The Dissertation Committee for Hyun Joon Jung
certifies that this is the approved version of the following dissertation:

Temporal Modeling of Crowd Work Quality for Quality Assurance in Crowdsourcing

Committee:

Matthew Lease, Supervisor

Paul Bennett

Ken Fleischmann

Raymond Mooney

Byron C. Wallace

Temporal Modeling of Crowd Work Quality for Quality Assurance in Crowdsourcing

by

Hyun Joon Jung, B.S.; M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2015

Dedicated to my wife, KyungHee, and lovely kids, SeoJin and SeoYun.

Acknowledgments

I would begin my acknowledgments with a deep sense of gratitude for Professor Matthew Lease. I thank him for providing me this great opportunity of being his student. Under him, I have learnt to first walk and then run among the deep and unknown jungles of research in crowdsourcing and information retrieval. I have no words to express my thanks to him for making me the researcher I am today.

I also thank my colleagues, Hohyon Ryu, Aashish Sheshadri, Donna Vakharia, Ivan Oropeza, Haofeng Zhou, and Yubin Park, for their helpful advice and support. I would like to express my gratitude to my committee members, Dr. Paul Bennett, Prof. Ken Fleischmann, Prof. Raymond Mooney, Prof. Byron C. Wallace and Prof. James Howison. Thanks to their guidance, my ideas and passions have come to fruition.

Finally, no one has provided more support, encouragement, and love than my wife, KyungHee, and my lovely kids, SeoJin and Seoyun. They encouraged and motivated me over my graduate years, and they have been a ray of my hope. I take the opportunity to thank my parents who have gone through several hardships to educate me. Words cannot say enough to thank them for all they have done and continue to do.

Temporal Modeling of Crowd Work Quality for Quality Assurance in Crowdsourcing

Publication No. _____

Hyun Joon Jung, Ph.D.
The University of Texas at Austin, 2015

Supervisor: Matthew Lease

While crowdsourcing offers potential traction on data collection at scale, it also poses new and significant quality concerns. Beyond the obvious issue of any new methodology being untested and often suffering initial growing pains, crowdsourcing has faced a very particular criticism since its inception: given anonymity of crowd workers, it is questionable whether we can trust their contributions as much as work completed by trusted workers. To relieve this concern, recent studies have proposed a variety of methods. However, while temporal behavioral patterns can be discerned to underlie real crowd work, prior studies have typically modeled worker performance under an assumption that a sequence of model variables is independent and identically distributed (i.i.d).

This dissertation focuses on the measurement and prediction of crowd work quality by considering its temporal properties. To better model such

temporal worker behavior, we present a time-series prediction model for crowd work quality. This model captures and summarizes past worker label quality, enabling us to better predict the quality of each worker's next label. Furthermore, we propose a crowd assessor model for predicting crowd work quality more accurately. By taking account of multi-dimensional features of a crowd assessor, we aim to build a better quality prediction model of crowd work. Finally, this dissertation explores how the proposed prediction models work under realistic scenarios. In particular, we consider a realistic use case in which limited gold labels are provided for learning our proposed model. For this problem, we leverage instance weighting with soft labels, which takes account of uncertainty of each training instance. Our empirical evaluation with synthetic datasets and a public crowdsourcing dataset has shown that our proposed models significantly improve prediction quality of crowd work as well as lead to an acquisition of better quality labels in crowdsourcing.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Jeff Howe, in an article for Wired magazine in 2006, coined the term "crowdsourcing". It has become a popular method to solve challenging and complex problems for computing systems by "outsourcing the task to a large group of people in the form of an open call" (Howe, 2008). Crowdsourcing offers potentially significant benefits for increasing scalability by collecting human labels from a globally-distributed online crowd via the Internet (Alonso et al., 2008). The crowd is always available, the cost of paying for labels appears to be relatively cheaper than that incurred by traditional labor practices, and similar to *Cloud Computing*'s provision of computing resources, the workforce can be engaged only when needed and scaled elastically on demand (Alonso and Baeza-Yates, 2011; Alonso and Mizzaro, 2009). Commercial crowdsourcing platforms such as Amazon's Mechanical Turk (MTurk)[1], and CrowdFlower[2] provide established infrastructure for engaging the online crowd to perform arbitrary tasks, thereby decreasing barriers to entry for researchers to collect human labels from the crowd. In sum, crowdsourcing has changed data collection practices in both industry and data-driven research areas such

[1] https://mturk.com
[2] http://crowdflower.com

as natural language processing (Snow et al., 2008), computer vision (Vijayanarasimhan and Grauman, 2011), and information retrieval (Law et al., 2011).

1.1 Quality Issues in Crowdsourcing

While crowdsourcing offers potential traction on data collection at scale, it also poses new and significant quality concerns. Beyond the obvious issue of any new methodology being untested and often suffering initial growing pains, crowdsourcing has faced a very particular criticism since its inception: given anonymity of crowd workers, it is questionable whether we can trust their contributions as much as work completed by trusted workers. It is also not trivial to measure crowd workers' quality with a simple metric. Moreover, beyond this simple scalability vs. reliability dichotomy, there are other potentially significant changes in data collection practice to be wrestled with. Unlike typical experiment participants and traditional workforces, crowd workers often perform only a small amount of work and then leave (Jung and Lease, 2012; Sheshadri and Lease, 2013b). Such differences give rise to data sparsity and uncertainty in crowdsourced labels. Finally, rudimentary crowdsourcing platforms like Amazon's Mechanical Turk (MTurk) limit interactions between task requesters and crowd workers. Limited search capabilities provided by current crowdsourcing platforms may not allow crowd workers to find the interesting tasks, potentially leading to decreased label quality (Chilton et al., 2010).

Given the breath of issues involved in crowdsourcing, researchers have

proposed various quality control methods, categorized here into four different methods: label aggregation, task design, workflow design, and worker management (recruitment and retention). Firstly, label aggregation methods minimize the effect of errors in labels by statistical machine-learning techniques (Snow et al., 2008; Sheng et al., 2008; Whitehill et al., 2009; Dekel and Shamir, 2009; Ipeirotis et al., 2010; Welinder et al., 2010; Raykar et al., 2010). These are widely used for label collection tasks such as relevance judgments and record matching. A basic assumption behind these methods is that multiple labeling is better than single labeling. Multiple labeling obtains multiple labels per example from more than one worker and can minimize the effect of a single worker's error (Sheng et al., 2008). Secondly, task design methods assume that a better designed task can improve the quality of crowdsourcing results (Shaw et al., 2011; Alonso, 2012; Kulkarni et al., 2012; Kazai et al., 2012a). Task design methods concentrate on developing best practices by taking into account multiple factors such as pay and task difficulty, which may influence the quality of crowdsourcing. Thirdly, workflow design methods differ from task design methods in how they attempt to change the overall structure of a crowdsourcing pipeline. While task design methods focus on an individual task design, workflow design methods aim to develop an operational workflow which is composed of a set of tasks (Bernstein et al., 2010; Little et al., 2010; Weld et al., 2011b; Kittur et al., 2011; Noronha et al., 2011; Lin and Weld, 2012; Kittur et al., 2012). Finally, worker management methods focus on the understanding of some characteristics of crowd workers rather than their out-

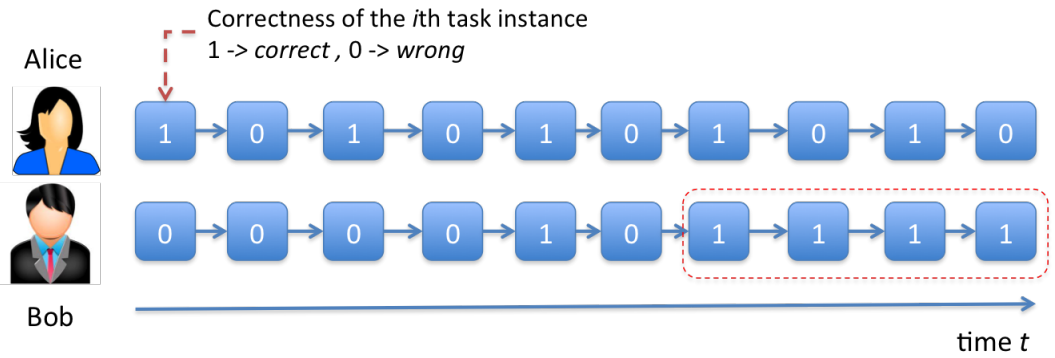

Figure 1.1: Example of two crowd workers showing the same label accuracy but different temporal patterns. While both Alice and Bob show the same accuracies of label quality, the correctness of Alice's label continuously changes but Bob's label correctness gradually improves over time. Considering the temporal dependencies of label correctness raises a difference issue of measuring crowd work quality as well as brings a potential opportunity of improving crowd work quality.

puts (Mason and Watts, 2009; Mason and Suri, 2012; Rzeszotarski and Kittur, 2011; Kazai et al., 2012b; Kaufmann et al., 2011; Dow et al., 2012; Eickhoff and Vries, 2012). Many methods in this category are strongly related to behavioral analysis, which investigate crowd workers' behavior principles on given tasks.

1.2 Research Scope and Methodology

Most prior studies assume that crowd workers' label quality is independent and identically distributed (i.i.d.). Hence, this thesis focuses on predicting crowd work quality by considering temporal aspects of crowd labels. Unlike most existing studies, the proposed algorithms consider temporal dependencies of crowd work quality. For instance, suppose two crowd workers showing

the same level of label annotation performance in **Figure 1.1**. Two workers' label accuracies are equivalent (0.5) to each other under the i.i.d. assumption. However, the temporal patterns of label quality of these two workers are quite different. While Alice's label correctness oscillates over time, that of Bob is gradually improved. Based on this recognition of underlying temporal patterns, we may consider Bob to be a better worker than Alice even though the two workers show the same level of label accuracy.

In reality, a worker may become tired, bored, or begin multi-tasking, leading to decreased work quality. Alternatively, work quality may improve as a worker's experience with a given task accumulates (Carterette and Soboroff, 2010). Although temporal effects are clearly evident in both cases, prior work in modeling crowd workers' performance have typically not considered such effects. Considering the temporal dependencies of label correctness raises a difference issue of measuring crowd work quality as well as brings a potential opportunity of improving crowd work quality. Based on this understanding of a gap in prior work, the thesis of this dissertation is that *better modeling of latent temporal factors behind crowd work enables more accurate measurement and prediction of crowd work quality and leads to better quality of label acquisition in crowdsourcing.*

For empirical validation of this hypothesis, the research scope of this study is concretized with the following questions:

RQ1: Adaptation of a time-series model to crowd work. How can we

measure and predict crowd work quality by taking account of underlying temporal aspects of crowd work quality? To what extent and why does a time-series model yield benefits in terms of measurement and prediction of crowd workers' next label quality? (Chapter 3)

RQ2: Discriminative Predictive model for crowd assessor accuracy. How can we formulate a discriminative, feature-based learning model for predicting crowd work quality? What features would be useful to include and what is their relative importance? (Chapter 4)

RQ3: Temporal Modeling with limited supervision. How can we best use limited gold labels for model training? When do the different methods considered perform better and why? (Chapter 5)

Prior to answering these questions, we first review the basics of crowdsourcing and its diverse use-cases (see in Chapter 2). In addition, we review state-of-the-art crowdsourcing techniques for quality assurance. In particular, we discuss four types of quality control methods based on how to alleviate quality concerns.

Next, for the investigation of the above research questions, we present three quality control methods based on a time-series model (See in **Figure 1.2** and **Figure 1.3**). Subsequently, Chapter 4 presents a discriminative model by considering multi-dimensional features about crowd workers for improving the quality of prediction (RQ2). Finally, Chapter 5 discusses an issue of learning a prediction model with limited supervision (RQ3).

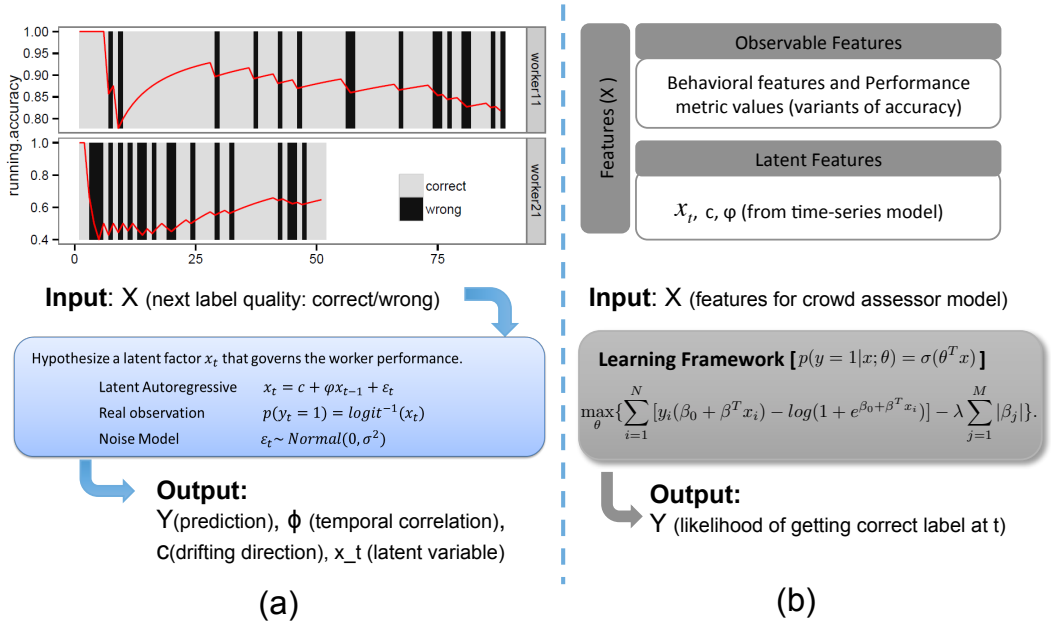

Figure 1.2: Overview of our time-series crowd model and discriminative model. (a) indicates a time-series model of crowd work for predicting next label correctness. (b) indicates a generalizable assessor model for better predictive power in crowdsourced relevance judgments.

The first proposed model in Chapter 3 focuses on latent dynamics of crowd work quality over time. In particular, we examine how to measure crowd worker's label correctness and then predict it (See in **Figure 1.2** (a)). While prior work in measuring crowd workers' annotation performance has typically assumed behavior of crowd workers to be independent and identically distributed (i.i.d.) over time (Yuen et al., 2012; Yi et al., 2013), we shed light on temporal dependencies between crowd workers' label correctness over time by proposing a time-series model of crowd work.

While the first proposed model in Chapter 3 motivates our time-series model for crowdsourcing, it may be not sufficient to build an accurate prediction model for high quality label acquisition. Hence, the second study in Chapter 4 aims at modeling crowd assessor accuracy with an expansion of the first proposed model. We examine a discriminative prediction model for predicting assessor accuracy based on multi-dimensional annotation-related features. While prior studies in modeling assessor behavior have relied upon a single generative model without considering a more flexible learning framework, we build a generalizable feature-based assessor model that allows us to capture a wider range of assessor behaviors by incorporating features which model different aspects of this behavior. Based on the model, we adopt a learning framework for the prediction of assessor accuracy over time (see **Figure 1.2 (b)**).

The proposed prediction models in Chapter 3 and 4 are shown to bring quality improvement in predicting crowd label correctness. However, these

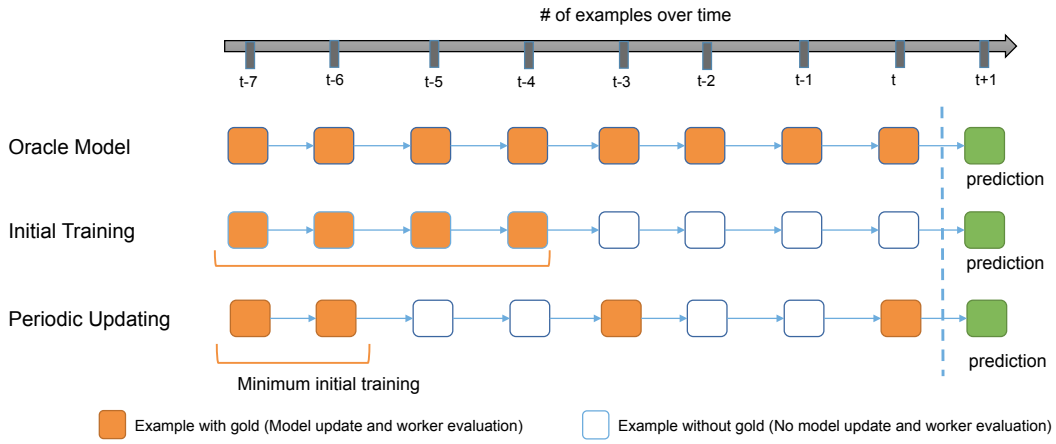

Figure 1.3: Overview of learning temporal prediction models with limited supervision.

two models still assume a training regime in which each crowd label has a corresponding gold label in order to judge whether the crowd label is correct or not. In reality, this assumption is not realistic. A fundamental reason for collecting labels is not already having them. Hence, our last study in Chapter 5 discusses how to best learn our prediction model when supervision is more realistically limited. Intuitively, if only a smaller sample of gold questions is available in order to check worker correctness, our estimate of worker accuracy will have larger variance (i.e., increased uncertainty). To address this, Chapter 5 explores how to maximally utilize limited gold labels and how to update a prediction model in the absence of gold labels. In particular, this study investigates two methods of utilizing limited gold labels. The first method, *initial training* (INIT), uses all of the given gold labels to estimate a worker's label correctness in the initial phase. A second, alternative approach, *periodic updating* (PER), uses gold labels to check label correctness periodically.

The key insight in *periodic updating* is that a worker's temporal performance may be non-stationary (i.e. exhibiting varying correctness over time), which may limit the effectiveness of training the model only on the worker's initial temporal patterns.

In sum, this dissertation presents new methods to better measure and predict the label correctness of each individual crowd worker by modeling latent temporal dynamics. The first study provides answers to RQ1 by modeling crowd work from a temporal perspective. To answer RQ2, the second study strengthens the predictive power of our time-series model by integrating multiple aspects of crowd assessor behaviors. The last study answers to RQ3, how to learn the proposed temporal model with limited supervision.

Chapter 2

Background

In this section, we review the basics of crowdsourcing, applications, and capabilities. In particular, we review state-of-the-art crowdsourcing techniques for quality assurance. Four types of quality control methods are discussed.

2.1 Introduction of Crowdsourcing

While computing systems have effectively automated many routine information processing tasks, computers are still unable to accurately solve a variety of "AI-hard" tasks such as understanding image contents or the meaning of texts. Such tasks often require some extent of human involvement (Chi and Bernstein, 2012). In this regard, "human computation" provides a new way to solve these tasks by outsourcing certain steps to humans. Luis von Ahn and his colleagues pioneered an impressive example of solving complicated problems which engage with human computation (Ahn, 2005).

Jeff Howe, in an article for Wired magazine in 2006, coined the term crowdsourcing. It has become a popular method to solve challenging problems by "outsourcing the task to a large group of people in the form of an open call" (Howe, 2008). Crowdsourcing facilitates the scalability and distribution

of human computation (Quinn and Bederson, 2011). For example, a number of web-based platforms such as Amazon Mechanical Turk (MTurk) and CrowdFlower now enable researchers to conduct paid "crowd work" at a low cost and on a large scale.

One great example of crowdsourcing is the *reCAPTCHA* project. The reCAPTCHA project transforms un-digitalized books to digitalized ones by asking multiple users to write out given words from their texts. Collected words are used to digitize scanned books while the words are used to verify whether users are real human beings (Ahn et al., 2008). Another of the most popular crowdsourcing examples is *Wikipedia*, which builds a large scale online encyclopedia by the voluntary support of hundreds of thousands of online users (Giles, 2005).

The advent of Amazon Mechanical Turk (MTurk) has accelerated a particular type of crowdsourcing: paid crowd work. MTurk provides an integrated online marketplace for crowd workers and task requesters. As shown in **Figure 2.1**, anyone in MTurk who wants to solve any problem can be a task requester. A task requester can design a task, known as a HIT (Human Intelligence Task) in MTurk, and then post it to the crowdsourcing platform. At the same time, anyone who wants to make money by solving tasks can be a crowd worker who selects one of the available tasks and completes it. The task requester pays the crowd worker for the effort. Finally, the task requester collects the results generated by crowd workers and uses them for various tasks such as image labeling (Whitehill et al., 2009; Welinder et al., 2010;

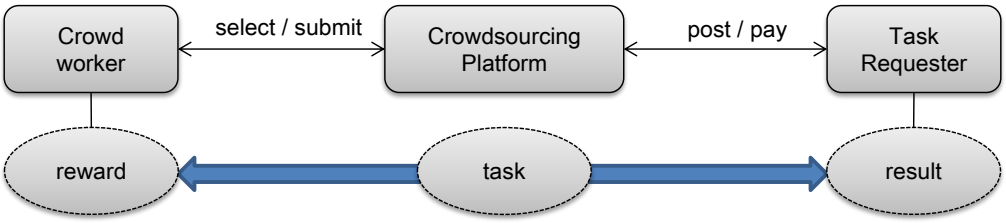

Figure 2.1: Overview of an interaction known as "crowd work". As a task requester, anyone can access an on-demand large workforce via several crowdsourcing platforms such as MTurk and quickly distribute tasks to the crowd. As a crowd worker, anyone chooses one of the available tasks and completes it. Workers are paid to complete the task. Finally, task requesters collect data from many crowd workers at a relatively low cost and large scale.

Kapoor et al., 2010; Vijayanarasimhan and Grauman, 2011), speech annotation (Callison-Burch and Dredze, 2010; Novotney and Callison-Burch, 2010; Jha et al., 2010; Voyer et al., 2010), search relevance assessment (Alonso and Mizzaro, 2012, 2009; Kazai et al., 2009; Kazai, 2011; Kazai et al., 2012b), and behavioral studies (Ross et al., 2010; Barbier et al., 2012; Kaufmann et al., 2011; Mason and Suri, 2012).

2.2 Benefits of Crowdsourcing

Given its benefits in time and cost savings, crowdsourcing has enabled a wide variety of research across multiple domains. It enables researchers to perform large-scale studies at a low cost; this helps overcome the scalability issues in data collection. In addition, these advantages potentially reduce sampling bias by increasing sample size. In the literature, typical scenarios of

using crowdsourcing for research can be roughly divided into three categories: label collection, creative problem solving, and behavioral studies. These are further described in turn below.

Label Collection. Label collection gathers labels for training or testing statistical AI models that are widely used in natural language processing (NLP), computer vision, and information retrieval (IR). For many applications in these fields, producing large-scale training, validation, and test sets is critical, though very time-consuming, expensive, and tedious. Crowdsourcing offers a great alternative to solving this problem. In NLP, researchers typically use MTurk for building a training and testing corpora, speech transcription, or annotation (Callison-Burch and Dredze, 2010; Novotney and Callison-Burch, 2010; Jha et al., 2010; Voyer et al., 2010). Moreover, crowdsourced labels are used for various tasks in computer vision such as image categorization and object recognition (Welinder et al., 2010; Welinder and Perona, 2010b; Kapoor et al., 2010; Vijayanarasimhan and Grauman, 2011). In IR, crowdsourcing has become an alternative for building test collections by collecting relevance judgments from crowd workers (Alonso and Mizzaro, 2012, 2009; Kazai et al., 2009; Kazai, 2011; Kazai et al., 2012b). Furthermore, it is also used to train models to rank documents (Law et al., 2010), identify popular blogs (McCreadie et al., 2012), and find topic experts in microblogs (Ghosh et al., 2012).

Creative Problem Solving. The second category of crowdsourcing research is the solving of complex problems. These might include creative drawing and planning. Numerous researchers in Human Computer Interaction

(HCI) and Computer Supported Cooperative Work (CSCW) report several examples of crowdsourcing for creative problem solving[1]. One such example is the *Sheep Market* that is a collection of 10,000 sheep images created by crowd workers[2]. Another example is a collaborative planning system called Mobie, enabling us to plan an itinerary by cooperative efforts of multiple crowd workers (Zhang et al., 2012). Moreover, crowdsourcing can be used as an alternative to other types of complex problems such as editing or correcting documents (Bernstein et al., 2010). Liu et al. (Liu et al., 2012) studied the use of crowdsourcing for usability testing by comparing it with traditional laboratory-based usability testing.

Behavioral Studies. Ross et al. investigated the temporal variations of the demographics of crowd workers (Ross et al., 2010). Suri and Mason studied the behavior of crowd workers compared with that of experts and human subjects in laboratory experiments (Mason and Suri, 2012). Kaufmann and Shulze investigated the motivation of crowd workers in crowdsourcing (Kaufmann et al., 2011).

2.3 Quality Issues in Crowdsourcing

While crowdsourcing provides significant benefits in terms of time and cost savings, many potential adopters worry about the quality of its results. Hence, a great deal of research has investigated effective quality control meth-

[1]Crowd Camp. http://crowdresearch.org/crowdcamp/
[2]The sheep market. http://www.thesheepmarket.com

ods for crowdsourcing (Lease, 2011).

Various methods can be applied in every stage of a crowdsourcing pipeline (see **Figure 2.2**). From qualification tests to worker retention, each stage in the pipeline involves quality control. These methods are largely divided into four categories: label-aggregation methods, task design methods, workflow design methods, and worker management (recruitment & retention). In the following subsection, we review details of these methods.

2.3.1 Label Aggregation for Quality Control

Label aggregation methods minimize the effect of errors in labels by statistical machine-learning techniques (Dawid and Skene, 1979; Snow et al., 2008; Sheng et al., 2008; Whitehill et al., 2009; Dekel and Shamir, 2009; Ipeirotis et al., 2010; Welinder et al., 2010; Raykar et al., 2010). These are widely used for labeling tasks such as relevance judging and record matching. A basic assumption behind these methods is that multiple labeling is better than single labeling. Multiple labeling obtains multiple labels per example from more than one worker to minimize the effect of a single worker's error (Sheng et al., 2008). Thus, the goal of this approach is to maximize the quality of labels by using statistical machine-learning models. One shortcoming is that it only exploits worker outputs, without seeking to improve the workers' actual performance.

Most studies in label aggregation methods apply several statistical models to minimize the effect of noisy labels. Sheng et al. investigated the usage of

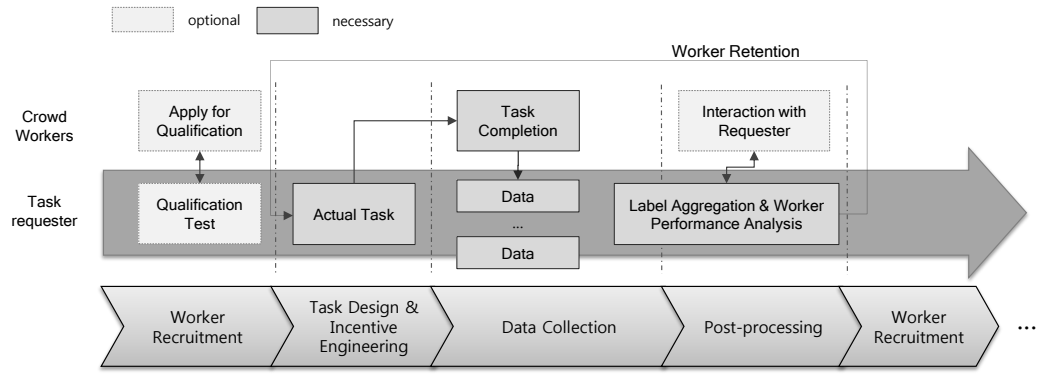

Figure 2.2: Quality control methods in an iterative crowdsourcing pipeline. A qualification test is a popular way to find qualified workers and filter out inaccurate workers early. It is optional but often effective. Task requesters design the interface of an actual task by taking account of several factors such as pay and task difficulty. In addition, an incentive strategy is determined at this step. During data collection, crowd workers choose one of the available tasks and complete it. Task requesters conduct label aggregation methods in two ways. One is conducted in an online fashion in which label collection and aggregation happens together. The other is performed in an offline fashion in which labels are aggregated once data collection is completed. Simultaneously, task requesters analyze crowd workers' performance in order to identify a pool of quality workers for future work and interact with crowd workers via feedback analysis in the post-processing stage. Finally, worker retention is connected to future crowdsourcing. Workflow design methods for quality control are not represented in this pipeline since they attempt to control quality in crowdsourcing by designing different types of pipelines.

multiple labeling under noisy labelers (Sheng et al., 2008). They argued for the advantages of multiple labeling strategies over single labeling in the context of supervised learning models. Through simulations, they tested the advantages of multiple labeling over single labeling and suggested that selective multiple labeling is desirable for improving the quality of labels.

