being either 0 or 1 are equivalent with each other.

A.8 Proof of Lemma II.10

$$P(\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) < \varepsilon) \leq P(\tilde{W}(\hat{S}) - W(\hat{S}) < -\varepsilon/2) + P(\tilde{W}(S \backslash \hat{S}) - W(S \backslash \hat{S}) > \varepsilon/2)\ .$$

Cause otherwise if

$$\tilde{W}(\hat{S}) - W(\hat{S}) \geq -\varepsilon/2,\ \tilde{W}(S \backslash \hat{S}) - W(S \backslash \hat{S}) < \varepsilon/2\ ,$$

we have

$$\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) \geq W(\hat{S}) - W(S \backslash \hat{S}) - \varepsilon \geq \varepsilon\ . \tag{A.25}$$

Consider each term above we have,

$$
\begin{aligned}
&P(\tilde{W}(\hat{S}) - W(\hat{S}) < -\varepsilon/2) \\
&\leq \sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| > \frac{\varepsilon}{2|\hat{S}|}) \\
&\leq \sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| > \frac{\varepsilon}{2|\hat{S}|} | |\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) \cdot P(|\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) \\
&+ \sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \geq \frac{\varepsilon}{2|\hat{S}|} | |\tilde{p}_i - p_i| < \frac{\varepsilon}{4C|\hat{S}|}) \cdot P(|\tilde{p}_i - p_i| < \frac{\varepsilon}{4C|\hat{S}|}) \\
&\leq \sum_{i\in S} P(|\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) \leq |\hat{S}|(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \,,
\end{aligned}
$$

since

$$
D_1(t) \geq 1/(\frac{1}{\max_m 4C\cdot m} - \alpha)\log t,\ \alpha < \frac{1}{\max_m 4C\cdot m} \,,
$$

as well as the fact that when $|\tilde{p}_i - p_i| \leq \frac{\varepsilon}{4C|\hat{S}|}$ and

$$
C > \max_i \max\{\frac{1+\varepsilon/4}{p_i}, \frac{1-\varepsilon/4}{1-p_i}, \frac{\varepsilon/4}{p_i}, \frac{\varepsilon/4}{1-p_i}\} \,,
$$

we have

$$
|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \leq 2C\cdot|\tilde{p}_i - p_i| < \frac{\varepsilon}{2|\hat{S}|} \,,
$$

where we have used the following results.

Lemma A.2. *With $\tilde{p}_i, p_i$ bounded away from 0 and 1 we have,*

$$
|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \leq 2C\cdot|\tilde{p}_i - p_i| \,, \tag{A.26}
$$

where C is a constant satisfying,

$$C > \max_i \max\{\frac{1}{p_i}, \frac{1}{\tilde{p}_i}, \frac{1}{1-p_i}, \frac{1}{1-\tilde{p}_i}\}. \tag{A.27}$$

Similarly we have

$$P(\tilde{W}(S\backslash\hat{S}) - W(S\backslash\hat{S}) > \varepsilon/2) \leq |S\backslash\hat{S}|(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \,.$$

Combine above we have

$$P(\tilde{W}(\hat{S})) - \tilde{W}(S\backslash\hat{S}) < \varepsilon) \leq |S|\cdot(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \,.$$

For the other case when $W(\hat{S})) = W(S\backslash\hat{S})$ we have,

$$\begin{aligned} P(|\tilde{W}(\hat{S}) - \tilde{W}(S\backslash\hat{S})| > \varepsilon) &\leq P(|\tilde{W}(\hat{S}) - W(\hat{S})| \geq \varepsilon/2) \\ &+ P(|\tilde{W}(S\backslash\hat{S}) - W(S\backslash\hat{S})| \geq \varepsilon/2) \\ &\leq |S|\cdot(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \,, \end{aligned} \tag{A.28}$$

where the second inequality is established similarly as in the first case.

A.9 Proof of Equation (A.19)

We prove the claim by induction. Notice when $m = 1$ the inequality holds trivially. When $m = 2$ we have

$$\begin{aligned}
&|l_1 \cdot l_2 - q_1 \cdot q_2| \\
&= |(l_1 - q_1) \cdot \frac{l_2 + q_2}{2} + (l_2 - q_2) \cdot \frac{l_1 + q_1}{2}| \\
&\leq |(l_1 - q_1) \cdot \frac{l_2 + q_2}{2}| + |(l_2 - q_2) \cdot \frac{l_1 + q_1}{2}| \\
&= |l_1 - q_1| \cdot |\frac{l_2 + q_2}{2}| + |l_2 - q_2| \cdot |\frac{l_1 + q_1}{2}| \\
&\leq |l_1 - q_1| + |l_2 - q_2| \ . \qquad \text{(A.29)}
\end{aligned}$$

The last inequality used the fact

$$|\frac{l_1 + q_1}{2}| \leq 1, \ |\frac{l_2 + q_2}{2}| \leq 1 \ .$$

Suppose the inequality holds for m.

$$\begin{aligned}
&|\prod_{i=1}^{m+1} l_i - \prod_{j=1}^{m+1} q_j| = |\prod_{i=1}^{m} l_i \cdot l_{m+1} - \prod_{j=1}^{m} q_j \cdot q_{m+1}| \\
&\leq |\prod_{i=1}^{m} l_i - \prod_{j=1}^{m} q_j| + |l_{m+1} - q_{m+1}| \\
&\leq \sum_{i=1}^{m+1} |l_i - q_i| \ , \qquad \text{(A.30)}
\end{aligned}$$

where the second inequality used the results for $m = 2$ which we proved, since

$$0 \leq \prod_{i=1}^{m} l_i, \prod_{j=1}^{m} q_j \leq 1 \ ,$$

and the last inequality used the induction hypothesis.

A.10 Proof of Lemma A.2

$$
\begin{aligned}
&|\log\frac{\tilde{p}_i}{1-\tilde{p}_i}-\log\frac{p_i}{1-p_i}| \\
&=|\log\tilde{p}_i-\log p_i+\log(1-p_i)-\log(1-\tilde{p}_i)| \\
&\leq|\log\tilde{p}_i-\log p_i|+|\log(1-p_i)-\log(1-\tilde{p}_i)| \\
&\leq 2C|\tilde{p}_i-p_i|\ , \qquad \text{(A.31)}
\end{aligned}
$$

since all four terms are bounded from 0 and the last inequality comes from classical inequality of $\log(\cdot)$ functions.

A.11 Proof for Proposition II.12

Let's ignore the $o(1)$ quantity for now: as we could easily show $\pi(\cdot)$ is linear in each p_i so the $o(1)$ change in each p_i will only result in a $o(1)$ change in $\pi(\cdot)$. First

$$
\pi(\frac{1}{2}+\delta,\frac{1}{2}+\varepsilon,\frac{1}{2}+\varepsilon)=\frac{1}{2}+\frac{\delta}{2}+\varepsilon-2\delta\varepsilon^2\ .
$$

Compare with $1-\delta$ we know we can find a (δ,ε) such that

$$
\begin{aligned}
&\frac{1}{2}+\frac{\delta}{2}+\varepsilon-2\delta\varepsilon^2<\frac{1}{2}+\delta \\
&\Leftrightarrow\varepsilon<\frac{\delta}{2}+2\delta\varepsilon^2\ .
\end{aligned}
$$

Now with error probability P_e we have the change in perception for each labelers' ac-

curacy as follows

$$\tilde{p}_1 = (\frac{1}{2}+\delta)(1-P_e)+(\frac{1}{2}-\delta)P_e = \frac{1}{2}+\delta(1-2P_e)\ ,$$
$$\tilde{p}_2 = (\frac{1}{2}+\varepsilon)(1-P_e)+(\frac{1}{2}-\varepsilon)P_e = \frac{1}{2}+\varepsilon(1-2P_e)\ ,$$
$$\tilde{p}_3 = (\frac{1}{2}+\varepsilon)(1-P_e)+(\frac{1}{2}-\varepsilon)P_e = \frac{1}{2}+\varepsilon(1-2P_e)\ .$$

First of all when $P_e > \frac{1}{2}$, we know $\tilde{p}_2 > \tilde{p}_1$, which will lead to the case that optimal set of labelers will be different from the case with p_1, p_2, p_3. When $P_e = \frac{1}{2}$, we will have $\tilde{p}_1 = \tilde{p}_2 = \tilde{p}_3 = \frac{1}{2}$. So the optimal solution does not equal to selecting labeler 1, which again leads to unbounded regrets. Now consider the case with $P_e < \frac{1}{2}$:

$$\pi(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3) = \frac{1}{2}+\frac{\delta(1-2P_e)}{2}+\delta(1-2P_e)-2\delta(1-2P_e)(\varepsilon(1-2P_e))^2\ .$$

Compare it with $\frac{1}{2}+\delta(1-2P_e)$ we know

$$\frac{1}{2}+\frac{\delta(1-2P_e)}{2}+\delta(1-2P_e)-2\delta(1-2P_e)(\varepsilon(1-2P_e))^2 > \frac{1}{2}+\delta(1-2P_e)$$
$$\Leftrightarrow \varepsilon > \frac{\delta}{2}+2\delta\varepsilon^2(1-2P_e)^2\ .$$

Depending on different P_e we know we could choose a pair of (ε, δ) such that

$$\varepsilon < \frac{\delta}{2}+2\delta\varepsilon^2\ ,$$
$$\varepsilon > \frac{\delta}{2}+2\delta\varepsilon^2(1-2P_e)^2\ ,$$

as above functions are all continuous in (ε, δ). So for any P_e we can find an example that based on $\tilde{p}_1, \tilde{p}_2, \tilde{p}_3$ the optimal solution set will be different from the one with p_1, p_2, p_3. Then following classical MAB results we will know the learning will converge to the sub-optimal solution which will make the learning regret being at the order of $O(T)$.

A.12 Proof for Proposition II.13

Now at each time t consider the hypothesis testing on whether a sub-optimal labeler is better than an optimal one, based on collected samples. Upto time t the number of making a wrong decision for above hypothesis is lower bounded by the summation of the event when a wrong ordering of the labelers occurs; as in cases with the top labeler being the optimal selection, a wrong ordering leads to a wrong selection.

Consider the following example with two hypothesis with parameters drawing from parameter space Θ. (Hypothesis H_i corresponds to parameter space θ_i.) Particularly suppose

$$\theta_0 = \{p_1, p_2, p_3 : p_1 > p_2 > p_3\} ,$$
$$\theta_1 = \{p_1', p_2, p_3 : p_2 > p_1' > p_3\} .$$

That is H_0 believes $p_1 > p_2$ while H_1 represents the hypothesis $p_2 > p_1$.

Denote by $T(t)$ as the number of sub-optimal arm selection upto time t. Then we have

$$\begin{aligned}
\sup_\theta E_\theta[T(t)] &= \sup_\theta \sum_{\tau=1}^{t} P_\theta(S(\tau) \neq S^*) \\
&\geq \sum_{\tau=1}^{t} \frac{P_{\theta_0}(S(\tau) \neq S^*) + P_{\theta_1}(S(\tau) \neq S^*)}{2} \\
&\geq \sum_{\tau=1}^{t} \frac{e^{-I(P^\tau_{H_0}, P^\tau_{H_1})}}{4} .
\end{aligned}$$

Denote the observation sequence as $X_1, ..., X_t$. Now consider each term in the summation

$$\begin{aligned}
I(P^\tau_{H_0}, P^\tau_{H_1}) &= E_{\theta_0}\left(\log(\frac{\tilde{f}(X_1, p_1)}{\tilde{f}(X_1, p_1')} \frac{\tilde{f}(X_2, p_1)}{\tilde{f}(X_2, p_1')} \cdots \frac{\tilde{f}(X_{T(\tau)}, p_1')}{\tilde{f}(X_{T(\tau)}, p_1')}) \right) \\
&= E_{\theta_0}[\sum_{t=1}^{T(\tau)} \log \frac{\tilde{f}(X_1, p_1)}{\tilde{f}(X_1, p_1')}] \\
&= \tilde{I}(p_1, p_1') E_\theta[T(\tau)] .
\end{aligned}$$

Notice each distribution we have used $\tilde{f}$ to denote this is rather a noisy observation.

There we have (as similarly argued in [11])

$$\begin{aligned}
S_t &\geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1,p_1')E_{\theta_0}[T(\tau)]} \\
&\geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1,p_1')\sup_\theta E_\theta[T(\tau)]} \\
&= \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1,p_1')S_\tau} \\
&\geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1,p_1')S_t} \\
&= \frac{t}{4} e^{-\tilde{I}(p_1,p_1')S_t} \ .
\end{aligned}$$

Take log on both sides and rearrange we know

$$S_t \geq \frac{\log t}{\tilde{I}(p_1,p_1')} + o(\frac{\log t}{\tilde{I}(p_1,p_1')}) \ . \tag{A.32}$$

Now consider $\sup_{P_e} S_t$ and we start with bounding $\tilde{I}(p_1,p_1')$. Consider the following fact.

$$\begin{aligned}
\tilde{I}(p_1,p_1') &= E_{\theta_0}(\log \frac{\tilde{f}(x,p_1)}{\tilde{f}(x,p_1')}) \\
&= E_{\theta_0}(\log \frac{f(x,p_1)(1-P_e)+(1-f(x,p_1))P_e}{f(x,p_1')(1-P_e)+(1-f(x,p_1'))P_e}) \\
&= E_{\theta_0}(\log \frac{f(x,p_1)+\frac{P_e}{1-2P_e}}{f(x,p_1')+\frac{P_e}{1-2P_e}})
\end{aligned}$$

For each $x \in \{0,1\}$, consider each function $\log \frac{f(x,p_1)+\frac{P_e}{1-2P_e}}{f(x,p_1')+\frac{P_e}{1-2P_e}}$. Denote $\delta = \frac{P_e}{1-2P_e}$ and

$$g(\delta) = \log \frac{f(x,p_1)+\delta}{f(x,p_1')+\delta} \ , \delta \geq 0. \tag{A.33}$$

By checking the second order derivative we can easily show that when $f(x,p_1) \geq f(x,p_1')$, $g(\delta)$ is convex in δ while when $f(x,p_1) < f(x,p_1')$, $g(\delta)$ is concave. Therefore

we have when $f(x,p_1) \geq f(x,p_1')$,

$$\begin{aligned} g(\delta) &\geq g(0) + g'(0)\delta \\ &= \log \frac{f(x,p_1)}{f(x,p_1')} + \frac{f(x,p_1) - f(x,p_1')}{f(x,p_1)f(x,p_1')}\delta \ . \end{aligned}$$

While when $f(x,p_1) < f(x,p_1')$,

$$\begin{aligned} g(\delta) &\geq g(0) - g'(\delta)(-\delta) \\ &= \log \frac{f(x,p_1)}{f(x,p_1')} + \frac{f(x,p_1) - f(x,p_1')}{(f(x,p_1)+\delta)(f(x,p_1')+\delta)}\delta \ . \end{aligned}$$

Denote

$$-C_1 = \min_x f(x,p_1) - f(x,p_1') \ ,$$

and

$$C_2 = \min_{x,\theta} f(x,\theta) \ .$$

Since we cannot have a probability measure being strictly larger than another on each outcome and for two different measures we know $C_1 > 0, C_2 > 0$. Then

$$g(\delta) \geq g(0) - \frac{C_1}{(C_2+\delta)^2}\delta. \tag{A.34}$$

Then

$$E_{\theta_0}(\log \frac{f(x,p_1) + \frac{P_e}{1-2P_e}}{f(x,p_1') + \frac{P_e}{1-2P_e}}) \geq E_{\theta_0}(\log \frac{f(x,p_1)}{f(x,p_1')}) - \frac{C_1}{(C_2+\delta)^2}\delta. \tag{A.35}$$

That is

$$\tilde{I}(p_1, p_1') \geq I(p_1, p_1') - \frac{C_1}{(C_2+\delta)^2}\delta \ . \tag{A.36}$$

Then

$$S_t \geq \frac{\log t}{I(p_1, p_1') - \frac{C_1}{(C_2+\delta)^2}\delta} \ . \tag{A.37}$$

We now see since

$$\delta = \frac{P_e}{1-2P_e} \ , \tag{A.38}$$

when $P_e \to 0$, $\delta \to 0$ and so is $\frac{C_1}{(C_2+\delta)^2}\delta$.

We can easily prove that $I(p_1, p_2 - x)$ is increasing in x; so the lower bound is again be bounded by setting $x = 0$ that is by setting $p_1' = p_2$. Combine above analysis we know we can achieve a lower bound as $\log t D_2(t)$.

A.13 Proof of Theorem II.14

The difference in selecting $D_2(t)$ (compared to setting $D_2(t) := O(\log t)$) lies in the following factor for inferring the ground-truth label for each tester:

$$\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \ .$$

Also we already know,

$$E[1\{y_k^* = 0\}] \leq e^{-2(\bar{p}-1/2)^2 D_2(t)} \ .$$

Now consider the following events

$$\omega_1(t) := \{\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \alpha\varepsilon\} \ , \tag{A.39}$$

and

$$\omega_2(t) := \{\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \alpha\varepsilon\} \ , \tag{A.40}$$

We plug (A.39) as in proof II.6 we show we have the gap become

$$(\frac{1}{n(S)\cdot|S|} - \alpha)\varepsilon \ , \tag{A.41}$$

with convergence rate being $e^{-2(\bar{p}-1/2)^2 D_2(t)}$. This part of analysis can go through similarly as we argued for previous analysis. Now consider (A.40). By Chernoff bound

$$\begin{aligned}
&P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \alpha\varepsilon) \\
&\leq P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} - e^{-2(\bar{p}-1/2)^2 D_2(t)} > \alpha\varepsilon - e^{-2(\bar{p}-1/2)^2 D_2(t)}) \\
&\leq e^{-2(\alpha\varepsilon - e^{-2(\bar{p}-1/2)^2 D_2(t)})O(\log t)} \\
&= O(\frac{1}{t^2}) \ ,
\end{aligned} \tag{A.42}$$

with appropriately selected parameter. Notice in order for inequality (A.42) to hold (to enable Chernoff bound), we need

$$\alpha\varepsilon - e^{-2(\bar{p}-1/2)^2 D_2(t)} > 0.$$

This is guaranteed to hold after a finite time T_0 as $e^{-2(\bar{p}-1/2)^2 D_2(t)}$ is decreasing to 0 with both α and ε stay as constants.