Snow et al. investigated the agreement between non-expert labels and existing expert labels (Snow et al., 2008). Through experiments using five different types of tasks, the authors demonstrate that non-expert labels showed high agreement with expert labels. In particular, through an affect recognition task, the authors showed that training machine-learning algorithms with non-expert labels could achieve effectiveness similar to that using gold standard labels from experts. Whitehill et al.'s study (Whitehill et al., 2009) proposed a graphical model of the labeling process that they called a Generative model of Labels, Abilities, and Difficulties (GLAD). GLAD modeled crowd workers varying in quality and the tasks varying in difficulty.

Ipeirotis et al.'s study on quality management on MTurk (Ipeirotis et al., 2010) demonstrates the effectiveness of multiple labeling by improving the quality of crowdsourced labels. When we aggregate multiple labels into one consensus label, majority voting is carried out in a very straightforward way since it is easy to use and understand. The form of expectation maximization (EM) algorithm proposed by Dawid and Skene for multiple label aggregation is typically more accurate than majority voting (Dawid and Skene, 1979). The EM algorithm exploits maximum likelihood to infer the error rates of

crowd workers that assign labels to the given documents in the absence of ground truth. Based on EM, Ipeirotis et al. (Ipeirotis et al., 2010) proposed an advanced model that separates error and bias. This is done by disentangling the true (un-recoverable) error rate from the recoverable biases due to workers' subjectivity.

Typical label aggregation methods for quality control in crowdsourcing are based a scenario in which label acquisition is followed by quality improvement. In other words, all of the above studies conduct label aggregation in an *offline* method. Here labels were collected in advance and aggregated only afterward. While this permitted a simple staged approach, with minimal coupling between label collection and aggregation stages, it precludes dynamic optimization of labeling effort. This means the same labeling effort is expended on all examples with no regard to examples of varying difficulty or a workforce of varying skill level.

To address this, Welinder and Perona proposed an *online* crowdsourcing approach that integrated label collection with aggregation. As labels were collected, the approach dynamically evaluate both aggregated labels and workers (Welinder and Perona, 2010b). This method finds and prioritizes high-quality crowd annotators when requesting additional labels and actively screens out unreliable annotators. In addition, to achieve a desired level of confidence, it dynamically judges the number of additional labels to request. In this way, this method achieves the savings of efforts for label acquisition and the total cost of labeling while keeping error rates low.

Venanzi et al. proposed a different way to aggregate labels by considering the similarity of confusion matrices between workers (Venanzi et al., 2014). Instead of merely relying on individual worker's labels, this study considered the community of crowds who share the similar annotation performance. By designing a confusion matrix for a community, this study presented a method to generate higher quality consensus labels from crowd work. The following study (Venanzi et al., 2015) further improved its label quality by considering temporal aspects (task duration) of crowd label quality.

While most studies in label aggregation have focused on how to aggregate multiple labels, Chen et al's study (Chen et al., 2013) propose a novel way to generate a better quality consensus label, especially for collecting relevance judgments. Instead of collecting absolute judgments indicating the degree of relevance of a given pair of document and query, this study collects a pairwise judgment which allows annotators to compare two different documents against a given query. Their empirical analysis demonstrates that the proposed method would improve the quality of relevance judgments significantly in comparison to existing methods.

Other quality control methods include spam worker filtering (Raykar and Yu, 2012; Eickhoff and Vries, 2012) and active learning-based methods using machine prediction alongside crowdsourced labels (Donmez and Carbonell, 2008; Donmez et al., 2009; Du and Ling, 2010; Law, 2011). Spam worker filtering methods typically measure the performance of each worker in a supervised way (via gold labels) or an unsupervised way (via majority vot-

ing), and filter out spam worker labels based on the performance during label aggregation. Active learning-based methods selectively pick the most useful labels to crowdsource rather than randomly selecting labels in a noisy crowdsourcing scenario. In this setting, machine learning is used for the prediction of the usefulness (uncertainty) of labels, and crowdsourcing is only used for making labels for the most useful (uncertain) ones.

Most label aggregation methods assume that labels from crowd workers are binary or at least discrete. However, there exist a number of tasks that are unstructured, with results from crowd workers which are neither binary nor discrete. Furthermore, all of these methods overlook that the quality of data collected from the crowd may vary significantly under different task or workflow designs. In the following section, we discuss how different methods attempt to control quality issues in detail.

2.3.2 Task Design for Quality Control

Task design methods fundamentally assume that a better designed task can improve the quality of crowdsourcing results (Shaw et al., 2011; Alonso, 2012; Kulkarni et al., 2012; Kazai et al., 2012a). Task design methods take account of multiple factors such as pay and task difficulty, which may influence the quality of crowdsourcing.

Shaw et al. studied the effects of different social and financial incentive schemes. They find, however, that the quality of work was influenced by the difficulty of the task (Shaw et al., 2011). Grady and Lease investigated

the impact of various aspects of the task design on MTurk, such as title, terminology, and pay (Grady and Lease, 2010). In addition to pay and task difficulty, other factors affecting the quality of crowdsourcing were studied in the literature. Ipeirotis (Ipeirotis, 2010) studied the completion time for tasks posted on MTurk. The researchers found that workers were limited by the user interface, and other workers' task selection was influenced by the tasks available through one of the existing sorting criteria. Chilton et al. reported similar findings. Such findings suggest that task requesters should consider how best to expose their tasks on crowdsourcing platforms (Chilton et al., 2010). This finding is also supported by Kucherbaev et al's survey results (Kucherbaev et al., 2014). Kazai et al. also conducted an analysis of human factors and label accuracy in crowdsourcing relevance judgments (Kazai et al., 2012a). Their findings, regarding the effect of the level of pay offered contrast with those of Mason and Watts (Mason and Watts, 2009).

One way to improve the quality of task results is to design a better task with qualification tests. To control quality, a simple but powerful method is to use "trap questions", also known as "canaries", "gold tests" and "verifiable answers". Trap questions have been reported as a useful tool in identifying unreliable workers and removing the effects of their noisy labels (Kazai et al., 2012a; Zhu and Carterette, 2010).

2.3.3 Workflow Design for Quality Control

Workflow design methods differ from task design methods in how they attempt to change the overall structure of a crowdsourcing pipeline. While task design methods focus on design of individual task, workflow design methods aim to develop an operational workflow which is composed of a set of tasks (Bernstein et al., 2010; Little et al., 2010; Weld et al., 2011b; Kittur et al., 2011; Noronha et al., 2011; Lin and Weld, 2012; Kittur et al., 2012).

The idea of structuring crowdsourcing tasks based on iterative workflows has also been explored. This is evidenced in Soylent (Bernstein et al., 2010), Turkit (Little et al., 2010), CrowdForge (Kittur et al., 2011), CrowdWeaver (Kittur et al., 2012), Platemate (Noronha et al., 2011), and Two-stage probabilistic generative model (Baba and Kashima, 2013). In addition, iterative workflow enables us to build strong quality control during a process that used to be hard to control (Dai et al., 2011; Weld et al., 2011b,a; Lin and Weld, 2012).

2.3.4 Worker Management for Quality Control

Finally, worker management methods focus on the understanding of some characteristics of crowd workers rather than their outputs (Mason and Watts, 2009; Mason and Suri, 2012; Rzeszotarski and Kittur, 2011; Kazai et al., 2012b; Kaufmann et al., 2011; Dow et al., 2012; Eickhoff and Vries, 2012). Many methods in this category are strongly related to behavioral analysis, which investigate crowd workers' behavior principles to the given tasks in

crowdsourcing.

Kaufmann et al. (2011) found that a significant share of the crowd's workforce is made up of a number of users motivated by financial gain. Some studies investigated the completion time of a task in crowdsourcing as a way of predicting the quality of the task (Wang et al., 2011; Rzeszotarski and Kittur, 2011). Rzeszotarski and Kittur found that the completion time provided by MTurk was not significantly effective at predicting the quality of the task performed. Therefore, the authors suggested that to correctly determine a worker's behavior, time information should be treated in a fine-grained way. Regarding the demographics and personalities of crowd workers, Kazai et al. (2012b) found in their particular study that Asian workers showed significantly poorer performance than did American and European workers. This result coincided with what is found by Ross et al. (2010). Mason and Watts studied the extrinsic incentives, especially the financial incentives. The authors demonstrated that increased financial incentives increases not the quality of work performed by crowd workers but just the quantity (Mason and Watts, 2009).

Dow et al. proposed a new way to improve quality of work by giving feedback to crowd workers (Dow et al., 2012). They investigated whether the effectiveness of timely, task-specific feedback helps crowd workers produce better results. Their experiments demonstrated that, over time, both external and self-assessment feedback influence the quality improvement of crowd workers. Jung and Lease also applied a similar approach to provide a crowd worker

with an error-report indicating the strength and weakness of each worker per task type. This study investigated how educating crowd workers with an error-report might over time improve the quality of work (Jung and Lease, 2013).

2.3.5 Summary

Several characteristics account for crowdsourcing's popularity: large-scale online workforces, low-cost labor, and fast completion (Mason and Suri, 2012). However, critical concerns persist about crowdsourcing's quality. This issue is caused by not only crowd workers' lack of expertise or concentration but also a lack of sufficient attention to multiple factors related to quality task design. Many studies have proposed a better task design, statistical models, and investigated multiple factors that bear on the quality of crowdsourced work. In addition, scholars have studied workflow design and worker management.

Chapter 3

Predicting Crowd Next Label Quality via a Time-series Model

In this Chapter, we investigate a new method for the measurement and prediction of a crowd worker's label quality over time. While temporal behavioral patterns can be discerned to underlie real crowd work, prior studies have typically modeled worker performance as independent and identically distributed (i.i.d.) over time. To better model such temporal dynamics of crowd work, we propose a time-series prediction model for crowd label correctness. This latent variable model captures and summarizes past worker behavior, enabling us to better predict the correctness of each worker's next label. Given inherent uncertainty in prediction, we also investigate a *decision reject option* to balance the tradeoff between prediction accuracy vs. coverage. Results show the proposed model improves accuracy of both prediction of real crowd worker data, as well as data quality overall. [1]

[1]This chapter is based on the published work (Jung et al., 2014) in the AAAI Conference of Human Computation and Crowdsourcing 2014, which is a joint research with Yubin Park and Matthew Lease who contributed to the implementation of our temporal learning algorithm.

3.1 Introduction

For online crowd work (Kittur et al., 2013), effective task recommendation and routing have potential to significantly improve the quality of data collected and worker experience by better matching workers to available work (Law et al., 2011; Li et al., 2014). Whereas preference-based recommendation model seeks to model varying worker interest for different task types, performance-based recommendation instead models varying worker accuracy as a function of task type (*macro-level* worker-task matching) or specific examples (*micro-quality prediction*). Prior work in performance-based recommendation has typically modeled behavior of crowd workers as i.i.d. over time (Yuen et al., 2012; Yi et al., 2013).

In practice, however, crowd worker behavior can be seen to dynamically vary over time, as shown in **Figure 3.1**. A worker may become tired or bored, or begin multi-tasking, leading to decreased work quality. Alternatively, work quality may improve as a worker's experience with a given task accumulates (Carterette and Soboroff, 2010). Regardless of cause, temporal effects are clearly evident.

The closest prior work we are aware of on temporal modeling of crowd work, by Donmez et al. (2010b), assumes that workers are weak learners who behave according to simple latent dynamics $x_t = x_{t-1} + \epsilon_t$. This approach, evaluated entirely by simulation, assumes a uniform offset and temporal correlation for the underlying dynamics, inconsistent with what we see in real data, such as in **Figure 3.1**. In psychometrics, other prior work in item re-

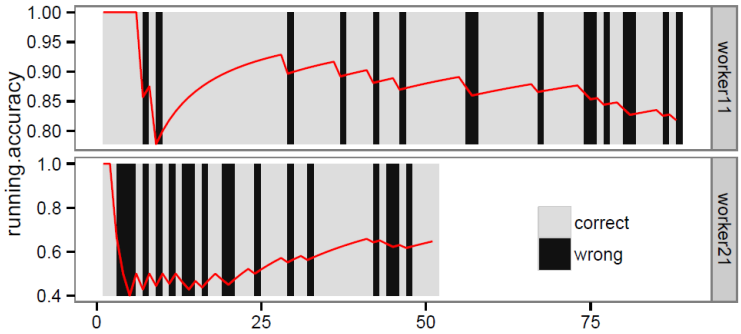

Figure 3.1: The work quality of two crowd workers is seen to vary over time. Gray area indicates correct responses while black stripes denote errors. The *running accuracy* of each (i.e., empirical accuracy up to a given response) is plotted in red. The top worker's accuracy is seen to decrease over time, while the lower worker's accuracy improves. Their temporal error patterns also differ: the top worker's errors become more frequent while the lower worker's become less so.

sponse theory (IRT)(Hambleton et al., 1991) seeks to assess each individual's temporal learning. However, our approach differs from IRT in that our models seek to capture latent dynamics by taking account of temporal correlation and additional variables.

To more faithfully model such temporal behavior, we present a time series-based prediction model for crowd workers' label correctness. This categorical time series model uses a temporally correlated latent variable which captures and summarizes the past behavior of a worker, enabling us to better predict the quality of the next label. To efficiently estimate model parameters, we adapt a recently developed technique (Park et al., 2014) to this crowdsourcing problem context. Given inherent uncertainty in prediction, we also investigate a *decision reject option* to balance the tradeoff between prediction

accuracy vs. coverage (Bartlett and Wegkamp, 2008; Nadeem et al., 2010).

The effectiveness of the proposed time-series model is evaluated on both a synthetic dataset and real data from the NIST TREC Crowdsourcing Track [2]. Results show the proposed time-series model improves the accuracy of prediction of workers' label correctness. Furthermore, empirical results demonstrate that the understanding of temporal dependency of crowd work quality leads to the improvement of prediction accuracy. The *decision reject option* enables further accuracy improvement by sacrificing coverage, providing a tuning parameter for aggressive vs. conservative prediction given model confidence. Additional simulation experiments show overall quality improvements achieved. We investigate the following research questions:

RQ1.1: Application of time-series modeling. *How can time-series models of crowd work be applied and interpreted?*

RQ1.2: Label prediction via a time series model. *How accurately does time-series prediction work in a crowdsourcing context? Why does time-series prediction work improve prediction accuracy?*

RQ1.3: Use of *decision reject option* for managing uncertainty *How effectively does the decision reject option let us tradeoff the proposed model's prediction accuracy vs. coverage?*

[2] http://sites.google.com/site/treccrowd/2011/

RQ1.4: Label quality improvement. *To what extent can time-series modeling yield overall improvement in quality of crowdsourced labels?*

3.2 Problem

On crowd work platforms such as Amazon's Mechanical Turk (MTurk), a worker selects a task as known as HIT group in MTurk by oneself. Relatively few studies have investigated task routing for micro-tasks, though work exists with other forms of crowdsourcing, such as Wikipedia (Cosley et al., 2007). Kamar et al. (2012) studied the cooperative refinement and task routing among on-line agents. In addition, both Kamar et al. and Dai et al. (2010) developed methods to predict the accuracy of the next label, but did not model workers' individual temporal profiles in making these predictions. Bernstein et al. (2012) investigated task routing in terms of real-time crowdsourcing. These studies do not address finding strong candidates for a particular task from the requester's viewpoint. Work on task markets seeks to chain together different worker competencies (Shahaf and Horvitz, 2010).

SFilter, proposed by Donmez et al. (2010b), is a Bayesian time series model that captures crowd workers' dynamically varying performance. The authors did not learn the parameters for the latent variable dynamics, but as mentioned earlier, assumed an uniform offset and temporal correlation for the underlying dynamics, with workers assumed to be weak learners following simple latent dynamics $x_t = x_{t-1} + \epsilon_t$. Based on the fixed parameters, the latent variable is estimated using a variation of a particle filter, cf. (Petuchowski and

Lease, 2014). This assumption does not seem to hold on the real crowd data we have observed, as evidenced in **Figure 3.1**.

This study attempts to relax special conditions ($c = 0$ and $\phi = 1$) by proposing a general time series model ($x_t = c + \phi x_{t-1} + \epsilon_t$). The principal difference is to capture and summarize the underlying dynamics of workers' labeling more efficiently and accurately. The goal of this study is to predict the next label of a crowd worker by estimating the latent variables governing the performance of the crowd worker. In addition, we would like to use the latent variables to better analyze the varying temporal performance of crowd workers. The formal definition of our problem is as follows:

Problem. Given an individual's label correctness history, 1) estimate the probability of correct labeling for the next task instance, and 2) provide meaningful summary statistics for the behavioral pattern of the worker.

We begin with a binary label annotation problem in crowdsourcing. As Park et al.'s study (2014) showed, the extension to multiple categorization is straightforward by changing our temporal prediction model, especially the link function. We first discuss the theoretical background of our approach and present the proposed model in the next Section. We aim to predict the next label of each worker, using this information to identify the best workers to which examples should be routed for labeling.

3.3 Method: Latent Autoregressive Model

Suppose that a worker has completed n task instances. The correctness of the ith instance is denoted as $y_i \in \{0, 1\}$, where 1 and 0 represent correct or not. Thus, the performance of a worker can be represented as a sequence of binary observations, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$. For example, if a worker completed three task instances and erred on the first only, then his *binary performance sequence* is encoded as $\mathbf{y} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$.

Assume two workers, Alice and Bob, have each labeled 10 instances with performance as follows:

$$\mathbf{y}_{\text{Alice}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$
$$\mathbf{y}_{\text{Bob}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

While both achieve 50% accuracy with respect to the ground truth, they exhibit quite different temporal profiles. Alice provides incorrect labels immediately after she marks correct labels. On the other hand, Bob shows a poor performance in the beginning, but he correctly labeled the last four tasks in a row. We must go beyond measuring accuracy to capture such temporal variation across workers.

Several statistical models can capture this kind of temporal variation. In a broad sense, such models fall into two classes depending on the use of latent variables: fully-observed vs. latent variable models. Fully-observed models include Mixture Transition Distribution Model (MTDM) (Raftery, 1985),

Markovian regression model (Kaufmann, 1987; Zeger et al., 1988), and Discrete Autoregressive Moving Average model (DARMA) (Jacobs and Lewis, 1983). Latent variable models have been successfully demonstrated in various applications such as decoding algorithms (Viterbi, 1967) and speech recognition (Juang and Rabiner, 1991). Such latent variable models can be further grouped into two sub-categories based on the representation of latent variables: Hidden Markov Models (HMM) (Zucchini and MacDonald, 2009) use discrete latent variables, whereas State-Space Models (SSM) (Zhen and Basawa, 2009) adopt continuous latent variables.

We adopt a Latent Autoregressive model (LAR) for categorical time series. The model is a state-space model for a categorical time series that has been less popular than HMM and SSM. This is partly because such continuous latent variables are notoriously difficult to reconstruct from categorical observations. However, the use of the LAR process provides two substantial advantages: *interpretability* and *extensibility*. Indeed, the autogregressive (AR) process has a rich history with parsimonious theoretical results (Canova and Cicarelli, 2013; Litterman, 1984). The interpretation on stationarity and spectral analyses (Burg, 1967) can be smoothly applied to the latent AR processes. Moreover, the latent VAR process can be easily extended to cover variants of the AR models such as ARMA (Box et al., 1994), Autoregressive Conditional Heteroskedasticity (ARCH) (Engle, 1982), and Generalized ARCH (GARCH) processes.

We hypothesize a latent factor x_t that governs the worker performance.

This latent factor evolves over time depending on the previous value in the sequence. The sequence dynamics of the latent factors is described by a set of parameters $\boldsymbol{\theta} = \{c, \phi\}$. In essence, the proposed model is described as follows:

$$\text{(Latent AR)} \qquad x_t = c + \phi x_{t-1} + \varepsilon_t \tag{3.1}$$

$$\text{(Observation)} \qquad p(y_t = 1) = \text{logit}^{-1}(x_t) \tag{3.2}$$

$$\text{(Noise model)} \qquad \varepsilon_t \sim \text{Normal}(0, \sigma^2) \tag{3.3}$$

where $\mathbf{y} = [y_t]_{t=1}^{T}$ with $y_t \in \{0, 1\}$, and $p(y_t = 1)$ indicates the probability that $y_t = 1$. Our model parameters, c and ϕ, are sequentially estimated over time. However, since these parameters only influence the estimation of a next latent variable x_t, we do not subscribe t for these two parameters.

Returning to our Alice and Bob example, we show that this LAR model captures the illustrated temporal patterns. For Alice's case, let us take $c = 0.1$, $\phi = -0.9$, and $x_0 = -1$ (initial latent state). For simplicity, c and ϕ are fixed over time in this example. In reality, we update both parameters over time once a new label comes. Ignoring the effect of noise, the latent variables propagate as follows:

$$\begin{aligned} x_1 &= c + \phi x_0 = 0.1 - 0.9 \times -1 = 1 \\ x_2 &= c + \phi x_1 = 0.1 - 0.9 \times 1 = -0.8 \\ x_3 &= c + \phi x_2 = 0.1 - 0.9 \times -0.8 = 0.82 \\ &\vdots \end{aligned}$$

where the sequence of latent variables oscillate. As a result, the probability of correct labeling also oscillates over time. On the other hand, for Bob's case,

let us assume that the parameters are $c = 0.1$, $\phi = 0.9$, and $x_0 = -1$. The temporal sequence of the latent factors are given as:

$$
\begin{aligned}
x_1 &= c + \phi x_0 = 0.1 + 0.9 \times -1 = -0.5 \\
x_2 &= c + \phi x_1 = 0.1 + 0.9 \times -0.5 = -0.35 \\
x_3 &= c + \phi x_2 = 0.1 + 0.9 \times -0.35 = -0.215 \\
&\vdots
\end{aligned}
$$

As can be seen, the probability of correct labeling improves over time. To estimate maximum likelihood parameters ϕ, c, x_t, we use a method known as *Low-resolution augmented Asymptotic Mean Regularized Expectation Maximization* (LAMORE) (Park et al., 2014).

Estimating the parameters from a categorical sequence involves several challenges. First, unlike continuous time series, categorical time series contain only finite bits of information. Categorical outputs can be viewed as lossy-compression from an information theoretic perspective, thus the reconstruction of the continuous latent variables suffers from a low signal-to-noise ratio. Furthermore, this noisy reconstruction increases the uncertainty of the estimated parameters, especially in the alternating minimization framework. As a result, classical alternating minimization techniques such as Expectation-Maximization become susceptible to various factors, including noisy reconstruction and multiple local optima of the log-likelihood function. LAMORE combines Method of Moments and Monte-Carlo Expectation Maximization algorithms to stabilize parameter estimation. The method can be extended to

general categorical time series, e.g. tertiary categorical time series.

3.4 Adaptation to Crowdsourcing

We next discuss how to use our time-series framework in the context of crowdsourcing [RQ1.1]. Prior to applying this framework, we first discuss the semantics of our time series model in crowdsourcing. We present the proposed label prediction algorithm based on this understanding of the semantics.

3.4.1 Interpretation of Parameters

The proposed time series framework takes a sequence of observations as input and generates four types of output values: latent variables ϕ, c, x_t and an observable variable $y = logit^{-1}(x_t)$. As input, we use the worker's *binary performance sequence*, as illustrated by the earlier Alice and Bob examples from the previous section.

3.4.1.1 Latent Variable (x_t)

The interpretation of x_t has an important meaning with regard to the analysis of workers' performance. First, x_t indicates the probability of making a correct label at a time point t. In our model, a link function, $logit^{-1}(x_t)$, transforms this probability to a soft label representing the polarity of a worker's next label correctness. With regard to task routing, this soft label is used as a criterion to judge an optimal candidate. If a soft label, $logit^{-1}(x_t)$, is close to 0 or 1, it indicates that the next label correctness of this worker is likely to be

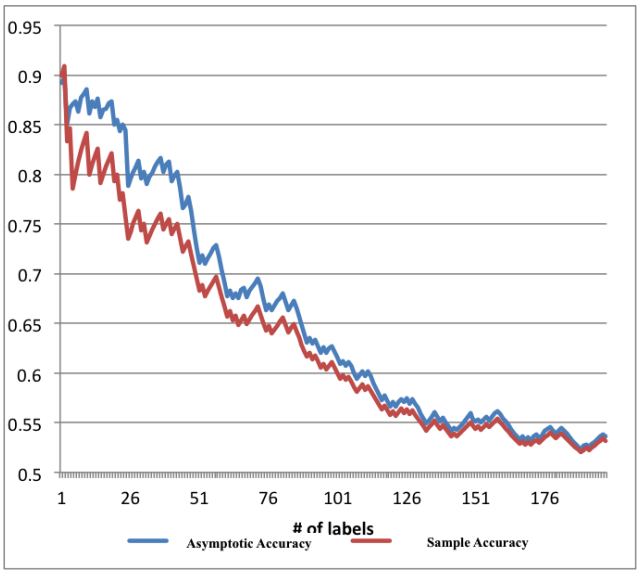

Figure 3.2: Relation over time between asymptotic accuracy and sample running accuracy (SA), where SA= $\frac{\text{\# correct labels}}{\text{\# submitted labels}}$.

confident. On the contrary, a label around 0.5 suggests that the confidence of the worker's next value is relatively low since the polarity of the label is weak.

Second, the dynamics of x_t is an autoregressive process with one lagged variable, AR(1). If the absolute value of the temporal correlation parameter ϕ is less than 1 i.e. $|\phi| < 1$, the underlying AR(1) process is a stationary ergodic process. The asymptotic theory of the autoregressive process provides that the asymptotic mean of x_t is given as $x_\infty = \frac{c}{1-\phi}$. This can be obtained by solving $E[x_t] = E[c] + E[\phi x_{t-1}] + E[\epsilon_t] = c + \phi E[x_t]$. Since y_t is fully determined by x_t in our model, we can extend the concept of the asymptotic mean to "asymptotic accuracy" which is defined as follows:

$$y_\infty = \text{logit}^{-1}(x_\infty) = \frac{\exp(x_\infty)}{1 + \exp(x_\infty)} \tag{3.4}$$

Provided modeling assumptions are met, the estimated asymptotic accuracy should converge to the sample accuracy (i.e. ergodicity). **Figure 3.2** empirically demonstrates the convergence of these two values over time where the data comes from a randomly selected worker. This suggests that our modeling assumptions fit well to the actual data.

3.4.1.2 Temporal Correlation (ϕ)

Temporal correlation ϕ indicates how frequently a sequence of correct/wrong observations has changed over time. A worker having $\phi = -0.8$ tends to follow a temporal pattern of regularly alternating between correct and wrong. On the other hand, another worker having $\phi = 0.8$ tends to follow a consistent pattern without a frequent switching, irrespective to being correct or wrong. Between these extremes, $\phi \approx 0$ indicates no temporal dependencies between sequential labels; a worker of having $\phi \approx 0$ does not show any regular temporal pattern. At each time point t, ϕ is updated. In sum, ϕ helps to characterize a worker's behavioral pattern and understand its underlying dynamics.