APPENDIX B

Proofs for Chapter III

B.1 Proof of Lemma III.1

For simplicity of presentation, in this proof we omit all sub- and super-scripts when there is no confusion; and further denote by $\delta(t)$ the estimate at time t and δ^* the true value. With a bit abuse of notation, throughout proofs we use $\hat{r}_k^j(l)$ to denote the estimated log sample mean of user j for option k with l samples, instead of being at time l as we defined earlier in the contexts. Also for simplicity of presentation we assume $\hat{r}_k^j(l), \hat{\mu}_k^j > 0$. Otherwise we could easily reverse the sign of $\hat{r}_k^j(l), \hat{\mu}_k^j$ and δ^* to make it consistent.

$$P(|\delta(t) - \delta^*| > \varepsilon) = P(\delta(t) > \delta^* + \varepsilon) + P(\delta(t) < \delta^* - \varepsilon) \ .$$

Consider $P(\delta(t) > \delta^* + \varepsilon)$.

$$\begin{aligned} P(\delta(t) > \delta^* + \varepsilon) &= P(\frac{\hat{r}_k^i(l)}{\hat{r}_k^j(l)} > \delta^* + \varepsilon) \\ &= P(\hat{r}_k^i(l) > (\delta^* + \varepsilon) \cdot \hat{r}_k^j(l)) \ . \end{aligned}$$

Choose $\xi := \frac{\hat{\mu}_k^j \varepsilon}{2(\delta^* + \varepsilon)}$. And consider the following two events.

$$E_1(t) = \{\omega : |\hat{r}_k^j(l) - \hat{\mu}_k^j| \geq \xi\} ,$$
$$E_2(t) = \{\omega : |\hat{r}_k^j(l) - \hat{\mu}_k^j| < \xi\} .$$

Clearly we have $E_1(t) \cap E_2(t) = \emptyset$ and $E_1(t) \cup E_2(t) = \mathscr{S}$, where $\mathscr{S}$ denotes the whole set.

$$\begin{aligned} & P(\hat{r}_k^i(l) > (\delta^* + \varepsilon) \cdot \hat{r}_k^j(l)) \\ & = P(\hat{r}_k^i(l) > (\delta^* + \varepsilon) \cdot \hat{r}_k^j(l) | E_1(t)) P(E_1(t)) \\ & + P(\hat{r}_k^i(l) > (\delta^* + \varepsilon) \cdot \hat{r}_k^j(l) | E_2(t)) P(E_2(t)) . \end{aligned}$$

Using Chernoff-Hoeffding bound we know

$$P(E_1(t)) \leq e^{-2\xi^2 \cdot l} . \tag{B.1}$$

Consider the second term with $E_2(t)$. We have

$$\begin{aligned} & P(\hat{r}_k^i(l) > (\delta^* + \varepsilon) \cdot \hat{r}_k^j(l) | E_2(t)) \\ & \leq P(\hat{r}_k^j(l) > (\delta^* + \varepsilon) \cdot (\hat{\mu}_k^j - \xi)) \\ & = P(\hat{r}_k^i(l) - \delta^* \cdot \hat{\mu}_k^j \geq \hat{\mu}_k^j \cdot \varepsilon - (\delta^* + \varepsilon) \cdot \xi) \\ & = P(\hat{r}_k^i(l) - \delta^* \cdot \hat{\mu}_k^j \geq \hat{\mu}_k^j \cdot \varepsilon - (\delta^* + \varepsilon) \cdot \frac{\hat{\mu}_k^j \varepsilon}{2(\delta^* + \varepsilon)}) \\ & = P(\hat{r}_k^i(l) - \delta^* \cdot \hat{\mu}_k^j \geq \frac{\hat{\mu}_k^j \varepsilon}{2}) \leq e^{-2(\frac{\hat{\mu}_k^j \varepsilon}{2})^2 l} . \end{aligned} \tag{B.2}$$

The other part of proving $P(\delta(t) < \delta^* - \varepsilon)$ can be similarly done due to symmetry and is thus omitted.

Now our task at hand is clear: we want to find the minimum number of samples that are needed (that is a minimum l) such that we will have $O(\frac{1}{t^2})$ probability bound for both

Eqn.(B.1) and Eqn.(B.2):

$$\begin{aligned} \min \quad & l \\ \text{s.t.} \quad & e^{-2\xi^2 \cdot l} \leq \frac{1}{t^2}, \qquad \text{(B.3)} \\ & e^{-2(\frac{\hat{\mu}_k^j \varepsilon}{2})^2 l} \leq \frac{1}{t^2}\ . \qquad \text{(B.4)} \end{aligned}$$

We now prove the three claims according to the number of samples l, respectively.

(1) $l = O(t)$. For this case suppose $l = C_1 t$. Take $\varepsilon = \frac{2}{\hat{\mu}_k^j} \cdot \sqrt{\frac{\log t}{C_1 t}}$, we have $e^{-2(\frac{\hat{\mu}_k^j \varepsilon}{2})^2 l} = \frac{2}{t^2}$. Moreover notice $e^{-2\xi^2 l} = e^{-2(\frac{\hat{\mu}_k^j \varepsilon}{2(\delta^* + \varepsilon)})^2 C_1 t}$, let $\varepsilon = \frac{2\delta^*}{\hat{\mu}_k^j - 2\sqrt{\frac{\log t}{C_1 t}}} \sqrt{\frac{\log t}{C_1 t}}$, we have $e^{-2\xi^2 l} \leq \frac{1}{t^2}$. In all in order for above two conditions to hold simultaneously, take

$$\begin{aligned} \varepsilon &= \max\{\frac{2\delta^*}{\hat{\mu}_k^j - 2\sqrt{\frac{\log t}{C_1 t}}} \sqrt{\frac{\log t}{C_1 t}}, \frac{2}{\hat{\mu}_k^j} \cdot \sqrt{\frac{\log t}{C_1 t}}\} \\ &\leq \frac{2\max\{\delta^*, 1\}}{\hat{\mu}_k^j - 2\sqrt{\frac{\log t}{C_1 t}}} \cdot \sqrt{\frac{\log t}{C_1 t}}\ . \end{aligned}$$

Notice since $\hat{\mu}_k^j - 2\sqrt{\frac{\log t}{C_1 t}}$ is readily lower bounded by a constant, we know with C_1 being a constant we can roughly achieve a $\sqrt{\frac{\log t}{t}}$ estimation error ε with $O(\frac{1}{t^2})$ error rate.

(2) $l = O(\log t)$. Suppose $l = C_2 \log t$, from Condition (B.3) (and plug in ξ) we need to have

$$\begin{aligned} & C_2 (\hat{\mu}_k^j)^2 \varepsilon^2 = 4(\delta^* + \varepsilon)^2 \\ \Leftrightarrow\ & \sqrt{C_2} \hat{\mu}_k^j \varepsilon = 2(\delta^* + \varepsilon) \\ \Leftrightarrow\ & \varepsilon = \frac{2\delta^*}{\sqrt{C_2}\hat{\mu}_k^j - 2}\ . \end{aligned}$$

On the other hand, take $\varepsilon = \frac{2}{\hat{\mu}_k^j \cdot \sqrt{C_2}}$, we have Condition (B.2) holds. Again we obtain a ε

satisfying both conditions as follows

$$\varepsilon = \max\{\frac{2}{\hat{\mu}_k^j \cdot \sqrt{C_2}}, \frac{2\delta^*}{\sqrt{C_2}\hat{\mu}_k^j - 2}\} \leq \frac{2\max\{\delta^*, 1\}}{\sqrt{C_2}\hat{\mu}_k^j - 2} \ .$$

Remember we have ignore the sign of $\hat{\mu}_k^j$ so we put the $|\cdot|$ to upper bound it.

B.2 Proof of Theorem III.2

We first convert the estimation error in the distortion to that in the collected samples.

$$X_k^j(t)^{\tilde{\delta}} = X_k^j(t)^{\delta} \cdot X_k^j(t)^{\tilde{\delta}-\delta} \ .$$

First consider the case with $\varepsilon := \tilde{\delta} - \delta \geq 0$ we have

$$X_k^j(t)^{\tilde{\delta}-\delta} \leq 1 + X_k^j(t) \cdot \varepsilon, \tag{B.5}$$

$\forall \varepsilon \in [0,1]$. The second inequality comes from the following fact

$$\varepsilon \log X_k^j(t) + (1-\varepsilon)\log 1 \leq \log(\varepsilon X_k^j(t) + 1) \ ,$$

where the concavity of log function is used. The case with $\varepsilon < 0$ can be similarly proved. The requirement of $\varepsilon \leq 1$ will be clear when the regret bound is proved; the idea is when number of samples is large enough, this criteria can be fairly easily achieved. Therefore since the error with each sample is bounded by $X_k^i(t) \cdot X_k^j(t) \cdot \varepsilon$.

We follow the idea used in [5] where UCB1 is first introduced and analyzed, and bound the number of times sub-optimal arms are played. Consider the total number of option $k \in \overline{N}_K^i$ has been used by user i up to time t and denote it by $T_k^i(t)$, and denote the following biase term for user i on its index for option k when the option has been sampled n_k^i times: $c_{n_k^i}^i(t) = \sqrt{\frac{2\log t}{n_k^i}}$. We use $r_k^i(n_k^i)$ to denote the estimated sample mean (as defined in U-

FULL index with more details) with no distortion errors and the corresponding $\tilde{r}_k^i(n_k^i)$ to denote the one with such error of option k for user i with n_k^i local plays under our algorithm. Then for any $\zeta \geq 0$ we have:

$$\begin{aligned}
T_k^i(t) &\leq \zeta + \sum_{s=\zeta+1}^{t} 1\{k \in a^i(s), T_k^i(s-1) \geq \zeta\} \\
&\leq \zeta + \sum_{s=\zeta+1}^{t} 1\{\min_{0<n_{k^*}^i<s} \tilde{r}_{k^*}^i(n_{k^*}^i) + c_{n_{k^*}^i}^i(s-1) \\
&\leq \max_{\zeta<n_k^i<s} \tilde{r}_k^i(n_k^i) + c_{n_k^i}^i(s-1), \exists k^* \in N_K^i\} \\
&\leq \zeta + \sum_{k^* \in N_K^i} \sum_{s=1}^{\infty} \sum_{n_{k^*}^i=1}^{s-1} \sum_{n_k^i=\zeta}^{s-1} 1\{\tilde{r}_{k^*}^i(n_{k^*}^i) + c_{n_{k^*}^i}^i(s) \leq \tilde{r}_k^i(n_k^i) + c_{n_k^i}^i(s)\} \ .
\end{aligned}$$

We will first show that at time t $\tilde{r}_{k^*}^i(n_{k^*}^i) + c_{n_{k^*}^i}^i(s) \leq \tilde{r}_k^i(n_k^i) + c_{n_k^i}^i(s)$ implies that at least one of the following must hold:

$$\begin{aligned}
r_{k^*}^i(n_{k^*}^i) \leq \mu_{k^*}^i - c_{n_{k^*}^i}^i(s),\ r_k^i(n_k^i) \geq \mu_k^i + c_{n_k^i}^i(s), \\
\mu_{k^*}^i \leq \mu_k^i + 2c_{n_k^i}^i(s) + 2\bar{X}_k^2 \varepsilon \ .
\end{aligned} \tag{B.6}$$

Otherwise if none of the three inequalities holds, we have

$$\begin{aligned}
\tilde{r}_{k^*}^i(n_*^i) + c_{n_{k^*}^i}^i(s) &\geq r_{k^*}^i(n_*^i) + c_{n_{k^*}^i}^i(s) - \bar{X}_k^2 \cdot \varepsilon > \mu_{k^*}^i - \bar{X}_k^2 \cdot \varepsilon \\
&> \mu_k^i + 2 \cdot c_{n_k^i}^i(s) + \bar{X}_k^2 \cdot \varepsilon > r_k^i(n_k^i) + c_{n_k^i}^i(s) + \bar{X}_k^2 \cdot \varepsilon \\
&> \hat{r}_k^i(n_k^i) + c_{n_k^i}^i(s) \ ,
\end{aligned} \tag{B.7}$$

which is a contradiction.

We now bound each of the three terms in (B.6). First via Chernoff-Holding bounds we

have

$$P\{r^i_{k^*}(n^i_{k^*}) \leq \mu^i_{k^*} - \sqrt{2}\sqrt{\frac{\log s}{\sum_{j\in\mathscr{U}} n^j_{k^*}}}\} \leq e^{-4\log s} = s^{-4} .$$

And similarly we have $P\{r^i_k(n^i_k) \geq \mu^i_k + c^i_{n^i_k}(s)\} \leq s^{-4}$. For the last term, firstly we argue at time t, w.h.p., $\forall k \in N^i_K$ we have $n^i_k(t) = O(t)$. Suppose there exists a $k \in N^i_K$ such that $n^i_k(t) = o(t)$ we know there must exist a $\bar{k} \in \overline{N}^i_K$, such that $n^i_{\bar{k}}(t) = O(t)$. However the probability of such event is bounded above by $O(\frac{1}{t^2})$, as can be shown following classical MAB results or can be adapted from this proof (that $E[n^i_{\bar{k}}(t)] = O(\log t)$). Since we have assumed $N^i_K \cap N^j_K \neq \emptyset$, select one $k \in N^i_K \cap N^j_K$. Then we know $\min\{n^i_k(t), n^j_k(t)\} = O(t)$. Denote the number as $C_1 t$ and based on results in Lemma 1 we have the estimation error for δ trained via data samples collected for option k given by

$$\varepsilon_k(t) = \frac{2\delta^{i,*}}{\min_{j\neq i}|\hat{\mu}^j_k| - 2\sqrt{\frac{\log t}{C_1 t}}} \cdot \sqrt{\frac{\log t}{C_1 t}} .$$

Clearly $\varepsilon_k(t)$ satisfies $\varepsilon_k(t) \leq 1$ when t is large enough.

Let's set $\zeta = \lceil L \log t \rceil$. Then we want a L such that

$$2\sqrt{2} \cdot \sqrt{\frac{\log t}{L \cdot \log t \cdot M}} + 2\bar{X}^2_k \cdot \varepsilon_k(t) \leq \Delta^i_k , \tag{B.8}$$

which gives us $L = \frac{8}{M(\Delta^i_k - 2\bar{X}^2_k \cdot \varepsilon_k(t))^2}$. With above L and ζ we have

$$\mu^i_{k^*} - \mu^i_k - 2c^i_{n^i_k}(s) - 2\bar{X}^2_k \cdot \sqrt{\varepsilon_k(t)} \geq \mu^i_{k^*} - \mu^i_k - \Delta^i_k \geq 0 .$$

Following similar steps used in proving UCB1 [5] we know the number of sampled

sub-optimal arm k is bounded by

$$
\begin{aligned}
E[T_k(t)] &\leq \lceil \frac{8}{M(\Delta_k^i - 2\bar{X}_k^2 \cdot \varepsilon_k(t))^2} \cdot \log t \rceil \\
&+ K \sum_{s=1}^{\infty} \sum_{n_{k^*}^i=1}^{s-1} \sum_{n_k^i=\zeta}^{s-1} (P\{r_{k^*}^i(n_{k^*}^i) \leq \mu_{k^*}^i - c_{n_{k^*}^i}^i(s)\} \\
&+ P\{r_k^i(n_k^i) \geq \mu_k^i + c_{n_k^i}^i(s)\}) \\
&\leq \lceil \frac{8}{M(\Delta_k^i - 2\bar{X}_k^2 \cdot \varepsilon_k(t))^2} \cdot \log t \rceil + K \sum_{s=1}^{\infty} \sum_{n_{k^*}^i=1}^{s} \sum_{n_k^i=1}^{s} 2s^{-4} \\
&\leq \lceil \frac{8}{M(\Delta_k^i - 2\bar{X}_k^2 \cdot \varepsilon_k(t))^2} \cdot \log t \rceil + \text{const.} ,
\end{aligned}
$$

and by multiplying Δ_k^i and adding up we have the regret of mistaking a sub-optimal for an optimal arm is bounded by

$$
R_{\text{CL-FULL}}^i(t) \leq \sum_{k \in \bar{N}_K^i} \lceil \frac{8\delta_k^{i,*}}{M(\Delta_k^i - 2\bar{X}_k^2 \cdot \varepsilon_k(t))^2} \cdot \log t \rceil + \text{const.}
$$

B.3 Proof of Theorem III.3

This part's proof is similar to what we presented in last proof. Again at time t $\tilde{r}_{k^*}^i(n_{k^*}^i) + c_{n_{k^*}^i}^i(s) \leq \tilde{r}_k^i(n_k^i) + c_{n_k^i}^i(s)$ implies that at least one of the following must hold with probability being at least $1 - O(\sum_{j \neq i} e^{-\frac{(\mu_k^i \cdot \mu_k^j)^2 \zeta}{4}})$:

$$
\begin{aligned}
r_{k^*}^i(n_{k^*}^i) \leq \mu_{k^*}^i - c_{n_{k^*}^i}^i(s),\ r_k^i(n_k^i) \geq \mu_k^i + c_{n_k^i}^i(s), \\
\mu_{k^*}^i \leq \mu_k^i + 2c_{n_k^i}^i(s) + 2 \cdot \frac{3}{2} \mu_k^i \cdot \mu_k^j \sqrt{\varepsilon} \, .
\end{aligned} \tag{B.9}
$$

First we know by Chernoff bound

$$
P(|\frac{\sum_{n=1}^{\zeta} X_k^i(n) X_k^j(n)}{\zeta} - \mu_k^i \cdot \mu_k^j| \geq \frac{\mu_k^i \cdot \mu_k^j}{2}) \leq 2e^{-\frac{(\mu_k^i \cdot \mu_k^j)^2 \zeta}{4}} \tag{B.10}
$$

And then otherwise if none of the three inequalities holds, we have

$$\begin{aligned}\tilde{r}^i_{k^*}(n^i_*) + c^i_{n^i_{k^*}}(s) &\geq r^i_{k^*}(n^i_*) + c^i_{n^i_{k^*}}(s) - \frac{3}{2}\mu^i_k\mu^j_k\varepsilon > \mu^i_{k^*} - \frac{3}{2}\mu^i_k\mu^j_k\varepsilon \\ &> \mu^i_k + 2\cdot c^i_{n^i_k}(s) + \frac{3}{2}\mu^i_k\mu^j_k\varepsilon > r^i_k(n^i_k) + c^i_{n^i_k}(s) + \frac{3}{2}\mu^i_k\mu^j_k\varepsilon \\ &> \hat{r}^i_k(n^i_k) + c^i_{n^i_k}(s)\ ,\end{aligned}$$

which is a contradiction.

Again after setting $\zeta = \lceil L\log t\rceil$ what we are willing to show is

$$2\sqrt{2}\cdot\sqrt{\frac{\log t}{L\cdot\log t\cdot M}} + 3\mu^i_k\mu^j_k\cdot\varepsilon_k(t) \leq \Delta^i_k\ .$$

By the results of Lemma 1, since we have $L\log t$ samples,

$$\varepsilon_k(t) = \frac{2\delta^{i,*}_k}{\sqrt{L}\min_{j\neq i}|\hat{\mu}^j_k| - 2}\ ,$$

which gives us

$$2\sqrt{2}\cdot\sqrt{\frac{\log t}{L\cdot\log t\cdot M}} + 3\mu^i_k\mu^j_k\cdot\frac{2\delta^{i,*}_k}{\sqrt{L}\min_{j\neq i}|\hat{\mu}^j_k| - 2} \leq \Delta^i_k\ . \tag{B.11}$$

Suppose $\frac{\sqrt{L}\min_{j\neq i}|\hat{\mu}^j_k|}{2} - 2 \geq 0,$ that is

$$L \geq \frac{16}{(\min_{j\neq i}|\hat{\mu}^j_k|)^2}\ , \tag{B.12}$$

which we will verify later with L. Then we have

$$\begin{aligned}&2\sqrt{2}\cdot\sqrt{\frac{\log t}{L\cdot\log t\cdot M}} + 3\mu^i_k\mu^j_k\cdot\frac{2\delta^{i,*}_k}{\sqrt{L}\min_{j\neq i}|\hat{\mu}^j_k| - 2} \\ &\leq 2\sqrt{2}\cdot\sqrt{\frac{1}{L\cdot M}} + 3\mu^i_k\mu^j_k\cdot\frac{4\delta^{i,*}_k}{\sqrt{L}\min_{j\neq i}|\hat{\mu}^j_k|}\ .\end{aligned}$$

Then set the following equality we have

$$2\sqrt{2}\cdot\sqrt{\frac{\log t}{L\cdot\log t\cdot M}}+3\mu_k^i\mu_k^j\cdot\frac{4\delta_k^{i,*}}{\sqrt{L}\min_{j\neq i}\hat{\mu}_k^j}=\Delta_k^i\ ,$$

we have

$$L=\frac{(2\sqrt{\frac{2}{M}}+\frac{12\mu_k^i\mu_k^j\cdot\delta_k^{i,*}}{\min_{j\neq i}|\hat{\mu}_k^j|})^2}{(\Delta_k^i)^2}\ .$$

Rest of the proofs is similar to the ones for Theorem 2. Now we verify the claim made in Eqn.(B.12). Notice

$$L\geq\frac{(\frac{12\mu_k^i\mu_k^j\cdot\delta_k^{i*}}{\min_{j\neq i}|\hat{\mu}_k^j|})^2}{(\Delta_k^i)^2}\geq\frac{16}{(\min_{j\neq i}|\hat{\mu}_k^j|)^2}\cdot\frac{9(\mu_k^i\mu_k^j)^2(\delta_k^{i,*})^2}{(\Delta_k^i)^2}\ . \tag{B.13}$$

Since $\delta_k^{i,*}\geq 1$ we know

$$L\geq\frac{16}{(\min_{j\neq i}|\hat{\mu}_k^j|)^2}\cdot\frac{9(\mu_k^i\mu_k^j)^2}{(\Delta_k^i)^2}\geq\frac{16}{(\min_{j\neq i}|\hat{\mu}_k^j|)^2}\ ,$$

when $\frac{9(\mu_k^i\mu_k^j)^2}{(\Delta_k^i)^2}\geq 1$, which is easy to satisfy.

Now consider $\varepsilon_k(t)=\frac{2\delta_k^{i,*}}{\sqrt{L}\min_{j\neq i}|\hat{\mu}_k^j|-2}$. Plug in L we have

$$\varepsilon_k(t)\leq\frac{4\delta_k^{i,*}}{\frac{12(\mu_k^i\mu_k^j)^2\delta_k^{i,*}}{\min_{j\neq i}|\hat{\mu}_k^j|}\min_{j\neq i}|\hat{\mu}_k^j|}=\frac{4}{12(\mu_k^i\mu_k^j)^2}<1,$$

when $\mu_k^i\mu_k^j>\sqrt{\frac{1}{3}}$. Now for $O(\sum_{j\neq i}e^{-\frac{(\mu_k^i\cdot\mu_k^j)^2\zeta}{4}})$, in order to have a $O(\frac{1}{t^2})$ rate, we need $L\geq\frac{8}{(\mu_k^i\cdot\min_{j\neq i}\mu_k^j)^2}$. Combine all above we have the results.

B.4 Proof of Theorem III.4

Again denote $\varepsilon := |\tilde{\delta}_k^{i,j} - \delta_k^{i,j}|$ as the estimation error for δ. Since for each converted signal we have

$$|(X_k^j(t))^{\tilde{\delta}_k^{i,j}} - X_k^i(t)| \leq \bar{X}_k^2 \varepsilon \ . \tag{B.14}$$

Notice when $\bar{X}_k^2 \varepsilon \leq \frac{\Delta_{\mathscr{X}_k^i}}{2}$ there will be no error in the sample converting procedure. Again via Lemma III.1 we know with $L \log t$ number of samples we have $\varepsilon = \frac{2\delta_k^{i,*}}{\sqrt{L}\min_{j\neq i}|\hat{\mu}_k^j| - 2}$, and after we set it to be equal to $\frac{\Delta_{\mathscr{X}_k^i}}{2\bar{X}_k^2}$ we have

$$L = (\frac{4\bar{X}_k^2 \delta_k^{i,*} + 2\Delta_{\mathscr{X}_k^i}}{\Delta_{\mathscr{X}_k^i} \min_{j\neq i}|\hat{\mu}_k^j|})^2 \ .$$

When the system is error free, following proofs for Theorem 3 we can similarly prove:

$$R^i_{\text{CL-FULL}}(t) \leq \sum_{k \in \overline{N}_K^i} \lceil \frac{8}{M\Delta_k^i} \cdot \log t \rceil + \text{const.}$$

Then

$$R^i_{\text{CL-FULL}}(t) \leq \sum_{k \in \overline{N}_K^i} \lceil \max\{\frac{8}{M\Delta_k^i}, (\frac{4\bar{X}_k^2 \delta_k^{i,*} + 2\Delta_{\mathscr{X}_k^i}}{\Delta_{\mathscr{X}_k^i} \min_{j\neq i}|\hat{\mu}_k^j|})^2 \Delta_k^i\} \cdot \log t \rceil + \text{const.}$$

B.5 Proof for Theorem III.5

First by Hoeffding's inequality when each user has $L\log t$ number of samples for option k we have

$$P(|(\hat{r}_k^i(t)+\sum_{l\neq i}\hat{r}_k^l(t)\tilde{\delta}_k^{i,l})-(\hat{\mu}_k^i+\sum_{l\neq i}\hat{\mu}_k^l\tilde{\delta}_k^{i,l})|\geq\varepsilon)\leq 2e^{-\frac{\varepsilon^2(ML\log t)^2}{ML\log t\bar{X}_k^2}}.$$

Set $\varepsilon=\frac{\bar{X}_k}{\sqrt{ML}}$ we have the RHS of above inequality reduces to $\frac{2}{t^2}$. Similarly we have this ε bound for the denominator.

Denote $L_{\Phi_k^0}$ and L_A, L_B as the matrix representation of operator defined on the quadratic matrix equations as similarly defined in [40]. We further have the following results from [40]:

$$|\tilde{\delta}_k^{i,j}-\delta_k^{i,j}|\leq\sum_l(|L_{\Phi_k^0}^{-1}L^A(i,l)|+|L_{\Phi_k^0}^{-1}L^B(i,l)|)\varepsilon\ . \tag{B.15}$$

When $L_{\Phi_k^0}^{-1}L^A$ and $L_{\Phi_k^0}^{-1}L^B$ are well conditioned such that each row sum is bounded up by a constant,

$$\sum_l L_{\Phi_k^0}^{-1}|L^A(i,l)|+\sum_l L_{\Phi_k^0}^{-1}|L^B(i,l)|\leq C_3, \tag{B.16}$$

we have

$$|\tilde{\delta}_k^{i,j}-\delta_k^{i,j}|\leq C_3\cdot\varepsilon\ . \tag{B.17}$$

The rest of the proof is similar with the one for Theorem III.3. This result also implies when $\sum_l|L_{\Phi_k^0}^{-1}L^{A(B)}(i,l)|$ is unbounded in M, the perturbation of QME solution cannot be bounded as desired, under which case we may not expect the M factor (but with a sub-linear term in M being possible).

$$P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq \sum_{s=1}^{t} 1\{k^* \in a^i(s)\}) \leq \underbrace{P(\sum_{s=1}^{t} 1\{k^* \in a^i(s)\} < O(t))}_{\textbf{Term 1}}$$
$$\cdot P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} | \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} < O(t))$$
$$+ P(\sum_{s=1}^{t} 1\{k^* \in a^i(s)\} \geq O(t))$$
$$\underbrace{\cdot P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} | \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} \geq O(t))}_{\textbf{Term 2}} . \tag{B.18}$$

B.6 Proof of Lemma B.6

The main idea is to bound the probability that the number of times sub-optimal options are selected being higher than that of the optimal options by time t. Consider $k \in \overline{N}_K^i$ and $k^* \in N_K^i$ we have Eqn.(B.18).

Now bound Term 1 and Term 2 separately. For Term 2,

$$P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} | \sum_{s=1}^{t} 1\{k^* \in a^i(s)\} \geq O(t))$$
$$\leq P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq O(t)) \leq \frac{C}{t^2} .$$

Consider the following fact that when $\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq O(t)$ happens, there must exists some time $t^* > O(t^c), 0 < c < 1$ such that an error in mis-selection is made, and the probability of such an event, as shown in the proof of Theorem 2,3,4, can be bounded up by $\frac{C}{t^2}$ for certain constant C. For Term 1,

$$P(\sum_{s=1}^{t} 1\{k^* \in a^i(s)\} < O(t))$$
$$\leq P(\sum_{s=1}^{t} 1\{k \in a^i(s)\} \geq O(t)) \leq \frac{C}{t^2} .$$

The reason for the first inequality above is when an optimal option has been used less than $O(t)$ times, there must exist a sub-optimal option k that has been used $O(t)$ times, because the total number of selections $K \cdot t$ up to time t is fixed.

Proof of Theorem III.7

Denote for user i,

$$\hat{c}^i_{n^i_k}(s) = \sqrt{\frac{2\log s}{n^i_k}} + \alpha(s) \cdot \tilde{\beta}^i_k(s) \sqrt{\frac{\log s}{n^i_k}} \ , \forall k \in \Omega, s,$$

which is a variant over $c^i_{n^i_k}(s)$ as introduced in Auer's work [5]. Again following the logic in the proof of full information case to bound the regret we need to bound the number of times the sub-optimal arms are played. Suppose $k \in \overline{N}_K$ we again have (similar as in the proof of Theorem III.2)

$$T^i_k(t) \leq \zeta + \sum_{s=1}^{\infty} \sum_{n^i_{k^*}=1}^{s-1} \sum_{n^i_k=\zeta}^{s-1} 1\{\tilde{r}^i_{k^*}(n^i_{k^*}) + \hat{c}^i_{n^i_{k^*}}(s) \leq \tilde{r}^i_k(n^i_k) + \hat{c}^i_{n^i_k}(s)\} \ .$$

Again we have that $\tilde{r}^i_{k^*}(n^i_{k^*}) + \hat{c}^i_{n^i_{k^*}}(s) \leq \tilde{r}^i_k(n^i_k) + \hat{c}^i_{n^i_k}(s)$ implies that at least one of the following must hold,

$$\begin{aligned} & r^i_{k^*}(n^i_{k^*}) \leq \mu^i_{k^*} - c^i_{n^i_{k^*}}(s), \ r^i_k(n^i_k) \geq \mu^i_k + c^i_{n^i_k}(s), \\ & \mu^i_{k^*} \leq \mu^i_k + 2c^i_{n^i_k}(s) - \alpha(t) \cdot (\tilde{\beta}^i_{k^*}(s) - \tilde{\beta}^i_k(s)) \cdot \sqrt{\frac{\log s}{n^i_k}} \ . \end{aligned} \tag{B.19}$$

We bound each term as follows.

$$P\{r^i_{k^*}(n^i_{k^*}) \leq \mu^i_{k^*} - c^i_{n^i_{k^*}}(s)\} \leq e^{-2\sqrt{2}^2 \log s} = s^{-4} \ .$$

And similarly $P\{r^i_k(n^i_k) \geq \mu^i_k + c^i_{n^i_k}(s)\} \leq s^{-4}$. Consider bounds on $\tilde{\beta}^i_{k^*}(s)$ and $\tilde{\beta}^i_k(s)$. Before we get into the following proof we will first show w.h.p., $N_K = B_K$, that is the sample

frequency based estimation on the optimal set is the same as the ground-truth. This can be similarly done as in Lemma B.6.

In particular following the proof from Theorem 2,3,4 (full information) we can prove

$$P(n_k^i(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \leq \frac{2}{t^2}.$$

(from which we can see the rationale behind the requirement $\alpha(t) < \sqrt{2}$)

Now consider the minimum over the whole crowd.

$$\begin{aligned}
&P(\min n_k^i(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&= P(\forall i, n_k^i(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&= \prod_i P(n_k^i(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&\leq (\frac{1}{t^2})^{M_i} = \frac{1}{t^{2M_i}} \ .
\end{aligned}$$

Clearly with crowd learning we have a much faster convergence rate. The independence of second equality is due to the relaxation of condition such that we consider the simpliest case when no recommendation information is taken account for. Or if we keep the same error rate $\frac{1}{t^2}$, we could have

$$P(\min n_k^i(t) > \frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2}) \leq \frac{1}{t^2} \ ,$$

where $\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2}$ is a much smaller quantity when M_i is large. Then we know that w.h.p., for $k \in \overline{N}_K$,

$$\tilde{\beta}_k^i(t) \leq \frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t} \ .$$

and for $k^* \in N_K$ we know

$$\begin{aligned}\tilde{\beta}^i_{k^*}(t) &= \frac{1-\sum_{k\in\overline{N}_K}\tilde{\beta}^i_k(t)}{K} \\ &\geq \frac{1-\sum_{k\in\overline{N}_K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t}}{K} \\ &= \frac{1}{K} - \frac{N-K}{K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t}\ .\end{aligned}$$

With this being ready now consider the bound invoking the use of $\tilde{\beta}$:

$$\begin{aligned}&4\left(\sqrt{2}-\alpha(t)(\tilde{\beta}^i_{k^*}(t)-\tilde{\beta}^i_k(t))\right)^2 \\ &\leq 4\left(\sqrt{2}-\alpha(t)(\frac{1}{K}-\frac{N-K}{K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t}-\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t})\right)^2 \\ &= 4\left(\sqrt{2}-\alpha(t)(\frac{1}{K}-\frac{N}{K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t})\right)^2 .\end{aligned}$$

And we would like to minimize above term, which is equivalent to maximizing the following one:

$$\begin{aligned}\max_\alpha \quad & \alpha(t)(\frac{1}{K}-\frac{N}{K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t}) \\ \text{s.t.} \quad & 0\leq\alpha(t)<\sqrt{2}\ .\end{aligned}$$

Notice the tradeoff obviously comes from the two terms $\alpha(t)$ and $\frac{N}{K}\frac{\log t}{M_i(\sqrt{2}-\alpha(t))^2 t}$. Solving above maximization problem directly is hard. We however show a viable solution: let

$$\alpha(t) = \sqrt{2}(1-\gamma)-\sqrt{N}\cdot\varepsilon\ ,$$

with $0<\gamma\leq 1$ being an arbitrarily small constant. Clearly we have $0\leq\alpha(t)<\sqrt{2},\ \forall t$.

Let $\zeta = \lceil \frac{4\left(\sqrt{2}-\frac{\sqrt{2}(1-\gamma)-\sqrt{N}\cdot\varepsilon(t)}{K}\cdot(1-\frac{1}{M_i})\right)^2}{(\Delta^i_k)^2}\log s\rceil$, we have the third term bounded as fol-

lows.

$$
\begin{aligned}
&\mu^i_{k^*} - \mu^i_k - \hat{c}^i_{n^i_{k^*}}(s) - \hat{c}^i_{n^i_k}(s) \\
&= \mu^i_{k^*} - \mu^i_k - 2c^i_{n^i_k}(s) - \alpha \cdot (\tilde{\beta}^i_{k^*}(s) - \tilde{\beta}^i_k(s)) \cdot \sqrt{\frac{\log s}{n^i_k}} \\
&\geq \mu^i_{k^*} - \mu^i_k - \Delta^i_k \geq 0 \, .
\end{aligned}
$$

The rest of the proof follows that used in proving Theorem 2, with the key step being to bound the number of times sub-optimal arms sampled.

B.7 Proof of Theorem III.8

This proof is also similar to the one for Theorem III.7. There are two key differences. First we could again show w.h.p., $N_K = B_K$, that is the sample frequency based estimation on the optimal set is the same as the ground-truth. This can be similarly done as in Lemma B.6. Notice for each user i, when j's top K option set does not align with user i, the sample frequency estimation will not be skewed by much due to the discount factors. So the claim holds in general.

The only subtle difference left is we are not screening out users that are completely

useless. Notice

$$
\begin{aligned}
&P(\min_j n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&= P(\forall j, n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&= \prod_j P(n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&= \prod_{j\in\mathscr{U}^i} P(n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \cdot \prod_{j\in\mathscr{U}_k^i\setminus\mathscr{U}^i} P(n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&\cdot \prod_{j\in \text{ else}} P(n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \\
&\leq (\frac{1}{t^2})^{M^i} \cdot \prod_{j\in\mathscr{U}_k\setminus\mathscr{U}^i} P(n_k^j(t) > \frac{\log t}{(\sqrt{2}-\alpha(t))^2}) \cdot 1 \\
&\leq (\frac{1}{t^2})^{M_k^i} .
\end{aligned}
$$

The rest of the proof follows the one for Theorem III.7, with the only difference being replacing M_i with M_k^i.