3.4.1.3 Offset (c)

The sign of offset c navigates the direction between correct and wrong. For example, if c is positive, at each time, the latent variable of a worker is drifted toward the positive direction, which implies better a correct rate. On the other hand, if c is negative, the latent variable is drifted toward the negative

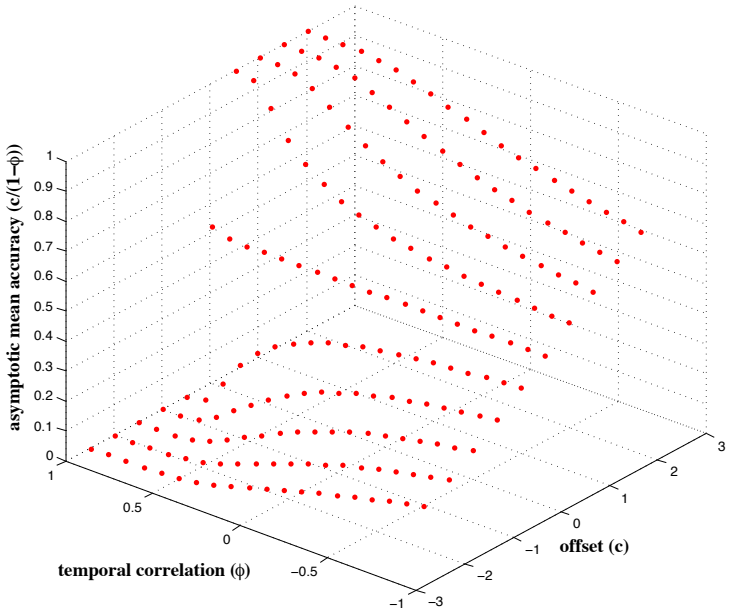

Figure 3.3: Relation between c and ϕ vs. asymptotic accuracy.

direction, implying that the performance of a crowdworker will degrades over time. The size of offset c combined with ϕ determines the asymptotic accuracy of a crowdworker. **Figure 3.3** shows the relationship between c and ϕ vs. asymptotic accuracy $logit^{-1}(\frac{c}{1-\phi})$. When offset c is positive, the higher ϕ indicates the higher accuracy. On the contrary, the lower ϕ indicates the higher accuracy when offset c is negative. This suggests that a worker of a higher temporal correlation ϕ shows extreme polarity with regard to accuracy. In contrast, a worker having low ϕ value shows an accuracy close to 0.5, which indicates less confidence in a worker's label.

3.4.2 Prediction with Decision Reject Option

For a real application of a time series model to crowdsourcing, we first consider workers' label predictions over time. The output of the proposed time series model can be easily applied toward label prediction as follows. A value of $logit^{-1}(x_t)$ can be used as a probabilistic label (*soft label*) indicating the strength of one direction, positive or negative. For label generation, we may use this value in two ways. First, it is straightforward to use a given soft label without any transformation. Second, we generate a *hard label* based on the value of a given soft label. For instance, in terms of predicting a binary label, if a predicted soft label is 0.76, a binary label of 1 is generated since the value of the given soft label is greater than 0.5.

In terms of label prediction, there exists room for improving the quality of label prediction by taking account of prediction confidence. For instance, if a soft label is close to 0.5, it fundamentally indicates very low confidence in terms of the polarity. Therefore, we may avoid the risk of getting noisy predictions by adopting a decision rejection option (Pillai et al., 2013). In this study, the following decision reject option is applied to our prediction model.

$$l(x_t) = \begin{cases} logit^{-1}(x_t) & \text{if } x \leq 0.5 - \delta \quad \text{or } x \geq 0.5 + \delta \\ \text{null} & \text{if } x > 0.5 - \delta \text{ and } x < 0.5 + \delta \end{cases}$$

where δ is a parameter to control the limits of the decision reject option, and $\delta \in [0, 0.5]$. High δ indicates conservative label prediction, which increases the range of decision rejection while sacrificing coverage. On the other hand, low δ allows label prediction in a permissive manner, decreasing the threshold of decision rejection and increasing coverage.

3.4.3 Preliminary Analysis

3.4.3.1 DataSet

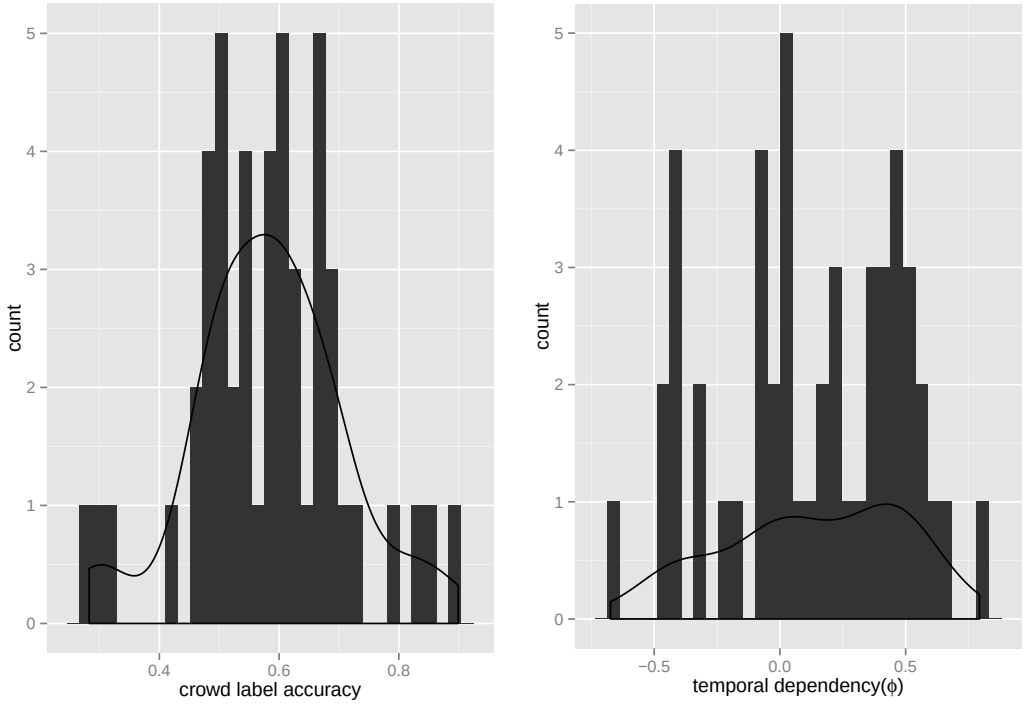

Figure 3.4: Characteristics of public crowdsourcing dataset. Left figure shows the histogram of crowd label accuracies across 49 crowd workers. Right figure shows the histogram of temporal dependencies (ϕ) of 49 crowd workers.

Prior to studying the effectiveness of our time-series model, we first discuss how an actual crowdsourcing data looks like from a temporal viewpoint. Our preliminary analysis is conducted with a public crowdsourcing dataset which is a subset of a public dataset created for the NIST TREC crowdsourcing Track 2011 Task 2. The dataset contains binary *relevance judgments* from workers rating the relevance of different Webpages to different search

queries (Buckley et al., 2010; Grady and Lease, 2010). The original dataset has 762 crowd workers who judged 19,033 query-document pairs of relevance judgment task (examples). We process this original dataset for our evaluation criteria. Firstly, we exclude workers making < 20 judgments to ensure stable estimation. Moreover, since the goal of our work is to be able to route work to specific workers, it is only worth modeling a given worker's behavior if we believe that worker will continue to do more work in the future, as suggested by their having already performed some minimal amount of work. Secondly, we include only examples which have ground truth labels. In addition, this dataset is processed to extract the original order of the workers' labels. We extract 49 sequential sets of binary label correctness, one set per crowd worker. The average number of labels (i.e., sequence length) per worker is 133. **Figure 3.4** shows the basic characteristics of this dataset. The left figure shows the distribution of crowd label accuracies. Most of the crowd label accuracies range from 0.45 and 0.7 and its mean is 0.58. The right figure shows the distribution of temporal dependencies (ϕ) of crowd workers' label correctness. As the above section introduced, ϕ indicates temporal dependencies of a given sequence of binary values. We estimate the ϕ of each crowd worker via our proposed model. Its mean across 49 workers is 0.129 and most of the crowd workers' ϕ 's range from 0 to 0.55.

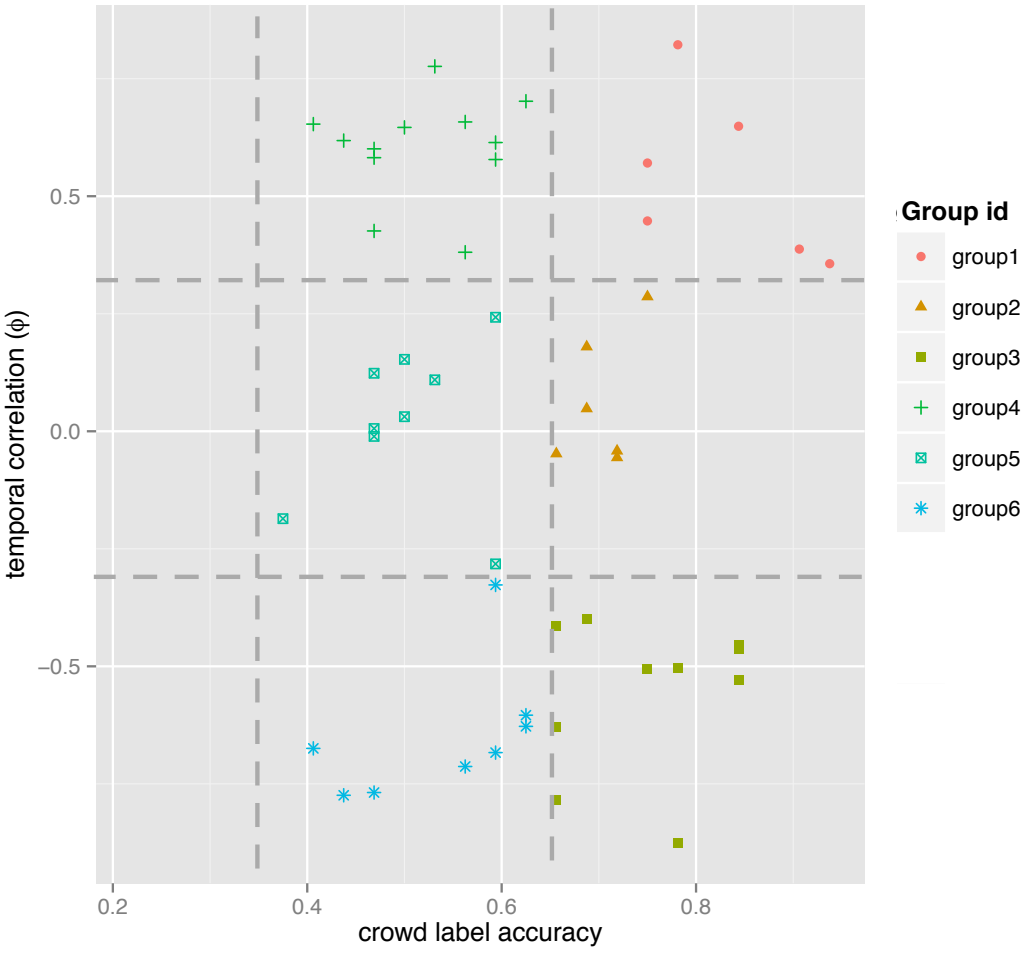

Figure 3.5: Crowd label accuracy vs. temporal correlation (ϕ). This plot groups crowd workers by crowd label accuracies and temporal correlations (ϕ) of workers' label sequences. For grouping, we use 0.35 and 0.65 as a threshold for accuracy and use 0.3 and -0.3 as a threshold for temporal correlation. While crowd label accuracy considers group 1, 2, and 3 (as well as group 4, 5, and 6) as one group of workers, temporal correlation differentiates these workers into three different groups based on different temporal dynamics.

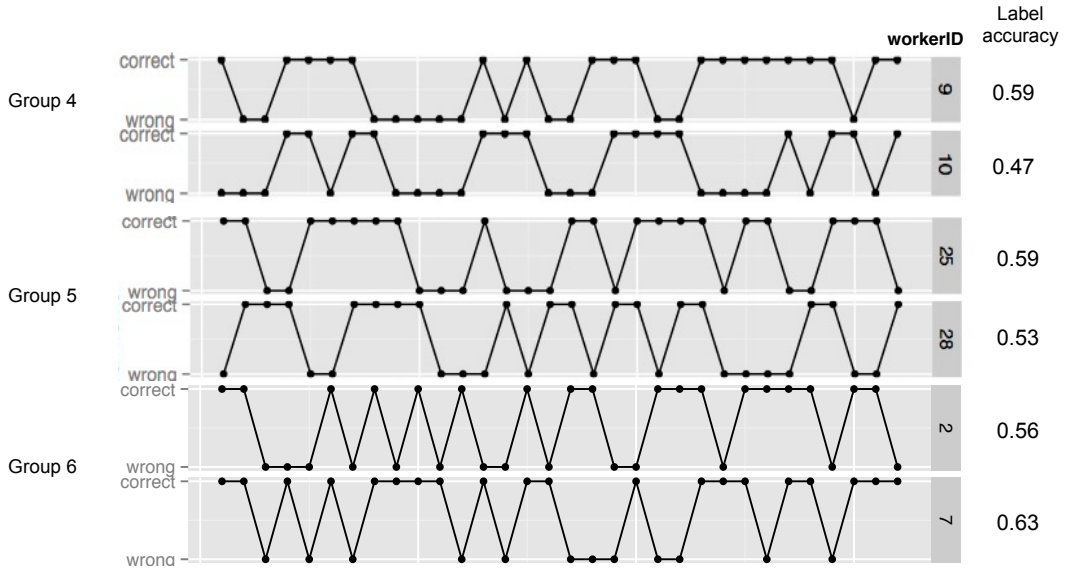

Figure 3.6: Example of crowd label correctness over time by three worker groups. While these workers show similar label accuracy ranging from 0.35 to 0.65, their sequence of label correctness vary considerably. Group 4 workers show rare changes of their label correctness. On the other hand, Group 6 workers show frequent transitions of label correctness over time.

3.4.3.2 Data Analysis

With this dataset, we first group workers by label accuracy and temporal correlation (ϕ) introduced in the previous section. We consider three worker groups by label accuracy: high quality (> 0.65), middle range (> 0.35 and < 0.65), and low quality (< 0.35). With regards to temporal correlation, we also consider three worker groups: positive strong (> 0.3), weak (< 0.3 and $>$ -0.3), and negative strong ($<$ -0.35).

Figure 3.5 shows a scatter plot of 49 workers by crowd label accuracy and its temporal correlation (ϕ). Our hypothesis is that workers showing sim-

ilar label accuracies may show different temporal patterns. This plot visually supports our hypothesis. While group 1, 2, and 3 show similar label accuracies, their temporal correlation ϕ vary among workers. Similarly, group 4, 5, and 6 shows different temporal correlations under one crowd label accuracy bin. How actually each group of workers show different patterns and statistics?

To answer this question, we conduct additional analysis with each group of workers. Firstly, we focus on the difference of binary sequences between groups. Figure 3.6 shows actual binary sequences of label correctness of three different groups. While these workers show similar label accuracies ranging from 0.35 and 0.65, their binary sequences tend to show a different pattern per group. Worker ID 9 and 10 who are in Group 4 show a relatively consistence pattern of label correctness over time. On the contrary, Worker ID 25 and 28 in Group 5 show more transitions of label correctness and Worker ID 2 and 7 in Group 6 show very frequent changes of label correctness over time. This suggests that accuracy is not sufficient metric to measure a worker's label quality.

Secondly, we compute basic statistics of binary sequences. In particular, we focus on the length of the same binary sequence in a given sequence. Our hypothesis is that if a worker's label correctness varies over time, the length of the same binary sequence would be shorter than others. Even for the workers showing the similar label accuracies, we expect that this pattern may happens differently. Table 3.1 presents the detailed statistics per worker group. To investigate how frequently a sequence of worker's label correctness s changes

over time, we measure three different statistics per worker, such as MLS(s), AS(s), and $\sigma(s)$. Next, we compute the average of these statistics per group. MLS(s) indicates the maximum length of the same value sequence in a given binary sequence s. For instance, for a worker who rarely changes its label correctness over time, MLS(s) is long. AS(s) indicates the average length of the same-value sequence. This metric indicates how frequently a worker's label correctness changes on average. For instance, a given binary sequence $\{0, 0, 1, 1, 1, 1, 0\}$, MLS(s) is 4 since the longest sequence of the same value is [1111]. AS(s) equals to (2+4+1)/3 = 2.33 since there exist three same-value sequence [00], [1111], and [0]. While these two metrics are used to measure the temporal characteristics of each worker, we measure the standard deviation σ of a worker's label correctness sequence. Since the standard deviation considers each label correctness as independent from others, this metric does not capture any temporal dependency in a given sequence.

Our analysis results show that MLS and AS vary across worker groups. With regards to the Group 1, 2, and 3, Group 1 shows the largest MLS and AS since workers of this group do not tend to change their label correctness frequently over time. On the contrary, Group 2 shows the smaller MLS and AS since these workers shows frequent changes of their label correctness. Workers in Group 3 have the smallest MLS and AS since their label correctness oscillates very frequently. The similar patterns are observed across Group 4, 5, and 6 even though these groups show the lower label accuracy in comparison to the former three groups. This result is consistent with that of Figure 3.6.

Worker Group	**MLS**	**AS**	$\sigma(S)$	NumWorkers	Avg. Label Accuracy
Group 1	15	4.15	0.36	6	0.82
Group 2	9.6	2.30	0.46	6	0.70
Group 3	9	2.26	0.43	9	0.75
Group 4	9.17	2.40	0.50	11	0.52
Group 5	7.22	2.12	0.50	9	0.50
Group 6	5.85	1.60	0.50	8	0.54

Table 3.1: Analysis on the sequences of crowd label correctness. MLS(s) indicates the maximum length of a same-value sequence in a given sequence s. AS(s) indicates the average length of a same-value sequence $\sigma(s)$ indicates the standard deviation of a given sequence s. For instance, a given binary sequence $\{0, 0, 1, 1, 1, 1\}$, MLS is 4, AS is 3, and $\sigma(S)$ is 0.51. This statistics indicates how frequently a worker's binary correctness is changed. While Group 4, 5, and 6 shows the similar level of crowd label accuracy, each group shows different statistics. Group 4 show the longest same-value sequence, but Group 6 shows the shorted same-value sequence.

However, note that $\sigma(s)$ does not capture any difference between workers of Group 4,5, and 6 since it does not capture temporal effects on crowd work quality.

To sum up, this preliminary analysis demonstrates that workers having similar label accuracies may show different temporal patterns. Furthermore, existing metric such as accuracy and the standard deviation may not capture such a temporal pattern properly. On the contrary, temporal correlation (ϕ) allows us to differentiate workers based on the understanding of underlying temporal dynamics of crowd label quality.

3.5 Evaluation

In this section, we describe the experimental evaluation and discuss its findings. We have tested the proposed time-series prediction model under various conditions of decision reject options with a synthetic dataset and a public crowdsourcing dataset.

3.5.1 Experimental Settings

3.5.1.1 Dataset

Our proposed time-series model takes a sequence of binary values indicating crowd labels' correctness as an input. For data acquisition, our evaluation uses two types of datasets, a synthetic dataset and a public crowdsourcing dataset. In general, a public crowdsourcing data is more realistic since it is based on crowd labels generated by real crowd users. However, a real crowdsourcing dataset can be insufficient to validate the effectiveness of the proposed model due to its small sample size or specific settings in the process of crowd label acquisition. On the contrary, a synthetic dataset allows us to explore the effectiveness of our proposed model under various conditions, but it may sacrifice realism. To take account of the pros and cons of these two datasets, we evaluate our time-series model over both.

Firstly, we generate a synthetic dataset that represents a set of sequential binary correctness $\big[1, 0, 1, ...\big]$. For realistic generation of a binary sequence, our data generation function g takes two input values: crowd label accuracy α and number of labels n. For instance, when the number of labels is

n and a crowd label accuracy is α, our function g generates all of the possible permutations of given two parameters, ${}_nP_k = \frac{n!}{(n-k)!}$, k indicating the number of correct labels based on $\alpha = \frac{k}{n}$. Next, function g randomly chooses s sample binary sequences from this permutation, ${}_nP_k = \frac{n!}{(n-k)!}$, without replacement. In order to generate a well-balanced dataset across different crowd label accuracies, we generate 10,000 binary sequence samples for each accuracy bin ranging from 0.5 to 0.9 by 0.1. As a result, the total number of samples are 50,000 across 5 crowd label accuracy bins.

Secondly, we use the dataset which is already introduced in our preliminary analysis, Section 3.4.3.

3.5.1.2 Models

We evaluate the proposed time series model (TS-prediction model) under various conditions of decision reject options. Our initial model uses no decision reject option, setting $\delta = 0$. In order to examine the effect of decision reject options, we vary $\delta \in [0, 0.25]$ by 0.05 step-size.

For the prediction of a worker's next label correctness, we use each worker's first 20 observed labels for training an initial individual model for each worker; we then update the parameter of each learned model once a new label comes. For instance, if a worker has 50 sequential labels, our prediction model takes the first 20 correct vs. incorrect observations in relation to the gold relevance judgments and then predicts the 21st label. Once the actual 21st label is submitted by the worker, we measure the accuracy and RMSE

over actual observations (correct/incorrect). For the following 29 judgments, we repeat the same process in a sequential manner, predicting each label one-by-one.

As a baseline, we compute a worker's *sample accuracy* at time t (SA_t) as observed accuracy up to time t, then use this value as the probability of the worker's next label being correct. While sample accuracy eventually converges to asymptotic accuracy, this baseline cannot capture shorter-term dynamics of workers' behavioral patterns.

3.5.1.3 Metrics

We evaluate the performance of our prediction model with two metrics. Firstly, we measure the prediction performance with Accuracy and Root Mean Square Error (RMSE). Probabilistic labels (soft labels) produced by our assessor model are measured with RMSE, indicating the absolute difference between a predicted label and ground truth: $RMSE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(pred_i - gold_i)}^2$. Rounded binary labels (hard labels) are evaluated by accuracy, defined as follows, where tp denotes the number of true positive classifications, fp the false positives, tn the true negatives, and fn the false negatives: $accuracy = \frac{tp+tn}{n}$. Secondly, we also evaluate the effect of our prediction method on the quality of relevance judgments with accuracy defined as above. Since our extracted dataset is well-balanced in terms of the ratio between relevant vs. non-relevant judgments, use of accuracy is appropriate. The score is individually computed for each worker, then averaged over all workers. Crowd label accuracy indi-

cates a crowd worker's label accuracy at the time point which each worker completed all the labeling task instances.

3.5.2 Evaluation with Synthetic Data

We evaluate the performance of the proposed time-series model with a synthetic dataset generated by the proposed method in the above section 3.5.1.1.

3.5.2.1 Experiment 1.1 (RQ1.2): Prediction without Rejection

How accurately does our time series prediction model infer workers' next label's correctness? We first measure prediction performance of the proposed model (TS-based prediction) and sample accuracy-based prediction model (SA-based prediction) over actual workers' correct/wrong observations without considering any decision reject option.

Figure 3.7 shows the difference of RMSE between two models by each accuracy bin. Each accuracy bin has 10,000 synthetic crowd workers, and thus each bin has two boxplots representing the prediction error (RMSE) of each model. This result demonstrates that TS model outperforms SA model across varying crowd label accuracies. In particular, the difference of RMSE between two models becomes larger as crowd label accuracies increase. **Figure 3.8** shows the comparison of prediction accuracies between two models across varying crowd label accuracies. Prediction accuracies of the TS model outperforms those of the SA model. In particular, the difference between the

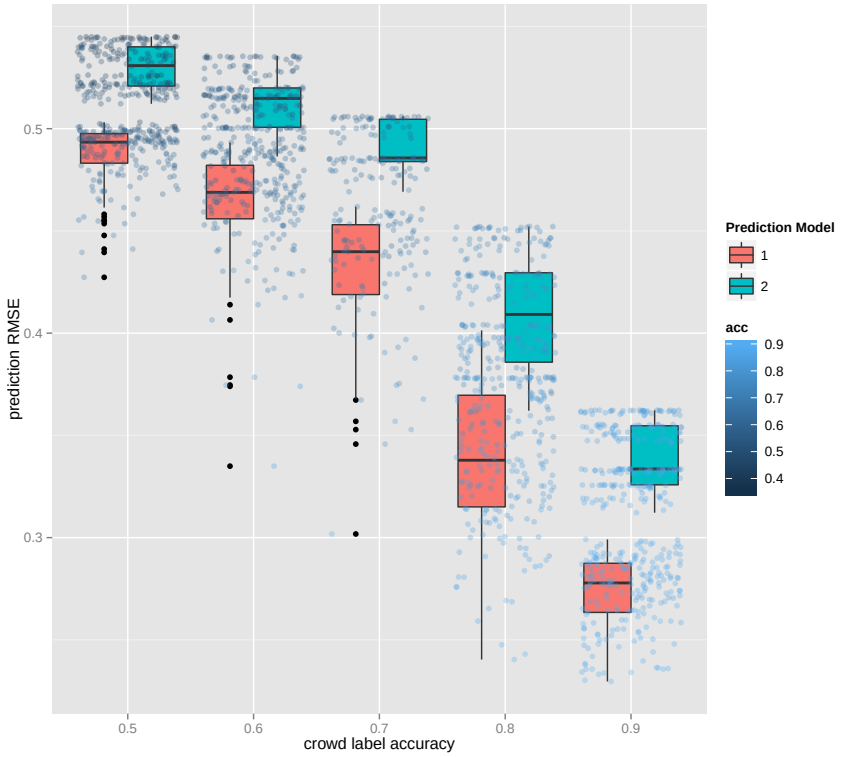

Figure 3.7: Crowd label accuracies vs. prediction RMSE of workers' label correctness. The x-axis indicates the accuracies of the crowd workers' labels and the y-axis indicates prediction quality RMSE. Crowd label accuracy indicates a crowd worker's label accuracy at the time point which the worker completed all the labeling task instances. Prediction Model 1 indicates our proposed time-series model (TS) and Model 2 refers to sample accuracy (SA). TS outperforms SA across all of the crowd label accuracy bins.

two models is greater when crowd label accuracies are close to random (0.5). This result demonstrates that the benefit of TS prediction becomes greatest when a crowd worker's label accuracy is random.

Next, we investigate how the TS model works under varying temporal correlations ϕ. **Figure 3.9** and **Figure 3.10** shows the prediction perfor-

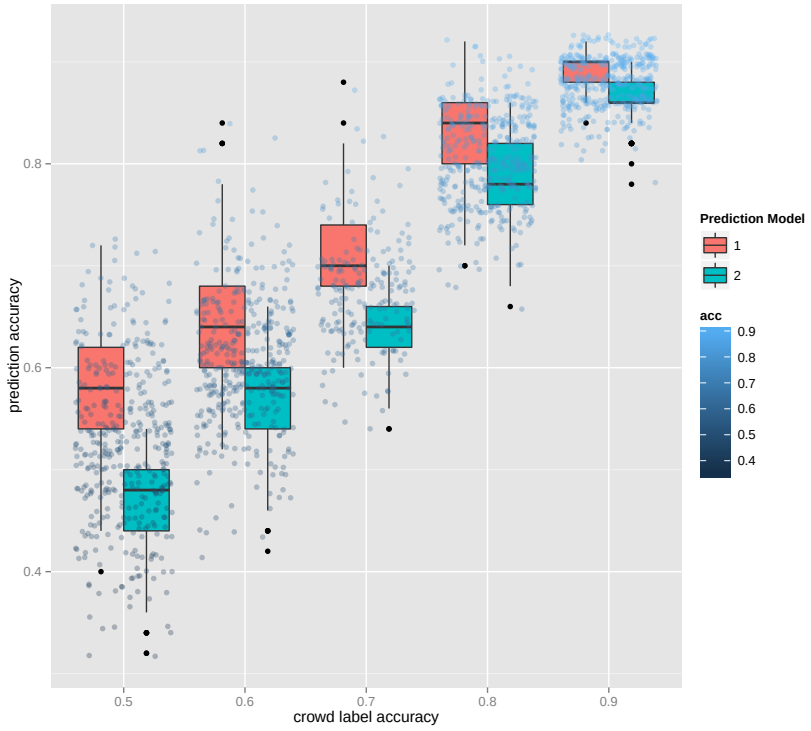

Figure 3.8: Crowd label accuracies vs. prediction accuracies of workers' label correctness. The x-axis indicates the accuracies of crowd workers' labels and the y-axis indicates prediction accuracies. Crowd label accuracy indicates a crowd worker's label accuracy at the time point which the worker completed all the labeling task instances. Prediction Model 1 indicates our proposed time-series model (TS) and Model 2 refers to sample accuracy (SA). Accuracy improvement of the time-series model (TS) tends to be relatively larger when crowd label accuracies are low (0.5). As crowd label accuracies increases, the difference between TS vs. SA becomes smaller.

mance (accuracy) of the TS and SA models across varying crowd label accuracy and temporal correlations. This experiments demonstrates why the TS model brings better prediction performance.