B.8 Performance analysis of (W-MF)

In order to bound the performance we need to prove the following aspects. (1) First we want to bound the error of a mis-association, that is we would like to associate users that are indeed coming from the same cluster. (2) Then though the association is correct, we do not want an imbalanced association to happen, i.e, consider two waypoints w_i and w_j that come from same cluster $\mathscr{C}_k$. We would like to see balanced associations for each one of them instead of seeing majority of others from $\mathscr{C}_k$ associates with any one of them (*balance*). (1) is quite clearly needed; to see the necessity of (2), consider the following facts [37]: in order to achieve (sub-) optimality for MF, for a sub-sampled rating matrix $\mathscr{M}$

the following condition must be satisfied:

$$(|\mathscr{M}|-1)\tau \geq c_1 \cdot \tau \cdot \mathbf{rank}(\mathscr{M}) \cdot \sqrt{\frac{|\mathscr{M}|}{\tau}} \cdot \max\{\mu_0 \log \tau, \mu^2 \mathbf{rank}(\mathscr{M}) \sqrt{\frac{|\mathscr{M}|}{\tau}} c_2\} \ ,$$

where c_1, c_2, μ_0, μ are all positive constants. It is straight-forward to check the algebra that above condition does not hold when $|\mathscr{M}|$ is small. In all, an imbalanced association or partition may result in unexpected performance.

Denote the following for each cluster k,

$$\bar{s}_k = \max_{i,j \in \mathscr{C}_k} S_{i,j}, \ \underline{s}_k = \min_{i,j \in \mathscr{C}_k} S_{i,j} \ ,$$

i.e., $\bar{s}_k, \underline{s}_k$ serve as upper and lower bound for in-cluster similarity. Similarly we have

$$\bar{s}_{k,l} = \max_{i \in \mathscr{C}_k, j \in \mathscr{C}_l} S_{i,j}, \ \underline{s}_{k,l} = \min_{i \in \mathscr{C}_k, j \in \mathscr{C}_l} S_{i,j} \ ,$$

i.e., $\bar{s}_{k,l}, \underline{s}_{k,l}$ bound the inter-cluster similarities. We consider the case with tight clustering as has been frequently adopted in literatures of clustering, e.g., [19]. So networks that are from the same cluster are more similar with each other.

Assumption B.1 (Tight Clustering)**.**

$$\underline{s}_i > \bar{s}_{i,j}, \forall i \ \& \ j \neq i. \tag{B.20}$$

(1) Mis-association

We bound the probability of a *mis-association*. We first assume the raw time series data $r_i^*(t)$ can be modeled as an ergodic Markov chain with transition probability matrix P_i on the state space $\mathscr{R}_i$, for otherwise the variation in the time series itself over time can

be arbitrary.[1] Suppose we adopt the cosine similarity measure and we have the following results:

Lemma B.2. *For any user $j \in \mathscr{C}_i$ we bound the probability of mis-association as follows*

$$P(j \notin \mathscr{W}_j) \leq O(\log N \cdot e^{-D\tau}) \ , \tag{B.21}$$

where $D > 0$ is a positive constant.

(2) Balanced association

Consider another user $j \in \mathscr{C}_k$ while j being not selected as Waypoint node. Then

$$j \text{ associates } w_i \in \mathscr{W}_k \text{ if } : \ w_i = \text{argmax}_{l \in \mathscr{W}_k} S_{j,l} \ . \tag{B.22}$$

Lemma B.3. *The waypoints based association process is balanced w.h.p.*

With above results, now suppose we have sampled $C \log N$ waypoints. Denote C_k as the number of waypoints that are from cluster k. Then we have

$$E[C_k] = E[\sum_{i=1}^{C\log N} 1\{i \in \mathscr{C}_k\}] = \sum_{i=1}^{C\log N} E[1\{i \in \mathscr{C}_k\}] = \sum_{i=1}^{C\log N} \rho_k = \rho_k C \log N \ . \tag{B.23}$$

Then we have by Chernoff bound

$$P(|C_k - E[C_k]| \geq \varepsilon \cdot \rho_k C \log N) \geq 1 - e^{-\frac{\varepsilon^2 \rho_i C \log N}{3}} = N^{-\frac{\varepsilon^2 \rho_i C}{3}} \ .$$

Denote these clusters with waypoint being from cluster k as the following $\mathscr{C}_{k,1}, ..., \mathscr{C}_{k,C_k}$. We now want to bound the number of users in each $\mathscr{C}_{k,j}$ that are not from cluster k. Based on the balanced association rule we have the following results (by symmetry we take $\mathscr{C}_{k,1}$

[1] This is not an overly restrictive assumption as the data generally fits well to the Markov model.

for example)

$$
\begin{aligned}
E[|\mathscr{C}_{k,1}|] &= E[\sum_{i=1}^{n_k} 1\{i \in \mathscr{C}_{k,1}\}] \\
&= E[E[\sum_{i=1}^{n_k} 1\{i \in \mathscr{C}_{k,1}\}|n_k]] \\
&= E[\sum_{i=1}^{n_k} E[1\{i \in \mathscr{C}_{k,1}\}|n_k]] \\
&= E[n_k \cdot \frac{1}{C_k}] = \frac{\rho_i N}{|C_k|}.
\end{aligned} \tag{B.24}
$$

Then w.h.p. we know the number of users within each cluster can be bounded at the order of $O(\frac{N}{\log N})$ which is sub-linear. Denote its order as $O(N^z)$, where $0 < z < 1$ is a constant. Also for each of the sub-cluster above we have the probability of mis-association is bounded as follows

$$
\begin{aligned}
&P(j \text{ mis-associates with } k) \leq P(\tilde{S}_{j,k} \geq \max S_{j,\mathscr{W}_j}) \\
&= \prod_{j' \in \mathscr{W}_j} P(\tilde{S}_{j,k} \geq \tilde{S}_{j,j'}) = O((\log N \cdot e^{-\varepsilon\tau})^{N^z}) \ ,
\end{aligned} \tag{B.25}
$$

Then the expected number of mis-associated user becomes

$$
\begin{aligned}
E[\#\text{mis-association}] &= \sum_j P(j \text{ mis-associates with } k) \\
&= O(N(\log N \cdot e^{-\varepsilon\tau})^{N^z}) \ .
\end{aligned}
$$

following which the rank of each sub-matrix $\mathscr{M}_k$ can be bounded by the intrinsic rank of subspace k plus the number of mis-associated user, which is a exponentially diminishing term in N.

B.9 Proof of Lemma B.2

We take cosine similarity for example. The targeted similarity score is equivalent with the following:

$$E[S_{i,j}] = \frac{E[X_i X_j]}{\sqrt{E[X_i^2]} \cdot \sqrt{E[X_j^2]}}.$$

Notice in the above, when $X_i = X_j, a.s.$ we have $E[S_{i,j}] = 1$; while for the other case when X_i and X_j being orthogonal to each other, we have the expected similarity score pinning down to 0.

Denote the following gap: $\delta := \min_{i,j \neq i} |\underline{s}_i - \overline{s}_{i,j}| > 0$. First

$$\begin{aligned}
&P(j \text{ is mis-associated}) \\
&= P(\exists k \neq i,\ j \text{ is mis-associated with } \mathscr{C}_k) \\
&\underset{\leq}{(e1)} \sum_{k \neq i} P(j \text{ is mis-associated with } \mathscr{C}_k) \\
&\underset{\leq}{(e2)} \sum_{k \neq i} P(\exists \hat{k} \in \mathscr{C}_k, \hat{i} \in \mathscr{C}_i, \text{ s.t. } \tilde{S}_{j,\hat{k}} \geq \tilde{S}_{j,\hat{i}}) \\
&\underset{\leq}{(e3)} \sum_{k \neq i} \sum_{\hat{k} \in \mathscr{C}_k} P(\tilde{S}_{j,\hat{k}} \geq \frac{\delta}{2} + S_{j,\hat{k}},\ \tilde{S}_{j,\hat{i}} \leq -\frac{\delta}{2} + S_{j,\hat{i}}) \\
&\underset{\leq}{(e4)} \sum_{k \neq i} \sum_{\hat{k} \in \mathscr{C}_k} P(\tilde{S}_{j,\hat{k}} \geq \frac{\delta}{2} + S_{j,\hat{k}}) + P(\tilde{S}_{j,\hat{i}} \leq -\frac{\delta}{2} + S_{j,\hat{i}}) \ .
\end{aligned} \tag{B.26}$$

(e1) and (e4) are due to union bound. (e2) is due to the equality and tie broker. For (e3) considering the following fact : if neither of the two argument is true we have

$$\tilde{S}_{j,\hat{i}} > -\frac{\delta}{2} + S_{j,\hat{i}} \geq -\frac{\delta}{2} + \underline{s}_i \geq -\frac{\delta}{2} + \delta + \overline{s}_{i,k} \geq \frac{\delta}{2} + S_{j,\hat{k}} > \tilde{S}_{j,\hat{k}} \ ,$$

which is a contradiction. Now bounding each term in the summation of Eqn.(B.26). By

Chernoff bound with n samples

$$P(|\tilde{X}_i^2 - E[X_i^2]| \geq \varepsilon) \leq 2e^{-2\varepsilon^2 n},$$
$$P(|\tilde{X}_j^2 - E[X_j^2]| \geq \varepsilon) \leq 2e^{-2\varepsilon^2 n},$$
$$P(|\widetilde{X_i X_j} - E[X_i X_j]| \geq \varepsilon) \leq 2e^{-2\varepsilon^2 n} .$$

Denote the following shorthand terms

$$\mu_i^2 = E[X_i^2], \mu_j^2 = E[X_j^2], \mu_{i,j} = E[X_i X_j] .$$

Denote the estimation terms as $\varepsilon_1, \varepsilon_2, \varepsilon_3$, then w.h.p.

$$|\varepsilon_1| \leq \varepsilon, |\varepsilon_2| \leq \varepsilon, |\varepsilon_3| \leq \varepsilon . \tag{B.27}$$

Also we have

$$\begin{aligned}
&|\frac{E[\widetilde{X_i X_j}]}{\sqrt{E[\tilde{X}_i^2]} \cdot \sqrt{E[\tilde{X}_j^2]}} - \frac{E[X_i X_j]}{\sqrt{E[X_i^2]}\sqrt{E[X_j^2]}}| \\
&\leq |\frac{(\mu_{i,j} + \varepsilon) \cdot \sqrt{\mu_i^2}\sqrt{\mu_j^2} - \mu_{i,j} \cdot \sqrt{\mu_i^2 - \varepsilon}\sqrt{\mu_j^2 - \varepsilon}}{\sqrt{\mu_i^2}\sqrt{\mu_j^2}\sqrt{\mu_i^2 - \varepsilon}\sqrt{\mu_j^2 - \varepsilon}}| \\
&\leq \frac{\varepsilon}{\sqrt{\mu_i^2 - \varepsilon}\sqrt{\mu_j^2 - \varepsilon}} + \frac{\mu_{i,j}}{\sqrt{\mu_i^2}\sqrt{\mu_j^2}\sqrt{\mu_i^2 - \varepsilon}\sqrt{\mu_j^2 - \varepsilon}} \qquad \text{(B.28)} \\
&\cdot |\sqrt{\mu_i^2}\sqrt{\mu_j^2} - \sqrt{\mu_i^2 + \varepsilon_1}\sqrt{\mu_j^2 + \varepsilon_2}| . \qquad \text{(B.29)}
\end{aligned}$$

Not hard to notice

$$
\begin{aligned}
&|\sqrt{\mu_i^2}\sqrt{\mu_j^2} - \sqrt{\mu_i^2+\varepsilon_1}\sqrt{\mu_j^2+\varepsilon_2}| \\
&\leq \max\{|\sqrt{\mu_i^2}\sqrt{\mu_j^2} - \sqrt{\mu_i^2+\varepsilon}\sqrt{\mu_j^2+\varepsilon}|, \\
&|\sqrt{\mu_i^2}\sqrt{\mu_j^2} - \sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}|\}
\end{aligned}
$$

Not hard to validate $\sqrt{\mu_i^2+x}\sqrt{\mu_j^2+x}$ is convex for $x \geq -\min\{\mu_i^2, \mu_j^2\}$. Denote it as $U(x)$. Then we have

$$
\begin{aligned}
\sqrt{\mu_i^2\mu_j^2} - \sqrt{(\mu_i^2+\varepsilon)(\mu_j^2+\varepsilon)} &\geq -\frac{\partial U(x)}{\partial x}|_{x=\varepsilon} \cdot \varepsilon = \frac{\mu_i^2+\mu_j^2}{\sqrt{\mu_i^2}\sqrt{\mu_j^2}} \cdot \varepsilon \ , \\
\sqrt{(\mu_i^2-\varepsilon)(\mu_j^2-\varepsilon)} - \sqrt{\mu_i^2\mu_j^2} &\geq -\frac{\partial U(x)}{\partial x}|_{x=0} \cdot \varepsilon = \frac{\mu_i^2+\mu_j^2}{2\sqrt{\mu_i^2}\sqrt{\mu_j^2}} \cdot \varepsilon \ .
\end{aligned}
$$

Thus

$$
|\sqrt{\mu_i^2}\sqrt{\mu_j^2} - \sqrt{\mu_i^2+\varepsilon_1}\sqrt{\mu_j^2+\varepsilon_2}| \leq \frac{\mu_i^2+\mu_j^2}{\sqrt{\mu_i^2}\sqrt{\mu_j^2}} \cdot \varepsilon \ .
$$

Choose ε such that

$$
\frac{\varepsilon}{\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} + \frac{\mu_{i,j}(\mu_i^2+\mu_j^2)\varepsilon}{\mu_i^2\mu_j^2\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} < \frac{\delta}{2} \ ,
$$

we have

$$
P(\tilde{S}_{j,\hat{k}} \geq \frac{\delta}{2} + S_{j,\hat{k}}) \leq 2e^{-2\varepsilon^2\tau} \ .
$$

Also notice

$$\sum_{k\neq i}\sum_{\hat{k}\in\mathscr{C}_k} 1 = O(\log N) \ ,$$

and together we proved the results.

Proof of Lemma B.3

We assume within each cluster of users $\mathscr{C}_k$, the similarity information between any two pair of users $S_{i,j}$s are i.i.d. on the range $[\underline{s}_k, \bar{s}_k]$. Then we observe

$$\begin{aligned} P(w_i &= \text{argmax}_{l\in\mathscr{W}_k} S_{j,w_i}) \\ &= P(S_{j,w_i} = \max_{w_l\in\mathscr{W}_k}\{S_{j,w_l}\}) \\ &= \int_x f(x)\cdot F^{n_k-1}(x)dx = \frac{F^{n_k}(x)|_0^1}{n_k} = \frac{1}{n_k} \ . \end{aligned} \tag{B.30}$$

$f(x), F(x)$ are the pdf and cdf for the similarity score between j and others in $\mathscr{C}_k$. The second equality is due to the i.i.d. assumption. With the expectation being bounded, we could very well bound the probability that a imbalanced association happens.

$$\begin{aligned} &P(|\frac{\sum_{j\in\mathscr{W}_k} 1\{j\Rightarrow w_i\}}{n_k} - E[\frac{\sum_{j\in\mathscr{W}_k} 1\{j\Rightarrow w_i\}}{n_k}]| \geq \varepsilon) \\ &= P(|\frac{\sum_{j\in\mathscr{W}_k} 1\{j\Rightarrow w_i\}}{n_k} - \frac{1}{n_k}| \geq \varepsilon) \\ &\leq 2e^{-2\varepsilon^2\cdot n_k} \ , \end{aligned} \tag{B.31}$$

via Chernoff bound.

APPENDIX C

Proofs for Chapter IV

C.1 Example of S

We show $S_{i,j} = 1 - \max_{x \in \mathscr{X}, y \in \mathscr{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2$ while setting $\beta_1 := 2\sum_{y \in \mathscr{Y}} y^2$ and $\beta_2 := 2\sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y} - y)^2$ is a feasible similarity measure according to our definition. For squared loss the optimal predictor is given by the conditional expectation; we thus have the following:

$$
\begin{aligned}
r_i(f_j) &= \sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^j_{x,\hat{y}} \hat{y} - y)^2 \\
&= \sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^j_{x,\hat{y}} \hat{y} - \sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y} + \sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y} - y)^2 \\
&\leq 2 \sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^j_{x,\hat{y}} \hat{y} - \sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y})^2 \\
&\quad + 2 \sum_{x \in \mathscr{X}} \pi^i_x \cdot \sum_{y \in \mathscr{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y} \in \mathscr{Y}} Q^i_{x,\hat{y}} \hat{y} - y)^2 \\
&\leq 2 \sum_{y \in \mathscr{Y}} y^2 \cdot (1 - S_{i,j}) + \beta_2 \, .
\end{aligned}
$$

□

C.2 To complete proof of Theorem IV.1

As we already bind $r_1(\cdot)$ with $\bar{r}^{\text{IID}}_{k(t)}(\cdot)$, we only need to bound $\bar{r}^{\text{IID}}_{k(t)}(\cdot)$ and consider the case with IID data and the expected prediction error at time t when combine with data from sources $k(t)$ for training. Denote $R_{|k(t)|t}(\mathscr{F})$ as the Rademacher complexity of space $\mathscr{F}$ with $|k(t)|t$ samples and f^* the optimal classifier trained on the set of data. Since we only have finite number of samples we first have the following lemma:

Lemma C.1. *With probability being at least* $1-\frac{2}{(|k(t)|t)^2}$,

$$\bar{r}^{IID}_{k(t)}(f^*) \leq \min_{f\in\mathscr{F}} \bar{r}^{IID}_{k(t)}(f) + 8y^* R_{|k(t)|t}(\mathscr{F}) + 8(y^*)^2\sqrt{\frac{\log(|k(t)|t)}{|k(t)|t}}\ .$$

The proof is standard following the VC theory and it can be derived from the results reported in [14]. Further we know from [14], for squared loss function we have $R_{|k(t)|t}(\mathscr{F}) \leq 2\sqrt{\frac{2d}{|k(t)|t}\cdot\log(\frac{2e|k(t)|t}{d})}$. Therefore

$$\begin{aligned}\bar{r}^{\text{IID}}_{k(t)}(f^*) &\leq \min_{f\in\mathscr{F}} \bar{r}^{\text{IID}}_{k(t)}(f) + 8(y^*)^2\sqrt{\frac{\log(|k(t)|t)}{|k(t)|t}} + 16y^*\cdot\sqrt{\frac{2d}{|k(t)|t}\cdot\log(\frac{2e|k(t)|t}{d})}\\ &\approx \min_{f\in\mathscr{F}} r^{\text{IID}}_{k(t)}(f) + 8y^*(2\cdot\sqrt{2d}+y^*)\cdot\sqrt{\frac{\log(|k(t)|t)}{|k(t)|t}}\ ,\end{aligned}$$

when t is sufficiently large (to thus ignore the $\log 2e$ in term $\sqrt{\frac{2d}{|k(t)|t}\cdot\log(\frac{2e|k(t)|t}{d})}$). Then

$$\begin{aligned}\bar{r}^{\text{IID}}_{1}(f^*) \leq\ & 2\cdot\min_{f\in\mathscr{F}} \bar{r}^{\text{IID}}_{k(t)}(f) + 2\beta_2 + \frac{2\beta_1}{|k(t)|}\cdot\sum_{i\in k(t)}(1-s_i)\\ &+ 8y^*(2\cdot\sqrt{2d}+y^*)\cdot\sqrt{\frac{\log(|k(t)|t)}{|k(t)|t}}\ .\end{aligned}$$

The last two terms are clear. In particular the 3rd term is a constant brought in by the disparities between data sources while the 4th term is the bias with finite number of samplings.

Consider the 1st term we have,

$$\min_{f\in\mathscr{F}} \bar{r}^{\mathrm{IID}}_{k(t)}(f) \leq \min_{f\in\mathscr{F}} \frac{\sum_{i\in k(t)} \alpha\cdot[r^{\mathrm{IID}}_i(f_1)+r^{\mathrm{IID}}_1(f)]}{|k(t)|}$$
$$\leq \frac{2\beta_1}{|k(t)|}\cdot\sum_{i\in k(t)}(1-s_i)+2\beta_2+\cdot\min_{f\in\mathscr{F}} r^{\mathrm{IID}}_1(f)\ .$$

Plug back the results we establish the theorem. □

C.3 Proof of Proposition IV.3

Suppose $i\in k^*(t)$ and there exists a $n<t$ such that $i\notin k^*(n)$. First consider the following fact: let $0<\delta<1$ we have

$$|\sqrt{\frac{\log\delta t}{\delta t}}-\sqrt{\frac{\log t}{t}}| = \frac{1}{\sqrt{\log t}+\sqrt{\frac{\log t+\delta}{\delta}}}\cdot\frac{|(1-1/\delta)\log t-\delta|}{\sqrt{t}}\ .$$

Easy to see the first term is strictly decreasing. For the second term since $\sqrt{t}$ is of a higher order compared with $\log t$ we expect this term to be decreasing when t passes certain threshold. Since $i\in k^*(t)$ and $i\notin k^*(n)$ and the fact we proved earlier that the optimal selection is always a continuous group we know $|k^*(n)|<|k^*(t)|$ and denote $\delta:=|k^*(n)|/|k^*(t)|$. Therefore reducing $k^*(t)$ to $k^*(n)$ will return a better strategy for time t : compared with time n, the loss from the term $\sqrt{\frac{\log t}{t}}$ to $\sqrt{\frac{\log\delta\cdot t}{\delta\cdot t}}$ is smaller, while the gain in average similarity is the same. Similar arguments hold for the term $(\lambda^i_2)^t$, which is also strictly decreasing with t. Proved. □

C.4 Proof of Theorem IV.5

In order to prove the results, we analyze the error of mis-calculating $k^*(t)$.

C.4.1 Error in ordering data sources

We have two steps towards calculating $k^*(t)$ in our algorithm we first need to order data sources $\{1,2,...,K\}$ in their similarity to user 1 to invoke the linear search. For simplicity of following analyses we assume $s_1 > s_2 > ... > s_K$, and denote by $\Delta_{\min} = \min_{i,j} |s_i - s_j|$. The error of mis-ordering at time t is bounded by the following event

$$P(\text{mis-ordering at time } t) \leq P(\omega_m(t)) \ ,$$

where

$$\omega_m(t) = \{\exists i : |\tilde{s}_i - s_i| \geq \frac{\Delta_{\min}}{2}\} \ .$$

this is easy to verify : otherwise the inaccurate measurements are not enough to leverage a sub-optimal option.

Then

$$\begin{aligned}
P(\omega_m(t)) &\leq \sum_{i=1}^{K} P(|\tilde{s}_i - s_i| \geq \frac{\Delta_{\min}}{2}) \\
&= \sum_{i \leq K} P(\max_{x,y} ||\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}|^2 - |Q^i_{x,y} - Q^1_{x,y}|^2| \geq \frac{\Delta_{\min}}{2}) \\
&\leq \sum_{i \leq K} \sum_{x} \sum_{y} P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |\tilde{Q}^1_{x,y} - Q^1_{x,y}| \geq \frac{\Delta_{\min}}{4}) \ ,
\end{aligned}$$

where the first inequality is due to union bound and the last inequality comes from the following fact

$$\begin{aligned}
&\left| |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}|^2 - |Q^i_{x,y} - Q^1_{x,y}|^2 \right| \\
&= \left| |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| + |Q^i_{x,y} - Q^1_{x,y}| \right| \cdot \left| |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \right| \\
&\leq 2 \left| |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \right| \ .
\end{aligned}$$

Consider $\left||\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}|\right|$.

$$\begin{aligned}
&|\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \\
&= |\tilde{Q}^i_{x,y} - Q^i_{x,y} + Q^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \\
&\leq |\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |Q^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \\
&\leq |\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |Q^i_{x,y} - \tilde{Q}^1_{x,y} - Q^i_{x,y} + Q^1_{x,y}| \\
&= |\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |Q^1_{x,y} - \tilde{Q}^1_{x,y}| \qquad \text{(C.1)}
\end{aligned}$$

and moreover we have

$$\begin{aligned}
&|\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}| \\
&= |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - \tilde{Q}^1_{x,y} + \tilde{Q}^1_{x,y} - Q^1_{x,y}| \\
&\geq |\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - \tilde{Q}^1_{x,y}| - |\tilde{Q}^1_{x,y} - Q^1_{x,y}| \\
&\geq -|\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y} - Q^i_{x,y} + \tilde{Q}^1_{x,y}| - |\tilde{Q}^1_{x,y} - Q^1_{x,y}| \\
&= -|\tilde{Q}^i_{x,y} - Q^i_{x,y}| - |Q^1_{x,y} - \tilde{Q}^1_{x,y}| \qquad \text{(C.2)}
\end{aligned}$$

From above two inequality we know

$$\left||\tilde{Q}^i_{x,y} - \tilde{Q}^1_{x,y}| - |Q^i_{x,y} - Q^1_{x,y}|\right| \leq |\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |\tilde{Q}^1_{x,y} - Q^1_{x,y}| \qquad \text{(C.3)}$$

Again via union bound we have

$$\begin{aligned}
&P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| + |\tilde{Q}^1_{x,y} - Q^1_{x,y}| \geq \frac{\Delta_{\min}}{4}) \\
&\leq P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| \geq \frac{\Delta_{\min}}{8}) + P(|\tilde{Q}^1_{x,y} - Q^1_{x,y}| \geq \frac{\Delta_{\min}}{8}) \ . \qquad \text{(C.4)}
\end{aligned}$$

Next we prove for each $i \in \mathscr{D}$, $n_{i,x}(t) = O(t)$ w.h.p. We invoke the following result, Theorem 3.3 from [46].

Lemma C.2. *For finite-state, irreducible Markov chain $X_i(t), t = 1,2,...$ with state space $\mathscr{X}^i$ and transition probability P^i, initial distribution q^i and stationary distribution π^i, denote $N_q^i = ||(\frac{q_x}{\pi_x})_x||_2$. Let $\hat{P}^i = P^{i,T} \cdot P^i$ be the multiplicative symmetrization of P^i where $P^{i,T}$ is the adjoint of P^i on $l_2(\pi)$. Let $\kappa = 1 - \lambda_2$ where λ_2 is the second largest eigenvalue of the matrix $\hat{P}^i$. ε is often referred to as the eigenvalue gap of $\hat{P}^i$. Let $f : \mathscr{X}^i \to R$ be such that $\sum_{y \in \mathscr{X}^i} \pi_y^i \cdot f(y) = 0$, $||f||_\infty \leq 1$ and $0 < ||f||_2^2 \leq 1$. For any positive integer n and $0 < \gamma \leq 1$ we have*

$$P\Big(\frac{\sum_{t=1}^n f(X_i(t))}{n} \geq \gamma\Big) \leq N_q \cdot e^{-\frac{n\gamma^2\kappa}{28}} \ . \tag{C.5}$$

Let $f(X_i(t)) = -1\{X_i(t) = x\} + \pi_x^i$. Not hard to verify such f satisfies all conditions required in Lemma C.2. Then we have

$$P(n_{i,x}(t) \leq (\pi_x^i - \gamma)t) \leq N_q \cdot e^{-\frac{t\gamma^2\kappa}{28}} \ , \tag{C.6}$$

which holds specifically for a constant $\gamma < \pi_x^i$.

With above (by Chernoff-Hoeffding bound),

$$P(|\tilde{Q}_{x,y}^i - Q_{x,y}^i| > 2\frac{\Delta_{\min}}{8}) \leq 2e^{-2\frac{\Delta_{\min}^2}{16} \cdot O(t)} \ .$$

Thus

$$P(\omega_m(t)) \leq O(e^{-Ct}) \ , \tag{C.7}$$

for a certain constant C.

C.4.2 Error in finding the best crowd

Denote the estimated error bound as $\tilde{\mathscr{U}}(t)$:

$$\tilde{\mathscr{U}}_{k(t)}(t) := \mathscr{U}_{k(t)}(t; \{\tilde{s}_i\}_{i \in k(t)}) \ ,$$

Now consider we are with correct ordering of all sources. We then further consider the following two events at step t:

$$\omega_1(t) = \{\forall k, |\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| \leq O(\sqrt{\frac{\log t}{t}})\},$$
$$\omega_2(t) = \{\exists k, |\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\sqrt{\frac{\log t}{t}})\}.$$

$\omega_1(t)$ is the event that estimation for $\mathscr{U}_{[k]}(t)$ is bounded by $O(\sqrt{\frac{\log t}{t}})$ from its true value for all k; while $\omega_2(t)$ is its complement set, i.e., we consider two cases regrading the estimation accuracy of the upper bound of prediction performance. Clearly $\omega_1(t) \cap \omega_2(t) = \emptyset$ and $\omega_1(t) \cup \omega_2(t) = \Omega$. Then we have the following

$$r_1(f_{\tilde{k}^*(t)}(t)) = r_1(f_{\tilde{k}^*(t)}(t)|\omega_1(t))P(\omega_1(t)) + r_1(f_{\tilde{k}^*(t)}(t)|\omega_2(t))P(\omega_2(t)) \ .$$

We first bound $P(\omega_2(t))$. First via union bound we have

$$\begin{aligned} &P(\{\exists k, |\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\sqrt{\frac{\log t}{t}})\}) \\ &\quad \leq \sum_{k=1}^{K} P(|\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\sqrt{\frac{\log t}{t}})) \ . \end{aligned}$$

Denote the mean of the top k similarities as $\bar{s}_{[k]} := \frac{\Sigma_{i=1}^{k} s_i}{k}$ and $\widetilde{\bar{s}_{[k]}}$ as its estimated version.

Consider each term in the summation we have by Chernoff-Hoeffding inequality,

$$
\begin{aligned}
&P(|\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\sqrt{\frac{\log t}{t}})) \\
&= P(|\widetilde{\bar{s}_{[k]}} - \bar{s}_{[k]}| > O(\sqrt{\frac{\log t}{t}})) \\
&\leq \sum_{i \in \mathscr{D}} P(|\tilde{s}_i - s_i| > O(\sqrt{\frac{\log t}{t}})) \\
&\leq \sum_{i \in \mathscr{D}} \left(P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| \geq O(\sqrt{\frac{\log t}{t}})) + P(|\tilde{Q}^1_{x,y} - Q^1_{x,y}| \geq O(\sqrt{\frac{\log t}{t}})) \right) ,
\end{aligned}
$$

as we have similarly argued in bounding ordering error. For each of the term above we have (again by Chernoff bound),

$$
P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| > O(\sqrt{\frac{\log t}{t}})) \leq 2e^{-2O(\frac{\log t}{t}) \cdot n_{i,x}(t)} = O(\frac{1}{t^2}) ,
$$

with appropriately chosen constants.

When $\omega_1(t)$ happens we know that the regret from choosing the incorrect maximum $\mathscr{U}(t)$ is bounded at most by $|\tilde{\mathscr{U}}_{\tilde{k}^*(t)} - \mathscr{U}_{k^*(t)}| \leq O(\sqrt{\frac{\log t}{t}})$ since when a sub-optimal set is chosen, its regret is bounded away from its true value by at most $O(\sqrt{\frac{\log t}{t}})$ and so is the optimal set. We thus proved the theorem: to summarize with probability being at least

$$
1 - O(e^{-Ct})(\text{mis-ordering}) - O(\frac{1}{t^2})(\omega_2(t)) = 1 - O(\frac{1}{t^2})
$$

we have

$$
r_1(f_{\tilde{k}^*(t)}(t)) = r_1(f_{\tilde{k}^*(t)}(t) | \omega_1(t), \text{correct ordering}) \leq \mathscr{U}_{k^*(t)} + O(\sqrt{\frac{\log t}{t}}) .
$$

□

C.5 Proof of Theorem IV.6

Most of this Section's proof is similarly to the ones in proving Theorem IV.5, but with limited number of sampling.

C.5.1 Bounding $E[\sum_{n=1}^{t} 1\{\mathscr{O}(n) \neq \emptyset\}]$

We start with bound the exploration errors $E[R_e(t)]$. In order to do so, we first establish the bounded number of exploration phases. Specifically we prove $E[\sum_{n=1}^{t} 1\{\mathscr{O}(n) \neq \emptyset\}] \leq O(t^z)$ w.h.p.. First notice at time t we have for each state i we have most $D(t)$ number of samples from exploration. Denote $\tau_{i,x}(n)$ as the length of regeneration cycle for n-th samples of each state x of user i. Then we have

$$\sum_{n=1}^{t} 1\{\mathscr{O}(n) \neq \emptyset\} \leq \sum_{i\in\mathscr{D}} \sum_{x\in\mathscr{X}} \sum_{n=1}^{D(t)} \tau_{i,x}(n) \ . \tag{C.8}$$

Consider each sum $\sum_{n=1}^{D(t)} \tau_{i,x}(n)$. Notice $\{\tau_{i,x}(n)\}_n$ forms an IID process due to the renewal properties of Markovian processes. And it is known $E[\tau_{i,x}(n)] = \frac{1}{\pi_x^i}$. Then we have for any $0 < \gamma < \frac{1}{\pi_x^i}$ we have by Chernoff-Hoeffding inequality that

$$P(\frac{\sum_{n=1}^{D(t)} \tau_{i,x}(n)}{D(t)} - \frac{1}{\pi_x^i} < -\gamma) \leq e^{-2\gamma^2 D(t)} \ , \tag{C.9}$$

which finishes the proof.

C.5.2 Bounding exploration errors $E[R_e(t)]/t$

Notice with training with data $k^*(t)$, compared to using only user 1's own data, the benefits come from a faster converging term $\sqrt{\frac{\log t}{t}}$, then

$$|\mathscr{U}_{[1]}(t) - \mathscr{U}_{k^*(t)}(t)| \leq O(\sqrt{\frac{\log t}{t}}) \ . \tag{C.10}$$

We then have

$$E[R_e(t)] \leq \sum_{n=1}^{t} 1\{\mathscr{O}(n) \neq \emptyset\} O(\sqrt{\frac{\log n}{n}}) \ . \tag{C.11}$$

Since the number of explorations have been bounded at the order of $O(t^z)$ and combine this with the fact that $\sqrt{\frac{\log t}{t}}$ is a decay function in t in general we have

$$\begin{aligned} E[R_e(t)] &\leq \sum_{n=1}^{O(t^z)} O(\sqrt{\frac{\log n}{n}}) \\ &\leq O(\sqrt{\log t^z} \cdot \sum_{n=1}^{O(t^z)} O(\sqrt{\frac{1}{n}}) \\ &\leq O(\sqrt{z \log t}) \cdot O((t^z)^{1-1/2}) \\ &= O(\sqrt{z \log t} \cdot t^{z/2}) \ , \end{aligned}$$

which gives us

$$\frac{E[R_e(t)]}{t} \leq O(\sqrt{z \log t} \cdot t^{z/2-1}) \ . \tag{C.12}$$

C.5.3 Bounding exploitation error

Now consider the exploitation errors. Again we first separate our discussions according two events.

$$\begin{aligned} \omega_1(t) &= \{\forall k, |\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| \leq O(\sqrt{\frac{\log t}{t^z}})\}, \\ \omega_2(t) &= \{\exists k, |\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\sqrt{\frac{\log t}{t^z}})\} \ , \end{aligned}$$

and

$$r_1(f_{k_1(t)}(t)) = r_1(f_{k_1(t)}(t)|\omega_1(t))P(\omega_1(t)) + r_1(f_{k_1(t)}(t)|\omega_2(t))P(\omega_2(t)) \ .$$

Similarly we could prove that with $O(\log t \cdot t^z)$ number of samples

$$\begin{aligned} P(\omega_2(t)) &\leq \sum_{1\leq k\leq K} P(|\tilde{\mathscr{U}}_{[k]}(t) - \mathscr{U}_{[k]}(t)| > O(\frac{1}{t^{z/2}})) \\ &\leq \sum_{i\in[k]} \sum_{x\in\mathscr{X}, y\in\mathscr{Y}s} P(|\tilde{Q}^i_{x,y} - Q^i_{x,y}| > O(\sqrt{\frac{\log t}{t^z}})) \\ &\leq 2e^{-2O(\frac{\log t}{t^z})\cdot D(t)} = O(\frac{1}{t^2}) \ , \end{aligned}$$

with appropriately selected constants.

Now we can focus on $r_1(f_{k_1(t)}(t)|\omega_1(t))$. When $\omega_1(t)$ happens we know that the regret from choosing the incorrect set of data sources is bounded at most by $|\tilde{\mathscr{U}}_{k(t)} - \tilde{\mathscr{U}}_{k^*(t)}| \leq O(\sqrt{\frac{\log t}{t^z}})$ since when a sub-optimal set is chosen, its regret is bounded away from its true value by at most $O(\sqrt{\frac{\log t}{t^z}})$ and so is the optimal set, i.e.,

$$|\mathscr{U}_{k_1(t)}(t) - \mathscr{U}_{k^*(t)}(t)| \leq O(\sqrt{\frac{\log t}{t^z}}) \ .$$

This observations leads to:

$$r_1(f_{k_1(t)}(t)|\omega_1(t)) \leq \mathscr{U}_{k^*(t)}(t) + O(\sqrt{\frac{\log t}{t^z}}) + |E[\hat{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t)]| \ , \quad \text{(C.13)}$$

where

$$E[\hat{\mathscr{U}}_{k_1(t)}(t)|\omega_1(t)] \leq E[\bar{\mathscr{U}}_{k_1(t)}(t)|\omega_1(t)] + E[e(t)||\omega_1(t)] \ , \quad \text{(C.14)}$$

and

$$\begin{aligned} \bar{\mathscr{U}}_{k_1(t)}(t) = 4 \min_{f\in\mathscr{F}} r_1^{\text{IID}}(f) + 6\beta_2 + 6\beta_1 \frac{\sum_{i\in k_1(t)} n_i(t)(1-s_i)}{N_{k_1(t)}(t)} \\ + \tilde{\rho}_{k_1(t)}(t) + 8y^*(2\sqrt{2d} + y^*) \cdot \sqrt{\frac{\log N_{k_1(t)}(t)}{N_{k_1(t)}(t)}} \ . \quad \text{(C.15)} \end{aligned}$$

Notice the subtle difference between $\hat{\mathscr{U}}_{k_1(t)}(t)$ and $\mathscr{U}_{k_1(t)}(t)$. $\hat{\mathscr{U}}_{k_1(t)}(t)$ is further bounded by two terms: one is $\bar{\mathscr{U}}_{k_1(t)}(t)$, the error bound with sub-sampled data and the other term $e(t)$ corresponds to the effects of dis-continuous samplings.