First, the understanding of temporal correlation (ϕ) gives a new horizon

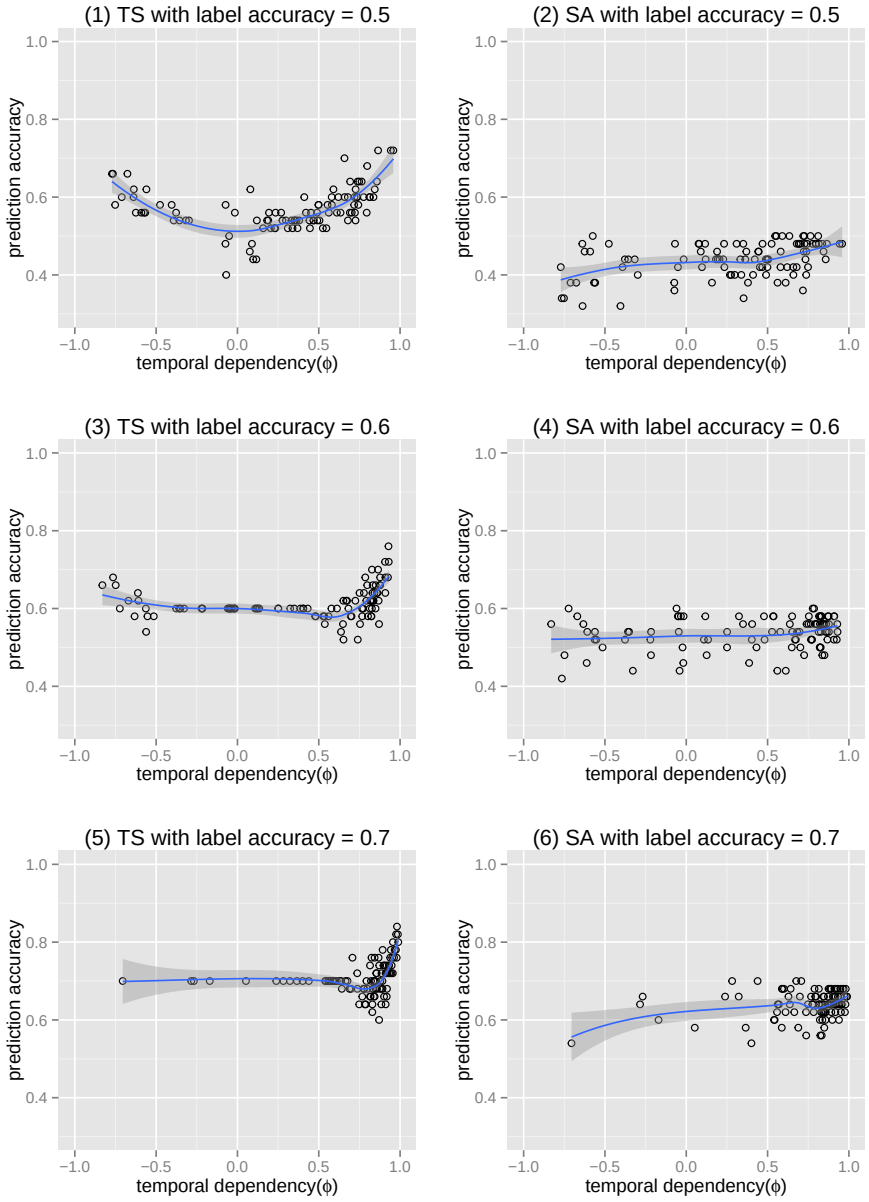

Figure 3.9: Temporal correlation (ϕ) vs. prediction accuracies of workers' label correctness. The x-axis indicates the temporal dependencies (ϕ) of crowd workers' label correctness and the y-axis indicates prediction accuracies. Prediction accuracy improvement of time-series model (TS) becomes larger when the absolute value of temporal dependencies ($|\phi|$) increases.

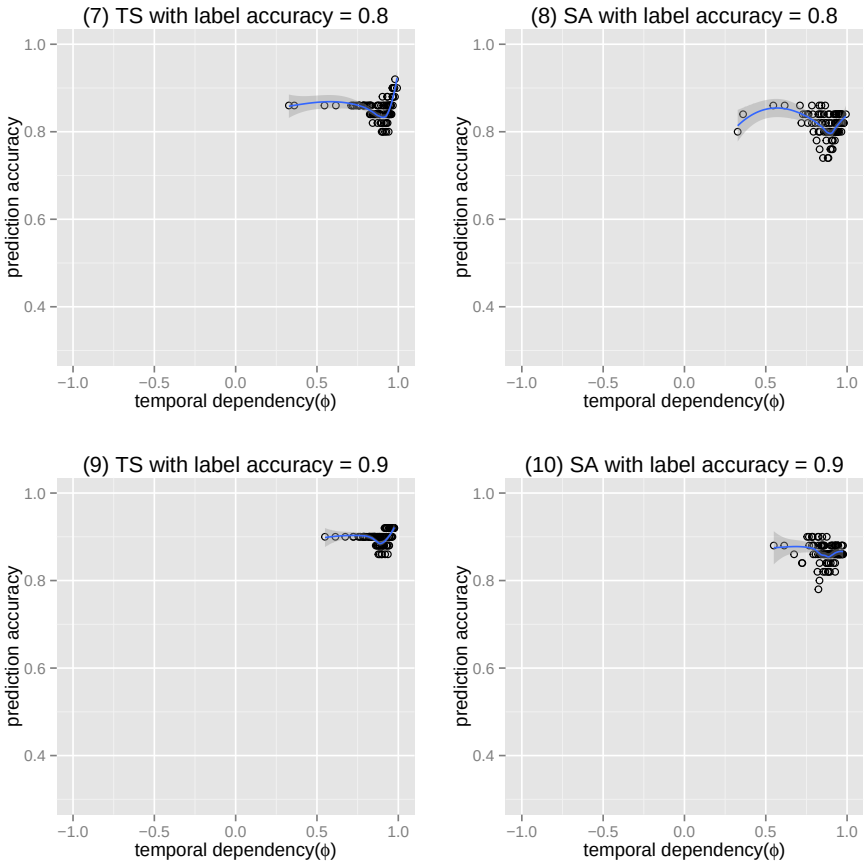

Figure 3.10: Temporal correlation (ϕ) vs. prediction accuracies of workers' label correctness. The x-axis indicates the temporal dependencies (ϕ) of crowd workers' label correctness and the y-axis indicates prediction accuracies. Prediction accuracy improvement of the time-series model (TS) becomes larger when the absolute value of temporal dependencies ($|\phi|$) increases.

to better measure crowd workers' label correctness. As both **Figure 3.9** and **Figure 3.10** show, each accuracy bin has a variety of crowd workers who show different temporal correlations. In particular, when crowd label accuracy is low (0.5 or 0.6), crowd workers show varying temporal correlation ϕ. Since the SA model is not able to capture such a temporal dynamic, it does not predict a crowd worker's label correctness accurately. On the other hand, the TS model allows us to measure a worker's label quality from a temporal perspective and build a better quality prediction model.

Second, the TS model improves its prediction accuracy of crowd workers whose absolute value of temporal correlation ϕ is close to 1. When a worker's label correctness tends to show strong temporal dependency over time (either or positive or negative), the predictive power increases significantly. The interpretation of having $|\phi| = 1$ is that a crowd workers' label correctness is predictable since it follows a certain temporal pattern. In the case of $\phi = 1$, this worker continuously generates correct or wrong labels over time. On the contrary, ϕ = -1 means that this worker tends to make a correct and wrong label in alternation. In sum, this result indicates that the TS model can effectively capture these temporal dynamics of crowd workers' label correctness which is not captured by the SA model.

3.5.2.2 Experiment 1.2 (RQ1.3): Prediction with Rejection

In the above section, we demonstrate the degree to which the TS model outperforms the SA model and why the TS model brings such improvement

Figure 3.11: Prediction accuracies of workers' next label correctness and its coverage across varying decision rejection options (δ=[0 0.25] by 0.05). The increase of δ improves the quality of the TS-based prediction while sacrificing the average number of predictions (coverage). In contrast, the coverage sacrifice of the SA-based predictions does not lead to the improvement of prediction accuracy. The TS-based prediction outperforms the SA-based prediction in terms of quality and coverage. Furthermore, decision reject options further improve the quality of TS-based prediction by trading-off prediction coverage.

without considering decision rejection. Our next question is to what extent decision reject options influence the quality of predicting workers' next labels? We conduct two experiments to examine the influence of varying the decision reject option parameter δ.

While further improvement of label predictions can be achieved by decision reject options, more conservative decisions not to predict naturally decrease the number of predictions made, as **Figure 3.11** shows. Without decision rejection, prediction coverage is 1.0 indicating 100%. However, increasing δ decreases prediction coverage since there are many ambiguous $logit^{-1}(x_t)$. For instance, in case of $\delta = 0.05$, both prediction models reject their predictions if $0.45 < logit^{-1}(x_t) < 0.55$. Therefore, the increase of δ naturally decreases the number of prediction labels. However, the accuracies substantially increase by rejecting uncertain predictions. In terms of accuracy, the proposed TS-based prediction improves its performance from 0.73 to 0.96 while the SA-based prediction does not achieve any performance improvement. The proposed model also shows similar quality improvement in terms of RMSE.

In addition to prediction coverage vs. prediction accuracy across varying decision reject options, we also investigate the effect of temporal dependencies (ϕ) on prediction accuracy with decision reject options. **Figure 3.12** shows the effect of decision reject options on prediction performance along with varying temporal dependencies (correlation, ϕ). The TS model outperforms the SA model in terms of prediction accuracy. Furthermore, the increase of decision reject option, δ (0.05 - 0.20) significantly improves the prediction

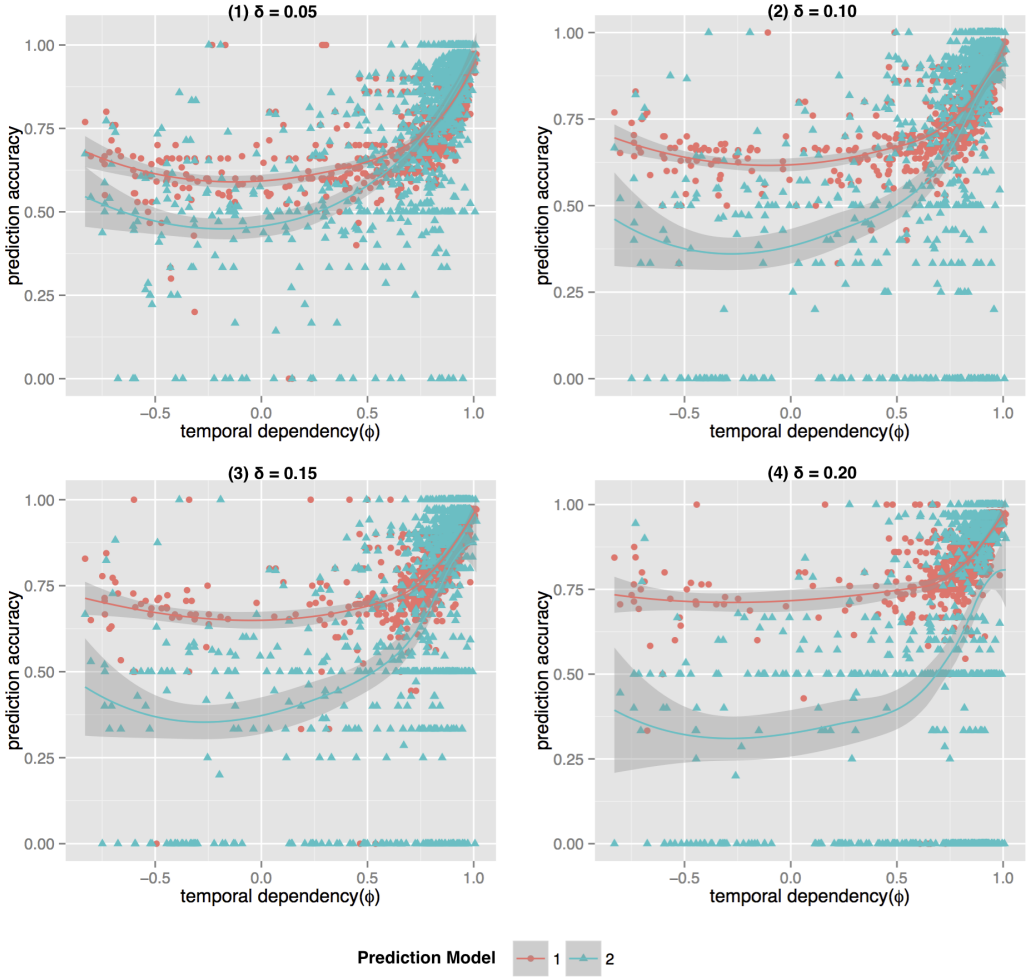

Figure 3.12: Temporal dependency (ϕ) vs. prediction accuracies of synthetic workers' label correctness across varying decision reject options (δ=[0 0.20] by 0.05). The x-axis indicates the temporal dependencies (ϕ) of crowd workers' label correctness and the y-axis indicates prediction accuracies. Model 1 indicates our proposed time-series model (TS) and Model 2 indicates sample accuracy-based prediction model (SA). As δ increases, the overall prediction accuracies of the TS-model increase while the SA-model does not bring such improvements. In particular, it is noticeable that the prediction accuracies of workers whose ϕ values are close to 0 increase significantly.

accuracy of the TS model when temporal correlations (ϕ) are close to 0. It suggests that decision reject options bring additional improvement of prediction accuracy by avoiding ambiguous predictions with a threshold δ.

3.5.3 Evaluation with Real Data

While a synthetic dataset allows us to evaluate the performance of the proposed time-series model with varying crowd label accuracies and temporal correlations, it still leaves a question of the effectiveness of the TS model on a real crowdsourcing dataset. In this section, we use the TREC crowdsourcing dataset discussed in Section 3.5.1.1 in order to further investigate the performance of the TS model.

3.5.3.1 Experiment 1.3 (RQ1.2): Prediction without Rejection

Experimental results on the public crowdsourcing dataset show similar performance to those of our synthetic dataset. **Figure 3.13** shows the comparison of two prediction models (TS and SA) over two metrics (accuracy and RMSE). Overall the TS model outperforms the SA model across all metrics and 49 workers. In particular, the TS model shows better prediction accuracies for 41 crowd workers and the same prediction accuracies for 8 crowd workers. Regarding the relationship between crowd label accuracy vs. prediction accuracy, the difference between the TS model vs. the SA model becomes smaller as crowd label accuracies increase. When a crowd worker's label accuracy is close to 0.5, the improvement of prediction accuracy tends to increase

substantially. In terms of the effect of temporal correlation, the TS model improves the prediction accuracies of crowd workers having $|\phi| \approx 1$. This result is the same as we observed in the experiment with our synthetic dataset. In sum, these results demonstrate that the understanding of temporal dynamics captured by the TS model leads to the improvement of predictive power.

Note that this prediction performance comparison does not consider any decision rejection options. In other words, both prediction algorithms use all predicted labels even though there exist many fewer confident predictions. In the following experiment, we investigate the effect of decision reject options on the prediction performance of models.

3.5.3.2 Experiment 1.4 (RQ1.3): Prediction with Rejection

We conduct two experiments to examine the influence of the decision reject option parameter δ on the TREC crowdsourcing dataset. First, we investigate the effect of the decision reject option on prediction accuracy under varying temporal correlations (ϕ). Second, we also investigate the tradeoff between prediction accuracy vs. coverage along with varying decision rejection options (δ).

Figure 3.14 shows the effect of decision reject options on prediction performance along with varying the temporal correlation of workers' label correctness. The x-axis indicates the temporal dependencies (correlation ϕ); the y-axis shows prediction accuracies. Results show that as decision reject option parameter δ increases, the gap between the two models becomes larger. In

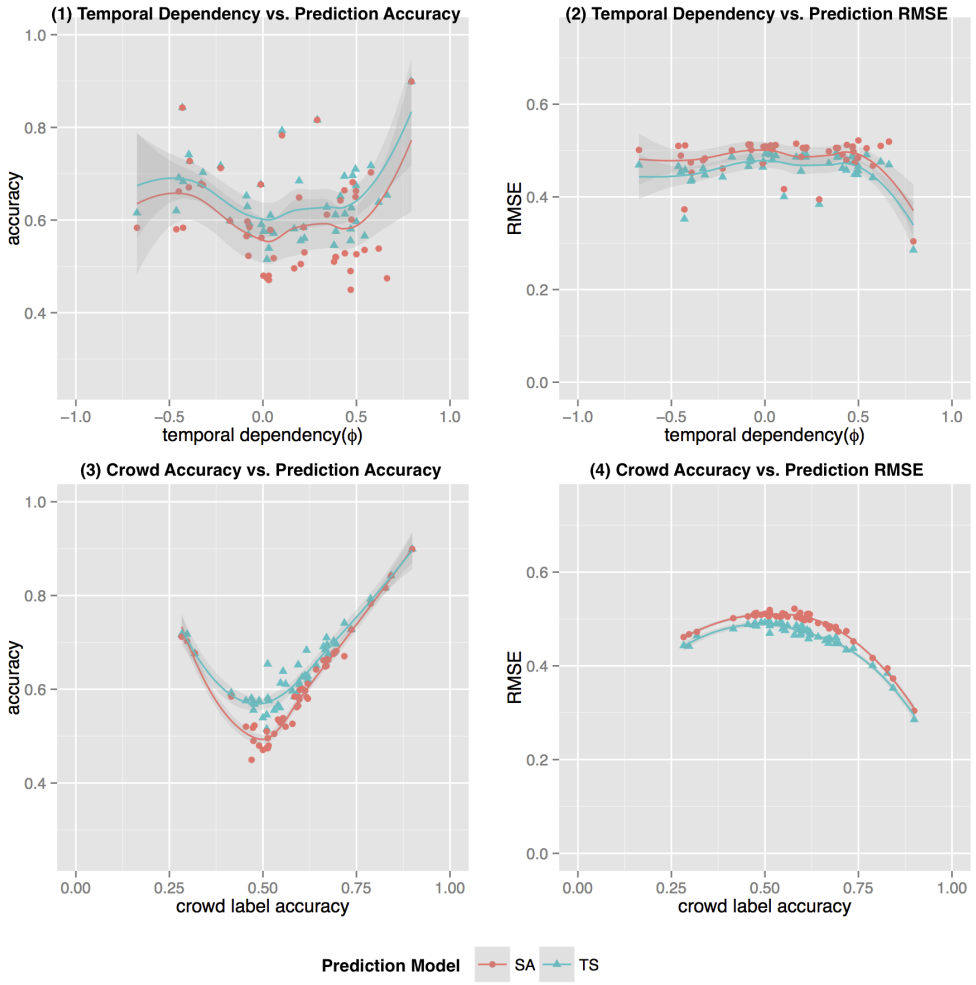

Figure 3.13: Figure (1) and (2) show the difference of prediction quality (accuracy and RMSE) between TS and AS across temporal dependencies ranging from -1 (frequent change) to 1 (infrequent change). Figure (3) and (4) shows the difference of prediction quality (accuracy and RMSE) between TS vs. SA across crowd label accuracies ranging from 0 to 1. TS indicates our proposed time-series model and SA refers to sample accuracy-based prediction model. Accuracy improvement by time-series model (TS) tends to be large when crowd label accuracies are close to 0.5 (random). As crowd label accuracies increase, the difference between TS vs. SA becomes smaller. In terms of temporal dependencies, TS model shows better prediction quality when the absolute values of temporal dependencies $|\phi|$ becomes close to 1.

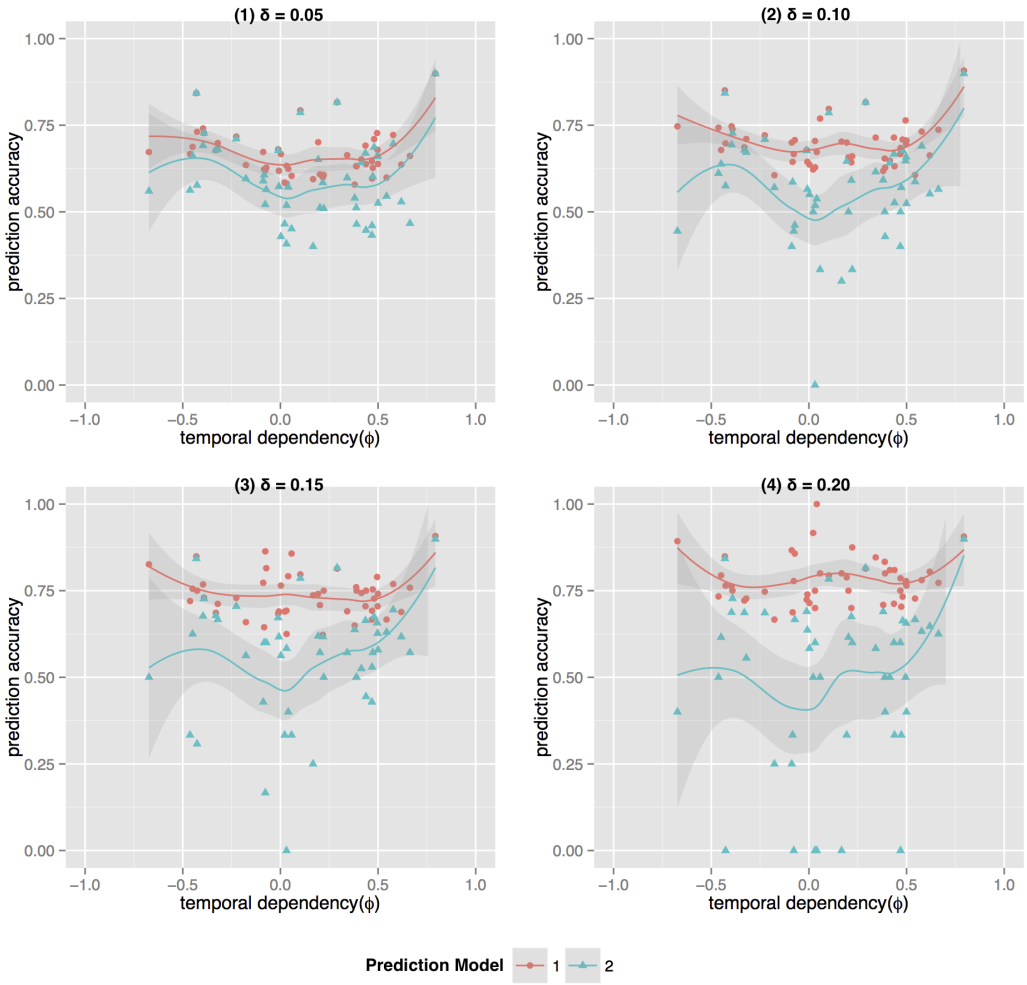

Figure 3.14: Effect of decision reject options (δ) on prediction accuracy under varying temporal correlations (ϕ). As decision reject option δ increases, the prediction accuracies of TS model improve substantially. However, the SA model does not show such improvement since it is not affected by decision reject options. In terms of temporal correlation, decision reject options (ϕ) further improve the prediction accuracies of the TS model when the absolute value of a crowd worker's temporal correlation (dependency) is close to 0.

particular, the TS model shows further improvement of prediction accuracies when crowd workers' temporal correlations ϕ are close to 0. Decision rejection options get rid of label predictions with less confidence. Therefore, overall prediction performance improves with increased rejection parameter δ.

SA-based prediction performance does not improve since the sample accuracy does not reflect the dynamics of workers' correct/wrong patterns. In other words, the oscillation of running accuracy becomes smaller over time and thus the SA model is not able to capture the dynamics. On the contrary, TS-based prediction considers workers' correct/wrong pattern at each time point and therefore this model is able to predict a short-term label more accurately than the SA-based predictions. In addition, decision rejection options even lead to the further improvement of predicted label quality by the predicted time series model.

While further improvement of label predictions can be achieved by decision reject options, more conservative decisions not to predict naturally decrease the number of predictions made, as **Figure 3.15** shows. The increase of δ reduces the number of predictions since there are many ambiguous $logit^{-1}(x_t)$. For instance, in case of $\delta = 0.05$, two prediction models reject their predictions if $0.45 < logit^{-1}(x_t) < 0.55$. Therefore, the increase of δ naturally decreases the number of prediction labels (coverage). However, the accuracies substantially increase by rejecting uncertain predictions. In terms of accuracy, the proposed TS-based prediction improves its performance from 0.65 to 0.82 while the RA-based prediction does not achieve any performance

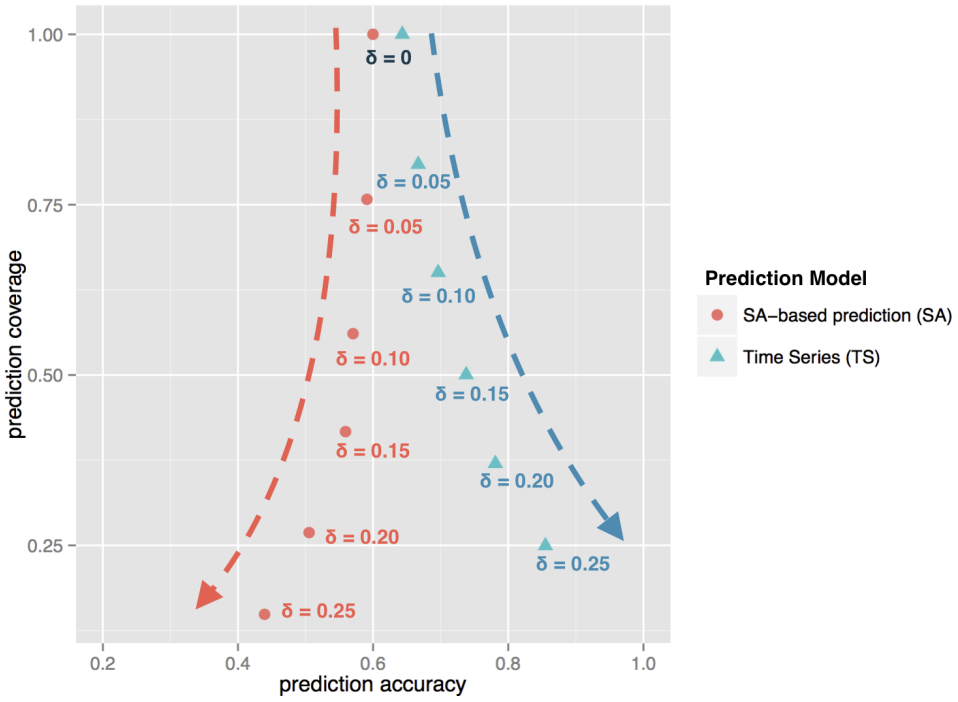

Figure 3.15: Prediction accuracies of workers' next label correctness and its coverage across varying decision rejection options (δ=[0 0.25] by 0.05). The increase of δ improves the quality of the TS-based prediction while sacrificing the average number of predictions (coverage). In contrast, the coverage sacrifice of the SA-based predictions does not lead to the improvement of prediction accuracy. The TS-based prediction outperforms the SA-based prediction in terms of quality and coverage. Furthermore, decision reject options further improve the quality of TS-based prediction by trading-off prediction coverage.

improvement. Besides, our proposed model shows similar quality improvement in terms of F1 score except with the highest setting of $\delta = 0.25$.