To make it more clear, we start the discussion by noticing that an incorrect calculation of $k_1(t)$ not only has effects on the prediction at time t, but also affects the learning process in all following steps due to this potential miss of collecting data. For details please refer to the difference between $\bar{\mathscr{U}}_{k(t)}(T)$ and $\mathscr{U}_{k(t)}(t)$: when a wrong decision is made at time t and data from the optimal sources have not been collected, the performance of the prediction for all following stages will suffer from sub-sampling. Denote $\bar{n}_i(t)$ as the number of missed data up to time t for $i \in k_1(t)$ and we first address two questions with sub-sampling for sequentially arriving Markovian data : 1). Under $\omega_1(t)$, is $\bar{n}_i(t)$ bounded above and how? (which affects $\bar{\mathscr{U}}_{k_1(t)}(t)$) 2). due to the dis-continuous sampling, there will be extra bias incurred for the distribution of sampled Markovian data. How to quantify this bias? (which affects $e(t)$).

In the rest of the proof, we show the following.

$$E[\bar{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t)] \leq O(\log t \cdot t^{-2/3}),\ E[e(t)||\omega_1(t)] \leq O(t^{-2/3})\ . \tag{C.16}$$

Under $\omega_1(t)$, we further consider two events defined as follows.

$$\omega_3(t) = \{\forall i \in k_1(t) : \bar{n}_i(t) < t^{\theta})\}\ , \tag{C.17}$$

$$\omega_4(t) = \{\exists i \in k_1(t) : \bar{n}_i(t) \geq t^{\theta})\}\ , \tag{C.18}$$

for a tunable constant $0 < \theta < 1$. Again $\omega_3(t) \cap \omega_4(t) = \emptyset$ and $\omega_3(T) \cap \omega_4(t) = \Omega$. Again we have

$$E[\bar{\mathscr{U}}_{k_1(t)}(t)|\omega_1(t)] = E[\bar{\mathscr{U}}_{k_1(t)}(t)|\omega_1(t),\omega_3(t)]P(\omega_3(t)) + E[\bar{\mathscr{U}}_{k_1(t)}(t)|\omega_1(t),\omega_4(t)]P(\omega_4(t)).$$

We first prove the boundedness of $\omega_4(t)$. Since at any time t, for $i \in k^*(t)$, we know we have $i \in k^*(n), n < t$ except for a constant based on Proposition IV.3, this is true for all $i \in k^*(t)$ and is also true for $i \in k'(t)$ such that having estimated learning error bound $\tilde{\mathscr{U}}_{k'(t)}(t)$ within $[\tilde{\mathscr{U}}_{k^*(t)} - \sqrt{\frac{\log t}{t^z}}, \tilde{\mathscr{U}}_{k^*(t)} + \sqrt{\frac{\log t}{t^z}}]$. Since as can be similarly argued in Proposition IV.3, if $i \in k'(t)$ then $i \in k'(n)$ for $n \leq t$. Then due to the construction of $k_2(t)$ and under $\omega_1(t)$(with appropriately selected constant), we know this also holds for $k_1(t)$. Denote the constant as C. That suggests the fact as long as $\omega_2(n)$ is NOT true, a data sources in the optimal set $k_1(n)$ will not be missed, which suggests the following

$$\begin{aligned} E[\bar{n}_i(t)] &\leq E[\sum_{n=1}^{t} 1\{\omega_2(n)\}] + C \\ &\leq \sum_{n=1}^{t} P(\omega_2(n) + C \\ &\leq \sum_{n=1}^{t} O(\frac{1}{n^2}) + C \\ &\leq C^{`} + C. \end{aligned}$$

Next we prove $E[\bar{n}_i^2(t)]$ is also bounded above.

$$\begin{aligned} E[\bar{n}_i^2(t)] &\leq E[(\sum_{n=1}^{t} 1\{\omega_2(n)\} + C)^2] \\ &= E[(\sum_{n=1}^{t} 1\{\omega_2(n)\})^2] + 2CE[\sum_{n=1}^{t} 1\{\omega_2(n)\}] + C^2 \\ &\leq E[\sum_{n=1}^{t} 1^2\{\omega_2(n)\}] + 2CC^{'} + C^2 \ . \end{aligned}$$

Now consider the square term. First notice

$$E[1\{\omega_2(t_1)\}1\{\omega_2(t_2)\}] \leq E[1\{\omega_2(\max\{t_1,t_2\})\}] \ .$$

We then have

$$\begin{aligned} E[(\sum_{n=1}^{t} 1\{\omega_2(n)\})^2] &\leq 2E[\sum_{n=1}^{t} n1\{\omega_2(n)\}] \\ &\leq 2\sum_{n=1}^{t} nO(\frac{1}{n^2}) \\ &\leq O(\log t) \ . \end{aligned}$$

Therefore we know $\text{var}(\bar{n}_i(t)) \leq O(\log t)$. Then via Chernoff bound we know

$$P(n_i(t) \leq t - t^\theta) = P(\bar{n}_i(t) \geq t^\theta) \leq O(\frac{\log t}{t^{2\theta}}) \ .$$

Now we analyze $|E[\hat{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t), \omega_3(t)]|$. As argued earlier we have

$$\begin{aligned} |E[\hat{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t), \omega_3(t)]| \leq & |E[\bar{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t), \omega_3(t)]| \\ & + E[e(t)|\omega_1(t), \omega_3(t)] \ . \end{aligned}$$

Notice

$$\begin{aligned} & |E[\bar{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t), \omega_3(t)]| \\ & \leq E\Bigg[\underbrace{6\beta_1 \sum_{i \in k^*(t)} (1 - s_i)|\frac{n_i(t)}{N_{k^*(t)}(t)} - \frac{1}{k}|}_{D_1(t)} + \underbrace{|\tilde{\rho}_{k^*(t)}(t) - \rho_{k^*(t)}(t)|}_{D_2(t)} \\ & + \underbrace{|8y^*(2\sqrt{2d} + y^*) \cdot |\sqrt{\frac{\log |k^*(t)|t}{|k^*(t)|t}} - \sqrt{\frac{\log N_{k^*(t)}(t)}{N_{k^*(t)}(t)}}|}_{D_3(t)}\Bigg| \omega_1(t), \omega_3(t)] \ , \end{aligned}$$

We have the following lemma.

Lemma C.3. *We have*

- $E[D_1(t)|\omega_1(t), \omega_3(t)] \leq O(\frac{1}{t^{1-\theta}})$.

- $E[D_2(t)|\omega_1(t),\omega_3(t)]$ *decays exponentially fast.*
- $E[D_3(t)|\omega_1(t),\omega_3(t)] \leq O(\frac{\sqrt{\log t}}{t^{3/2-\theta}})$.
- $E[e(t)|\omega_1(t),\omega_3(t)] \leq O(\frac{1}{t^{1-\theta}})$.

Adding up the terms we have

$$|E[\hat{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t),\omega_3(t)]| \leq O(\frac{\log t}{t^{2\theta}}) + O(\frac{1}{t^{1-\theta}}) + O(\frac{\sqrt{\log t}}{t^{3/2-\theta}}) ,$$

and the optimal upper bound occurs when $2\theta = 1-\theta$ which gives us $\theta^* = 1/3$ and the optimal bound follows as

$$|E[\hat{\mathscr{U}}_{k_1(t)}(t) - \mathscr{U}_{k_1(t)}(t)|\omega_1(t),\omega_3(t)]| \leq O(\frac{\log t}{t^{2/3}}) .$$

□

C.6 Proof for Lemma C.3

C.6.1 Bound on $E[D_1(t)|\omega_1(t),\omega_3(t)]$

Shorthand $|k^*(t)|$ as k. Take the difference between the coefficients for each term $1-s_1$ satisfies the following,

$$\frac{t}{N_{k^*(t)}(t)} - \frac{1}{k} = \frac{kt - N_{k^*(t)}(t)}{k \cdot N_{k^*(t)}(t)} \geq 0 .$$

For all others $i \neq 1$, we have

$$\frac{n_i(t)}{N_{k^*(t)}(t)} - \frac{1}{k} = \frac{kn_i(t) - N_{k^*(t)}(t)}{k \cdot N_{k^*(t)}(t)} .$$

Therefore we have

$$|\frac{\sum_i n_i(t)\cdot(1-s_i)}{N_{k^*(t)}(t)} - \frac{\sum_i(1-s_i)}{k}| \leq \frac{kT - N_{k^*(t)}(t)}{k\cdot N_{k^*(t)}(t)}\cdot(1-s_1) + \sum_{i\geq 2}|\frac{kn_i(t) - N_{k^*(t)}(t)}{k\cdot N_{k^*(t)}(t)}|\cdot(1-s_i)\ . \tag{C.19}$$

Notice under $\omega_3(t)$, $N_{k^*(t)}(t) \geq kt - k\cdot t^\theta$ and we know

$$\frac{kt - N_{k^*(t)}(t)}{k\cdot N_{k^*(t)}(t)}\cdot(1-s_1) \leq \frac{kt^z}{k(kt-k\cdot t^\theta)}\cdot(1-s_1) = \frac{t^\theta}{k(t-t^\theta)}\cdot(1-s_1)\ , \tag{C.20}$$

and moreover

$$\begin{aligned}
&|\frac{kn_i(t) - N_{k^*(t)}(t)}{k\cdot N_{k^*(t)}(t)}|\cdot(1-s_i)\\
&\leq \sum_j \frac{|n_i(t)-n_j(t)|}{k(kt-k\cdot t^\theta)}\cdot(1-s_i)\\
&\leq k\cdot\frac{t^\theta}{k(kt-k\cdot t^\theta)}\cdot(1-s_i)\\
&= \frac{t^\theta}{kt-k\cdot t^\theta}\cdot(1-s_i) = O(\frac{1}{t^{1-\theta}})
\end{aligned} \tag{C.21}$$

here we have used the fact that if $t - t^\theta \leq n_i(t) \leq t$ and $t - t^\theta \leq n_j(t) \leq T$ we must also have $|n_i(t) - n_j(t)| \leq t^\theta$. So is the expectation bounded. Proved . □

C.6.2 Bound on $E[D_2(t)|\omega_1(t),\omega_3(t)]$

This one is fairly simple to prove. Consider each of the difference term.

$$|(\lambda_2^i)^t - (\lambda_2^i)^{n_i(t)}| \leq |(\lambda_2^i)^{n_i(t)}|\ ,$$

since $n_i(t) \leq t$. However

$$|(\lambda_2^i)^{n_i(t)}| \leq (\lambda_2^i)^{t-t^\theta} ,$$

as $n_i(t) \geq t - t^\theta$. Proved. □

C.6.3 Bound on $E[D_3(t)|\omega_1(t), \omega_3(t)]$

Denote function $g(x) := \sqrt{\frac{\log x}{x}}$ and by mean-value theorem we have

$$|g(x-\delta) - g(x)| \leq |\delta| \cdot \max_{y \in [x-\delta, x]} \frac{\partial g(y)}{\partial y} .$$

Notice

$$\frac{\partial g(x)}{\partial x} = \frac{1}{2} \cdot \frac{1}{\sqrt{\frac{\log x}{x}}} \cdot |\frac{1 - \log x}{x^2}| .$$

For $x \geq 3$ we have

$$\frac{\partial g(x)}{\partial x} \approx \frac{1}{2e} \cdot \frac{1}{\sqrt{\frac{\log x}{x}}} \cdot \frac{\log x}{x^2} = \frac{1}{2e} \cdot \sqrt{\frac{\log x}{x^3}} ,$$

Since $\sqrt{\frac{\log x}{x^3}}$ is a strictly decreasing function when $x \geq 3$, we have (under $\omega_3(t)$) the worst case bound is given by

$$\begin{aligned} &|\sqrt{\frac{\log(kt - (k-1)\cdot t^\theta)}{kt - (k-1)\cdot t^\theta}} - \sqrt{\frac{\log kt}{kt}}| \\ &\leq (k-1)\cdot t^\theta \cdot \frac{1}{2e} \cdot \sqrt{\frac{\log(kt - (k-1)\cdot t^\theta)}{(kt - (k-1)\cdot t^\theta)^3}} , \end{aligned}$$

which is decreasing sub-linearly as long as $\theta < 1$. Moreover when t is large enough we have

$$\sqrt{\frac{\log(kt-(k-1)\cdot t^{\theta})}{(kt-(k-1)\cdot t^{\theta})^3}} \leq \sqrt{\frac{\log t}{t^3}} \ ,$$

which leads us to the fact that

$$|\sqrt{\frac{\log(kt-(k-1)\cdot t^{\theta})}{kt-(k-1)\cdot t^{\theta}}} - \sqrt{\frac{\log kt}{kt}}| \leq O(\frac{\sqrt{\log t}}{t^{3/2-\theta}}) \ .$$

So is the expectation bounded. Proved. □

C.6.4 Bound on $E[e(t)|\omega_1(t),\omega_3(t)]$

Due to discontinous sampling of Markovian data (the fact that $n_{i,x}(t) > 0$) the resultant data collection, in the form of its empirical distributions $\tilde{\pi}_x^i$ will be biased. To see this more clearly. Suppose $\tilde{f}$ is trained on a dataset $\tilde{D}$ and f is on D, where $\tilde{D}$ is a biased version of D. Then we have $E[e(t)|\omega_1(t),\omega_3(t)]$ bounded as follows (we omit the conditioning on $\omega_1(t),\omega_3(t)$ for brevity)

$$\begin{aligned} E_{z\sim D}[\mathscr{L}(\tilde{f},z)] - E_{z\sim D}[\mathscr{L}(f,z) &= E_{z\sim\tilde{D}}[\mathscr{L}(\tilde{f},z)] - E_{z\sim\tilde{D}}[\mathscr{L}(f,z)] \\ &+ (E_{z\sim D}[\mathscr{L}(\tilde{f},z)] - E_{z\sim\tilde{D}}[\mathscr{L}(\tilde{f},z)]) + (E_{z\sim D}[\mathscr{L}(f,z)] - E_{z\sim\tilde{D}}[\mathscr{L}(f,z)]) \end{aligned}$$

By definition of $\tilde{f}$ (also optimality) we know $E_{z\sim\tilde{D}}[\mathscr{L}(\tilde{f},z)] - E_{z\sim\tilde{D}}[\mathscr{L}(f,z)] \leq 0$. For $E_{z\sim D}[\mathscr{L}(\tilde{f},z)] - E_{z\sim\tilde{D}}[\mathscr{L}(\tilde{f},z)]$ we have

$$|E_{z\sim D}[\mathscr{L}(\tilde{f},z)] - E_{z\sim\tilde{D}}[\mathscr{L}(\tilde{f},z)]| \leq \max\mathscr{L} \cdot \sum_{x\in\mathscr{X}} E[|\tilde{\pi}_x^i - \pi_x^i|] \ .$$

Notice under $\omega_3(t)$,

$$E[|\tilde{\pi}_x^i - \pi_x^i|] \leq O(\frac{t^\theta}{t-t^\theta}) = O(\frac{1}{t^{1-\theta}}) \ ,$$

where the upper bound comes from the extreme cases all missed samples are from one specific state transition. Therefore we proved

$$|E_{z\sim D}[\mathscr{L}(\tilde{f},z)] - E_{z\sim \tilde{D}}[\mathscr{L}(\tilde{f},z)]| \leq O(\frac{1}{t^{1-\theta}}) \ ,$$

and similar analysis applies to $(E_{z\sim D}[\mathscr{L}(f,z)] - E_{z\sim \tilde{D}}[\mathscr{L}(f,z)])$. Proved. □

C.7 Proof of Theorem IV.7

We now analyze the difference in cost for requesting additional data. There are mainly two sources for this extra cost : 1). first of all, we know the there is unnecessary cost for exploration phases. This is the cost mainly for requesting enough samples to train or learn the similarity information between user 1 and any other users. 2). second is the unnecessary cost at exploitation phases when bad decisions are made (requesting data from a source that is outside the optimal set). More rigorously we have

$$\begin{aligned} E[R_c(t)] &= cE[\sum_{n=1}^{t}\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\}] \\ &= cE[\sum_{n=1}^{t}\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1\{\mathscr{O}(n) \neq \emptyset\}] \\ &\quad + cE[\sum_{n=1}^{t}\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1^c\{\mathscr{O}(n) \neq \emptyset\}] \end{aligned}$$

Consider the first term above we have

$$E[\sum_{n=1}^{t}\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1\{\mathscr{O}(n) \neq \emptyset\}] \leq KE[\sum_{n=1}^{t}\sum_{i=1}^{K} 1^c\{\mathscr{O}(n) \neq \emptyset\}] \leq O(ct^z) \ .$$

The last inequality is due to the fact number of exploration rounds are bounded above by $O(t^z)$.

Now consider the second term. For exploitation phases, consider the possibility of requesting redundant samples. We again decompose our discussion into the cases corresponding to events $\omega_1(t)$ and $\omega_2(t)$ (as defined in the proof for Theorem IV.6). Then

$$\begin{aligned}
&E[\sum_{n=1}^{t}\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1^c\{\mathscr{O}(n) \neq \emptyset\}] \\
&= \sum_{n=1}^{t} E[\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1^c\{\mathscr{O}(n) \neq \emptyset\} | \omega_1(n)] P(\omega_1(n))] \\
&+ \sum_{n=1}^{t} E[\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1^c\{\mathscr{O}(n) \neq \emptyset\} | \omega_2(n)] P(\omega_2(n))] \ .
\end{aligned}$$

As we showed probability for cases $\omega_2(t)$ is bounded above and the cost regret associated with the case is also bounded above:

$$\begin{aligned}
&\sum_{n=1}^{t} E[\sum_{i=1}^{K} 1\{i \notin k^*(n), i \in k_2(n)\} \cdot 1^c\{\mathscr{O}(n) \neq \emptyset\} | \omega_2(n)] P(\omega_2(n))] \\
&\qquad \leq \sum_{n=1}^{t} K \cdot O(\frac{1}{n^2}) = O(1) \ .
\end{aligned}$$

Now consider the case with $\omega_1(t)$. Clearly at time t, if $k_2(t) \subseteq k^*(t)$ there would be no extra cost for redundant data. Consider the case $k^*(t) \subset k_2(t)$. Based on our sampling policy, with bounded probability as long as we have

$$\mathscr{U}_{k_2(t)} > \mathscr{U}_{k^*(t)} + O(\sqrt{\frac{\log t}{t^z}}) \ , \tag{C.22}$$

there will be no error in requesting data from users in the set $k_2(t)$ by observing the follow-

ing fact

$$\begin{aligned}\tilde{\mathscr{U}}_{k_2(t)} &> \mathscr{U}_{k_2(t)} - O(\sqrt{\frac{\log t}{t^z}}) > \mathscr{U}_{k^*(t)} + O(\sqrt{\frac{\log t}{t^z}}) - O(\sqrt{\frac{\log t}{t^z}}) \\ &= \mathscr{U}_{k^*(t)} + O(\sqrt{\frac{\log t}{t^z}}) \geq \tilde{\mathscr{U}}_{k^*(t)} + \sqrt{\frac{\log t}{t^z}} \ ,\end{aligned}$$

with appropriately chosen constants. So is

$$\tilde{\mathscr{U}}^{tr}_{k_2(t)} > \tilde{\mathscr{U}}^{tr}_{k^*(t)} + \sqrt{\frac{\log t}{t^z}} \ .$$

Since $\frac{\sum_{i=1}^{k_2(t)} s_i}{|k_2(t)|} < \frac{\sum_{i=1}^{k^*(t)} s_i}{|k^*(t)|}$ as $k^*(t) \subset k_2(t)$, there exists a constant T_0 such that the Eqn.(C.22) holds (the gain in Term 4 becomes less than the loss in similarity Term 2.). Therefore the cost regret with over-sampling is bounded up by cKT_0 . Again summing up all above we have the bounds for $E[R_c(T)]$. Proved. □

APPENDIX D

Proofs for Chapter V

D.1 Justification of Linear Fitting for Multi-layer Inference

We start by introducing an underlying *factor selection/controller* random variable Z that determines the generation of H from $\{A_i\}$ by acting as a random switch, and consider this simplified model. This is a method commonly employed in Bayesian inference, see e.g., [60]. This model typically comes with the conditional independence assumption $P_z(A_i|H) = P_z(A_i), \forall i \neq z$. The rationale is that H is in general a random matrix due to the unknown nature of its generation. This randomness may be captured by a random variable Z, that controls whether each of the latent variables is active. By doing so, the role of each A_i is completely determined by Z, and not by H if observed; this is the conditional independence assumption given above.

With this simplification, the posterior probability distribution of H after observing $\{A_i\}$ is given by:

$$P(H|\{A_i\}) \sum_{Z=z} P(H, Z=z|\{A_i\}) = \sum_{Z=z} \eta_z \cdot P_z(H|\{A_i\}, Z=z) \ .$$

where $\eta_z = P(Z = z|\{A_i\})$. Based on our assumption on conditional independence, we have:

$$\begin{aligned} P_z(H|\{A_i\}, Z = z) &= \frac{P(H, \{A_i\}, Z = z)}{\sum_{\tilde{H}} P(\tilde{H}, \{A_i\}, Z = z)} \\ &= \frac{P(H)P_z(A_z|H)\prod_{\tilde{z}\neq z} P_z(A_z)}{\sum_{\tilde{H}} P(\tilde{H})P_z(A_z|\tilde{H})\prod_{\tilde{z}\neq z} P_z(A_z)} \\ &= \frac{P(H)P_z(A_z|H)}{P_z(A_z)} \ . \end{aligned} \tag{D.1}$$

Let $\gamma_z = \frac{\eta_z}{P_z(A_z)}$. Then:

$$P(H|\{A_i\}) = \sum \gamma_z P(H) P_z(A_z|H) \ . \tag{D.2}$$

Assuming that the prior on H is uniform, the maximum a-posteriori probability (MAP) estimation rule reduces to:

$$\text{argmax}_H \sum \gamma_z P_z(A_z|H) \ , \tag{D.3}$$

which is also the same as the maximum likelihood problem. Similarly, we can reduce the RHS of the inference problem to:

$$\text{argmax}_H \sum \psi_z P_z(\mathscr{S}_z|H) \ ,$$

with a possibly different number of states z.

In general, solving these problems is hard without assuming the conditional distributions of the observations $\{A_i\}$ and $\{S_j\}$. However, we can make the analysis more tractable by assuming that these matrices follow an isometric Gaussian distribution, as is commonly done in the literature, see e.g., [60], i.e.,

$$P_z(A_z|H) = \mathscr{N}(H, \sigma_z^2 \mathbf{I}), \ P_z(\mathscr{S}_z|H) = \mathscr{N}(H, \sigma_z^2 \mathbf{I}) \ .$$

Denote the objective function in the MAP problem (D.3) as $f(\mathbf{x})$. Take an arbitrary $\mathbf{x} \in \mathbb{R}^n$, and project $\mathbf{x}$ onto the following sphere (here we are using three latent matrices A_1, A_2, A_3 to illustrate):

$$S_\lambda = A_1 + \lambda_2(A_2 - A_1) + \lambda_3(A_3 - A_1) \ , \tag{D.4}$$

where $\lambda_2, \lambda_3 \in R$. Denote $\mathbf{x}^*$ as the projection of $\mathbf{x}$ onto S_λ, and let $\tilde{\mathbf{x}}$ be the orthogonal component. By plugging in the multivariate Gaussian distribution in (D.3), each term A_i appears in an argument of an exponential function. To solve the MAP problem, these terms need to be minimized. Consider for instance the term associated with A_1. We have:

$$(\tilde{\mathbf{x}} + \mathbf{x}^* - A_1)^T(\tilde{\mathbf{x}} + \mathbf{x}^* - A_1) = (\mathbf{x}^* - A_1)^T(\mathbf{x}^* - A_1) + 2\tilde{\mathbf{x}}(\mathbf{x}^* - A_1) + \tilde{\mathbf{x}}^T\tilde{\mathbf{x}} \ . \tag{D.5}$$

Notice that due to orthogonality, $\tilde{\mathbf{x}}(\mathbf{x}^* - A_1) = 0$, and therefore:

$$(\tilde{\mathbf{x}} + \mathbf{x}^* - A_1)^T(\tilde{\mathbf{x}} + \mathbf{x}^* - A_1) \geq (\mathbf{x}^* - A_1)^T(\mathbf{x}^* - A_1) \ . \tag{D.6}$$

The other two terms can be similarly analyzed. We can then conclude that $f(\mathbf{x}^*) \geq f(\mathbf{x})$. It is also easy to show (by geometry or induction) that the optimum must be between A_1, A_2, A_3, i.e., $\lambda_2, \lambda_3 \in [0, 1]$, as $f(\cdot)$ decreases when the distance between $\mathbf{x}$ and A_1, A_2, A_3 increases. In other words, under the assumptions stated earlier, the optimal inferences of H from both sides are given by a linear model.

D.2 Proof of Theorem V.1

Since the per-element error is bounded by δ, by the definition of the Frobenius norm (with the normalization), we have $||\tilde{A}_i - A_i||_F \leq \delta$. For any $(\{\alpha_i\}, \{\beta_j\})$ we have

$$||\tilde{H}_\alpha - \tilde{H}_\beta||_F = ||H_\alpha - H_\beta + N_\alpha - N_\beta||_F \leq ||H_\alpha - H_\beta||_F + ||N_\alpha - N_\beta||_F \ , \tag{D.7}$$

where N_α, N_β are the error terms

$$N_\alpha = \sum_i \alpha_i(\tilde{A}_i - A_i),\ N_\beta = \sum_j \beta_j(\tilde{B}_j - B_j), \tag{D.8}$$

and the inequality is due to the triangle inequality of norm. Repeating the triangle inequality we have

$$\begin{aligned} ||N_\alpha - N_\beta||_F &\leq ||N_\alpha||_F + ||N_\beta||_F \\ &\leq \sum_i \alpha_i ||\tilde{A}_i - A_i||_F + \sum_j \beta_j ||\tilde{B}_i - B_i||_F \\ &\leq \sum_i \alpha_i \cdot \delta + \sum_j \beta_j \cdot \delta = 2\delta\ , \end{aligned} \tag{D.9}$$

i.e., $||\tilde{H}_\alpha - \tilde{H}_\beta||_F - ||H_\alpha - H_\beta||_F \leq 2\delta$. Similarly we can prove

$$||\tilde{H}_\alpha - \tilde{H}_\beta||_F - ||H_\alpha - H_\beta||_F \geq -2\delta\ . \tag{D.10}$$

Therefore, the difference between the two optimal objective values are bounded as follows:

$$\left|||\tilde{H}_\alpha - \tilde{H}_\beta||_F - ||H_\alpha - H_\beta||_F\right| \leq 2\delta\ . \tag{D.11}$$

Then we can further show the following:

$$\left|||\tilde{H}_{\tilde{\alpha}^o} - \tilde{H}_{\tilde{\beta}^o}||_F - ||H_{\alpha^o} - H_{\beta^o}||_F\right| \leq 2\delta\ , \tag{D.12}$$

for otherwise a new optimal solution could be found for either of the optimization problems which is a contradiction.

Suppose $||(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})-(\{\alpha_i^o\},\{\beta_j^o\})||_2 \geq \varepsilon$. Notice we have

$$\begin{aligned}
&|\mathscr{L}(\tilde{H}_{\tilde{\alpha}^o},\tilde{H}_{\tilde{\beta}^o})-\mathscr{L}(H_{\alpha^o},H_{\beta^o})| \\
&= |\mathscr{L}(\tilde{H}_{\tilde{\alpha}^o},\tilde{H}_{\tilde{\beta}^o})-\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})+\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})-\mathscr{L}(H_{\alpha^o},H_{\beta^o})| \\
&\geq |\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})-\mathscr{L}(H_{\alpha^o},H_{\beta^o})|-|\mathscr{L}(\tilde{H}_{\tilde{\alpha}^o},\tilde{H}_{\tilde{\beta}^o})-\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})| \\
&\geq |\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})-\mathscr{L}(H_{\alpha^o},H_{\beta^o})|-2\delta\ , \qquad \text{(D.13)}
\end{aligned}$$

where the last inequality is due to (D.11).

By the sub-gradient inequality we have

$$\begin{aligned}
&|\mathscr{L}(H_{\tilde{\alpha}^o},H_{\tilde{\beta}^o})-\mathscr{L}(H_{\alpha^o},H_{\beta^o})| \\
&\geq \triangledown\mathscr{L}^T|_{(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})}\cdot\Big((\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})-(\{\alpha_i^o\},\{\beta_j^o\})\Big).
\end{aligned}$$

Consider $|\triangledown\mathscr{L}|_{(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})}|$. Denote n_a as the number of matrices As (LHS) and n_b as the number for Ss (RHS). Since $\sum_i \alpha_i = 1$, choose one dummy variable as α_{n_a}. By taking derivative we could easily show

$$\triangledown\mathscr{L}|_{\tilde{\alpha}_i^o} = \frac{||A_i||_F^2\tilde{\alpha}_i^o - ||A_{n_a}||_F^2(1-\sum_{j\neq n_a}\tilde{\alpha}_j^o)}{C_1}\ ,$$

where $C_1 > 0$ is some bounded (from below) constant. Similar observation holds for β_is. Now denote each perturbation in solution as δ^i, that is $\delta_i = \tilde{\alpha}_i^o - \alpha_i^o$; then we have

$$\triangledown\mathscr{L}|_{\tilde{\alpha}_i^o} = \frac{||A_i||_F^2\delta_i + ||A_{n_a}||_F^2\sum_{j\neq n_a}\delta_j}{C_1}\ ,$$

since $\triangledown\mathscr{L}|_{\alpha_i^o} = 0$. Then we have

$$\triangledown\mathscr{L}|_{\tilde{\alpha}_i^o}\cdot\delta_i = \frac{||A_i||_F^2\delta_i^2 + ||A_{n_a}||_F^2\sum_{j\neq n_a}\delta_j\delta_i)}{C_1}\ .$$

Summing up we have

$$\nabla \mathscr{L}^T|_{(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\})} \cdot \Big((\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\}) - (\{\alpha_i^o\},\{\beta_j^o\}) \Big) \\ \geq \frac{\min\{\min_i ||A_i||_F^2, ||S_j||_F^2\} \sum \delta_i^2 + (\sum_j \delta_j)^2 \min\{||A_{n_a}||_F^2, ||S_{n_b}||_F^2\}}{C_1} \geq C\varepsilon^2 \ ,$$

for certain constant $C > 0$.

However, earlier we have shown in (D.12)

$$|\mathscr{L}(\tilde{H}_{\tilde{\alpha}^o}, \tilde{H}_{\tilde{\beta}^o}) - \mathscr{L}(H_{\alpha^o}, H_{\beta^o})| = \qquad \Big| ||\tilde{H}_{\tilde{\alpha}^o} - \tilde{H}_{\tilde{\beta}^o}||_F - ||H_{\alpha^o} - H_{\beta^o}||_F \Big| \leq 2\delta \ .$$

Therefore $\varepsilon \leq ||(\{\tilde{\alpha}_i^o\},\{\tilde{\beta}_j^o\}) - (\{\alpha_i^o\},\{\beta_j^o\})||_2 \leq \sqrt{\frac{2\delta}{C}}$.

D.3 Proof of Lemma V.2

$||\tilde{A}_i - A_i||_F$ can be written as follows.

$$||\tilde{A}_i - A_i||_F = \sqrt{\frac{\sum_{k,l}(\tilde{A}_i(k,l) - A_i(k,l))^2}{N^2}} = \sqrt{\frac{2\sum_{k<l}(\tilde{A}_i(k,l) - A_i(k,l))^2}{N^2}} \ , \qquad \text{(D.14)}$$

where the second equality holds due to the adjacency matrices $\{A_i\}$ being symmetric and with zero diagonal entries. Since

$$(\tilde{A}_i(k,l) - A_i(k,l))^2 = \begin{cases} 0, & \text{w.p. } 1-\varepsilon \\ 1, & \text{w.p. } \varepsilon \ , \end{cases} \qquad \text{(D.15)}$$

we have $E[(\tilde{A}_i(k,l) - A_i(k,l))^2] = \varepsilon \ , k \neq l$. Noting the errors are independent, using the Chernoff-Hoeffding bound we have

$$P(|\frac{\sum_{k<l}(\tilde{A}_i(k,l) - A_i(k,l))^2}{(N^2-N)/2} - \varepsilon| \geq \delta) \leq 2e^{-2\delta^2 \cdot \frac{N^2-N}{2}} \ . \qquad \text{(D.16)}$$

Noting $N^2 \geq \frac{N^2-N}{2}$ (for $N \geq 2$), we have

$$||\tilde{A}_i - A_i||_F \leq \sqrt{2\frac{\sum_{k\neq l}(\tilde{A}_i(k,l) - A_i(k,l))^2}{N^2-N}} = \sqrt{2\frac{\sum_{k<l}(\tilde{A}_i(k,l) - A_i(k,l))^2}{(N^2-N)/2}}\ , \quad \text{(D.17)}$$

and the lemma follows immediately.

D.4 Proof of Theorem V.5

We start with bounding the errors in estimating the similarity measures. The true similarities between i and j are given by

$$S_{i,j} = \frac{2E[R_iR_j]}{E[R_i^2]+E[R_j^2]}\ ,$$

and

$$S_{i,j}^q = \frac{E[R_iR_j]}{\sqrt{E[R_i^2]}\cdot\sqrt{E[R_j^2]}}\ .$$

Let $\mu_i^2 := E[R_i^2]$ and $\mu_{i,j} := E[R_iR_j]$. We have the following.

Lemma D.1. *Denote by $\tilde{S}_{i,j}$ the similarity measure computed using a length-τ time series $r_i^*(t)$ and $r_j^*(t)$, both in steady state. Further assume (R_i,R_j) form a two-dimensional Markov chain. Then*

$$P(|\tilde{S}_{i,j} - S_{i,j}| \geq f_1(\varepsilon)) \leq \sum_{r_i\in\mathscr{R}_i, r_j\in\mathscr{R}_j} N_q\cdot e^{-\frac{\tau\varepsilon^2\cdot\kappa}{28\cdot(|\mathscr{R}_i|\cdot|\mathscr{R}_j|\cdot r_i\cdot r_j)^2}} + \sum_{r_i\in\mathscr{R}_i} N_q\cdot e^{-\frac{\tau\cdot\delta^2\cdot\kappa}{28\cdot(|\mathscr{R}_i|\cdot r_i)^2}}\ , \quad \text{(D.18)}$$

$$P(|\tilde{S}_{i,j}^q - S_{i,j}^q| \geq f_2(\varepsilon)) \leq \sum_{r_i\in\mathscr{R}_i, r_j\in\mathscr{R}_j} N_q\cdot e^{-\frac{\tau\varepsilon^2\cdot\kappa}{28\cdot(|\mathscr{R}_i|\cdot|\mathscr{R}_j|\cdot r_i\cdot r_j)^2}} + \sum_{r_i\in\mathscr{R}_i} N_q\cdot e^{-\frac{\tau\cdot\delta^2\cdot\kappa}{28\cdot(|\mathscr{R}_i|\cdot r_i)^2}}\ , \quad \text{(D.19)}$$

for constant $\kappa > 0$ and $\forall\varepsilon > 0$ and $f_1(\varepsilon), f_2(\varepsilon)$ are both monotonically increasing function in ε and $f_1(0) = f_2(0) = 0$.