Predicion method	TS	SA	Original Label
RMSE	0.27**	0.34*	0.45
Accuracy	0.80**	0.71*	0.59

Table 3.2: Label quality over ground truth. Decision rejection option was set $\delta = 0.2$. (**) indicates that TS-based prediction method outperforms the other two methods with high statistical significance ($p<0.05$). (*) indicates that SA-based method outperforms the quality of original labels with high statistical significance ($p<0.05$).

3.5.3.3 Experiment 1.5 (RQ1.4): Label Quality Improvement

The previous two experiments showed our TS model better predicting the correctness of each worker's next label than the baseline (SA-based prediction). Moreover, we demonstrated that decision reject options further improve the prediction performance by avoiding less confident predictions.

Next, we conduct an experiment on the quality of crowdsourced labels over ground truth generated by expert annotators. We measure three metrics (RMSE and accuracy) in order to compare the quality of actual labels generated by two prediction methods (TS-based prediction vs. SA-based prediction) to original labels collected from workers without any task recommendation. For the experiment, we use *soft* (i.e., probabilistic) labels for RMSE and *hard* (i.e., rounded, binary) labels for accuracy (*proper scoring rules* (Gneiting and Raftery, 2007)). In addition, we conduct a paired-sample t-test in order to assess significance between results of the prediction methods. A decision reject option ($\delta = 0.2$) is used for this experiment. Each score indicates average prediction score across all workers. For simplicity, we do not consider any

aggregation methods.

Table 3.2 shows prediction scores of each method with respect to ground truth. Temporal modeling is seen to outperform the baseline (SA-based prediction) by 10-20% and significantly improve upon original labels by 20-30%. This suggests that label generation via our time-series prediction model leads to quality improvement of crowdsourced labels.

3.6 Conclusion and Future Work

Predicting the correctness of workers' next label can helpfully support successful task recommendation in crowdsourcing. While the existing studies make i.i.d. assumption in terms of analyzing the patterns of crowd workers' label correctness for finding the best worker, we propose a time-series prediction model in order to take account of the dynamics of workers' temporal patterns.

Our experiments with a synthetic dataset and a public datastset demonstrate that the proposed model not only predicts the actual workers' label correctness more accurately (RQ1.2) but also improves the quality of crowdsourced labels over ground truth (RQ1.4). In addition, we show that decision reject options would be an useful method to manage uncertainty of prediction (RQ1.3). In particular, TS-based prediction with decision reject options outperforms SA-based prediction with them in terms of prediction quality and coverage.

While our experiments with a public crowdsourcing dataset have some

limitation of its external validity, our empirical analysis with a synthetic dataset allows us to delve into the detailed answer to our RQ1.1. In particular, we presents how and why the understanding of temporal correlation ϕ brings benefit to better quality prediction. This finding presents a promising direction of time-series modeling to improve crowd work quality.

While this study opens a new horizon of time-series model for predicting crowd workers' label correctness, there is still room for further improvement. One direction to extend this study would be to consider multiple features about the crowd assessor for better quality prediction. In addition, we also consider how to learn our models with limited supervision. The following sections will further investigate and develop these ideas.

Chapter 4

Generalized Assessor Model

In this section, we aim at modeling crowd assessor accuracy with an extension of our time-series model. While the previous chapter presents a time-series prediction model based on the understanding of temporal dynamics of crowd work quality, it still relies on single generative framework for prediction. We now describe a generalizable feature-based assessor model that allows us to flexibly capture a wider range of assessor behaviors by incorporating features which model different aspects of this behavior. Based on the model, we adopt a learning framework for predicting assessor accuracy over time. [1]

4.1 Introduction

While estimating and predicting crowd assessors' performance may bring benefits of quality improvement, it has gained relatively little attention in IR evaluation. Most prior work in crowd assessor modeling has focused on simple estimation of assessors' performance via metrics such as accuracy and F1 (Kazai, 2011; Smucker and Jethani, 2011). Unlike other studies, Caterette

[1] This chapter is based on the published work (Jung and Lease, 2015a) in the European Conference of Information Retrieval 2015, which is guided by a co-author, Matthew Lease.

and Soboroff presented several assessor model based on Bayesian style accuracy with various types of Beta priors (Carterette and Soboroff, 2010). Recently, Ipeirotis and Gabrilovich presented a similar type of Bayesian style accuracy with a different Beta prior in order to measure assessors' performance (Ipeirotis and Gabrilovich, 2014). However, neither investigated prediction of an assessor's judgment quality.

In crowdsourcing and human computation, some research has focused on the estimation or prediction of crowd workers' behavior or performance (Raykar and Yu, 2012; Rzeszotarski and Kittur, 2011); however, most studies assumed that each annotation is independent and identically distributed (i.i.d.) over time while crowd worker behavior can have temporal dynamics as shown in **Figure 4.1**. While our previous chapter presents a time-series model, this model still relies upon a single generative feature for prediction. For this reason, our previous time-series model remains limited in terms of predicting an assessor's next judgment quality, as shown in **Figure 4.1**.

For this problem, we propose a generalizable feature-based assessor model (GAM) that allows us to flexibly capture a wider range of assessors' behaviors by incorporating features which model different aspects of this behavior. We integrate various features from prior studies which were used mainly for the estimation of a crowd assessor's annotation performance in the previous chapter or judgment simulation (Carterette and Soboroff, 2010). In addition, we devise several behavioral features indicating an assessor's annotation performance over time and integrate them with the features selected from prior

studies. Based on this model, we build a predictive model for an assessor's next judgment quality. By this ability to flexibly model more aspects of assessor behavior, we expect greater predictive power and an opportunity for more accurate predictions.

We investigate this predictive model from three viewpoints. In the first experiment, we evaluate the prediction performance of the proposed model with the similar evaluation methodology as we used in the previous chapter. We measure prediction accuracy and MAE and investigate the effect of a decision reject option. Second, we conduct an in-depth feature analysis in order to find which features most influence success of our predictive model as well as present an optimal feature selection. Finally, we evaluate the effectiveness of the proposed prediction model for crowdsourced judgment quality improvement under a realistic scenario assuming task routing and label aggregation. Empirical evaluations demonstrate that the proposed model improves prediction accuracy by 15-47% across 49 qualified assessors. In addition, our experiments show that the quality of relevance judgments by the proposed prediction model-based task routing improves its accuracy by 17-47% with 54-83% of the original assessment cost. The research questions of this study are as follows:

RQ2.1: Prediction Performance Improvement *To what extent does the proposed prediction model improve prediction performance? Why does prediction performance improve? How does decision rejection balance coverage vs. accuracy of the proposed prediction model?*

RQ2.2: Relative Feature Importance *How can we formulate a discriminative, feature-based learning framework for predicting work quality, what features would be useful to include, and what is their relative importance?*

RQ2.3: Impact on judgment quality and cost. *Can the proposed prediction model improve the quality of relevance judgments and/or decrease the cost of collecting judgments?*

4.2 Problem

Figure 4.1 shows two real examples of failures of existing assessor models in predicting assessor's judgment quality. The more accurate left assessor (a) begins with very strong accuracy (0.8) which continually degrades over time, whereas accuracy of the right assessor (b) hovers steadily around 0.5. Suppose that a crowd worker's next label quality (y_t) is binary (correct/wrong) with respect to ground truth. While y_t oscillates over time, the existing models (Carterette and Soboroff, 2010)(Ipeirotis and Gabrilovich, 2014) are not able to capture such temporal dynamics and thus prediction based on these models is almost always wrong. In particular, when an assessor's labeling accuracy is greater than 0.5 (e.g. avg. accuracy = 0.67 in **Figure 4.1 (a)**), the prediction based on the existing models are always 1 (correct) even though the actual assessor's next label quality oscillates over time. A symmetric problem happens in **Figure 4.1 (b)** with another worker whose average accuracy is below 0.5.

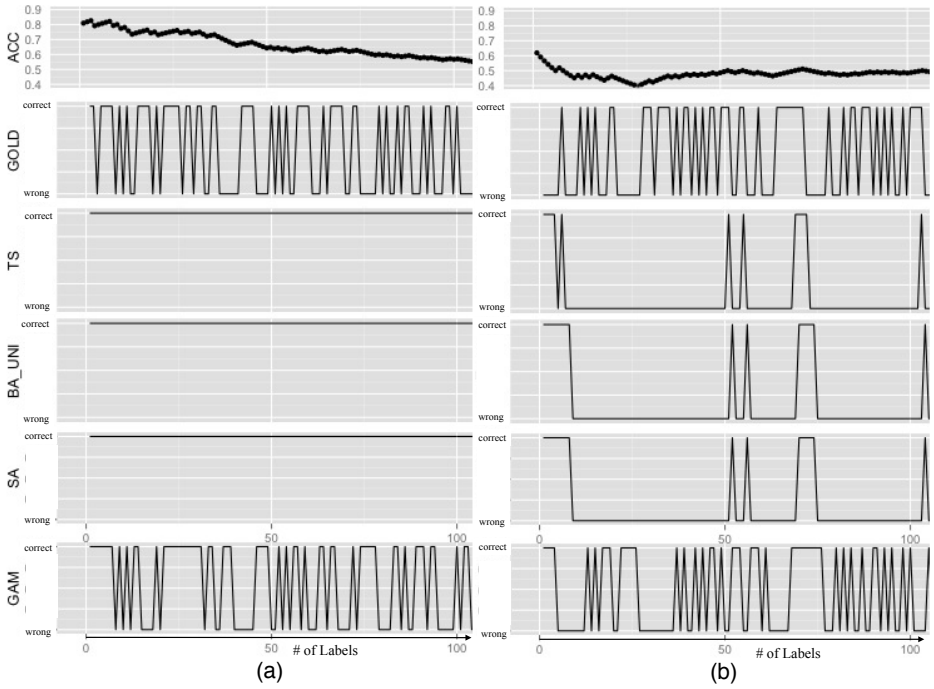

Figure 4.1: Two examples of failures of existing models and success of GAM in predicting assessors' next label correctness ((a) high accuracy assessor and (b) low accuracy assessor). While an actual assessor's next label correctness (GOLD) oscillates over time, the existing assessor models (Time-series (TS)), Sample Accuracy (SA), Bayesian uniform beta prior (BA-UNI (Ipeirotis and Gabrilovich, 2014)) do not track the temporal variation of the gold labels since they are not capable of modeling it. In contrast, the proposed model, GAM, is very sensitive to such dynamics of labels over time for higher quality prediction.

4.2.1 Problem Setting

Suppose that an assessor has completed n relevance judgments. As in Chapter 3, the correctness of the ith judgment is denoted as $y_i \in \{0, 1\}$, where 1 and 0 represent correct or not. Thus, the performance of an assessor can be represented as a sequence of binary observations, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$.

For example, if an assessor completed five relevance judgments and erred on the first and third respectively, then his *binary performance sequence* is encoded as $\mathbf{y} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$. GOLD in **Figure 4.1** indicates the $\mathbf{y}$ value of each assessor. Next, we generate a multi-dimensional feature vector, $\mathbf{x_i} = \begin{bmatrix} x_{1i} & x_{2i} & \dots & x_{mi} \end{bmatrix}$ per time i and use x_i as an input of a prediction function f. At this point, prior assessor models only consider a simple feature measure x_i by a single metric, accuracy, and then use this feature an input of a simple function $y_{i+1} = \mathit{roundOff}(x_i)$. While the previous chapter proposed to use a decision reject option instead of the existing simple round off function, it sacrifices prediction coverage. Instead, the proposed model incorporates a multi-dimensional feature vector $\mathbf{x_i}$ and uses this feature vector with a learning framework $f(x_i, y_i) = y_{i+1}$. The bottom plot of **Figure 4.1** shows how GAM is able to track the assessor's varying correctness with greater fidelity.

4.3 Generalized Assessor Model (GAM)

In this section, we describe the GAM model, which incorporates various observable and latent features modeling different aspects of assessors' behaviors. Feature generation and integration are examined and then learning a predictive model with the generated features is discussed.

4.3.1 Feature Generation and Integration

An assessor's behavior and annotation performance may be captured by various types of features. In this study, we generate and integrate two

	Feature Name	Description
Observable	Bayesian Optimistic Accuracy (BA_{opt}) [1]	a Bayesian style accuracy with a prior $Beta$ (16,1) $BA_{opt} = (x_t + 16)/(n_t + 17)$
	Bayesian Pessimistic Accuracy (BA_{pes}) [1]	a Bayesian style accuracy with a prior $Beta$ (1,16) $BA_{pes} = (x_t + 1)/(n_t + 17)$
	Bayesian Uniform Accuracy (BA_{uni}) [2]	a Bayesian style accuracy with a prior $Beta$ (0.5,0.5) $BA_{uni} = (x_t + 0.5/(n_t + 1)$
	Sample Running Accuracy (SA)	$SA_t = x_t/n_t$
	CurrentLabelQuality	a binary value indicating whether a current label is correct or wrong.
	TaskTime	time to spend in completing this judgment task. (ms)
	AccuracyChangeDirection (ACD)	a binary value indicating the absolute difference between SA_{t-1} - SA_t.
	TopicChange	a binary value indicating a topic change between time $t-1$ and time t.
	NumLabels	a cumulative number of completed relevance judgments at time t.
	TopicEverSeen	a real value [0∼1] indicating the familiarity of a topic. $\frac{1}{a\ number\ of\ judgments\ on\ topic\ k\ at\ time\ t}$
Latent	Asymptotic Accuracy (AA)	a time-series accuracy estimated by latent time-series model proposed in Chapter 3 $\frac{c}{1-\phi}$.
	ϕ	a temporal correlation indicating how frequently a sequence of correct/wrong observations has changed over time.
	c	a variable indicating the direction of judgments between correct and wrong.

Table 4.1: Features of generalized assessor model (GAM). n is the number of total judgments and x is the number of relevance judgments at time t. [1] indicates (Carterette and Soboroff, 2010) and [2] indicates (Ipeirotis and Gabrilovich, 2014)

types of features: observable and latent features. Bayesian-style features have various forms in prior work according to different Beta prior settings. Among them, we adopt *optimistic* (a Beta prior $\alpha = 16, \beta = 1$) and *pessimistic* (a Beta prior $\alpha = 1, \beta = 16$) assessor models from Carterette and Soboroff's study (Carterette and Soboroff, 2010). In addition, we adopt a Bayesian style accuracy from Ipeirotis and Gabrilovich's study which assumes a Beta prior ($\alpha = 0.5, \beta = 0.5$) named $uniform$ assessor model. In these assessor models, each Beta prior characterizes each assessor's annotation performance. For instance, the *optimistic* assessor model indicates that an assessor is likely to make a relevance judgment in a permissive fashion, while the *pessimistic*

model tends to make more non-relevant judgments than relevant judgments. The $uniform$ model has an equal chance of making a relevant or non-relevant judgment. Note that Bayesian style accuracies (BA_{opt}, BA_{pes}, BA_{uni}) were only used as a way of simulating judgments or estimating an assessor's performance in the original studies. In this study, we instead used these accuracies as a feature of estimating an assessor's annotation performance as well as predicting an assessor's next judgment's quality. Other observable features include measurable features from a sequence of relevance judgments from an assessor. Among them, $TaskTime$ and $NumLables$ are designed to capture an assessor's behavioral transition over time. In addition, we leverage the topic of a given task, which is predefined by TREC dataset. For instance, a topic in our dataset is a search query such as "growing tomatoes". Under a topic, we have multiple relevance judgment tasks for a set of pairs of the query and different documents. $TopicChange$ checks the sensitivity of an assessor to topic variation over time. $TopicEverSeen$ feature is designed in order to consider the effect of topic familiarity over time. The value is discounted by increased exposure to topic k.

Latent features are adopted from the proposed time-series model in Chapter 3. While the previous study only used *asymptotic accuracy* (AA) as a indicator of an assessor's annotation performance, we integrate all three features (AA, ϕ, and c) of the time-series model into our generalized assessor model. Our intuition of integrating various features is that each feature may capture a different aspect of an assessor's annotation performance and thus

the integration of various features gives rise to greater predictive power and an opportunity for more accurate predictions.

4.3.2 Predicting Judgments Quality

To select a learning model, we adopt L1 regularized logistic regression due to several reasons. First, it supports probabilistic classification as well as binary prediction by logistic function. In this problem setting, we conflate relevance judgments into binary values (0 or 1), and thus logistic regression is the best fit in order to handle such a binary classification problem. In addition, a logistic regression model allows us obtain the odds ratio, defined as the ratio of the probability of correct over incorrect relevance judgments. Second, L1-regularized logistic regression prevents over-fitting in learning models due to either co-linearity of the covariates or high-dimensionality. The regularized regression shrinks the estimates of the regression coefficients towards zero relative to the maximum likelihood estimate. Finally, L1-regularized logistic regression is relatively simple and fast. In practice, one of the challenging issues to run learning algorithms is that it takes too much time to update parameters and predict output values once a new label comes. However, this model is quite efficient.

In prediction, we consider a supervised learning task where N training instances $\{(x_i, y_i), i = 1, ..., N\}$ are given. Here, each $x_i \in \mathbb{R}^M$ is an M-dimensional feature vector, and $y_i \in 0, 1$ is a class label indicating whether an assessor's next judgment is correct (1) or wrong (0). Before fitting a model to

the features and target labels, we first normalize the features in order to ensure that normalized feature values implicitly weight all features equally in a model learning process. Logistic regression models the probability distribution of the class label y given a feature vector X as follows:

$$p(y = 1|x; \theta) = \sigma(\theta^T x) = \frac{1}{1 + exp(-\theta^T x)} \tag{4.1}$$

Here $\theta = \{\beta_0, \beta_1^T, ..., \beta_M^T\}$ are the parameters of the logistic regression models, and $\sigma(\cdot)$ is the sigmoid function, defined by the second equality. We then maximize the log-likelihood in order to fit a model to the given training data.

$$\max_{\theta}\{\sum_{i=1}^{N}[y_i(\beta_0 + \beta^T x_i) - log(1 + e^{\beta_0 + \beta^T x_i})] - \lambda \sum_{j=1}^{M} |\beta_j|\}. \tag{4.2}$$

4.3.3 Prediction with a Decision Reject Option

Our predictive model can generate two types of outputs: a probabilistic label ($y_{i+1} \in [0, 1]$) indicating the degree of polarity and a binary label (0 or 1). While binary labels (*hard label*) can be directly used as it is, probabilistic labels (*soft label*) can be used after a transformation, such as rounding off. For instance, if a predicted soft label is 0.76, we would round this to a binary label of 1. In term of soft label prediction, we can exploit prediction confidence. For instance, if a soft label is close to 0.5, it fundamentally indicates very low confidence in terms of the polarity. Therefore, we may avoid the risk of getting noisy predictions by adopting a decision rejection option (Pillai et al.,

2013). In this study, we round off a probabilistic label with a decision reject option as follows. If $y_{i+1} \leq 0.5 - \delta$ or $y_{i+1} \geq 0.5 + \delta$, then y_{i+1} does not need any transformation and we use its original value. If $y_{i+1} > 0.5 - \delta$ or $y_{i+1} < 0.5 + \delta$, then y_{i+1} is *null*, and we decide not to predict. $\delta \in [0, 0.5]$ is a parameter to control the limits of the decision reject option. High δ indicates a conservative label prediction, which increases the range of decision rejection while sacrificing coverage. On the other hand, low δ allows more aggressive label prediction, decreasing the threshold of decision rejection and increasing coverage.

4.3.4 Operational Flow of GAM

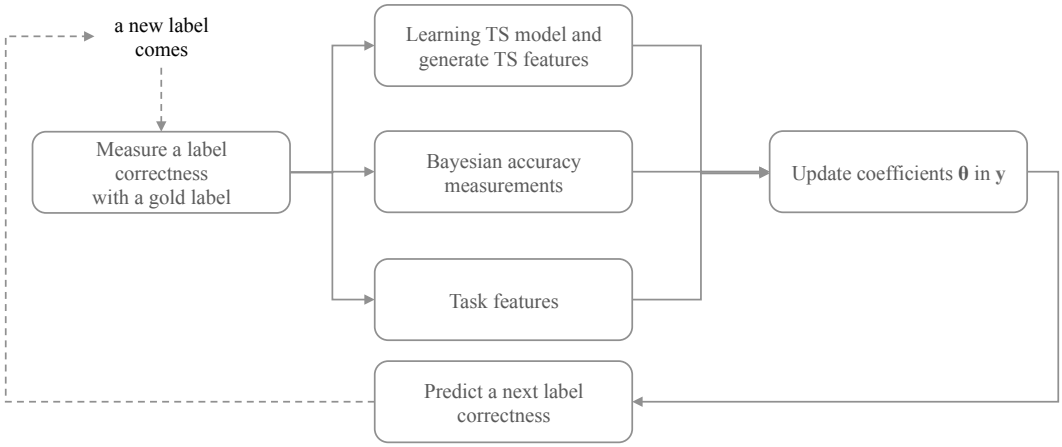

Figure 4.2: Operational flow of GAM learning process. Once a new label (a worker's label correctness) comes, all features are generated. Based on the feature, our learning model updates its coefficients and generates a predicted value of the worker's next label correctness.

Figure 4.2 summarizes an operational flow of GAM learning process. A GAM learning process begins with a new label. With a given new label,

GAM measures the correctness of the given label over its corresponding gold label. Next, three different feature sets are generated in parallel. To generate TS features, we adopt TS model proposed in the previous chapter. In the meantime, three different Bayesian accuracies are measured and the rest of task-dependent features are generated. Once all of the features and its corresponding label correctness are ready, GAM updates its coefficients θ and predict a next label correctness. This process repeatedly occurs once a new label comes.

4.4 Evaluation

4.4.1 Experimental Settings

4.4.1.1 Metrics and Dataset

We evaluate the performance of our prediction model with two metrics. Firstly, we measure the prediction performance with Accuracy and Mean Absolute Error (MAE). Probabilistic labels (soft labels) produced by our assessor model are measured with MAE, indicating the absolute difference between a predicted label vs. ground truth: $MAE = \frac{1}{n}\sum_{i=1}^{n}|pred_i - gold_i|$. Rounded binary labels (hard labels) are evaluated by accuracy, defined as follows, where tp denotes the number of true positive classifications, fp the false positives, tn the true negatives, and fn the false negatives: $accuracy = \frac{tp+tn}{n}$. Secondly, we also evaluate the effect of our prediction method on the quality of relevance judgments with accuracy defined in the above. Since our extracted dataset is well-balanced in terms of a ratio between relevant vs. non-relevant judgments,

use of accuracy is appropriate. Data from the NIST TREC crowdsourcing track 2011 Task 2 is used, as defined in Chapter 3.

4.4.1.2 Models

We evaluate the proposed Generalized Assessor Model (GAM) under various conditions of decision reject options with two metrics. Our initial model uses no decision reject option, setting $\delta = 0$. In order to examine the effect of decision reject options, we vary $\delta \in [0, 0.25]$ by 0.05 step-size. Since we have 49 workers, we build 49 different predictive models and evaluate prediction performance and final judgment quality improvement.

For the prediction of a worker's next label correctness, we adopt the same strategy as Chapter 3 does. We use each worker's first 20 observed labels for training an initial individual model for each worker; we then update the parameter of each learned model once a new label comes.

As a baseline, we adopt our times-series model in Chapter 3 and several assessor models proposed by prior studies (Carterette and Soboroff, 2010) (Ipeirotis and Gabrilovich, 2014). We adopt two assessor models from Carterette and Soboroff's study, *optimistic* assessor (BA_{opt}) and *pessimistic* assessor (BA_{pes}), and one assessor model of Bayesian accuracy (BA_{uni} used in Ipeirotis and Gabrilovich's study (see **Table 4.1**). In addition, we test the performance of the time-series model (TS) proposed in Chapter 3, and sample running accuracy (SA) as defined by **Table 4.1**. All of the baseline methods predict next judgment quality y_{i+1} based on the accuracy at time i by rounding off.

Metric	GAM	TS	$\mathbf{BA}_{uni}$	$\mathbf{BA}_{opt}$	$\mathbf{BA}_{pes}$	SA
Accuracy	0.782**	0.643*	0.593	0.603	0.531	0.593
NumWins in Accuracy	NA	44	47	47	49	48
NumTies in Accuracy	NA	4	1	2	0	1
NumLosses in Accuracy	NA	1	1	0	0	0
MAE	0.338**	0.387	0.449	0.439	0.482	0.448
NumWins in MAE	NA	44	49	49	49	49
NumLosses in MAE	NA	5	0	0	0	0

Table 4.2: Prediction performance (Accuracy and MAE) of different predictive models. **NumWins** indicates the number of assessors for which GAM outperforms a baseline method, while **NumLosses** indicates the opposite. **NumTies** indicates the number of assessors that a given method shows the same prediction performance as GAM for an assessor. (**) indicates that GAM prediction outperforms the other six methods with a high statistical significance ($p<0.01$). (*) indicates that a prediction method outperforms SA with a statistical significance ($p<0.05$).

Decision reject options are equally applied to all of the baseline methods. To learn the L1-regularized logistic regression model, we set the regularization parameter $\lambda = 0.01$ based on evaluation on the first 20 observed labels across workers with a sweep of $\lambda \in \{0.001, 0.01, 0.1\}$. For feature normalization, we apply standard min-max normalization, $\frac{max-feature}{max-min}$, to the 13 features proposed in the section 4.3.1. Note that λ is the only model parameter we tune, and all settings of the decision-reject parameter are reported in the results.

4.4.2 Experiment 2.1 (RQ2.1): Prediction Performance Improvement

To answer our first research question, we compare the overall prediction performance (Accuracy, MAE) of GAM with the baseline models across

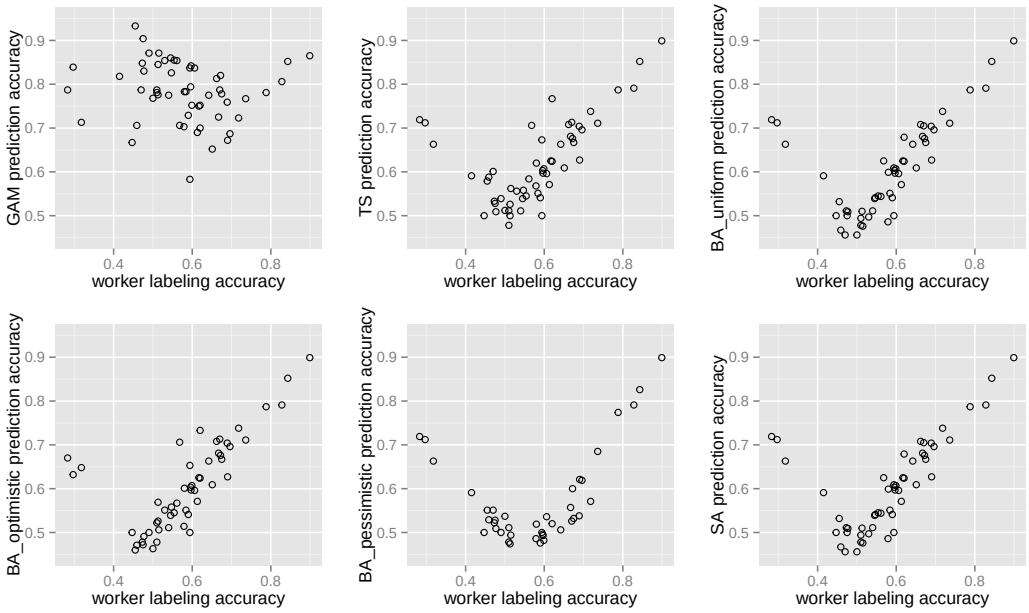

Figure 4.3: Prediction accuracy of workers' next labels by different methods. While the other methods show low accuracy against assessors whose labeling accuracy are near 0.5, the proposed model (GAM) shows significant improvement of predicting those workers' next judgments.

49 crowd assessors. **Table 4.2** shows that GAM prediction performance outperforms all of the baseline methods across 44-49 assessors in accuracy and 44-49 assessors in MAE. GAM improves the prediction accuracy (hard label) and MAE (soft label) by 15-47% on average. GAM prediction performs worse for only one of the assessors. In particular, we observe that GAM prediction performance improves very quickly as the number of cumulative judgments increases while the baseline methods only slowly improve.

Figure 4.3 shows the relationship between crowd assessors' label accuracies (sample accuracy) vs. prediction accuracy of GAM and the baseline

models. While the other models show low accuracy against assessors whose labeling accuracy is near 0.5, GAM significantly improves prediction error for those assessors in particular.

Finally, we examine the effects of decision reject options on GAM prediction. **Figure 4.4** demonstrates that the baseline models show sharp decline of coverage in order to improve prediction accuracy. However, the coverage of GAM prediction only gently decreases; even with the strongest reject option ($\delta = 0.25$), it still covers the half of prediction. In sum, GAM prediction not only outperforms baseline models in terms of prediction accuracy, but also shows less loss in coverage when using the decision reject option.

4.4.3 Experiment 2.2 (RQ2.2): Feature Selection & Importance

Our next experiment is to figure out individual GAM prediction models across 49 workers and which features are relatively more important than others. Intuitively, having more features leads to more predictive power. However, in practice, excessive features may lead to overfitting. Thus, we investigate relative feature importance by evaluating feature subsets.

We adopt the *bestglm* r package [2] and run the BICg model in order to find the best subset regression models. Since we have 49 assessors, we run this method for all the 49 original regression models. Next, we observe the selected features of each subset model, and count the cumulative selection of

[2]http://cran.r-project.org/web/packages/bestglm

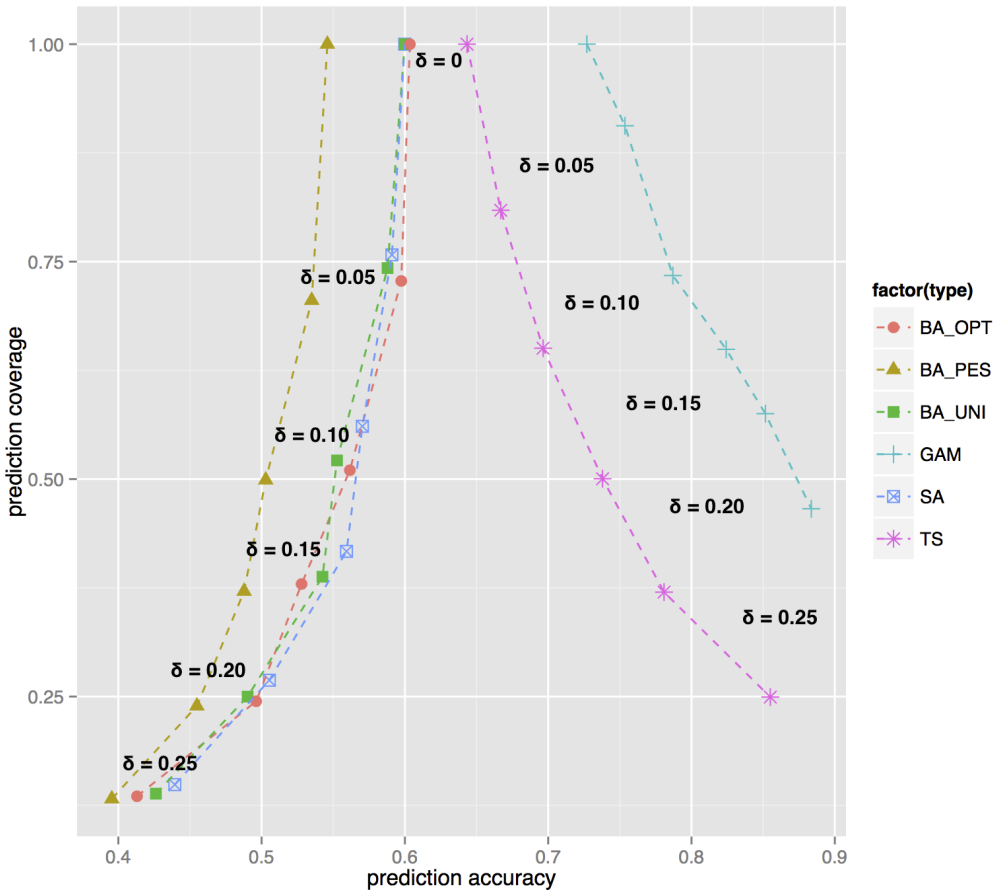

Figure 4.4: Prediction accuracies of assessors' next judgments and corresponding coverage across varying decision rejection options (δ=[0∼0.25] by 0.05). While the other methods show a significant decrease in coverage, under all the given reject options, GAM shows better coverage as well as prediction performance.

each feature across 49 regression models. **Figure 4.5** shows the relative feature importance across 49 regression models for all of the assessors. Asymptotic accuracy (AA) is selected in 47 of 49 models, followed by BA_{opt} and BA_{pes} at 42 and 38, respectively. *Numlabels* is selected in almost half of the cases

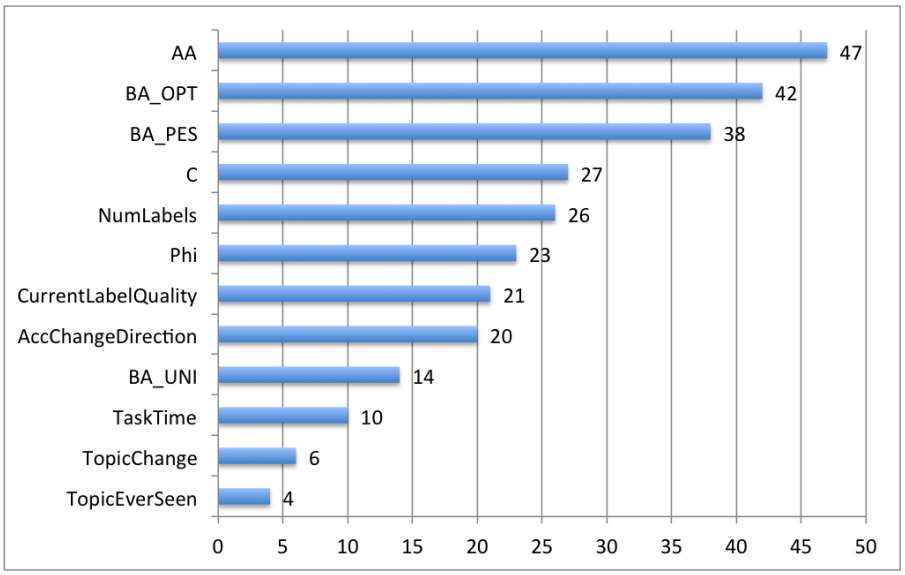

Figure 4.5: Relative feature importance across 49 regression models.

(26), which implicitly indicates that task familiarity affects an assessor's next judgment quality. Features based on our time-series model (C and phi) from Chapter 3 are also selected in the half of the models. On the contrary, prediction quality appears insensitive to topic change and topic familiarity. In addition, this result suggests that sample accuracy (SA) is not relatively important feature to GAM prediction. Interestingly, a GAM model with only the top five features still shows little degraded performance (7-10% less) vs. the original regression models and outperforms all of the baselines.

4.4.4 Experiment 2.3 (RQ2.3): Impact on judgment quality and cost

Our last experiment is to examine quality effects on relevance judgments via the proposed prediction model. We conduct an experiment based

	Task routing							No Routing
NumJudge	**GAM**	**TS**	**BA$_{uni}$**	**BA$_{opt}$**	**BA$_{pes}$**	**SA**	**Random**	**All labels**
1	0.786**	0.671	0.577	0.585	0.557	0.568	0.555	0.592
2	0.817**	0.687*	0.591	0.592	0.572	0.581	0.571	
3	0.877**	0.708*	0.610	0.624*	0.596	0.605	0.582	

Table 4.3: Accuracy of aggregated relevance judgments via predictive models. Majority voting is used for all the prediction methods. Accuracy is measured against NIST expert gold labels. **NumJudge** indicates the number of judges per query-document pair. Avg. number of judges per query-document pair is almost 3.7. (**) indicates that GAM prediction outperforms the other six methods with very high statistical significance ($p<0.01$). (*) indicates that a prediction method outperforms the quality of aggregated labels with all labels with high statistical significance ($p<0.05$).

on task routing. For instance, if the prediction of an assessor's next judgment quality on a particular example indicates that the assessor is expected to be correct, we route the given topic-document pair to this assessor and measure actual judgment quality against ground truth labeled by NIST. From our dataset, we only use 826 topic-document pairs that have more than three judgments per topic-document pair. Since the avg. number of judges per query is about 3.7, we test the effect of cost saving under three task routing scenarios ($NumJudge = \{1, 2, 3\}$) indicating lower labeling cost. Judgment quality is measured with accuracy, and a paired t-test is conducted to check whether quality improvement is statistically significant.

Table 4.3 shows the results of judgment quality via predictive model-based task routing. GAM substantially outperforms the other baselines across three task routing cases. The improvement of final judgment quality grows with the increase of the number of judges per query-document pair (*Num-*

Judge) from 17-32% to 23-47%. Notice that GAM with only two routed judges achieves 17% quality improvement. Moreover, GAM provides high-quality relevance judgments (accuracy > 0.8) with only $54\% = (\frac{2}{3.7})$ of the original assessment cost. Additionally, we see that task routing with baselines alone (BA_{uni},BA_{pes},SA) may not be any better than random assignment.

4.5 Conclusion

In general, it is widely known that having more features brings a better predictive power when building a prediction model. However, prior work in crowd assessor modeling relies only a single feature, which limits its predictive power. To address this, we propose a general discriminative learning framework for integrating arbitrary and diverse evidence for temporal modeling and prediction of crowd work accuracy.

Our experiments with a public crowdsourcing dataset answers our three research questions. First, our empirical results demonstrates that GAM outperforms the other baseline models in terms of prediction performance (RQ2.1). Second, our following analysis answers the reason of GAM's huge improvement of prediction quality. In particular, the in-depth feature analysis shows which feature is relatively more important than the other features (RQ2.2). Finally, GAM demonstrates that it can improve relevance judgment quality as well as cost saving (RQ2.3). Based on a simple task routing scenario, we demonstrate that GAM significantly improves the quality of relevance judgments with fewer number of workers per query-document pair.

To sum up, this study suggests that GAM can improve the predictive power of TS model by considering multi-dimensional features about a crowd worker. Feature design and generation would be a critical role for improving a prediction model of crowd worker quality.

Chapter 5

Temporal Modeling of Crowd Work Quality without Supervision

While the two proposed models in Chapter 3 and 4 have shown that a worker's performance can be more accurately modeled by temporal correlation in task performance, a fundamental challenge remains in the need for expert gold labels to evaluate a workers performance. To solve this problem, we explore two methods of utilizing limited gold labels: *initial training* (INIT) and *periodic updating* (PER). Furthermore, we present a novel way of learning a prediction model in the absence of gold labels with uncertainty-aware learning and soft-label updating. Our experiments with a real crowdsourcing dataset demonstrate that *periodic updating* tends to show better performance than *initial training*. [1]

5.1 Introduction

Our previous chapters have shown that a worker's performance can be more accurately modeled by abandoning traditional (i.i.d.) assumptions

[1] This chapter is based on the published work (Jung and Lease, 2015b) in the AAAI Conference of Human Computation and Crowdsourcing 2015, which is guided by a co-author, Matthew Lease.

between tasks and instead modeling temporal correlation in task performance. However, a fundamental challenge remains in the need for expert "gold" labels to evaluate a worker's performance. As Bragg et al. (2014) opined, prior work has often made a strong assumption that all examples have known gold labels readily available to immediately evaluate each worker response as it arrives. Of course, if we already had gold labels in-hand for all examples, there would be no need for collecting additional labels from the crowd.

A common alternative strategy is to ask multiple workers to answer the same question, aggregate responses, and then evaluate each individual's agreement with the aggregate. This poses a fundamental trade-off in *plurality*: asking more workers to answer the same question increases aggregate accuracy at the cost of increased redundancy. Also, unlike use of expert gold, it cannot safeguard against systematic crowd bias or crowd collusion. Most pertinent in this work, this strategy is difficult to employ in an online setting because it is unrealistic to assume that all workers assigned a given example will label it at the same time, or that a worker would happily wait for all others to complete the task before anyone could proceed to the next task.

We consider how to best estimate a temporal model of worker performance when supervision is more realistically limited. Intuitively, if we have only a smaller sample of gold questions with which to check worker correctness, our estimate of worker accuracy will have larger variance (i.e., increased uncertainty).

Methodology. To solve this problem, this chapter explores how to

maximally utilize limited gold labels and how to update a prediction model in the absence of gold labels. This study begins with an investigation of two methods of utilizing limited gold labels. The first method, *initial training* (INIT), uses all of the given gold labels to estimate a worker's label correctness in the initial phase. The second alternative approach, *periodic updating* (PER), uses gold labels to check label correctness periodically. The key insight in PER is that a worker's temporal performance may be non-stationary (i.e. exhibiting varying correctness over time), which may limit the effectiveness of training the model only on the worker's initial temporal patterns.

This chapter also presents a novel way of learning a prediction model in the absence of gold labels with *uncertainty-aware learning* (Bootkrajang and Kaban, 2013b) and *soft-label updating*. The idea is, for training examples without a known gold label, to generate a pair of positive and negative training examples with instance weights based on a probability of the worker producing correct and incorrect labels. We consider two approaches for estimating uncertainty: one based on model prediction scores and one based on the confidence interval of worker accuracy.

The proposed methods are evaluated on a real crowdsourcing dataset that is used for the above chapters. Results demonstrate that PER tends to show better prediction than *initial training* across a varying the number of gold labels as well as decision reject options. Furthermore, we find that *uncertainty-aware* learning with *soft-label updating* brings further improvement to prediction accuracy with limited supervision.

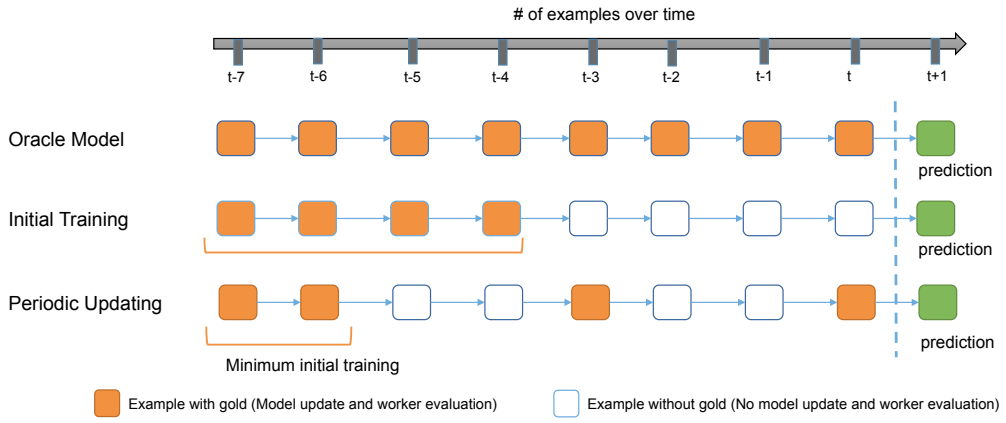

Figure 5.1: Three sequential learning methods of a prediction model for crowd work quality with limited supervision.

We investigate the following research questions:

RQ3.1: Initial training (INIT) vs. Periodic updating (PER) *How can we best use limited gold labels for model training? When do different methods perform better?*

RQ3.2: Uncertainty-aware learning *To what extent can we effectively update models without supervision? How does uncertainty-aware learning impact prediction accuracy?*

5.2 Problem

We begin with a binary label acquisition problem in crowdsourcing. Suppose that a worker has produced a label set L of n labels ($|L| = n$), and that each label l_i may or may not have a corresponding gold label g_i, which belongs to a gold label set G. Our task is to predict whether or not a worker's

next judgment will be correct, as defined by agreement with gold labels. In this work, we assume an objective labeling task for which each example has only a single correct label, indicated by the gold label set.

The correctness of the ith label is denoted as $y_i \in \{0, 1\}$, where 1 and 0 represent correct or not. Label correctness y_i is computed by comparing a worker's label l_i to its corresponding gold label g_i. Thus, the labeling performance of a worker can be represented as a sequence of binary observations, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$. For example, if a worker produced five labels and erred on the first and third respectively, then her *binary performance sequence* is encoded as $\mathbf{y} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$.

We generate a multi-dimensional feature vector, $x_i = \begin{bmatrix} x_{1i} & x_{2i} & \dots & x_{mi} \end{bmatrix}$ per time i and use x_i as an input of a prediction function f. We adopt the same features used in Chapter 4: observable and latent features about crowd assessors' annotation performance and behavior. However, our feature generation process is different from their study in a sense that feature generation relies upon the availability of gold labels. For instance, when a gold label is provided, we generate the same features as our previous study. If a gold label is not provided, we include only the subset of features which do not require gold labels to be computed. Specifically, in such cases we omit their accuracy-based features and compute only their behavioral features. Our final goal is to find a prediction function f for each worker and use the function f for predicting each worker's next label correctness.

Prior work has typically assumed the existence of gold labels associated

with all of the labels, ($|L| = |G|$) (Donmez et al., 2010; Krause and Porzel, 2013). However, this assumption does not hold true in practice since one of fundamental reasons for crowdsourcing is collecting labels that we do not have. Furthermore, many studies on online algorithms in quality assurance in crowdsourcing (Tran-Thanh et al., 2014; Welinder and Perona, 2010a) make a very strong assumption that a worker's label correctness can be checked instantly at each time step (Bragg et al., 2014). We aim to relax these unrealistic assumptions by limiting the number of gold labels ($|G| < |L|$) to be used for measuring the label correctness of crowd labels. This is consistent with common practice of injecting occasional questions with known answers into each worker's task queue in order to assess performance. This also resembles a traditional semi-supervised setting in which we seek to learn from unlabeled examples as well as labeled examples, though here we have an additional temporal dimension.

Prior work in *item response theory* (IRT) seeks to assess each individual's temporal learning (Hambleton et al., 1991). However, our approach differs from IRT in that our models seek to capture latent dynamics by taking account of temporal correlation and additional variables. In addition, IRT typically assumes that pairs of questions and answers are provided ahead of a test. This assumption may not be directly applicable to crowdsourcing settings since if gold labels are in-hand for all examples, there would be no need to collect additional labels from the crowd.

Knowledge tracing has been also devised to model learner's mastery of knowledge being tutored (Corbett and Anderson, 1994). While it looks similar

to our models in the sense that knowledge tracing also observes the correctness of each student's answers over time, its actual usage is different since it focuses on measuring the level of overall mastery rather than supporting instance-level prediction. Furthermore, it also assumes that pairs of questions and answers are provided ahead of a test, as IRT does.

The closest prior work we are aware of on temporal modeling of crowd work with limited supervision, by Krause et al. (2013), proposes a method to estimate a worker's response quality by measuring agreement with gold labels as well as using a sliding window over time. However, they assume plurality-based gold estimation, which, as discussed earlier, is difficult to employ in an online setting. Furthermore, this study only leverages gold labels to estimate worker performance in the beginning. There may exist further ways to use gold labels for worker performance estimation, such as the *periodic updating* (PER) we propose.

5.2.1 Challenges from limited supervision

Limiting the number of gold labels raises critical questions about how to measure label correctness for model updating. Firstly, we face a challenge of measuring label correctness without gold labels. If we assume offline analysis (after data collection) and a reasonably high-degree of plurality (the number of worker assigned to the same question), then it is possible to measure a worker's label correctness with *pseudo-gold* labels which can be generated by aggregating multiple labels from workers (Ipeirotis and Provost, 2013; Sheshadri and

Lease, 2013a).

However, when it comes to measuring a worker's label correctness online (during data collection), as noted earlier, it is unrealistic to assume that all workers label the same task and they are willing to wait for a next task to be assigned. Moreover, the confidence of pseudo-gold labels is sensitive to the number of workers per task. Ideally, we would avoid requiring any plurality and be able to rely upon an individual worker with reliable (predictable) behavior.

Secondly, we should consider how to best use limited gold for training a prediction model. Using all of the given gold labels for *initial training* is simple, but prediction performance may suffer when a worker's temporal performance drifts dynamically over time (non-stationary). A prediction function f trained in this fashion may drift further from the true distribution as the number of labels from this worker $|L|$ increases over time.

5.3 Method

We present two methods to address the problems raised in the previous section. Firstly, we explore how to use limited gold for learning a prediction model. Secondly, we introduce a method for learning a prediction model in the absence of gold labels by *soft-label updating*.

5.3.1 Initial Training vs. Periodic Updating

While offline batch learning does not consider the order of training examples, an online learning algorithm is sensitive to order since it processes

training examples in a sequential fashion. For this reason, it matters when gold labels are used to check a worker's label correctness with limited gold. In this study, we compare two different methods of using limited gold labels for model training. The first method, *initial training* (INIT), uses all of the given number of gold labels G at the start to estimate model parameters. *Initial training* seems appropriate if we assume a sequence of a worker's label correctness follows the property of a stationary process. From a temporal perspective, a sequence of worker's label correctness $\mathbf{y}$ can be described as a stationary process if statistical parameters such as mean, variance, and autocorrelation of $\mathbf{y}$ are all constant over time. However, *initial training*'s prediction performance may suffer if a sequence of a worker's label correctness violates this stationary property.

To address this concern, we propose another method, referred to as *periodic updating* (PER), which updates a learning model periodically in order to remain in sync with any temporal drift of a worker's label correctness. Whereas INT uses all k gold labels at the start, PER saves limited gold for later checks. In practice, since a learning model requires some amount of training labels in the initial learning phase, PER also uses some number of gold labels at the start. However, remaining gold labels are reserved for periodic checking. This method is hypothesized to perform better than INIT when worker correctness is not stationary over time.

Figure 5.1 presents a conceptual example contrasting *initial training* (INIT) with *periodic updating* (PER). While the former uses 4 gold labels

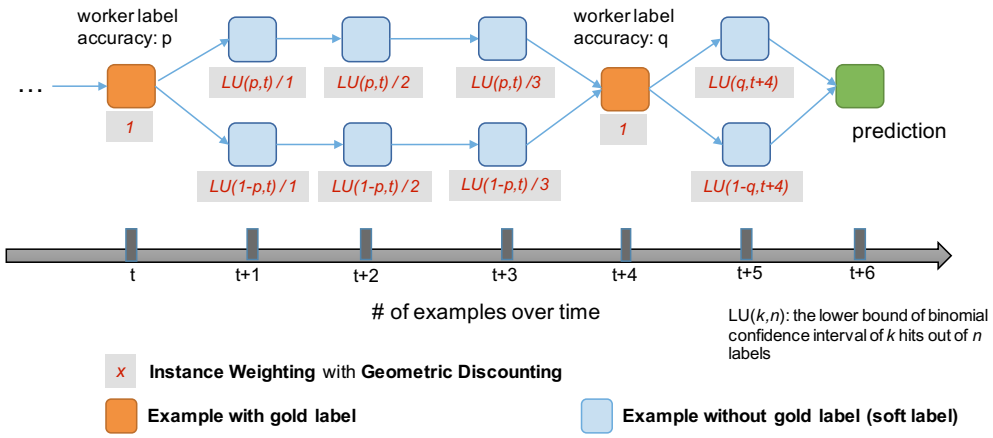

Figure 5.2: An example of soft labels with lower bound-based approach. Geometric discounting is applied to reduce instance weights with increasing time as a guard against temporal drift.

initially, the latter uses two gold labels for *periodic updating*. As the number of crowd labels increases, the two methods are expected to show different performance.

5.3.2 Instance Weighting with Soft Labels

While the two proposed methods in the previous section investigate how to effectively utilize limited gold labels for building a prediction model, due to the absence of gold labels, some labels cannot be checked for their correctness. While it is possible to measure the quality of labels offline via pseudo gold labels (generated by aggregating labels), it is problematic in practice to rely on pseudo-gold labels while data collection is ongoing because workers do not all label the same example at the same time.

Instead, our idea is to estimate and utilize soft labels based on a prob-

ability of the worker producing a correct label at time t. For an example with unknown gold, we generate two soft labels: a positive training example and a negative example. We assign instance weights to these training examples: a probability of getting a correct label from this worker at time k, $p(correct)_k$, and 1- $p(correct)_k$.

In order to derive this probability, we first consider a *model score-based* approach, which assigns $p(correct)_k$ and 1- $p(correct)_k$ generated from the model at time t to instance weights at time t+1. Once instance weights are assigned to a pair of two soft labels at time t+1, we update our model and obtain new model scores at time t+1, which are used for instance weights at time t+2.

While the first approach relies upon actual model scores, we consider another way to derive instance weights, a *lower bound-based* approach. In this approach, both the quality of the worker's label accuracy as well as the quantity of labels are considered. Based on the most recent accuracy measured over gold labels, we estimate the probability of a worker producing a correct label. For instance, if the latest worker's accuracy is measured over gold at time t, we use this value for instance weighting.

For lower bound estimation, we adopt the Clopper and Pearson Interval (Clopper and Pearson, 1934), the so-called *binomial exact confidence interval*. While the upper bound of it is widely used for exploration and exploitation in Multi-armed Bandit approaches (Auer, Peter and Cesa-Bianchi, Nicolò and Fischer, Paul, 2002; Tran-Thanh et al., 2014), we focus on the

lower bound of accuracy estimation since our primary goal is to estimate the weight of each training example with minimal uncertainty. The lower bound of the Clopper and Pearson Interval is defined by $B(\frac{\alpha}{2}; x, n-x+1)$ where x is the number of correct labels made by a worker w_j at time i, n is the total number of labels by a worker w_j, and $B(a; b, c)$ is the ath quartile from a beta distribution (b,c).

In addition to lower bound estimation, we also consider *temporal geometric discounting* of the training weight of soft labels. Our intuition is that the confidence of a worker's labeling accuracy at time i diminishes over time given possible non-stationary in labeling performance. Assuming X_t has a gold label, then for k>0, discount $\gamma_{t+k}= \frac{1}{k}$ So for k=1,2,3, ... , we have $\gamma_{t+k} = 1$, $\frac{1}{2}$, $\frac{1}{3}$, ...

Figure 5.2 shows an example of training a prediction model with soft labels using *periodic updating* (PER). Whenever a gold label comes, the weight of a training example is reset to 1. However, when no gold is available, we do instance weighting using soft-labels.

5.3.3 Uncertainty-aware Learning

Recent studies in machine learning have investigated how to learn with noisy training examples (Bootkrajang and Kabn, 2012; Bootkrajang and Kaban, 2013b). We adopt such *uncertainty-aware learning*, which trains a prediction model by including instance weights of each training example.

To select a learning model, we adopt a variant of the *Adaptive Boosting*

(Adaboost) model proposed by Bootkrajan and Kaban (2013a) for several reasons (Freund and Schapire, 1997). Firstly, since we need to differentiate the weight of each training example, *weighted Adaboost* exactly fits this need. Secondly, it is a well-known ensemble algorithm that obtains better predictive performance by combining multiple weak learners. Thirdly, a weak learner to be used for this boosting model is logistic regression which is relatively simple to implement and not prone to overfitting.

In classical logistic regression, the log likelihood is defined as:

$$\sum_{n=1}^{N} y_n \log p(y=1|x_n, w) + (1-y_n) \log p(y=0|x_n, w) \tag{5.1}$$

where w is the coefficient vector. If all of the class labels (y) were presumed to be correct, we would have $p(y=1|x_n, w) = \sigma(w^T x_n) = \frac{1}{1+e(-w^T x_n)}$ and if this value is greater than 0.5, the predicted value of x_n is class 1. However, when class label noise is present, this approach may not hold true. Thus, *uncertainty-aware learning* introduces a latent variable $\bar{y}$ to consider uncertainty of having an incorrect class label. We model $p(\bar{y}=k|x_n, w)$ as follows:

$$S_n^k \; p(\bar{y}=k|x_n, w) = \sum_{j=0}^{1} p(\bar{y}=k|y=j) p(y=j|x_n, w) \tag{5.2}$$

where k $\in 0, 1$. Hence, the log likelihood of *uncertainty-aware* learning is defined as:

$$\sum_{n=1}^{N} \bar{y}_n \log S_n^1 + (1-\bar{y}_n) \log S_n^0. \tag{5.3}$$

We omit the details of mathematical proof of this method since it is provided in (Bootkrajang and Kaban, 2013a). In prediction, we consider a

semi-supervised sequential learning task where we are given N training instances $\{(x_i, y_i), i = 1, ..., N\}$. Here, each $x_i \in \mathbb{R}^M$ is an M-dimensional feature vector adopted from our previous study in Chapter 4, and $y_i \in 0, 1$ is a class label indicating whether an worker's next label is correct (1) or wrong (0). Before fitting a model to our features and target labels, we first normalize feature values using min-max normalization as defined in Section 4.4.1.

5.4 Evaluation

5.4.0.1 Dataset and Metric

We adopt the same NIST TREC 2011 Crowdsourcing Track Task 2 dataset as in the previous chapters.The performance of our prediction model is evaluated with accuracy. Since we build a prediction model per worker, we report mean prediction accuracy across 49 workers.

5.4.0.2 Methods and Learning Models.

We investigate the prediction accuracy of the proposed methods with a varying supervision ratio. Eight different combinations are considered in this experiment. First, vanilla *periodic updating* (PER) and vanilla *initial training* (INIT) do not update with soft labels. Next, both PER with *uncertainty-aware* learning (PER+UNC(MD)) and INIT with *uncertainty-aware* learning (INIT+UNC(MD)) update a learning model with soft labels whose instance weights are based on model prediction (MD) scores. Finally, PER with *uncertainty-aware* learning based on lower bound (PER+UNC(LB)), PER

with *uncertainty-aware* learning based on lower bound and geometric discounting (PER+UNC(LB+GD)), INIT with *uncertainty-aware* learning based on lower bound (INIT+UNC(LB)), and INIT with *uncertainty-aware* learning based on lower bound and geometric discounting (INIT+UNC(LB+GD)) use soft labels based on the lower bound of worker accuracy with or without geometric discounting. As a baseline method, we also report an oracle which runs with all known gold labels.

To investigate the effect of supervision fairly, we vary the number of gold labels for each worker based on a fixed supervision ratio. For instance, when a supervision ratio equals 0.4 and a worker labeled 50 examples, we use 20 gold labels for *initial training* (INIT) ($50 \times 0.4 = 20$). In terms of *periodic updating* (PER), initial 10 labels are used for minimal training and the other 10 labels are used for periodic updating. If the number of gold labels based on a supervision ratio is equal to the number of gold labels for minimal training (10), no periodic updating happens.

As our base model, we adopt the generalized assessor model (GAM) proposed in Chapter 5. While it uses L1-regularized logistic regression, we instead use a variant of AdaBoost, as discussed in the previous section. To learn the AdaBoost model, we use default parameter settings from Scikit-learn (Pedregosa et al., 2011), though setting a learning rate 0.3 after varying parameter values between 0.1 and 1 over the *initial training* set of each worker. In terms of *uncertainty-aware* learning, unsupervised features such as $TaskTime$, $NumberOfLabels$, $TopicChange$, and $TopicEverSeen$ are only considered.

	Supervision Ratio (%)						
Method	20	25	30	35	40	45	50
0: Oracle	78.2						
1: INIT	56.5	59.2	61.8	63.3	66.0	67.7	69.4
2: PER	64.1*	65.9*	67.1*	69.2*	71.6*	73.3*	75.1*
3: INIT+UNC(MD)	55.4	59.7	61.3	63.4	67.2	67.2	69.8
4: INIT+UNC(LB)	57.8*	60.8*	62.7*	64.3*	67.6 *	68.9	70.5*
5: INIT+UNC(LB+GD)	58.4*	61.5*	63.6*	65.2*	68.7*	70.0*	71.6*
6: PER+UNC(MD)	65.1*	67.1*	68.9*	**70.9***	72.1*	74.1	**75.1***
7: PER+UNC(LB)	65.3*	66.4*	68.2*	70.3*	72.2*	74.3*	74.8*
8: PER+UNC(LB+GD)	**65.9***	**67.4***	**69.2***	70.3*	**72.6***	**74.4***	74.6*

Table 5.1: Mean prediction accuracy of eight methods of learning model (INIT and PER) over 49 workers with a varying supervision ratio. A two-tailed pairwise t-test is conducted to examine whether one method significantly outperforms Method 1 (INIT). (*) indicates that one method outperforms Method 1 (INIT) with statistical significance ($p<0.05$). Overall, PER outperforms INIT. Uncertainty-aware learning further improves prediction accuracies. In particular, the effect of *uncertainty-aware* learning is greater when a supervision ratio is small (20-30%).

5.4.1 RQ3.1: Initial training vs. Periodic updating

What is the best way to utilize limited gold labels for building a more accurate prediction model? To answer this question, we compare the difference of prediction accuracy between INIT and PER under various conditions. First, we compare the prediction accuracy of two methods under varying the number of gold examples and crowd accuracies. Second, we investigate the difference between the two methods under a varying number of gold examples and temporal dependencies (correlation ϕ). The prediction accuracy is initially measured by each worker and then we compute the overall average score across 49 workers. To examine if the results are significantly different from

each other, we conduct a two-tailed paired t-test.

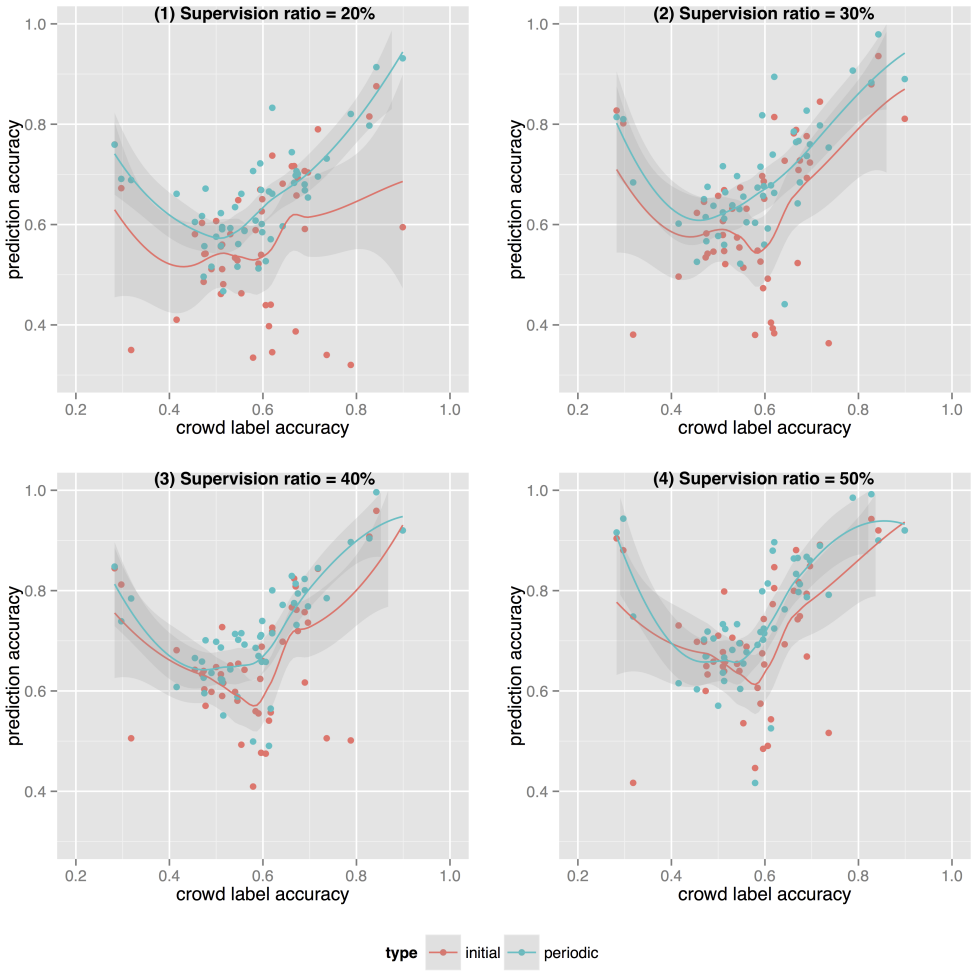

Figure 5.3: Comparison between initial training (INIT) vs. periodic updating (PER) across varying supervision ratios and different crowd label accuracies. Both methods work without *uncertainty-aware* learning. In general, PER outperform INIT across varying crowd label accuracies and varying number of gold labels. Both methods show better prediction accuracies when crowd label accuracies are far from 0.5. The increase in the number of gold labels (supervision) leads to less differentiation between the two methods.

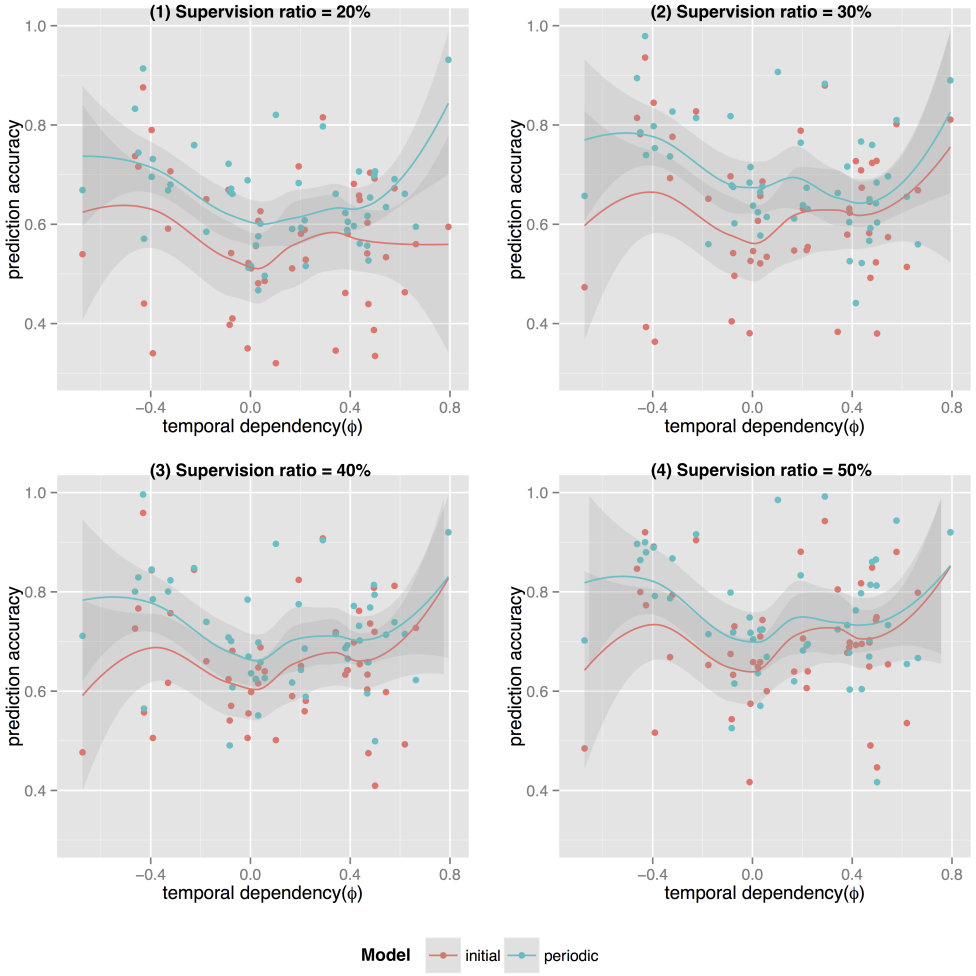

Figure 5.4: Comparison between initial training (INIT) vs. periodic updating (PER) across varying supervision ratios and different temporal dependencies (ϕ). Both methods work well without *uncertainty-aware* learning. As the number of gold labels increases, the gap between the two methods decreases. PER shows better prediction accuracies with $|\phi| \approx 1$. On the contrary, the prediction accuracies of initial updating does not show the same pattern when the number of gold labels is small.

Table 5.1 summarizes mean accuracy of the two model training methods (1:INIT and 2:PER), as well as six different updating methods with varying combinations of *uncertainty-aware* learning. The results show that INIT underperforms PER across varying supervision ratios. In particular, the gap between the two methods tends to be larger when the supervision ratio is small (20-25%). This result corresponds with the difference of updating strategy between the two methods. Since INIT does not update its model after initial training, it is natural that its predictive power is very limited by limited supervision. On the contrary, the nature of PER allows learning models over time. Hence, in comparison to INIT, the prediction accuracy of PER tends to be greater when supervision is very limited (20-25%).

Our next experiment examines the difference of prediction accuracies between INIT vs. PER across varying supervision ratio and crowd accuracies. **Figure 5.3** demonstrates that PER outperforms INT across varying crowd label accuracies and number of gold labels. In addition, both methods show better prediction accuracies when crowd label accuracies are far from 0.5. The increase in the number of gold labels (supervision) leads to less differentiation between the two methods.

Another experiment demonstrates the comparison of prediction accuracies between INIT vs. PER across varying temporal correlations (dependencies) and supervision ratios. **Figure 5.4** shows that PER outperforms INIT across varying temporal dependencies. In particular, PER shows better prediction accuracies when $|\phi| \approx 1$. It indicates that if a worker's label correctness is

very consistent ($\phi \approx 1$) or repeatedly oscillated with a pattern ($\phi \approx -1$) over time, PER tends to be more sensitive to such temporal dynamics than INIT. In terms of crowd workers with temporal dependencies $\phi \approx 0$, the prediction accuracies of two methods are influenced by the more supervision ratio rather than by temporal dependency.

In sum, our comparison between INIT vs. PER shows that PER outperforms INIT across varying conditions such as supervision ratios, temporal correlations, and crowd label accuracies. Furthermore, PER tends to show better predictive power when the temporal dependencies of crowd workers' label correctness are larger.

5.4.2 RQ3.2: Uncertainty-aware Learning

The previous experiments demonstrated that PER works more accurately than INIT under limited supervision. Next, how do we update our prediction model when a gold label is not available? And to what extent does this method benefit improving prediction accuracy? We examine the efficacy of *uncertainty-aware learning* with *soft labels* and *geometric discounting* in this experiment.

The below six methods in **Table 5.1** show the effect of *uncertainty-aware* learning (UNC). This result shows that *uncertainty-aware* learning brings improvement in prediction accuracies to both methods. In particular, the effect of *uncertainty-aware* learning is greater when the supervision ratio is small (20-30%). While the effect of this method decreases as supervision in-

creases, Method 5 (INIT+UNC(LB+GD)) and Method 8 (PRED+UNC(LB+GD)) show statistically significant improvement of prediction accuracies in comparison to vanilla INIT (Method 1) and vanilla PER (Method 2). In regard to the method of producing soft labels, *lower bound*-based methods (Method 4, 5, 7, and 8) outperform *model score*-based methods (Method 3 and 6) across INIT and PER. This result suggests that *lower bound*-based methods tend to provide more accurate soft labels, which leads to greater improvement of prediction accuracy.

Next, we investigate the cause of performance improvement by *uncertainty-aware* learning with soft-labeling. Our idea in soft labels and instance weighting is that a probability of getting a correct label can be derived from model scores or the lower bound of a worker's accuracy. We expect that if a worker shows low entropy of labeling accuracy (far from accuracy of 0.5), then the benefit of soft labeling increases. To examine this hypothesis, we conduct an additional experiment which focuses on correlation between the prediction accuracy improvement by uncertainty-aware learning vs. workers' label accuracy. To measure the improvement of prediction accuracy, we compare Method 2, vanilla PER, with Method 8 (PER+UNC(LB+GD)), PER with *uncertainty-aware* learning based on lower bound and geometric discount. **Figure 5.5** shows that *uncertainty-aware* learning achieves better prediction accuracies across 64% of all workers (26 out of 42, 7 ties). Note that a prediction model for a worker with accuracy > 0.6 or < 0.4 achieves prediction accuracy improvement. For a worker whose accuracy ranges around 0.5 (higher entropy

of labeling accuracy), *uncertainty-aware* learning based on lower bound and geometric discount tends to show slightly weaker performance improvement.

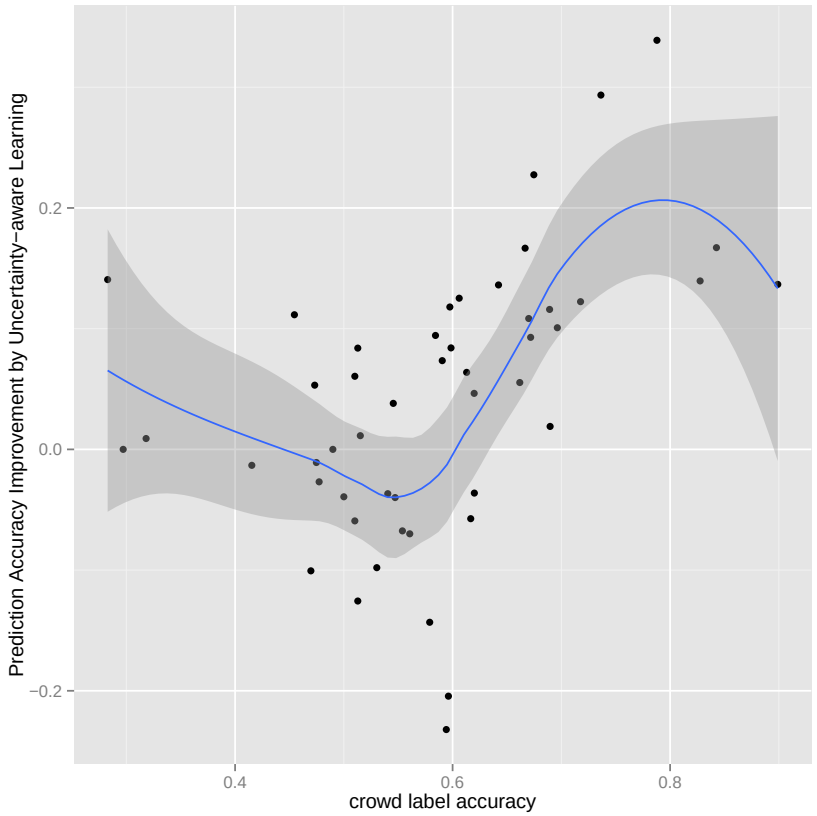

Figure 5.5: Prediction accuracy improvement by *uncertainty-aware* learning vs. crowd worker label accuracy (supervision ratio = 20%). Prediction accuracy improvement by *periodic updating* (PER) is computed by $\frac{Method8-Method2}{Method2}$ where Method 2 indicates PER and Method 8 indicates PER+UNC(LB+GD) as defined in **Table 5.1**. Method 8 improves the prediction accuracy of vanilla PER (Method 2). In particular, when label accuracy is reliable (> 0.6 or < 0.4), it is superior to vanilla PER.

Finally, we conduct an experiment to investigate how the geometric discount influences prediction accuracy. We hypothesize that if the benefit of

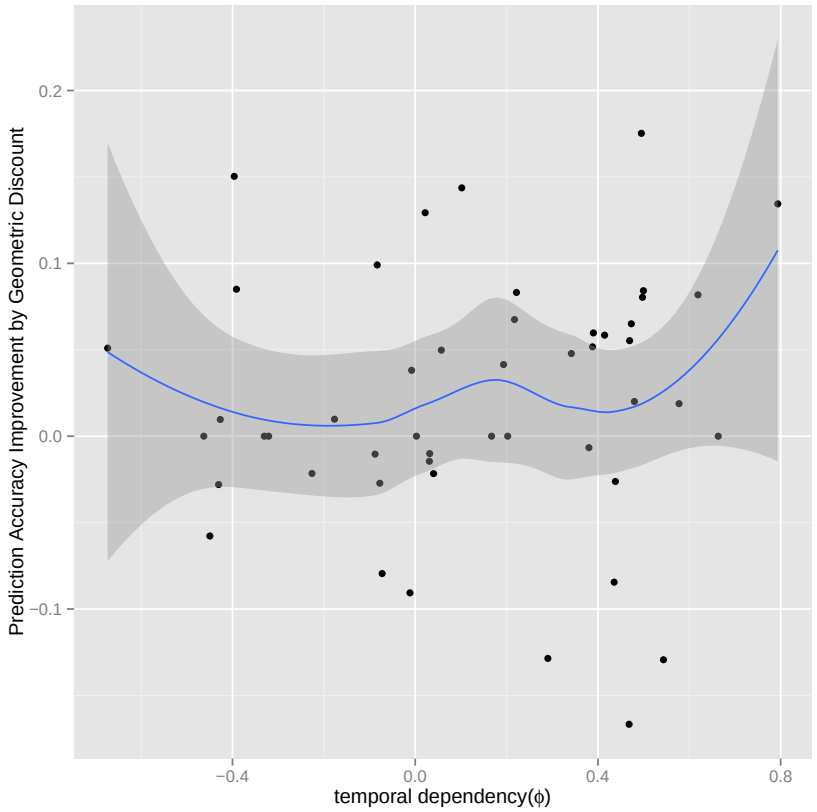

Figure 5.6: Prediction accuracy improvement by geometric discount vs. temporal dependencies (supervision ratio = 20%). Prediction accuracy improvement by *geometric discounting* is computed by $\frac{Method8-Method7}{Method7}$, where Method 7 indicates PER+UNC(LB) and Method 8 indicates PER+UNC(LB+GD) as defined in **Table 5.1**.

geometric discount increases when temporal correlations are large. We measure the relative improvement from the geometric discount by comparing Method 7 (PER+UNC(LB)) and Method 8 (PER+UNC(LB+GD)). To measure the degree of being stationary, we measure the variance of a worker's label correctness. **Figure 5.6** shows that the geometric discount brings more benefit when

temporal dependencies tend to be larger. This result supports our hypothesis since the geometric discount approach shows greater improvement for a worker showing higher temporal dependencies.

In sum, this result demonstrates that *uncertainty-aware* learning shows substantial improvement in prediction accuracy with soft labels. Furthermore, our additional experiments confirm why and when *uncertainty-aware* learning with soft labels brings the improvement of prediction accuracy.

5.5 Conclusion

While Chapter 3 and 4 present a novel way to predict a next label correctness, it basically assumes the existence of gold labels to judge a crowd worker's label correctness. In practice, the number of gold labels may be limited, which raises a critical question of how to learn a temporal prediction model of crowd work quality.

To address this problem, this chapter investigates two methods of learning a temporal prediction model with limited supervision (initial training (INIT) vs. periodic updating (PER)). Our experiments demonstrate that PER outperforms INIT across varying supervision ratios (RQ3.1). This result suggests that PER would be recommended to capture the non-stationary property of a worker's label quality over time.

In addition to the investigation of a comparison between INIT vs. PER, we present a novel way to learn a temporal prediction model by uncertainty-

aware learning (UNC) (RQ3.2). We demonstrates that UNC further improves prediction accuracy based on soft labels with instance weighting.

To sum up, this study suggests that a temporal prediction model can be learned with limited supervision. However, it is strongly recommended to update a model periodically in order to catch up with a non-stationary property of a worker's label correctness over time.

Chapter 6

Discussion

In this chapter, we summarize the contributions and limitations of this dissertation. In light of the findings and limitations, many questions arise that require future work.

6.1 Theoretical Contributions

For quality assurance in crowdsourcing, it is critical to find reliable workers among crowds. This process typically relies on the measurement and prediction of crowd work quality. Even though there exists evidence of temporal effect on crowd work quality such as learning effect, boredom, and fatigue, prior work tends to disregard this issue. To address this problem, this dissertation studies builds a learning model by considering underlying temporal perspectives of crowd work quality.

The primary contribution of this dissertation is to measure and model temporal aspects of crowd work quality. While prior work typically measures or models crowd work performance based on the assumption of i.i.d. distribution, the proposed time-series models relax such a strong assumption by considering underlying temporal dynamics of crowd work. Our study in Chapter

3 empirically demonstrates the effectiveness of the proposed model and discusses why it outperforms baselines. In particular, our experiments show that prediction performance significantly improves when crowd work quality (label correctness) tends to have a strong temporal correlation. Unlike a typical measurement of crowd work quality, such as accuracy and RMSE, TS model allows us to capture a stationary process of crowd work quality.

Secondly, this dissertation presents a novel way to leverage multiple features of crowd labeling behavior to improve the prediction performance of crowd work quality. Chapter 4 investigates various features of crowd workers annotation performance, and then integrates these features into a generalized assessor model. Our empirical analysis demonstrates the effectiveness of the proposed model in addition to the relative importance of the proposed features.

Finally, this dissertation examines how to learn the proposed temporal models (TS and GAM) with limited supervision. In particular, our empirical analysis demonstrates that periodic updating outperforms initial training when crowd work quality shows a strong temporal correlation. In addition, Chapter 5 shows that learning with soft-labels can potentially result in improving the prediction accuracy.

To sum up, this dissertation proposes a novel way to measure and model crowd work quality by taking the temporal aspects into account. Chapter 3 presents an innovative approach to model and predict crowd work quality. Chapter 4 shows the effectiveness of multiple feature dimensions for improving prediction quality as well as crowd label quality. Finally, Chapter 5 demon-

strates that the proposed model works in a realistic condition.

6.1.1 Practical Contributions

Given the increasing popularity of collecting crowdsourced labels at scale, a need for better quality assurance methods is on the rise. With regards to quality control methods such as label aggregation, spam worker filtering, and good worker retention, measuring a crowd workers performance is critical. Prior work in quality assurance mostly assumes that observed crowd workers labeling performance is independently and identically distributed despite the fact that crowd worker behavior can dynamically vary over time. For this reason, this dissertation opens a new possibility of analyzing and understanding crowd work quality from a temporal viewpoint. By relaxing the limits of i.i.d assumption, we have a better predictive model which offers a variety of ways to improve quality of crowdsourced labels, such as routing judging tasks to workers most likely to produce reliable judgments, label aggregation by considering workers behavioral pattern, and filtering out spam crowd workers. Furthermore, the proposed models can be used for educating crowd workers as well as managing a reliable crowd worker pool over time.

Beyond this practical applications of our proposed models and methods, this dissertation allows us to have better understanding of temporal effects on crowd work quality. For practitioners who want to leverage the wisdom of crowds, this dissertation enlighten the importance of temporal effects on crowd work quality. For researchers who analyze and measure crow work quality, this

dissertation allows them to have a better model and method which may make their works easy. Finally, for crowd workers, the proposed models and methods in this dissertation may give them a chance to be educated over time.

To sum up, this dissertation brings various benefits to crowdsourcing practices.

6.2 Limitations and Future Work

While this dissertation presents novel models and methods to measure, model, and predict crowd work quality, the proposed models and methods still rely on certain assumptions and constraints. In this section, we discuss the limitations of this dissertation and provide future research directions.

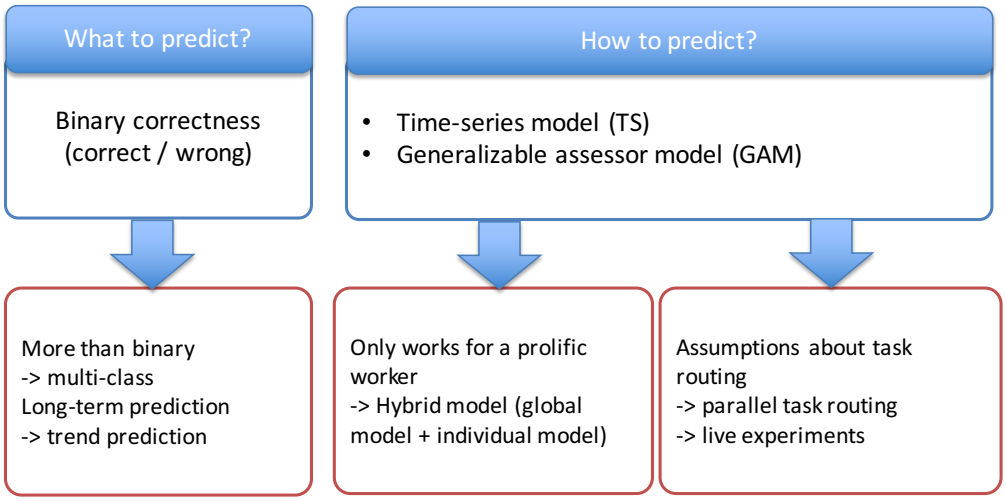

Figure 6.1: Limitations and possibilities of the proposed models and methods. Blue boxes indicate what we achieved in this dissertation and red boxes discuss future work.

Figure 6.1 shows the limitations and possibilities of the proposed meth-

ods (TS and GAM).

Firstly, with regards to the prediction variable, TS and GAM predict the binary correctness of a worker's next label. In practice, we may need to have a different use-case for predicting crowd work quality. For instance, if a task has more than two values as a target variable, this model may be extended to predict a multi-class variable such as an ordinal value or a categorical value. Furthermore, our proposed models can be extended to predict a long-term trend instead of a next label's correctness. For instance, we may use a simple moving average of recent 5 label quality or a weighted moving average of recent 10 label quality.

Secondly, the proposed models are designed for a prolific crowd worker whose number of labels are greater than 20. In other words, TS and GAM may not work properly for an ad-hoc worker who has annotated less than 20 labels. For future work, researchers may consider a hybrid model which integrates a *global* model and an*individual* model. In this case, a global model handles an issue about an ad-hoc worker by taking an average model across workers, whereas an individual model enhances the performance of a global model by incorporating a workers individual characteristic.

Thirdly, TS and GAM still rely on sequential task routing. While parallelization outperforms sequentialization in general scheduling or routing scenarios, the majority of prior work on task routing assumes a sequential task assignment which does not efficiently utilize all available workers. If a task routing framework only hires a specific number of crowd workers in a

sequential manner, this will irritate the current workers who have devoted their time to the task.ta Therefore, it is necessary to have a task routing framework which aims at an efficient resource allocation.

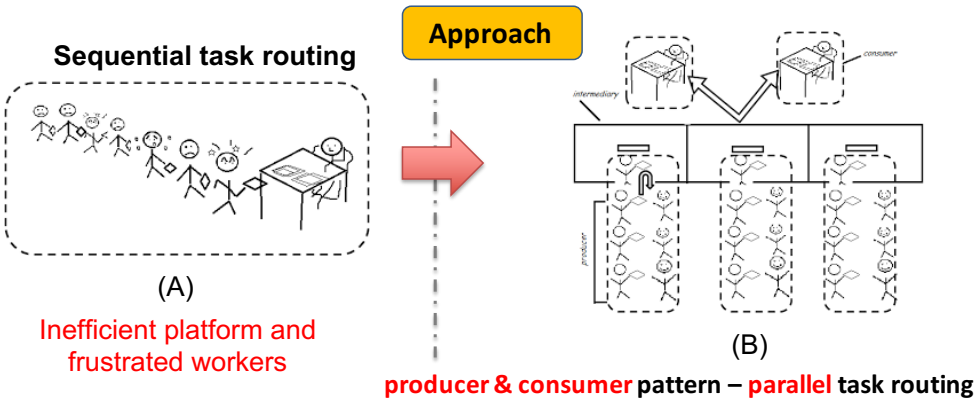

Figure 6.2: Sequential routing vs. Parallel task routing.

For parallel task routing, we may consider a producer & consumer pattern which effectively maintains asynchronous task allocation in order to serve a parallel routing Laws et al. (2011). As shown in Figure 6.2, a worker queue (producer) keeps track of available crowd workers in real time while a task queue (consumer) monitors a set of available tasks. In this setting, future work can investigate a "lazy update" strategy (Laws et al., 2011). Instead of updating each worker's model immediately upon label submission, we delay the update, after all other peer-labels have been received for that example and the consensus label has been established.

Finally, our empirical analysis is based on a retrospective experiment whereby past data, an improper instrument under a task routing scenario, is

used to learn the model. To address this issue, we conducted our experiments with a synthetic dataset. However, our synthetic dataset-based experiments may not be generalizable since the dataset was not generated under a real task routing scenario. For future work, one may extend this study with a live experiment which can capture some additional factors related to prediction quality and task routing.

Chapter 7

Conclusion

Crowdsourcing has appeared to be an attractive alternative for several challenging problems to computers. It enables us to collect labels for training and testing as well as capture users information seeking behaviors on a large scale at a low cost. In particular, crowdsourcing relieves a burden of academic researchers that calls for large-scale experiment samples.

Crowdsourcing, however, is no panacea for all the problems. Quality concerns are accompanied with it. Due to the lack of expertise, dedication, interest and the failure of task design, the quality of crowdsourced results are often still dubious. This is one reason many researchers remain hesitant to adopt crowdsourcing in their research. Through this research, we have found that quality concerns can be mitigated by taking into account temporal factors behind crowd work. In particular, most studies did not focus underlying characteristics behind crowd work for worker quality prediction and task routing. From this intuition, we envisioned a more accurate, efficient, and cost-effective crowdsourcing by modeling latent features behind crow work.

For its concretization, this dissertation investigates how to build a prediction model of temporal dynamics of crowd work. Each study from Chapter

3 to Chapter 5 answers our research questions raised in Chapter 1. For RQ1, adaptation of time-series model for quality control in crowdsourcing, Chapter 3 demonstrates that a novel time-series model allows us to measure and predict crowd work quality more accurately than baselines. Next, for RQ2, discriminative Predictive model for crowd assessor accuracy, Chapter 4 answers this question by leveraging a variety of features about crowd label quality. Finally, for RQ3, temporal modeling with limited supervision, Chapter 5 examines how to learn a temporal model with limited supervision.

Despite the benefit of the proposed methods, there is still much room for improvement. However, this study opens a new horizon of modeling and measuring crowd work quality by taking account of the dimension of time. This finding would be a critical foundation to enable more accurate measurement and prediction of crowd work quality and lead to better quality of label acquisition in crowdsourcing.

Bibliography

Luis von Ahn. *Human computation*. PhD thesis, Pittsburgh, PA, USA, 2005. AAI3205378.

Luis von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: human-based character recognition via web security measures. *Science*, 321(12):1465 – 1468, 2008.

Omar Alonso. Implementing crowdsourcing-based relevance experimentation: an industrial perspective. *Information Retrieval*, pages 1–20, 2012. ISSN 1386-4564. doi: 10.1007/s10791-012-9204-1. URL `http://dx.doi.org/10.1007/s10791-012-9204-1`.

Omar Alonso and Ricardo Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 153–164, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. URL `http://dl.acm.org/citation.cfm?id=1996889.1996910`.

Omar Alonso and Stefano Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of the 32nd annual international ACM SIGIR conference on Research and development*

in information retrieval, pages 5–6, 2009. URL http://www.science.uva.nl/~kamps/publications/2009/geva:futu09.pdf#page=25.

Omar Alonso and Stefano Mizzaro. Using crowdsourcing for TREC relevance assessment. *Information Processing & Management*, 48(6):1053–1066, November 2012. ISSN 0306-4573. doi: 10.1016/j.ipm.2012.01.004. URL http://dx.doi.org/10.1016/j.ipm.2012.01.004.

Omar Alonso, Daniel E. Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, November 2008. ISSN 0163-5840. doi: 10.1145/1480506.1480508. URL http://doi.acm.org/10.1145/1480506.1480508.

Auer, Peter and Cesa-Bianchi, Nicolò and Fischer, Paul. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.*, 47(2-3):235–256, May 2002. ISSN 0885-6125. doi: 10.1023/A:1013689704352. URL http://dx.doi.org/10.1023/A:1013689704352.

Yukino Baba and Hisashi Kashima. Statistical Quality Estimation for General Crowdsourcing Tasks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 554–562, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487600. URL http://doi.acm.org/10.1145/2487575.2487600.

Geoffrey Barbier, Reza Zafarani, Huiji Gao, Gabriel Fung, and Huan Liu. Maximizing benefits from crowdsourced data. *Computational and Mathemati-*

cal Organization Theory, 18(3):257–279, June 2012. ISSN 1381-298X. doi: 10.1007/s10588-012-9121-2. URL http://www.springerlink.com/index/10.1007/s10588-012-9121-2.

Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.*, 9:1823–1840, June 2008. ISSN 1532-4435.

Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 313–322, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866078. URL http://doi.acm.org/10.1145/1866029.1866078.

Jakaramate Bootkrajang and Ata Kaban. Boosting in the presence of label noise. In *Proceedings of the 29rd International Conference on Uncertainty in Artificial Intelligence*, UAI '13, 2013a.

Jakramate Bootkrajang and Ata Kaban. Boosting in the presence of label noise. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013b.

Jakramate Bootkrajang and Ata Kabn. Label-noise robust logistic regression and its applications. In PeterA. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523

of *Lecture Notes in Computer Science*, pages 143–158. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33459-7. doi: 10.1007/978-3-642-33460-3_15. URL http://dx.doi.org/10.1007/978-3-642-33460-3_15.

George Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, third edition, 1994.

Jonathan Bragg, Andrey Kolobov, Mausam, and Daniel S. Weld. Parallel Task Routing for Crowdsourcing. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*, HCOMP '14, pages 11–21, 2014.

Chris Buckley, Matthew Lease, and Mark D. Smucker. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *19th Text Retrieval Conference (TREC)*, 2010.

J. P. Burg. Maximum entropy spectral analysis. In *Proc. 37th Meeting of the Society of Exploration Geophysicists*, 1967.

Chris Callison-Burch and Mark Dredze. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12, Los Angeles, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-0701.

Fabio Canova and Matteo Cicarelli. Panel vector autoregressive models: A survey. *European Central Bank: Working Paper Series*, 2013.

Ben Carterette and Ian Soboroff. The effect of assessor error on IR system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and Development in Information Retrieval*, SIGIR '10, pages 539–546, 2010. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835540. URL http://doi.acm.org.ezproxy.lib.utexas.edu/10.1145/1835449.1835540.

Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise Ranking Aggregation in a Crowdsourced Setting. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 193–202, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1869-3. doi: 10.1145/2433396.2433420. URL http://doi.acm.org/10.1145/2433396.2433420.

EH Chi and MS Bernstein. Leveraging online populations for crowdsourcing. *IEEE Internet Computing*, pages 10–12, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6319289.

Lydia B. Chilton, John J. Horton, Robert C. Miller, and Shiri Azenkot. Task search in a human computation market. In *ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 1–9, 2010.

C. J. Clopper and Egon Sharpe Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):pp. 404–413, 1934. URL http://www.jstor.org/stable/2331986.

AlbertT. Corbett and JohnR. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994. ISSN 0924-1868. doi: 10.1007/BF01099821. URL http://dx.doi.org/10.1007/BF01099821.

Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. Suggestbot: using intelligent task routing to help people find work in wikipedia. In *12th ACM conference on Intelligent user interfaces*, pages 32–41, 2007.

Peng Dai, Mausam, and Daniel S. Weld. Artificial intelligence for artificial artificial intelligence. In *Proc. AAAI HComp workshop*, 2011.

A P Dawid and A M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C Applied Statistics*, 28(1):20–28, 1979. ISSN 00359254. doi: 10.2307/2346806. URL http://www.jstor.org/stable/2346806.

Ofer Dekel and Ohad Shamir. Vox populi: Collecting high-quality labels from a crowd. In *In Proceedings of the 22nd Annual Conference on Learning Theory*, 2009. URL http://eprints.pascal-network.org/archive/00005406/.

P. Donmez, J. Carbonell, and J. Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the SIAM International Conference on Data Mining*, pages 826–837, 2010.

Pinar Donmez and JG Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008. ISBN 9781595939913. URL http://dl.acm.org/citation.cfm?id=1458165.

Pinar Donmez, Jaime G Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009. ISBN 9781605584959.

Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of ACM Computer Supported Cooperative Work (CSCW)*, New York, New York, USA, 2012. ISBN 9781450310864. doi: 10.1145/2145204.2145355. URL http://dl.acm.org/citation.cfm?doid=2145204.2145355.

Jun Du and Charles X. Ling. Active learning with human-like noisy oracle. In *2010 IEEE International Conference on Data Mining*, pages 797–802. Ieee, December 2010. doi: 10.1109/ICDM.2010.114. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5694041.

Carsten Eickhoff and Arjen P. Vries. Increasing cheat robustness of crowdsourcing tasks. *Information Retrieval*, 2012. ISSN 1386-4564. doi: 10.1007/s10791-011-9181-9. URL http://www.springerlink.com/index/10.1007/s10791-011-9181-9.

Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: http://dx.doi.org/10.1006/jcss.1997.1504. URL http://www.sciencedirect.com/science/article/pii/S002200009791504X.

S Ghosh, N Sharma, and F Benevenuto. Cognos: crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 575–590, 2012. ISBN 9781450314725. URL http://dl.acm.org/citation.cfm?id=2348283.2348361.

Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, December 2005. ISSN 1476-4687. doi: 10.1038/438900a. URL http://dx.doi.org/10.1038/438900a.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007. URL http://EconPapers.repec.org/RePEc:bes:jnlasa:v:102:y:2007:p:359-378.

Catherine Grady and Matthew Lease. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010*

Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10, pages 172–179, 2010. URL http://dl.acm.org/citation.cfm?id=1866696.1866723.

R.K. Hambleton, H. Swaminathan, and H.J. Rogers. *Fundamentals of Item Response Theory*. Measurement Methods for the Social Science. SAGE Publications, 1991. ISBN 9780803936478. URL https://books.google.com/books?id=cmJU9SHCzecC.

J. Howe. *Crowdsourcing: Why the power of the crowd Is driving the futhre of business.* Crown Publishing Group, 2008.

P Ipeirotis and Foster Provost. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 2013. URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1688193.

Panagiotis G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21, December 2010. ISSN 1528-4972. doi: 10.1145/1869086.1869094. URL http://doi.acm.org/10.1145/1869086.1869094.

Panagiotis G Ipeirotis and Evgeniy Gabrilovich. Quizz: targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World Wide Web*, WWW '14, pages 143–154, 2014.

P.G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on*

human computation, pages 64–67. ACM, 2010. URL http://dl.acm.org/citation.cfm?id=1837906.

P. A. Jacobs and P. A. W. Lewis. Stationary discrete autoregressive-moving average time series generated by mixtures. *Journal of Time Series Analysis*, 4(1):19–36, 1983.

Mukund Jha, Jacob Andreas, Kapil Thadani, Sara Rosenthal, and Kathleen McKeown. Corpus creation for new genres: A crowdsourced approach to PP attachment. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 13–20, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1866696.1866698.

B. H. Juang and L. R. Rabiner. Hidden Markov Models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

Hyun Joon Jung and Matthew Lease. Improving Quality of Crowdsourced Labels via Probabilistic Matrix Factorization. In *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*, pages 101–106, 2012. URL http://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/download/5258/5609.

Hyun Joon Jung and Matthew Lease. UT Austin in the TREC 2012 Crowdsourcing Track's Image Relevance Assessment Task. In *Proceedings of*

the 21st NIST Text Retrieval Conference (TREC), 2013. URL `http://trec.nist.gov/pubs/trec21/papers/UTAustin.crowd.final.pdf`.

Hyun Joon Jung and Matthew Lease. A Discriminative Approach to Predicting Assessor Accuracy. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2015a.

Hyun Joon Jung and Matthew Lease. Modeling Temporal Crowd Work Quality with Limited Supervision. In *Proceedings of the 3rd AAAI Conference on Human Computation (HCOMP)*, 2015b.

Hyun Joon Jung, Yubin Park, and Matthew Lease. Predicting Next Label Quality: A Time-Series Model of Crowdwork. In *Proceedings of the 2nd AAAI Conference on Human Computation*, HCOMP '14, pages 87–95, 2014.

Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Gaussian processes for object categorization. *International Journal of Compututer Vision*, 88(2):169–188, June 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0268-3. URL `http://dx.doi.org/10.1007/s11263-009-0268-3`.

Heinz Kaufmann. Regression models for nonstationary categorical time series: asymptotic estimation theory. *The Annals of Statistics*, 15(1):79–98, 1987.

Nicolas Kaufmann, T Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcinga study on mechanical turk. In *Proceedings of the Seventeenth Americas Conference on Information*

Systems, pages 1–11, 2011. URL http://schader.bwl.uni-mannheim.de/fileadmin/files/publikationen/Kaufmann_Schulze_Veit_2011_-_More_than_fun_and_money_Worker_motivation_in_Crowdsourcing_-_A_Study_on_Mechanical_Turk_AMCIS_2011.pdf.

Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Proceedings of the 30th European Conference on Advances in Information Retrieval*, ECIR '11, pages 165–176, 2011. URL http://www.springerlink.com/index/U33373T17P56HR2L.pdf.

Gabriella Kazai, J J Thomson Ave, and Jamie Costello. Towards Methods for the Collective Gathering and Quality Control of Relevance Assessments. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 452–459, 2009. ISBN 9781605584836.

Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information Retrieval*, July 2012a. doi: 10.1007/s10791-012-9205-0. URL http://www.springerlink.com/index/10.1007/s10791-012-9205-0.

Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. The Face of Quality in Crowdsourcing Relevance Labels: Demographics, Personality and Labeling Accuracy. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2583–

2586, 2012b. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398697. URL http://doi.acm.org/10.1145/2396761.2398697.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. Crowdforge: crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 43–52, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047202. URL http://doi.acm.org/10.1145/2047196.2047202.

Aniket Kittur, Susheel Khamkar, Paul André, and R Kraut. CrowdWeaver: visually managing complex crowd work. In *Proceedings of the 2012 ACM conference on Computer Supported Cooperative Work*, 2012. ISBN 9781450310864. URL http://dl.acm.org/citation.cfm?id=2145204.2145357.

Aniket Kittur, Jeff Nickerson, Michael S. Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matthew Lease, and John J. Horton. The future of crowd work. In *In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 1301–1318, February 2013.

Markus Krause and Robert Porzel. It is About Time: Time Aware Quality Management for Interactive Systems with Humans in the Loop. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 163–168, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1952-

2. doi: 10.1145/2468356.2468386. URL http://doi.acm.org/10.1145/2468356.2468386.

Pavel Kucherbaev, Florian Daniel, Maurizio Marchese, Fabio Casati, and Brian Reavey. Toward Effective Tasks Navigation in Crowdsourcing. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 401–404, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2775-6. doi: 10.1145/2598153.2602249. URL http://doi.acm.org/10.1145/2598153.2602249.

Anand Kulkarni, Philipp Gutheim, Prayag Narula, David Rolnitzky, Tapan Parikh, and Bjorn Hartmann. MobileWorks: Designing for Quality in a Managed Crowdsourcing Architecture. *IEEE Internet Computing*, 16 (5):28–35, September 2012. ISSN 1089-7801. doi: 10.1109/MIC.2012.72. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6226337.

Edith Law. Towards Large-Scale Collaborative Planning : Answering High-Level Search Queries Using Human Computation. In *Association for the Advancement of Artificial Intelligence*, 2011.

Edith Law, Burr Settles, and Tom Mitchell. Learning to Tag using Noisy Labels. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part II*, 2010. URL http://www.ml.cmu.edu/research/dap-papers/dap_law.pdf.

Edith Law, PN Bennett, and Eric Horvitz. The effects of choice in routing relevance judgments. In *Proceedings of the 34th ACM SIGIR conference on Research and development in Information*, SIGIR '11, pages 1127–1128, 2011. URL http://dl.acm.org/citation.cfm?id=2010082.

Florian Laws, Christian Scheible, and Hinrich Sch. Active Learning with Amazon Mechanical Turk. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, 2011.

Matthew Lease. On quality control and machine learning in crowdsourcing. In *Proceedings of the 3rd Human Computation Workshop*, 2011. URL http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewPDFInterstitial/3906/4255.

Hongwei Li, Bo Zhao, , and Ariel Fuxman. The Wisdom of Minority: Discovering and Targeting the Right Group of Workers for Crowdsourcing. In *Proceedings of the 23rd WWW conference*, 2014.

CH Lin and DS Weld. Dynamically switching between synergistic workflows for crowdsourcing. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. URL http://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/viewPaper/5334.

Robert B. Litterman. Specifying vector autoregressions for macroeconomic forecasting. *Federal Reserve Bank of Minneapolis Staff report*, 1(92), 1984.

Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 57–66, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866040. URL http://doi.acm.org/10.1145/1866029.1866040.

Di Liu, Ranolph Bias, Matthew Lease, and Rebecca Kuipers. Crowdsourcing for usability testing. In *Proceedings of the 75th Annual Meeting of the American Society for Information Science and Technology (ASIS&T)*, October 28–31 2012. URL ../papers/liu-asist12.pdf.

Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior research methods*, 44(1):1–23, March 2012. ISSN 1554-3528. doi: 10.3758/s13428-011-0124-6. URL http://www.ncbi.nlm.nih.gov/pubmed/21717266.

Winter Mason and DJ Watts. Financial incentives and the performance of crowds. *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2009. URL http://dl.acm.org/citation.cfm?id=1600150.1600175.

R McCreadie, Craig Macdonald, and Iadh Ounis. Crowdterrier: Automatic crowdsourced relevance assessments with terrier. *Proceedings of the 35th annual international ACM SIGIR conference on Research and development*

in information retrieval, page 4503, 2012. doi: 10.1007/s10791-012-9186-z. URL http://dl.acm.org/citation.cfm?id=2348430.

Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Machine Learning in System Biology, Journal of Machine Learning*, volume 8 of *JMLR Proceedings*, pages 65–81, 2010.

Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z. Gajos. Platemate: Crowdsourcing nutrition analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 1–12, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1.

Scott Novotney and Chris Callison-Burch. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Proceedings of HLT-NAACL*, pages 207–215, Los Angeles, California, June 2010. URL http://www.aclweb.org/anthology/N10-1024.

Yubin Park, Carlos Carvalho, and Joydeep Ghosh. Lamore: A stable, scalable approach to latent vector autoregressive modeling of categorical time series. In *17th International conference AISTAT*, 2014.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Ethan Petuchowski and Matthew Lease. TurKPF: TurKontrol as a Particle Filter. Technical report, University of Texas at Austin, April 2014. arXiv:1404.5078.

Ignazio Pillai, Giorgio Fumera, and Fabio Roli. Multi-label classification with a reject option. *Pattern Recognition*, 46(8):2256 – 2266, 2013. ISSN 0031-3203.

Alexander J. Quinn and Benjamin B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1403–1412, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979148. URL `http://doi.acm.org/10.1145/1978942.1979148`.

Adrian E. Raftery. A model for high-order markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(3):528–539, 1985.

VC Raykar and S Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13: 491–518, 2012. URL `http://dl.acm.org/ft_gateway.cfm?id=2188401&type=pdf`.

VC Raykar, Shipeng Yu, and LH Zhao. Learning from crowds. *Journal of*

Machine Learning Research, 11:1297–1322, 2010. URL http://dl.acm.org/citation.cfm?id=1859894.

Joel Ross, L Irani, and M Silberman. Who are the crowdworkers?: shifting demographics in mechanical turk. *Proceedings of the 28th CHI '10 Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872, 2010. URL http://dl.acm.org/citation.cfm?id=1753873.

Jeffrey M. Rzeszotarski and Aniket Kittur. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 13–22, 2011. ISBN 978-1-4503-0716-1.

Dafna Shahaf and Eric Horvitz. Generalized task markets for human and machine computation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 986–993, 2010.

Aaron D. Shaw, John J. Horton, and Daniel L. Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, pages 275–284, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0556-3. doi: 10.1145/1958824.1958865. URL http://doi.acm.org/10.1145/1958824.1958865.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proc. ACM KDD*, pages 614–622, 2008.

Aashish Sheshadri and Matthew Lease. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*, pages 156–164, 2013a. URL `http://ir.ischool.utexas.edu/square/documents/sheshadri.pdf`.

Aashish Sheshadri and Matthew Lease. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*, 2013b. URL `http://ir.ischool.utexas.edu/square/documents/sheshadri.pdf`.

Mark D Smucker and Chandra Prakash Jethani. Measuring assessor accuracy: a comparison of NIST assessors and user study participants. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1231–1232, 2011. ISBN 9781450307574.

R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast – but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263, 2008.

Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214(0):89 – 111, 2014. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/j.artint.2014.04.005. URL `http://www.sciencedirect.com/science/article/pii/S0004370214000538`.

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. ACM, 2014.

Matteo Venanzi, John Guiver, Pushmeet Kohli, and Nick Jennings. Time-Sensitive Bayesian Information Aggregation for Crowdsourcing Systems. *CoRR*, abs/1510.06335, 2015. URL `http://arxiv.org/abs/1510.06335`.

Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2011. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5995430`.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269, 1967.

Robert Voyer, Valerie Nygaard, Will Fitzgerald, and Hannah Copperman. A hybrid model for annotating named entity training corpora. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 243–246, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-72-5. URL `http://dl.acm.org/citation.cfm?id=1868720.1868759`.

Jing Wang, S Faridani, and P Ipeirotis. Estimating the completion time of crowdsourced tasks using survival analysis models. In *Proceedings of*

the Workshop on Crowdsourcing for Search and Data Mining (WSDM11-CSDM), 2011. ISBN 9781450302227. URL http://ir.ischool.utexas.edu/csdm2011/proceedings/csdm2011_proceedings.pdf#page=31.

Daniel S. Weld, Mausam, and Peng Dai. Execution control for crowdsourcing. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, UIST '11 Adjunct, pages 57–58, New York, NY, USA, 2011a. ACM. ISBN 978-1-4503-1014-7. doi: 10.1145/2046396.2046421. URL http://doi.acm.org/10.1145/2046396.2046421.

D.S. Weld, P. Dai, and Others. Human Intelligence Needs Artificial Intelligence. In *Proc. AAAI HComp workshop*, 2011b. URL http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewPaper/3972.

P. Welinder and P. Perona. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 25–32, 2010a.

Peter Welinder and Pietro Perona. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 25–32. Ieee, June 2010b. ISBN 978-1-4244-7029-7. doi: 10.1109/CVPRW.2010.5543189. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5543189.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The

Multidimensional Wisdom of Crowds. In *Proceeding of the conference on Neural Information Processing Systems (NIPS)*, pages 1–9, 2010. URL `http://www.vision.caltech.edu/visipedia/papers/WelinderEtalNIPS10.pdf`.

Jacob Whitehill, Paul Ruvolo, Ting fan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS'09)*, pages 2035–2043, 2009.

Jinfeng Yi, Rong Jin, Shaili Jain, and Anil K. Jain. Inferring Users' Preferences from Crowdsourced Pairwise Comparisons: A Matrix Completion Approach. In *1st AAAI Conference on Human Computation (HCOMP)*, 2013.

Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. Task recommendation in crowdsourcing systems. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, pages 22–26, 2012.

Scott L. Zeger, Kung-Yee Liang, and Paul S. Albert. Models for longituidinal data: A generalized estimating equation approach. *Biometrics*, 44:1049–1060, 1988.

Haoqi Zhang, Edith Law, R Miller, and K Gajos. Human computation tasks with global constraints. In *Proc. CHI*, pages 217–226, 2012. ISBN 9781450310154. URL `http://dl.acm.org/citation.cfm?id=2207708`.

X. Zhen and I. V. Basawa. Observation-driven generalized state space models for categorical time series. *Statistics and Probability Letters*, 79:2462–2468, 2009.

Dongqing Zhu and Ben Carterette. An analysis of assessor behavior in crowdsourced preference judgments. In *SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE2010)*, pages 21–26, 2010. URL `http://ir.ischool.utexas.edu/cse2010/materials/zhucarterette.pdf`.

Walter Zucchini and Iain L. MacDonald. *Hidden Markov Models for Time Series: An Introduction Using R*. Chapman and Hall/CRC, 2009.

Index

Vita

Hyun Joon Jung was born in Seoul, South Korea, on October, 1979. He received a B.S. in Computer Science from Korea University in August 2003, and a M.S.E. in Computer Science and Engineering from Seoul National University in February 2006. He receives a PhD in the University and Texas at Austin.

Permanent address: 10146 Alpine Dr.
Cupertino, CA 95014

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.