Proof. For the analysis we suppress $*$ for each signal and use $r_i(t)$ to denote any of the

aggregate. Define the following two functions g_1, g_2

$$g_1 : (R_i, R_j) \to R_i \cdot R_j \ , g_2 : R_i \to R_i^2 \ . \tag{D.20}$$

We first prove the following.

$$\begin{aligned}
&P(|\tilde{g}_1(R_i,R_j) - E[g_1(R_i,R_j)]| \geq \varepsilon) \\
&= P(|\sum_{r_i \in \mathscr{R}_i, r_j \in \mathscr{R}_j} \frac{\sum_{t=1}^{\tau} I_{r_i,r_j}(r_i(t), r_j(t))}{\tau} \cdot r_i \cdot r_j - \sum_{r_i \in \mathscr{R}_i, r_j \in \mathscr{R}_j} \pi_{r_i,r_j} \cdot r_i \cdot r_j| \geq \varepsilon) \\
&\leq \sum_{r_i \in \mathscr{R}_i, r_j \in \mathscr{R}_j} P(|\frac{\sum_{t=1}^{\tau} I_{r_i,r_j}(r_i(t), r_j(t))}{\tau} - \pi_{r_i,r_j}| \geq \frac{\varepsilon}{|\mathscr{R}_i| \cdot |\mathscr{R}_j| \cdot r_i \cdot r_j}) \\
&\leq \sum_{r_i \in \mathscr{R}_i, r_j \in \mathscr{R}_j} N_q \cdot e^{-\frac{\tau \varepsilon^2 \cdot \kappa}{28 \cdot (|\mathscr{R}_i| \cdot |\mathscr{R}_j| \cdot r_i \cdot r_j)^2}} \ .
\end{aligned} \tag{D.21}$$

where the last inequality is due to Lemma C.2 with

$$f(r_i, r_j) = \frac{\sum_{t=1}^{\tau} I_{r_i,r_j}(r_i(t), r_j(t))}{\tau} - \pi_{r_i,r_j} \ , \tag{D.22}$$

and $I_{r_i,r_j}(r_i(t), r_j(t))$ is the indicator function for any pair of observations (r_i, r_j). The required properties of f can be easily verified. Similarly

$$\begin{aligned}
&P(|\tilde{g}_2(R_i) - E[g_2(R_i)]| \geq \varepsilon) \\
&= P(|\sum_{r_i \in \mathscr{R}_i} \frac{\sum_{t=1}^{\tau} I_{r_i}(r_i(t))}{\tau} \cdot r_i - \sum_{r_i \in \mathscr{R}_i} \pi_{r_i} \cdot r_i| \geq \varepsilon) \\
&\leq \sum_{r_i \in \mathscr{R}_i} P(|\frac{\sum_{t=1}^{\tau} I_{r_i}(r_i(t))}{\tau} - \pi_{r_i}| \geq \frac{\varepsilon}{|\mathscr{R}_i| \cdot r_i}) \\
&\leq \sum_{r_i \in \mathscr{R}_i} N_q \cdot e^{-\frac{\tau \cdot \varepsilon^2 \cdot \kappa}{28 \cdot (|\mathscr{R}_i| \cdot r_i)^2}} \ .
\end{aligned} \tag{D.23}$$

We then have

$$\begin{aligned}
&\left| \frac{E[\widetilde{R_i R_j}]}{E[\tilde{R}_i^2]+E[\tilde{R}_j^2]} - \frac{E[R_i R_j]}{E[R_i^2]+E[R_j^2]} \right| \\
&\leq \left| \frac{(\mu_{i,j}+\varepsilon)\cdot(\mu_i^2+\mu_j^2) - \mu_{i,j}\cdot(\mu_i^2+\mu_j^2+\varepsilon+\varepsilon)}{(\mu_i^2+\mu_j^2)\cdot(\mu_i^2+\mu_j^2-\varepsilon-\varepsilon)} \right| \\
&\leq \frac{\mu_i^2+\mu_j^2+2\mu_{i,j}}{(\mu_i^2+\mu_j^2)\cdot(\mu_i^2+\mu_j^2-2\varepsilon)} \cdot \varepsilon \ , \qquad \text{(D.24)}
\end{aligned}$$

where ε is sufficiently small. Denote

$$f_1(\varepsilon) := \frac{\mu_i^2+\mu_j^2+2\mu_{i,j}}{(\mu_i^2+\mu_j^2)\cdot(\mu_i^2+\mu_j^2-2\varepsilon)} \cdot \varepsilon \ .$$

Similarly for $\tilde{S}_{i,j}^q - S_{i,j}^q$, following the arguments in Proof B.9 and we could show

$$\begin{aligned}
&\left| \frac{E[\widetilde{R_i R_j}]}{\sqrt{E[\tilde{R}_i^2]}\cdot\sqrt{E[\tilde{R}_j^2]}} - \frac{E[R_i R_j]}{\sqrt{E[R_i^2]}\sqrt{E[R_j^2]}} \right| \\
&\leq \frac{\varepsilon}{\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} + \frac{\mu_{i,j}(\mu_i^2+\mu_j^2)\varepsilon}{\mu_i^2\mu_j^2\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} \ .
\end{aligned}$$

Denote

$$f_2(\varepsilon) := \frac{\varepsilon}{\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} + \frac{\mu_{i,j}(\mu_i^2+\mu_j^2)\varepsilon}{\mu_i^2\mu_j^2\sqrt{\mu_i^2-\varepsilon}\sqrt{\mu_j^2-\varepsilon}} \ ,$$

we finish the proof. □

The bias terms introduced by limited sampling can thus be uniformly bounded. Noting that the RHS of both the inequalities D.18 and D.19 approaches 0 when $\tau \to \infty$, and applying the same technique used in Theorem V.1 we can bound the difference in the solutions.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Adams, T. M., and A. B. Nobel (2010), Uniform Convergence of VapnikChervonenkis Classes Under Ergodic Sampling, *The Annals of Probability*, *38*(4), 1345–1367, doi: 10.1214/09-AOP511.

[2] Agrawal, R. (1995), The Continuum-Armed Bandit Problem, *SIAM journal on control and optimization*, *33*(6), 1926–1951.

[3] AMT (2014), AMT dataset, `http://tamaraberg.com/importanceDataset/`.

[4] Anantharam, V., P. Varaiya, and J. Walrand (1986), Asymptotically Efficient Allocation Rules for the Multiarmed Bandit Problem with Multiple Plays Part I: I.I.D. Rewards, Part II: Markovian Rewards, *Tech. Rep. UCB/ERL M86/62*, EECS Department, University of California, Berkeley.

[5] Auer, P., N. Cesa-Bianchi, and P. Fischer (2002), Finite-time Analysis of the Multiarmed Bandit Problem, *Mach. Learn.*, *47*, 235–256, doi:http://dx.doi.org/10.1023/A:1013689704352.

[6] Bishop, C. M., et al. (2007), *Pattern Recognition and Machine Learning*, vol. 1, springer New York.

[7] Bonchev, D., and G. A. Buck (2005), Quantitative Measures of Network Complexity, in *Complexity in Chemistry, Biology, and Ecology*, pp. 191–235, Springer.

[8] Cai, J., E. Candès, and Z. Shen (2010), A Singular Value Thresholding Algorithm for Matrix Completion.

[9] Candès, E. J., and B. Recht (2009), Exact Matrix Completion via Convex Optimization, *Foundations of Computational mathematics*, *9*(6), 717–772.

[10] CBL (2013 - 2014), Composite Blocking List, `http://cbl.abuseat.org/`.

[11] Chandramouli, S. S. (2011), Multi armed bandit problem: some insights.

[12] Chapelle, O., V. Sindhwani, and S. S. Keerthi (2008), Optimization Techniques for Semi-supervised Support Vector Machines, *The Journal of Machine Learning Research*, *9*, 203–233.

[13] Choffnes, D. R., F. E. Bustamante, and Z. Ge (2010), Crowdsourcing Service-level Network Event Monitoring, *SIGCOMM Comput. Commun. Rev.*, *40*(4), 387–398, doi: 10.1145/1851275.1851228.

[14] Crammer, K., M. Kearns, and J. Wortman (2008), Learning from Multiple Sources, *The Journal of Machine Learning Research*, *9*, 1757–1774.

[15] Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009), ImageNet: A Large-Scale Hierarchical Image Database, in *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*.

[16] Dhamdhere, A., and C. Dovrolis (2008), Ten Years in the Evolution of the Internet Ecosystem, in *Proceedings of ACM IMC*, pp. 183–196, ACM, New York, NY, USA, doi:10.1145/1452520.1452543.

[17] DShield (2013 - 2014), DShield, `http://www.dshield.org/`.

[18] Du, N., B. Wu, X. Pei, B. Wang, and L. Xu (2007), Community Detection in Large-scale Social Networks, in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007*, WebKDD/SNA-KDD '07, pp. 16–25, ACM, New York, NY, USA, doi: 10.1145/1348549.1348552.

[19] Eriksson, B., G. Dasarathy, A. Singh, and R. Nowak (2011), Active Clustering: Robust and Efficient Hierarchical Clustering Using Adaptively Selected Similarities, *arXiv preprint arXiv:1102.3887*.

[20] Galesic, M., and D. Barkoczi (2014), Wisdom of Small Crowds for Diverse Real-World Tasks, *Available at SSRN 2484234*.

[21] Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2003), *Bayesian Data Analysis, Second Edition*, 2 ed., Chapman and Hall/CRC.

[22] Ghosh, A., S. Kale, and P. McAfee (2011), Who Moderates the Moderators?: Crowdsourcing Abuse Detection in User-generated Content, in *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pp. 167–176, ACM, New York, NY, USA, doi:10.1145/1993574.1993599.

[23] Goldstein, D. G., R. P. McAfee, and S. Suri (2014), The Wisdom of Smaller, Smarter Crowds, in *Proceedings of the fifteenth ACM conference on Economics and computation*, pp. 471–488, ACM.

[24] Guille, A., H. Hacid, C. Favre, and D. A. Zighed (2013), Information Diffusion in Online Social Networks: A Survey, *ACM SIGMOD Record*, *42*(2), 17–28.

[25] Haklay, M., and P. Weber (2008), Openstreetmap: User-generated Street Maps, *Pervasive Computing, IEEE*, *7*(4), 12–18.

[26] Hao, S., N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser (2009), Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine, in *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, pp. 101–118, USENIX Association, Berkeley, CA, USA.

[27] He, M., L. Yang, J. Zhang, and V. Vittal (2014), A Spatio-temporal Analysis Approach for Short-term Forecast of Wind Farm Generation, *IEEE Trans. Power Syst*, pp. 1–12.

[28] Ho, C.-J., and J. W. Vaughan (2012), Online Task Assignment in Crowdsourcing Markets, in *AAAI'12*.

[29] Holz, T., C. Gorecki, K. Rieck, and F. Freiling (2008), Measuring and Detecting Fast-Flux Service Networks, in *In Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, ISOC.

[30] hpHosts (2013 - 2014), hpHosts for your pretection, `http://hosts-file.net/`.

[31] Hua, G., C. Long, M. Yang, and Y. Gao (2013), Collaborative Active Learning of a Kernel Machine Ensemble for Recognition, in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1209–1216, IEEE.

[32] Hyup Roh, T. (2007), Forecasting the Volatility of Stock Price Index, *Expert Systems with Applications*, *33*(4), 916–922.

[33] Javed, M., and V. Paxon (2013), Detecting Stealthy, Distributed SSH brute-forcing, in *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security*, pp. 85–96, ACM.

[34] Kanich, C., C. Krebich, K. Levchenko, B. Enright, G. Voelker, and V. Paxon (2008), Spamalytics: An empirical analysis of spam marketing conversion, in *Proceedings of the 15th ACM conference on Computer and communications*, pp. 3–14, ACM.

[35] Karger, D. R., S. Oh, and D. Shah (2011), Iterative Learning for Reliable Crowdsourcing Systems, in *Advances in neural information processing systems*, pp. 1953–1961.

[36] Karger, D. R., S. Oh, and D. Shah (2013), Efficient Crowdsourcing for Multi-class Labeling, in *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, pp. 81–92, ACM.

[37] Keshavan, R. H., A. Montanari, and S. Oh (2010), Matrix Completion from a Few Entries, *Information Theory, IEEE Transactions on*, *56*(6), 2980–2998.

[38] Kim, K.-j. (2003), Financial Time Series Forecasting using Support Vector Machines, *Neurocomputing*, *55*(1), 307–319.

[39] KONECT (2014), Movielens 1m network dataset – KONECT.

[40] Konstantinov, M., D. Gu, V. Mehrmann, and P. Petkov (2003), *Perturbation Theory for Matrix Equations*, Studies in Computational Mathematics, Elsevier Science.

[41] Koren, Y. (2008), Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model, in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, ACM.

[42] Koren, Y., R. Bell, and C. Volinsky (2009), Matrix Factorization Techniques for Recommender Systems, *Computer*, (8), 30–37.

[43] Kulis, B., S. Basu, I. Dhillon, and R. Mooney (2009), Semi-supervised Graph Clustering: a Kernel Approach, *Machine learning*, *74*(1), 1–22.

[44] Lai, T. L., and H. Robbins (1985), Asymptotically Efficient Adaptive Allocation Rules, *Advances in Applied Mathematics*, *6*, 4–22.

[45] Langford, J., and T. Zhang (2007), The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information, in *NIPS*.

[46] Lezaud, P. (1998), Chernoff-type Bound for Finite Markov Chains, *The Annals of Applied Probability*, *8*(3), 849–867, doi:10.1214/aoap/1028903453.

[47] Liu, X., and K. Aberer (2013), SoCo: A Social Network Aided Context-aware Recommender System, in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 781–802, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland.

[48] Liu, Y., and M. Liu (2013), Group Learning and Opinion Diffusion in a Broadcast Network, in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 1509–1516, doi:10.1109/Allerton.2013.6736706.

[49] Liu, Y., and M. Liu (2014), Detecting Hidden Propagation Structure and its Application to Analyzing Phishing, in *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pp. 184–190, IEEE.

[50] Liu, Y., and M. Liu (2015), Crowd-Learning: Improving Online Learning Using Crowdsourced Data, submitted.

[51] Liu, Y., and M. Liu (2015), An Online Learning Approach to Improving the Quality of Crowd-Sourcing, in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, pp. 217–230, ACM, New York, NY, USA, doi:10.1145/2745844.2745874.

[52] Liu, Y., and M. Liu (2016), Finding One's Best Crowd: Online Learning By Exploiting Source Similarity, in *AAAI'16, to appear*.

[53] Liu, Y., J. Zhang, M. Liu, M. Karir, and M. Bailey (2014), Enhancing Multi-source Measurement Using Similarity and Inference: A Case Study of Network Security Interdependence, submitted.

[54] Long, C., G. Hua, and A. Kapoor (2013), Active Visual Recognition with Expertise Estimation in Crowdsourcing, in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 3000–3007, IEEE.

[55] Lu, T., D. Pl, and M. Pal (2010), Contextual Multi-Armed Bandits, *Journal of Machine Learning Research*, *9*, 485–492, doi:http://www.jmlr.org/proceedings/papers/v9/lu10a.html.

[56] Massoulié, L., M. I. Ohannessian, and A. Proutière (2015), Greedy-Bayes for Targeted News Dissemination, in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, pp. 285–296, ACM, New York, NY, USA, doi:10.1145/2745844.2745868.

[57] Natarajan, N., I. Dhillon, P. Ravikumar, and A. Tewari (2013), Learning with Noisy Labels, in *Advances in Neural Information Processing Systems*, pp. 1196–1204.

[58] Ng, A. Y., M. I. Jordan, and Y. Weiss (2001), On Spectral Clustering: Analysis and an algorithm, in *NIPS*, pp. 849–856.

[59] OpenBL (2013 - 2014), OpenBL, `http://www.openbl.org/`.

[60] Oselio, B., A. Kulesza, and A. Hero (2014), Multi-layer Graph Analytics for Dynamic Social Networks, *Selected Topics in Signal Processing, IEEE Journal of*, *8*(4), 514–523.

[61] PhishTank (2013 - 2014), PhishTank, `http://www.phishtank.com/`.

[62] Prelec, D. (2004), A Bayesian Truth Serum for Subjective Data, *Science*, *306*(5695), 462–466, doi:10.1126/science.1102081.

[63] Prelec, D., and H. S. Seung (2006), An Algorithm that Finds Truth Even If Most People Are Wrong.

[64] Rea, L. M., and R. A. Parker (2012), *Designing and Conducting Survey Research: A Comprehensive Guide*, John Wiley & Sons.

[65] RIPE (2014), RIPE Routing Information Service (RIS) Raw data Project, `http://www.ripe.net/data-tools/stats/ris/ris-raw-data`.

[66] Rosenthal, J. S. (1995), Convergence Rates for Markov Chains, *SIAM Review*, *37*(3), pp. 387–405.

[67] Russell, B. C., A. Torralba, K. P. Murphy, and W. T. Freeman (2008), LabelMe: A Database and Web-Based Tool for Image Annotation, *Int. J. Comput. Vision*, *77*(1-3), 157–173, doi:10.1007/s11263-007-0090-8.

[68] Sheng, V. S., F. Provost, and P. G. Ipeirotis (2008), Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers, in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, ACM.

[69] SpamCop (2013 - 2014), SpamCop Blocking List, `http://www.spamcop.net/`.

[70] SPAMHAUS (2013 - 2014), The SPAMHAUS project: SBL, XBL, PBL, ZEN Lists, `http://www.spamhaus.org/`.

[71] Spirtes, P., C. Meek, and T. S. Richardson (2013), Causal Inference in the Presence of Latent Variables and Selection Bias, *CoRR*, *abs/1302.4983*.

[72] SURBL (2013 - 2014), SURBL: URL REPUTATION DATA, http://www.surbl.org/.

[73] Tekin, C., and M. Liu (2012), Online Learning of Rested and Restless Bandits, *Information Theory, IEEE Transactions on*, *58*(8), 5588–5611.

[74] Tekin, C., and M. Liu (2013), Online Learning in a Contract Selection Problem, *CoRR*, *abs/1305.3334*.

[75] Tsybakov, A. B. (2008), *Introduction to nonparametric estimation*, Springer Science & Business Media.

[76] UCEPROTECTOR (2013 - 2014), UCEPROTECTOR Network, http://www.uceprotect.net/.

[77] University of Oregon (), Route Views Project, http://www.routeviews.org/.

[78] Vapnik, V. N. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., New York, NY, USA.

[79] Wang, C.-C., S. R. Kulkarni, and H. V. Poor (2005), Bandit Problems with Side Observations, *IEEE Transactions on Automatic Control*, *50*, 338–355.

[80] WPBL (2013 - 2014), WPBL: Weighted Private Block List, http://www.wpbl.info/.

[81] Xu, K. S., M. Kliger, and A. Hero (2010), Tracking Communities of Spammers by Evolutionary Clustering , *ICML*.

[82] Zhang, J., A. Chivukula, M. Bailey, M. Karir, and M. Liu (2013), Characterization of Blacklists and Tainted Network Traffic, in *Proceedings of PAM*, Hong Kong.

[83] Zhong, E., W. Fan, and Q. Yang (2012), Contextual Collaborative Filtering via Hierarchical Matrix Factorization, in *SDM'12*, pp. 744–755.

[84] Zhu, X., and A. B. Goldberg (2009), *Introduction to Semi-Supervised Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers.