Toward a Robust and Universal Crowd Labeling Framework

Faiza Khan Khattak

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Toward a Robust and Universal Crowd Labeling Framework

Faiza Khan Khattak

The advent of fast and economical computers with large electronic storage has led to a large volume of data, most of which is unlabeled. While computers provide expeditious, accurate and low-cost computation, they still lag behind in many tasks that require human intelligence such as labeling medical images, videos or text. Consequently, current research focuses on a combination of computer accuracy and human intelligence to complete labeling task. In most cases labeling needs to be done by domain experts, however, because of the variability in expertise, experience, and intelligence of human beings, experts can be scarce.

As an alternative to using domain experts, help is sought from non-experts, also known as *Crowd*, to complete tasks that cannot be readily automated. Since crowd labelers are non-expert, multiple labels per instance are acquired for quality purposes. The final label is obtained by combining these multiple labels. It is very common that the ground truth, instance difficulty, and the labeler ability are unknown entities. Therefore, the aggregation task becomes a "chicken and egg" problem to start with.

Despite the fact that much research using machine learning and statistical techniques has been conducted in this area (e.g., [Dekel and Shamir, 2009; Hovy *et al.*, 2013a; Liu *et al.*, 2012; Donmez and Carbonell, 2008]), many questions remain unresolved, these include: (a) What are the best ways to evaluate labelers? (b) It is common to use expert-labeled instances (ground truth) to evaluate labeler ability (e.g., [Le *et al.*, 2010; Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012; Khattak and Salleb-Aouissi, 2013]). The question is, what should be the cardinality of the set of expert-labeled instances to have an accurate evaluation? (c) Which factors other than labeler expertise (e.g., difficulty of instance, prevalence of class, bias of a labeler toward a particular class) can affect the labeling accuracy? (d) Is there any optimal way to combine multiple labels to get the

best labeling accuracy? (e) Should the labels provided by oppositional/malicious labelers be discarded and blocked? Or is there a way to use the "information" provided by oppositional/malicious labelers? (f) How can labelers and instances be evaluated if the ground truth is not known with certitude?

In this thesis, we investigate these questions. We present methods that rely on few expert-labeled instances (usually 0.1% -10% of the dataset) to evaluate various parameters using a frequentist and a Bayesian approach. The estimated parameters are then used for label aggregation to produce one final label per instance.

In the first part of this thesis, we propose a method called *Expert Label Injected Crowd Estimation* (ELICE) and extend it to different versions and variants. ELICE is based on a frequentist approach for estimating the underlying parameters. The first version of ELICE estimates the parameters i.e., labeler expertise and data instance difficulty, using the accuracy of crowd labelers on expert-labeled instances [Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012]. The multiple labels for each instance are combined using weighted majority voting. These weights are the scores of labeler reliability on any given instance, which are obtained by inputting the parameters in the logistic function.

In the second version of ELICE [Khattak and Salleb-Aouissi, 2013], we introduce entropy as a way to estimate the uncertainty of labeling. This provides an advantage of differentiating between good, random and oppositional/malicious labelers. The aggregation of labels for ELICE version 2 flips the label (for binary classification) provided by the oppositional/malicious labeler thus utilizing the information that is generally discarded by other labeling methodologies.

Both versions of ELICE have a cluster-based variant in which rather than making a random choice of instances from the whole dataset, clusters of data are first formed using any clustering approach e.g., K-means. Then an equal number of instances from each cluster are chosen randomly to get expert-labels. This is done to ensure equal representation of each class in the test dataset.

Besides taking advantage of expert-labeled instances, the third version of ELICE [Khattak and Salleb-Aouissi, 2016], incorporates pairwise/circular comparison of labelers to labelers and instances to instances. The idea here is to improve accuracy by using the crowd labels, which unlike expert-labels, are available for the whole dataset and may provide a more comprehensive view of the labeler ability and instance difficulty. This is especially helpful for the case when the domain

experts do not agree on one label and ground truth is not known for certain. Therefore, incorporating more information beyond expert labels can provide better results.

We test the performance of ELICE on simulated labels as well as real labels obtained from Amazon Mechanical Turk. Results show that ELICE is effective as compared to state-of-the-art methods. All versions and variants of ELICE are capable of delaying phase transition. The main contribution of ELICE is that it makes the use of all possible information available from crowd and experts. Next, we also present a theoretical framework to estimate the number of expert-labeled instances needed to achieve certain labeling accuracy. Experiments are presented to demonstrate the utility of the theoretical bound.

In the second part of this thesis, we present *Crowd Labeling Using Bayesian Statistics* (CLUBS) [Khattak and Salleb-Aouissi, 2015; Khattak *et al.*, 2016b; Khattak *et al.*, 2016a], a new approach for crowd labeling to estimate labeler and instance parameters along with label aggregation. Our approach is inspired by Item Response Theory (IRT). We introduce new parameters and refine the existing IRT parameters to fit the crowd labeling scenario. The main challenge is that unlike IRT, in the crowd labeling case, the ground truth is not known and has to be estimated based on the parameters. To overcome this challenge, we acquire expert-labels for a small fraction of instances in the dataset. Our model estimates the parameters based on the expert-labeled instances. The estimated parameters are used for weighted aggregation of crowd labels for the rest of the dataset. Experiments conducted on synthetic data and real datasets with heterogeneous quality crowd-labels show that our methods perform better than many state-of-the-art crowd labeling methods.

We also conduct significance tests between our methods and other state-of-the-art methods to check the significance of the accuracy of these methods. The results show the superiority of our method in most cases. Moreover, we present experiments to demonstrate the impact of the accuracy of final aggregated labels when used as training data. The results essentially emphasize the need for high accuracy of the aggregated labels.

In the last part of the thesis, we present past and contemporary research related to crowd labeling. We conclude with future of crowd labeling and further research directions. To summarize, in this thesis, we have investigated different methods for estimating crowd labeling parameters and using them for label aggregation. We hope that our contribution will be useful to the crowd labeling community.

This page is intentionally left blank.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This thesis exists because of my honorable advisor *Ansaf Salleb-Aouissi*. There is no way it would have existed otherwise. I am really blessed to have her as my advisor. She has been the greatest source of inspiration and motivation. I really cannot thank her enough for believing in me when no one else did, not even myself. Had she not been my advisor, I might have quitted long ago. I cannot say enough to appreciate her, it is all beyond words, in undiscovered dimensions of heart and soul.

I also thank my thesis committee who agreed to spend their valuable time reviewing my thesis. I am truly grateful to Julia Hirschberg, Rebbecca Passonneau, Anita Raja, Panos Ipeirotis and Ansaf Salleb-Aouissi.

I am thankful to all my teachers from Kindergarten to Columbia University. I would also like to thank my friends everywhere and at Columbia including Tayyaba Sharif, Faiza Yousaf, Asma Tanvir, Ume Kalsoom, Summera Asif, Sara Alkuhlani, Heba Elfardy and Boyi Xie.

I also like to convey my special thanks to Dr. Matthews. I have learned so much from her during these years. She has made me a stronger human being and has changed the perception of life for me. It is a blessing that I have known her in this life time.

I would also thank my father for his prayers, my mother who did not live long enough to see me complete my thesis, my sister for cheering me up at all times, my brother for asking me that annoying question everyday “When are you completing your Ph.D.?”, my husband for living all the successes and disappointments with me and many thanks to my lifeline my beloved son.

Last but not least, I would like to thank God for brining all the above-mentioned people in my life and helping me through all hard times.

Dedicated to
my parents,
my sister,
my brother,
my husband,
and my beloved son.

Chapter 1

Introduction to Crowd Labeling

With the advent of digitization, Big Data became available everywhere, affecting almost every field in our daily life. While data is abundant, most of it still remains in an unlabeled form and not readily available for prediction tasks through machine learning algorithms. Although computers provide expeditious, accurate and low-cost computation, they still lag behind in many tasks that require human intelligence such as labeling medical images, videos or text to cite a few. In most cases labeling needs to be done by domain experts; however, because of the variability in expertise, experience and intelligence of human beings, experts can be scarce. As an alternative to using domain experts, help is sought from non-experts, also known as *Crowd*, to complete tasks that can't be readily automated.

In a crowd labeling process, multiple labels are acquired for each data instance from the crowd workers (also called labelers or annotators). Labels can be binary, categorical, ordinal or continuous. Multiple labels are acquired for quality assurance and then aggregated to get one final label. Different approaches are commonly used in the aggregation process. Crowd labeling is generally done through an open call and nowadays on website platforms. Examples include labeling an image and choosing the meaning of a word on Amazon Mechanical Turk (AMT). In a typical crowd labeling scenario, the identity of the labeler and the requester of the task are not known to each other.

Crowd labeling is a subfield of crowdsourcing but mostly is referred to as crowdsourcing in the literature. While crowd labeling focuses on labeling task done by the crowd, crowdsourcing is the process of hiring crowd services for a variety of tasks including designing a logo, writing an essay

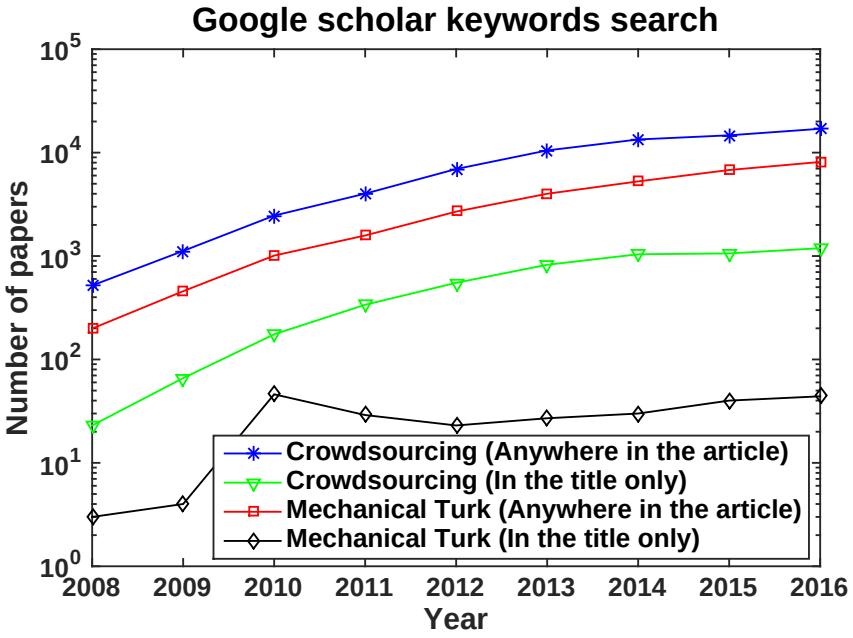

Figure 1.1: Keyword search on Google Scholar

other than usual labeling tasks [Doan *et al.*, 2011].

The idea of crowd labeling is not new. In 1714, the British government offered a prize to invent a method to measure the longitude [Lynch, 2012]. Relatively recent historical evidence recorded by Nelson [Nelson, 2008], is from the 1880's when Harvard Observatory in Cambridge, Massachusetts took the images of thousands of stars on photographic plates. A team of untrained women, hired at very low pay, labeled about half a million of such photographic plates. They analyzed these photographic plates using magnifying glass to catalog the stars [Nelson, 2008].

Due to the availability of crowd services at low rates, crowd labeling has attracted the attention of many researchers. Therefore, crowd labeling literature has increased at an exponential rate in the past few years (see Graph 1.1) and cannot be thoroughly summarized in one paper. In this chapter, we give a very general overview about the crowd labeling motivation, process, techniques, logistics and future.

1.1 Motivation of the Crowd

While crowd labeling is popular today, it is interesting to analyze what motivates the crowd to complete labeling tasks. Crowd motivation can be categorized as follows [Quinn and Bederson, 2011].

(i) **Fun or Virtual Money:** Many crowd labeling tasks are done just for enjoyment or virtual rewards e.g., ESP game [von Ahn and Dabbish, 2004] and FoldIt.

(ii) **Embedded work:** Sometimes the crowd labeling task is embedded in some other tasks, making labeling mandatory e.g., the reCAPTCHA project [Ahn *et al.*, 2008].

(iii) **Voluntary or Pastime:** Sometimes crowd volunteer to work. For instance, Family Search Indexing aims to create searchable family history digital indexes from scanned images of historical documents. These include birth and death certificates, marriage licenses and property records.

(iv) **Altruism:** Another motivation for the crowd is altruism e.g., a search for a missing computer scientist Jim Gray [Hellerstein and Tennenhouse, 2010]. Satellite images of the area of disappearance were uploaded on AMT for labeling the possible locations to search.

(v) **Reputation/Getting Noticed:** Sometimes, crowd labeling is done to get recognition or earn reputation e.g., volunteer translators at childrenlibrary.org. Also many games come into this category for which getting a high score and publishing it on social media can be fulfilling.

(vi) **Payment:** A large part of crowd labeling is done for payment, e.g., Amazon Mechanical Turk (AMT), Crowdtask and Clickworkers.

(vii) **Flexible job & Task Autonomy:** Crowd labeling provides workers with a flexible work schedule without any pressure from an employer.

1.2 Crowd Labeling Process

Crowd labeling mainly consists of the following steps (also see Figure 1.3.)

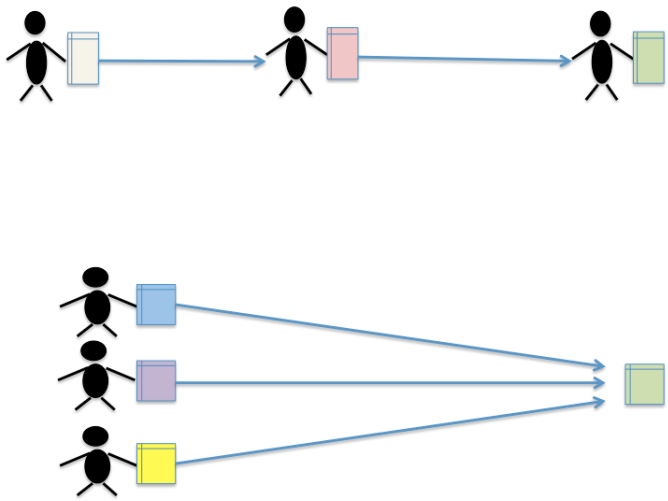

Figure 1.2: (Top) Sequential workflow (Bottom) Parallel workflow.

1.2.1 Task Design

Task design is one of the crucial parts of crowd labeling process. While designing the task, the following points should be considered:

(a) **Query Formulation:** The query should be clearly stated and accompanied with instructions and examples. External links can be provided for more information especially when the problem is field-specific and labelers are not expected to know about it in advance. This reduces the chance of the task to be misunderstood and hence improve the labels accuracy [Kittur *et al.*, 2008].

(b) **Task Division** It is important to divide crowd tasks into smaller subsets so that a worker can do it without being overwhelmed and many workers can work on the task in parallel. Task division is extra work on the requesters' part, reducing the benefit obtained from utilizing the crowd. Some researchers have proposed methods for easier task division such as Turkomatic [Kulkarni *et al.*, 2011] and Turkit [Little *et al.*, 2009; Little *et al.*, 2010b].

(c) **Types of workflow** There are two main types of workflows for crowd labeling: parallel workflow, and iterative workflow [Kulkarni *et al.*, 2011]. In parallel workflow, all the workers do the tasks independently of each other, and tasks are combined afterwards. While in iterative workflow, each worker completes his task, which is then passed on to another worker to improve the

outcome. This process goes on for a predefined number of iterations. A combination of the two workflows can also be used.

(d) **Task Assignment** Most of the crowd labeling tasks are published as an open call. But sometimes qualification tests for labelers are required for a more appropriate task assignment. Moreover, research is also being done to improve the task assignment using optimization methods [Ho *et al.*, 2013] . This adaptive task assignment can help in improving labeling accuracy and reducing the labeling cost.

(e) **Number of labels needed** Research shows that getting more crowd labels increases the cost but may lead to an improvement in accuracy [Sheng *et al.*, 2008]. Therefore, care must be taken while deciding the number of requested crowd labels. Deciding on the number of labels depends on the task budget, task nature, quality of the crowd and the size of the dataset. There is no well-known general rule for deciding about the number of labels needed.

1.2.1.1 Worker Problems & Solutions

Sometimes crowd workers face labeling-related issues, which should be kept in mind while designing the task [Silberman *et al.*, 2010b; Irani and Silberman, 2013]. Worker problems include:

(a) **Low pay and Long pay delays:** Payment for most of the tasks ranges from $0.01-1$.

(b) **Work rejection:** Requesters have the right to reject the work if it is below standard but some requesters may reject the work to avoid payment.

(c) **Task time:** Sometimes task completion time is too short and workers are not able to complete the task and do not get paid.

(d) **Lack of communication:** Uncommunicative requesters who do not resolve the issues discourage the workers.

(e) **Error in tasks:** Sometimes task posted has some errors, e.g., worker is unable to submit the completed task due to a website issue. As a result the worker does not get paid and has to bear the cost of the requester's mistake or technical issues.

(f) **Fraudulent tasks:** There is a minor but not negligible risk in tackling crowd labeling task as some may be fraudulent. These may damage the computer of the worker or cause other kinds of threats.

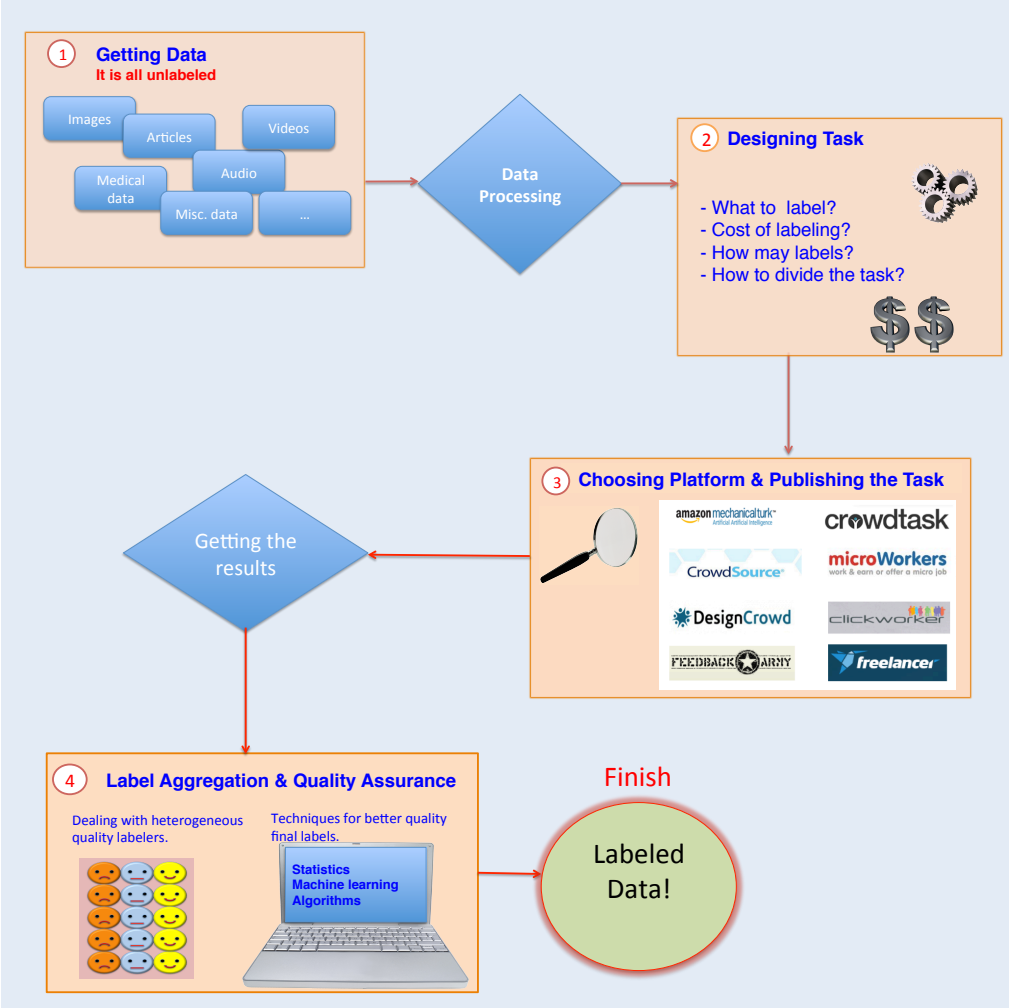

Figure 1.3: Crowd labeling Process

Proposed solutions to the worker problems [Bederson and Quinn, 2011; Donmez and Carbonell, 2008] include defining hourly pay, giving feedback about work quality, improving communication between requester and worker, providing more details about the task and limiting anonymity. A more practical solution to worker problems is Turkopticon [Kulkarni *et al.*, 2011], which is used to get workers' reviews about requesters. This feedback helps the workers to know about the requesters

beforehand and also helps the requesters improve the quality of the task design.

1.2.1.2 Demographics

Crowd demographics should be kept in mind while designing the tasks. The website MTurk Tracker (http://demographics.mturk-tracker.com/) provides live hourly and daily details of AMT crowd demographics. The information on this website shows that crowd labelers contribute from different parts of the world with a large percentage of workers from the United States and India. Since the crowd workers have different cultural and social background, their perception about the same problem can be quite different.

1.2.2 Choice of a Crowd Labeling Platform

Crowd labeling has led to the development of several websites, which provide many possibilities of a platform for publishing and accomplishing crowd work to choose from. There are many general-purpose platforms, such as Amazon Mechanical Turk (AMT) and CrowdFlower [Le *et al.*, 2010]. AMT is a big market place for posting crowd labeling tasks and getting crowd labels. CrowdFlower is another website for posting tasks, but unlike many other platforms it provides aggregated labels as the final outcome. Some platforms are special-purpose and do not allow posting tasks by unauthorized people. Examples include Galaxy zoo developed by Oxford university researchers for online classification of astronomical data. FoldIt [Cooper *et al.*, 2010] is another example developed by the University of Washington researchers. FoldIt is a puzzle video game with the underlying purpose for folding the protein structure.

1.2.3 Labeler Types

Crowd labelers are non-experts. Hence, it is important to check the quality of the labeling work. In general, labelers can be categorized as follows [Raykar and Yu, 2012; Khattak and Salleb-Aouissi, 2013].

- **Good/Trained:** A labeler who is good at the labeling task as well as diligent is considered a good/trained worker.

- **Untrained/Novice:** A labeler who is new and inexperienced or does not have the enough knowledge to complete the task is considered to be untrained or novice worker.
- **Random/Lazy:** A random labeler is a careless labeler who chooses the labels randomly or semi-randomly without paying much attention to his task.
- **Oppositional/Malicious/Biased:** These are the workers who have a biased opinion maybe because of a misunderstanding of the task or due to personal preferences. This category also includes the workers who have an intention to make the labeling noisy.

Each type of labelers provides a certain amount of information about the instances through their labels. Good/trained workers are most informative while information level obtained from untrained/novice labelers is low. Random/lazy workers do not provide any information at all and are merely wastage of resources. Oppositional/malicious/biased labelers provide labels, which are not good in their raw form but once adjusted can be as informative as the labels provided by the good/trained labelers. The reason is that oppositional/malicious/biased labelers work hard to identify the instance and label it according to their inclination. Therefore, it is important to identify the oppositional/malicious/biased labelers and correct their labels. Otherwise, they can be as non-informative as the random/lazy labelers.

1.2.4 Common Techniques for Quality Assurance

Quality assurance for crowd labeling has received a lot of attention from researchers. Many methods have been developed for improving the accuracy of the final label. Generally, the proposed approaches use one or more of the following techniques [Quinn and Bederson, 2011].

1.2.4.1 Redundancy

Redundancy refers to acquiring multiple opinions for each instance to label to improve accuracy. We discuss two methods, which are based on redundancy.

- **Majority Voting (MV):** Multiple labels are acquired and majority wins. The main drawback of this method is assigning equal weights to the opinions of labelers regardless of the above-mentioned categories of workers. Moreover, in the case of even number of labelers with

evenly split votes, it becomes impossible to decide. Increasing the number of labels improves the final label quality only when labeler accuracy is above 50% while overall accuracy is deteriorated if the labelers have accuracy below 50%. For labelers with 50% accuracy, no improvement is observed and random results are obtained, which can be acquired just by tossing a coin [Sheng *et al.*, 2008].

- **Inter-annotator Agreement (ITA):** This method is widely used in Natural Language Processing (NLP). Two or more annotators work independently. If agreement [Passonneau *et al.*, 2012] between any pair of workers is above a certain threshold, then each annotator in that pair is considered an expert. For the rest of the data, labeling of these annotators is considered as true labels. The main problem with this method is that only pairwise agreements are considered while comparing each annotator to the majority of the other labelers can be a better option. Moreover, there is a high chance of agreement on wrong labels for difficult instances.

1.2.4.2 Ground Truth Seeding

To keep a check on the workers quality, an intuitive and straightforward method is to use the instances for which we have ground truth (true label). But generally, ground truth is not readily available. In most cases, it can be acquired from domain experts, who are scarce, busy, and expensive. Therefore, true labels are normally obtained only for a small subset of the data. The choice of the instances for acquiring ground truth is an interesting topic. Some researchers suggest active learning for making this choice [Yan *et al.*, 2010] and some prefer randomly choosing the instances either from the whole dataset or its clusters [Khattak and Salleb-Aouissi, 2013]. Since only a small amount of true labels is available, these should be used intelligently.

CrowdFlower [Le *et al.*, 2010] suggests testing the labelers using ground truth instances before the actual labeling task and rejecting the labelers who do not pass the test. But this approach has certain problems:

- Workers can perform well in the test and then can be careless or even oppositional/malicious while doing the actual task.
- Labelers are discouraged if rejected.
- Labelers can have more than one account and can even collude.

- Designing testing phase can take time and energy.

A solution to this problem is to embed the ground truth instances in the task itself such that labelers cannot identify them. In this case, the following challenges emerge.

- Deciding the number of ground truth instances.
- A composition of ground truth instances to make it balanced i.e., same amount of instances from each class.
- Ground truth instances should not be identifiable.

Some researchers have suggested making up the ground truth instances called *Programmatic Gold* [Oleson *et al.*, 2011] . This is done by injecting known type of errors or using previously collected labels for which the workers have high confidence. This approach cannot be applied to all types of crowd labeling tasks e.g., ground truth data for cancer diagnosis cannot be created by injecting errors.

1.2.4.3 Labeler Ability

The ability of the labeler is a measure to identify the skill level and type of labeler. It can also be used for assigning weight to the labeler opinion in the process of label aggregation. One way to obtain the ability of a labeler is through ground truth [Khattak and Salleb-Aouissi, 2013]. But sometimes, obtaining ground truth is not possible either due to the experts' disagreement or because alternative options for acquiring ground truth are not feasible or very expensive.

Researchers have tried to mediate this problem by developing Expectation Maximization (EM) based methods. It is a maximum likelihood method, which iteratively learns the unknown parameters and latent variables. In crowd labeling, EM is used to learn the final label and the ability of the labeler. In this context, a seminal paper was written by Dawid & Skene [Dawid and Skene, 1979]. EM can also be used when ground truth is available for few instances, which can help in boosting the accuracy. Other methods include message passing [von Ahn and Dabbish, 2004], variational inference [Liu *et al.*, 2012], support vector machines [Dekel and Shamir, 2009] and proactive learning [Wallace *et al.*, 2011]. Another approach corrects the labels according to labeler category [Ipeirotis and Paritosh, 2011].

1.2.4.4 Instance Difficulty

While labeler ability is of crucial importance in the labeling process, there are other factors, which cannot be overlooked. These include the difficulty of the instance. Remarkably, this factor has received less attention in the crowd labeling literature, assuming that all instances in a dataset have the same difficulty level. This assumption is not always true. The difficulty of the instance may include the difficulty level of the question itself as well as the level of clarity of the question. Note that the ability of a labeler is also affected by the instance difficulty. The performance of a good labeler can decrease if the instances are really challenging or if the problem is not well formulated. Therefore, while aggregating the labels, instance difficulty should be taken into account [Whitehill *et al.*, 2009]. The instance difficulty factor can be estimated through ground truth labels [Khattak and Salleb-Aouissi, 2013], by adding instance difficulty parameters into the EM method [Whitehill *et al.*, 2009], using the features/attributes of the instances [Karger *et al.*, 2011] or acquiring feedback from labelers about difficult instances [Welinder *et al.*, 2010a].

1.3 Challenges and Phase Transition

In a crowd labeling scene, an object is usually annotated by more than one labeler. The multiple labels obtained per object are then combined to produce one final label for quality assurance. Since the ground truth, instance difficulty and the labeler ability are generally unknown entities, the aggregation task becomes a "chicken and egg" problem to start with. While significant progress has been made on the process of aggregating crowd labeling results, e.g., [Karger *et al.*, 2014; Sheng *et al.*, 2008; Whitehill *et al.*, 2009], it is well-known that the precision and accuracy of labeling can vary due to differing skill sets. The labelers can be *good/experienced*, *random/careless* or even *oppositional*.

Oppositional labelers can include both intentionally or unintentionally oppositional. Intentionally oppositional labelers are those who identify the correct labels and change them strategically while unintentionally oppositional labelers demonstrate the same labeling behavior due to some misunderstanding about the labeling task. Throughout the rest of this paper, we will refer to both kinds as oppositional labelers.

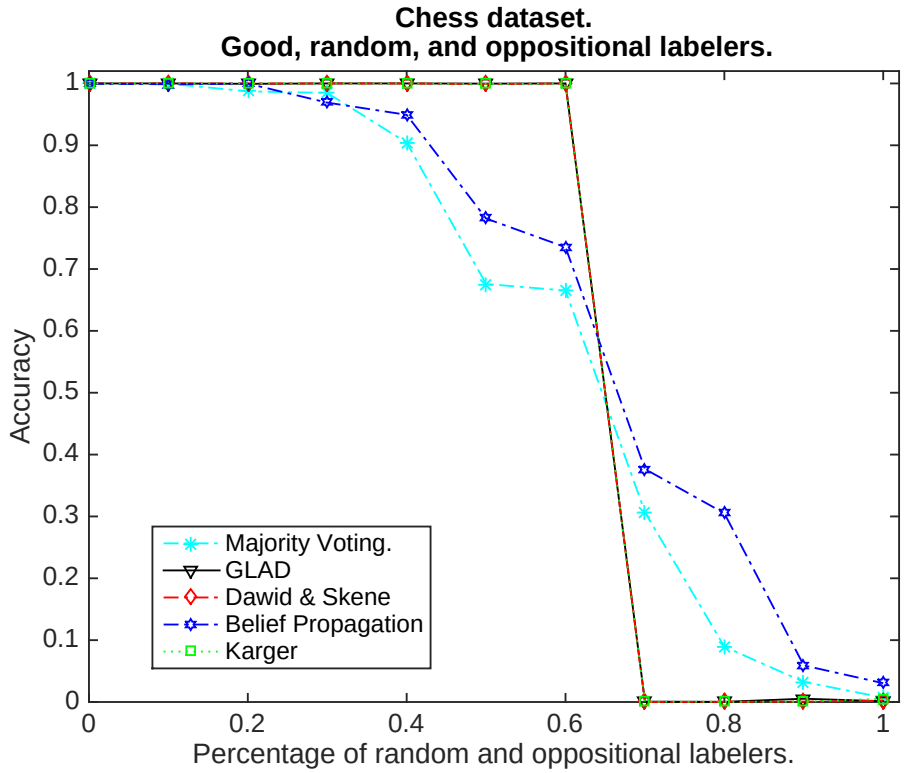

Figure 1.4: Phase transition in the performance of majority voting, GLAD [Whitehill *et al.*, 2009], Dawid and Skene's method [Dawid and Skene, 1979], Belief Propagation [Liu *et al.*, 2012] and Karger's iterative method [Karger *et al.*, 2014] on the University of California Irvine (UCI) Machine Learning Repository Chess dataset.

One of the main challenges in crowd labeling is that the proportion of low-quality/oppositional labelers is unknown. High proportion of low quality (random and oppositional) labelers can often result into a *phase transition* leading to a steep, non-linear drop in labeling accuracy as noted by [Karger *et al.*, 2014].

We observed a similar phenomenon in the experiments we conducted on five benchmark datasets from the University of California Irvine (UCI) Machine Learning Repository [Asuncion and Newman, 2007]. We used majority voting, GLAD (Generative model of Labels, Abilities, and Difficulties) by [Whitehill *et al.*, 2009], Dawid and Skene's method [Dawid and Skene, 1979], Karger's iterative method [Karger *et al.*, 2014] and Belief Propagation [Liu *et al.*, 2012]. The crowd labels for all these datasets were simulated. Figure 1 illustrates the phase transition for the UCI Chess dataset of 3,196 instances. We assume that a *good labeler* makes less than 35% mistakes, a *random*

labeler makes between 35% to 65% mistakes, while a *bad labeler* makes more than 65% mistakes. This highlights the larger challenge of producing an objective assessment to measure the quality of the crowd for a given task.

Other than phase transition, many basic questions remain unresolved that make crowd labeling a prevailing research topic, e.g., [Dekel and Shamir, 2009; Hovy *et al.*, 2013a; Liu *et al.*, 2012; Donmez and Carbonell, 2008]. The unresolved questions include:

1. What are the best ways to evaluate labeler ability and instance difficulty?
2. It is common to use expert-labeled instances or ground truth to evaluate labelers and instances [Le *et al.*, 2010; Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012; Khattak and Salleb-Aouissi, 2013]. The question is, how many expert-labeled instances should be used in order to obtain an accurate evaluation?
3. How can labelers and instances be evaluated if ground truth is not known with certitude?
4. Is there any optimal way to combine multiple labels to get the best labeling accuracy?
5. Should the labels provided by oppositional labelers be discarded and blocked? Or is there a way to use the "information" provided by oppositional labelers?

1.4 Summary of Contributions

In this thesis, we have presented two different approaches to improve crowd labeling accuracy (a) A Frequentist Approach (b) A Bayesian Approach. Both of which are based on parameter estimation. In the first approach parameters are learned by taking the frequency of correct labels provided for the expert-labeled instances, therefore this approach is named as frequentist approach. The second approach involves parameter estimation using a Bayesian method. In both approaches, the following assumptions are made.

Through out this thesis we assume:

- Classes are predefined and presented to the labelers to choose from.
- Domain experts are available to label a small fraction of dataset, usually 0.1%- 10%.

- Expert labels are assumed to be ground truth unless otherwise stated.
- We categorize the crowd labelers into three categories; good, random and oppositional. This is done based on their accuracy level. A good crowd labeler is assumed to make less than 35% mistakes, a random crowd labeler makes 35% to 65% mistakes and an oppositional crowd labeler makes 65% to 100% mistakes. Categorization is done for better visualization of results. More details about labeler categorization are available in Table 2.1.

A brief introduction of our frequentist and Bayesian approaches is given below.

1.4.1 Frequentist Approach

We present a framework called Expert Label Injected Crowd Estimation (ELICE). ELICE has three different versions along with their respective variants. The goal of ELICE is to provide better accuracy for the labeling/annotation tasks for which predefined options of answers are available. We have assumed the scenario of labeling to be questions with multiple choices provided to the labelers.

All versions of ELICE rely on expert labels for a small subset of randomly chosen instances from the dataset. However, it can be noted that instead of random choice of the instances, experts can also help in identifying the representative instances of each class. These expert-labeled instances are used to evaluate labeler ability and data instance difficulty that help to improve the accuracy of the final labels. For the first two versions of ELICE, we assume that expert-labels are ground truth labels. In the third version of ELICE however, we assume that expert-labels may not be ground truth either because the experts do not agree on the same label or because the instances are difficult and alternative methods to get ground truth are infeasible.

Earliest versions of ELICE were published in the workshop papers [Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012], which are presented as ELICE 1 in this thesis. ELICE 1 estimates the parameters, i.e., labeler expertise and data instance difficulty, using the accuracy of crowd labelers on expert-labeled instances [Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012]. The multiple labels for each instance are combined using weighted majority voting. These weights are the scores of labeler reliability on any given instance, which are obtained by inputting the parameters in the logit function.

We also present ELICE 2 [Khattak and Salleb-Aouissi, 2013] with a *new and improved* aggregation method that genuinely takes advantage of the labels provided by oppositional labelers. In the second version of ELICE [Khattak and Salleb-Aouissi, 2013], we introduce entropy as a way to estimate the uncertainty of labeling. This provides an advantage of differentiating between good, random and oppositional labelers. The aggregation method for ELICE version 2 flips the label (for binary classification case) provided by the oppositional labeler thus utilizing the information that is generally discarded by many other labeling methods.

Both versions of ELICE have a cluster-based variant in which rather than making a random choice of instances from the whole dataset, clusters of data are first formed using any clustering approach e.g., K-means. Then equal number of instances from each cluster are chosen randomly to get expert-labels. This is done to ensure equal representation of each class in the test-dataset.

Besides taking advantage of expert-labeled instances, the third version of ELICE, incorporates pairwise/circular comparison of labelers to labelers and instances to instances. The idea here is to improve the accuracy by using the crowd-labels, which unlike expert-labels, are available for the whole dataset and may provide a more comprehensive view of the labeler ability and instance difficulty. This is especially helpful for the case when the domain experts do not agree on one label and ground truth is not known for certain. Therefore, incorporating more information beyond expert-labels can provide better results.

We show empirically that our approaches are robust even in the presence of a large proportion of low-quality labelers in the crowd. This procedure also helps in stabilizing labeling process and delaying the phase transition to inaccurate labels. Furthermore, we derive a lower bound of the number of expert labels needed [Khattak and Salleb-Aouissi, 2013]. This lower bound is a function of the overall quality of the crowd and difficulty of the dataset. We present experiments showing the effectiveness of the lower bound to get better accuracy of the final label.

1.4.2 Bayesian Approach

In the second part of this thesis, we explore a Bayesian approach to the labeling. We present Crowd Labeling Using Bayesian Statistics (CLUBS), a new approach for improved crowd labeling to estimate labeler and instance parameters along with label aggregation. Our approach is inspired by Item Response Theory (IRT) Lord [1952], which is used to design and analyze test scoring strategies to

evaluate students. We introduce new parameters and refine the existing IRT model parameters to fit the crowd labeling scenario. The main challenge is that unlike IRT, in the crowd labeling case, the ground truth is not known and has to be estimated based on the parameters. To overcome this challenge, we acquire expert labels (ground truth) for a small fraction of instances in the dataset. Our model estimates the parameters based on the expert-labeled instances. The estimated parameters are used to perform weighted aggregation of crowd labels for the rest of the dataset. Experiments are conducted on synthetic data and two real datasets, which show that overall our method performs better than state-of-the-art crowd labeling methods.

1.5 Thesis Outline

This thesis is summarized and organized as follows:

- We present the ELICE framework in chapter 2.
- Empirical evaluation of ELICE is reported in Chapter 3.
- In Chapter 4, we present the theoretical framework to derive a lower bound on the number of expert labels needed for ELICE along with empirical evaluation.
- We present the Bayesian framework CLUBS in Chapter 5.
- Results for experiments using CLUBS are presented in Chapter 6. This chapter also contains the significance test results to check the significance of the accuracy of different methods. Moreover, the impact of using noisy final-labels as training data is evaluated.
- Chapter 7 summarizes the past and contemporary work in the crowd labeling area.
- Chapter 8 concludes the thesis by discussing the future directions of crowd labeling.

Part I

The Frequentist Approach

Chapter 2

Expert Label Injected Crowd Estimation (ELICE)

In this chapter, we present our frequentist approach called Expert Label Injected Crowd Estimation (ELICE). ELICE has three versions with two variants each. All versions of ELICE use expert labels. For the first two versions of ELICE, we assume that expert labels are equivalent to ground truth but for the third version, we assume that expert labels may not be gold labels. For all versions of ELICE labeling classes are pre-defined. We present empirical evaluations on different datasets with simulated and real labels in the next chapter.

2.1 Notation & Scenario

Throughout the first part of this thesis, the following notation and scenario are used.

Dataset: $\mathcal{D}$,

Cardinality of the dataset $\mathcal{D}$: N,

Label categories: $\{-1, 1\}$,

Number of crowd labelers: M,

Index i: Represents instances $i \in \{1, 2, 3, ..., N\}$,

Index j: Represents labelers $j \in \{1, 2, 3, ..., M\}$,

Crowd label for i^{th} instance by the j^{th} labeler: l_{ij},

A subset of randomly chosen instances to get expert labels: $\mathcal{D}'(\subset \mathcal{D})$,

Number of expert-labeled instance: $n << N$,

Expert-labels for instance i: L_i,

Labeler ability of labeler j**:** α_j,

Instance difficulty of instance i: β_i.

Scenario: Let $\mathcal{D}$ be a dataset of N unlabeled instances. We assign M crowd labelers to label the whole dataset; each instance i will receive a label $l_{ij} \in \{\pm 1\}$ from labeler j, where $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$. To evaluate the performance of the labelers, we get "ground truth" labels L_i for a random sample $\mathcal{D}'(\subset \mathcal{D})$ of cardinality $n << N$ (usually 0.1% -10% of the dataset). Instances of $\mathcal{D}'$ are labeled by one or more experts getting one expert-label each.

2.2 Labeler Categories

Labelers can be of different ability level and need to be categorized accordingly for a better understanding of labeling process. Initially, we categorized the labelers into 11 different categories based on their performance, as given in the first column of Table 2.1. But for convenience and reduced complexity as well as a better understanding of the labeler performance, we reduced the categories to only three that is good, random and oppositional, mentioned in the third column of Table 2.1.

Good labelers: It should be noted that the labeler categorized as "good" does not necessarily have the exceptional performance and can make up to 35% mistakes. The reason this labeler is categorized as "good" is because such labeler will be correct more than 65% of the time and the labels provided by this labeler can potentially help in the labeling task.

Random labelers: These labelers make 36% to 65% mistakes. Random labelers are labelers who randomly label without paying any attention to the instances. These labelers are either lazy or want to get more work done in a short time to be able to earn more money. They provide least or no information as their labeling is random (or nearly random) and cannot help in the labeling process.

Oppositional labelers: Similarly, oppositional labeler category includes all the labelers with less than 65% of correct answers, which means these labelers provide wrong labels most of the time. It is important to note that this category can be subdivided into two kinds of labelers (a) the labelers being oppositional because they misunderstood the task (b) the labelers being oppositional because

Category	% of mistakes	Combined category
Exceptional	0% -5%	Good
Excellent	6% -15%	
Competent	16% -25%	
Fair	26% -35%	
Nearly random	36% -45%	Random
Totally Random	46% -55%	
Nearly random	56% -65%	
Bad	66% -75%	Oppositional
Incompetent	76% - 85%	
Malicious	86% - 95%	
Totally oppositional	96% - 100%	

Table 2.1: Categorization of labelers

they are really malicious and deliberately provide wrong labels. The outcome of both subcategories is the same hence they are combined and dealt with in the same manner throughout this thesis. Since in this thesis, we have focused on binary labeling only, the oppositional labelers will be providing flipped labels, irrespective of the reason for their oppositional behavior.

It is also worth noting that in all our methods presented in this chapter, the labels provided by all the labelers categorized as "good" are not treated in the same way in the final aggregation. The higher the mistakes level of the labeler the lower weight his labels will have. This automatically adjusts the impact of any labeler in the final aggregation of labels. The same is true for the other two categories, that is random and oppositional labelers.

2.3 ELICE 1 Framework

In this section, we present the first version of ELICE [Khattak and Salleb-Aouissi, 2011] along with its variant called ELICE with clustering. The detailed methodology is described below.

2.3.1 ELICE 1

We start by calculating the parameters: labeler ability and instance difficulty based on n expert-labeled instances. The ability of labeler j, denoted by α_j, can have a value between -1 and 1, where 1 is the score of a labeler who labels all instances correctly and -1 is the score of a labeler who labels everything incorrectly. This is because the expertise of a crowd labeler is penalized by subtracting 1 when he makes a mistake but it is incremented by 1 when he labels correctly. At the end, the sum is divided by n. Similarly, β_i denotes the difficulty level of instance i, which is calculated by adding 1 when a crowd labeler labels that particular instance correctly. The sum is normalized by dividing by M. It can have a value between 0 and 1, where 0 is for difficult instances and 1 is for the easy ones. We calculate α_j's and β_i's as follows,

$$\text{Labeler ability} = \alpha_j = \frac{1}{n}\sum_{i=1}^{n}[\mathbf{1}(L_i = l_{ij}) - \mathbf{1}(L_i \neq l_{ij})] \tag{2.1}$$

$$\text{Instance difficulty} = \beta_i = \frac{1}{M}\sum_{j=1}^{M}[\mathbf{1}(L_i = l_{ij})] \tag{2.2}$$

where $j = 1, \ldots, M$ and $i = 1, \ldots, n$.

We infer the rest of the $(N - n)$, β's based on α's . As the true labels for the rest of instances are not available, we try to find an approximation which we name as *hypothesized label* (HL),

$$HL_i = sign(\frac{1}{M}\sum_{j=1}^{M}\alpha_j * l_{ij}) \tag{2.3}$$

These hypothesized labels are used to approximate the β's ,

$$\beta_i = \frac{1}{M}\sum_{j=1}^{M}[\mathbf{1}(HL_i = l_{ij})] \tag{2.4}$$

The logistic function denoted by σ is used to calculate the score associated with the correctness of a label, based on the level of expertise of the crowd labeler and the difficulty of the instance. This score gives us the approximation of the true labels (F) using the following formula:

$$F_i = sign(\frac{1}{M}\sum_{j=1}^{M}\sigma(\alpha_j\beta_i) * l_{ij}) \tag{2.5}$$

Here i denotes the instances for which expert-labels are not available.

2.3.2 ELICE 1 with Clustering

We propose a variation of ELICE called ELICE with clustering. Instead of picking the instances randomly from the whole dataset $\mathcal{D}$ to acquire expert labels, clusters of instances in $\mathcal{D}$ are first formed by applying k-means clustering using the features (if available); then equal numbers of instances are chosen from each cluster and given to the expert to label. This allows us to have expert-labeled instances from different groups in the data, particularly when the dataset is highly skewed. Another possibility is to use any other method of clustering, for instance, K-means++ [Arthur and Vassilvitskii, 2007].

2.3.3 Test case

We use the UCI Chess dataset as a test case to see the effectiveness of our method (see Figure 2.1). Chess dataset consists of 3196 instances out of which we used 20 as expert-labeled instances for ELICE 1. It can be seen that ELICE 1 performs better than state-of-the-art methods by delaying

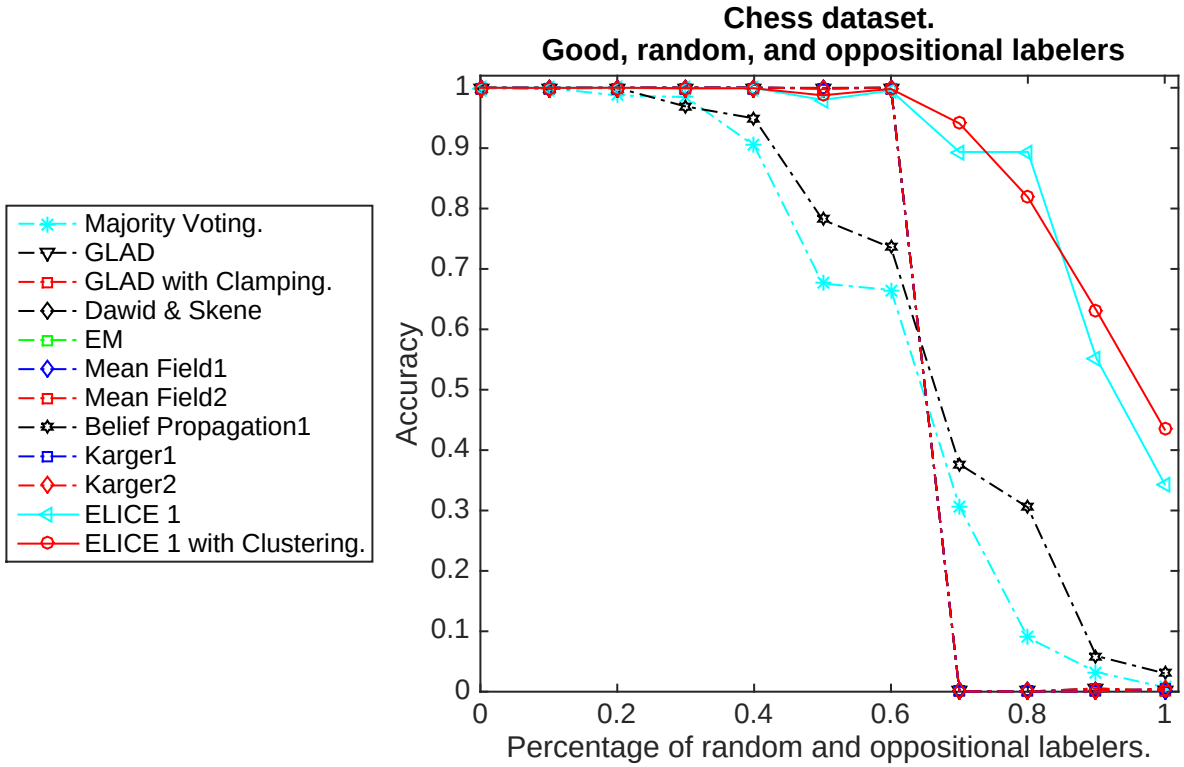

Figure 2.1: Performance on the UCI Chess dataset. We start with all good labelers and keep on increasing the percentage of random and oppositional labelers. Number of expert labels used for ELICE and all its versions is 20.

phase transition. As the percentage of good labelers decreases the performance of state-of-the-art methods deteriorates while ELICE stays stable. State-of-the-art methods include majority voting, GLAD [Whitehill *et al.*, 2009], Dawid and Skene's method [Dawid and Skene, 1979], Belief Propagation [Liu *et al.*, 2012], and Karger's iterative method [Karger *et al.*, 2014].

2.3.4 Summary

This version of ELICE is simple and easy to implement. It provides better results than state-of-the-art methods. The key factor in this version of ELICE is that it relies on the judgment of the good labelers minimizing the effect of the random or oppositional labelers. This can especially be helpful when at least one good labeler is available. The low computational cost and effectiveness of the approach as compared to state-of-the-art methods are the main advantages of this methodology.

2.4 ELICE 2 Framework

In the first version of ELICE, the random and oppositional labelers are treated in the same way i.e., their opinion is weighted less than the good labelers. However, it is known from the crowd labeling literature, e.g., [Raykar and Yu, 2012] that oppositional labelers can be informative in their own way and once they are identified, their labels can be adjusted to get the underlying possibly correct labels.

The random labelers, on the other hand, are those who label without paying attention to instances. Therefore, their labels merely add noise to the labeling process. The oppositional labelers are not random in their labels. They take time to identify the instance, try to infer the correct label and then flip it intentionally or unintentionally (assuming binary classification). Therefore, if we know the underlying intentions of a labeler in advance, we can obtain the correct label by decrypting the provided label.

In the second version of ELICE, we have incorporated the idea of utilizing the labels provided by the oppositional labelers. Just like the previous version of ELICE, the labeler ability and instance difficulty are evaluated but this time the evaluation involves the concept of entropy. Entropy measures the uncertainty of the information provided by the labelers (or uncertainty about the information obtained for the instances). A random labeler will have a high entropy while the good or oppositional labeler will have a low entropy.

This lets us differentiate between random vs. oppositional or good labelers. Then the oppositional and good labelers are separated. ELICE 2 assigns low weights to the labels of a random labeler and high weights to the labels of a good labeler. Oppositional labelers' annotations are also highly weighted but after adjusting the labels provided by them. This helps us in using the information that is discarded by many other label aggregation methods. Clustering method can also be used for this version of ELICE.

2.4.1 ELICE 2

In this section, we present methodology for ELICE 2.

2.4.1.1 Labeler Expertise

We use expert-labeled instances to evaluate the labelers by finding the probability of getting correct labels. This estimation of labeler's performance has a factor of uncertainty since it is based on a sample. Therefore, the entropy function can be a natural way to measure this uncertainty. Entropy is high when the probability is around 0.5 as we are least certain about such a labeler and it is low when the probability is close to 0 or 1. The formula for the entropy for a worker j is given by:

$$E_j = -p_j log(p_j) - q_j log(q_j) \quad \text{such that,} \quad p_j = \frac{n_j^+}{n} \qquad q_j = 1 - p_j \tag{2.6}$$

$n_j^+ = |$correctly labeled instances from $\mathcal{D}'$ by labeler j$|$

Since we are more interested in the reliability of the assessment, we take $(1 - E_j)$. In order to differentiate between good and bad labelers, we multiply by $(p_j - q_j)$. This assigns a negative value to the bad labeler and positive value to the good one. We define the expertise of the labeler as

$$\alpha_j = (p_j - q_j)(1 - E_j) \tag{2.7}$$

where $\alpha_j \in [-1, 1]$. The multiplication by $(p_j - q_j)$ also allows for less variability in α_j when the number of correct and incorrect labels is close, assuming that it can be due to the choice of the instances in $\mathcal{D}'$. This Equation 2.7 is described as the difference of the probability of getting correct labels and probability of getting incorrect labels by labeler j times the measure of certainty of the label provided by labeler j.

We can use α to categorize the labelers as follows:

- *Random guesser* is the labeler with α close to zero. This labeler is either a lazy labeler who randomly assigns the labels without paying any attention to the instances or an inexperienced labeler.
- *Good labeler* is the labeler with α close to 1. He does a good job of labeling.
- *Oppositional labeler* is the labeler with α close to -1. He guesses the correct label and then flips it.

2.4.1.2 Instance Difficulty

Similarly, the difficulty of an instance is defined as:

$$\beta_i = (p'_i - q'_i)(1 - E'_i) + 1 \tag{2.8}$$

where $p'_i = \frac{M_i^+}{M}$ $q'_i = 1 - p'_i$, p'_i is the probability of getting a correct label for instance i, from the crowd labeler and M_i^+ is the number of correct labels given to the instance i. Also,

$$E'_i = -p'_i log(p'_i) - q'_i log(q'_i) \tag{2.9}$$

represents the entropy for the instance i which measures the uncertainty in our assessment of the difficulty of the instance. All these values are calculated using the expert labeled instances. We have added 1 to the formula in (2.8) because we find it more convenient mathematically to make the value of β positive. Another reason for adding 1 is that we cannot assume the difficulty level to be negative, just because the labelers did a bad job of labeling. This Equation 2.8 is described as the difference of the probability of getting correct labels and the probability of getting incorrect labels for instance i times the measure of certainty of the label provided for instance i plus1.

We have $\beta_i \in [0, 2]$ which is used to categorize the instances as follows:

- *Easy instance* is the one with β close to 2.
- *Average difficulty instance* is the instance with β around 1.
- *Difficult instance* is the instance with β close to 0.

To judge the difficulty level of the remaining $(N - n)$ instances, we define *hypothesized labels* W_i as:

$$HL_i = sign(\sum_{j=1}^{M} \alpha_j * l_{ij}) \tag{2.10}$$

The rest of β's are estimated by:

$$\beta_i = (p''_i - q''_i)(1 - E''_i) + 1 \tag{2.11}$$

where p''_i, q''_i, E''_i are calculated using the hypothesized labels.

2.4.1.3 Label Aggregation

The parameters α and β are used to aggregate the labels. As a first step for this aggregation, we calculate the probability of getting a correct label for instance i from the labeler j defined as

$$P(L_i = l_{ij}|\alpha_j, \beta_i) = \sigma(c\alpha_j\beta_i), \tag{2.12}$$

where T_i is the true but unknown label for the instance i. In this function, c is a scaling factor with value 3. The reason for multiplying with this scaling factor is to span the range of the function to [0,1], otherwise the values only map to a subinterval of [0,1]. The value 3 is chosen due to the fact that $\alpha_j\beta_i \in [-2, 2]$ and $c\alpha_j\beta_i \in [-6, 6]$, the latter choice maps to all values in the interval [0,1].

Since in this version of ELICE, we are able to identify random and oppositional labelers separately, we can make use of this information. We have incorporated this aspect of knowledge in the aggregation formula.

$$FL_i = sign(\sum_{j=1}^{M} \sigma(|c\alpha_j\beta_i|) * L_{ij} * sign(\alpha_j\beta_i)) \tag{2.13}$$

This formula flips the label when the product $\alpha\beta$ is negative, which means α is negative (as β is always positive) and the labeler is on the oppositional side. If the product $|\alpha\beta|$ has large value, logistic function will weight the label higher and for small value of $|\alpha\beta|$ the weight is small. So for a given instance, when the labeler is random the weight assigned to the label will be low, when the labeler is good or oppositional the weight is high. But for the oppositional labeler, label is automatically flipped because of being multiplied to $sign(\alpha\beta)$. This case is specially helpful when many labelers are oppositional.

2.4.2 ELICE 2 with Clustering

ELICE 2 also has a cluster-based variant. We cluster the data and choose equal number of instances from each cluster, to get expert-labels. The rest of the method remains the same.

2.4.3 Test case

Figure 2.2 shows the performance of majority voting, GLAD [Whitehill *et al.*, 2009], Dawid and Skene's method [Dawid and Skene, 1979], Belief Propagation [Liu *et al.*, 2012], Karger's iterative

method [Karger *et al.*, 2014], ELICE 1, ELICE 1 with clustering, ELICE 2 and ELICE 2 with clustering on the University of California Irvine (UCI) Machine Learning Repository Chess dataset. We start with all good labelers and keep on increasing the percentage of random and oppositional labelers. Number of expert labels used for ELICE and all its versions is 20

We can see that ELICE 2 performs not only better than all state-of-the-art methods but also better than ELICE 1, especially when all or most labelers are oppositional. The reason is that ELICE 2 is able to utilize the information provided by the oppositional labelers, which is wasted in most cases.

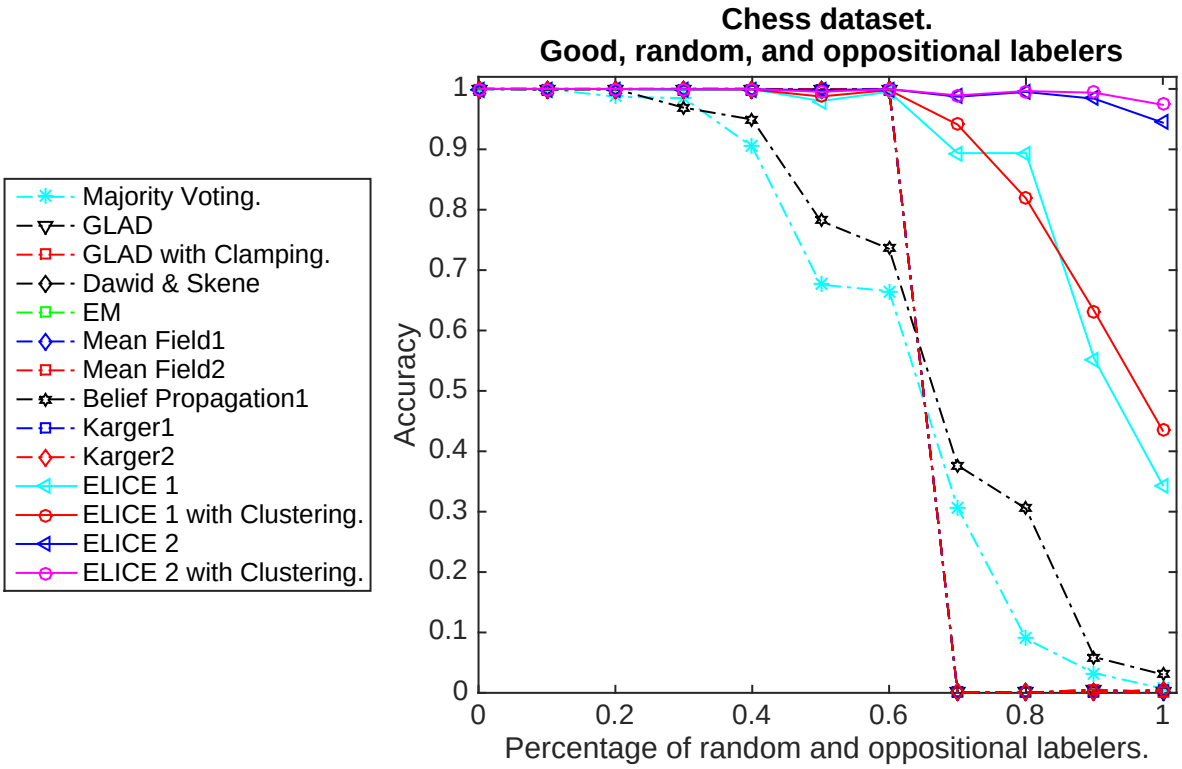

Figure 2.2: Accuracy of state-of-the-art methods along with ELICE 1 and 2 on UCI chess dataset.

2.4.4 Summary

This version of ELICE provides a better accuracy as compared to the previous version as can be seen in Figure 2.2. The main reason is that the entropy helps in identifying the good, random and oppositional labelers. A better aggregation of labels leads to incorporating the information from the

oppositional labelers, which improves the labeling accuracy.

2.5 ELICE 3 Framework

In the previous versions of ELICE, we have assumed the availability of domain experts who provide correct labels without making mistakes. Therefore, expert-labeled instances serve as ground truth. But sometimes, ground truth is not known for certain due to one or more of the following problems in the crowd labeling scenario:

- Expert-labels can be wrong due to the complexity of the task.
- Experts do not agree on one label and have diverse opinions.
- A ground truth cannot be obtained using methods other than expert-evaluation or has a high acquisition cost (e.g., a biopsy in the case of a brain tumor.)

In this situation, we propose to add more information other than expert-labeled instances by involving labeler to labeler and instance to instance comparisons. Since the expert labels are available for a subset of instances and have a chance of being wrong, incorporating crowd labels, which are available for the whole dataset can help. This can increase the chance of refining the estimates of the labeler ability and instance difficulty.

In this version of ELICE, the initial inputs of α and β are taken from ELICE 2 with the only difference that the expert labels are not necessarily ground truth. Based on this information the pairwise comparison is performed. While this version of ELICE is computationally more expensive than ELICE 1 and 2, it can be helpful when ground truth is not known with certainty. To reduce the computational complexity, we also propose ELICE with circular comparison.

2.5.1 ELICE 3 with Pairwise Comparison

In this variant of ELICE, we use a generalization of the model in [Bradley and Terry, 1952; Huang *et al.*, 2006]. In this generalized model, pairwise comparison is used to rank teams of players of a game based on their abilities. The approach uses the previous performance of the players as an input to the model. We use a similar idea to find the expertise of the labelers and difficulty of the instances.

We obtain the average score of the labelers and instances that is calculated using the α's and β's, which we get through the expert evaluation. In our approach we compare labeler to labeler and instance to instance. There are $M^{'} = \binom{M}{2}$ pairwise comparisons for M labelers and $N^{'} = \binom{N}{2}$ pairwise for N instances. The level of ability of a labeler j based on a pairwise comparison is denoted by α'_j. Similarly, the difficulty of instance i based on the pairwise comparison is denoted by β'_i.

The procedure for finding α'_js is described as follows. We assume that the actual performance of labeler j, which is represented by a random variable X_j, has some unknown distribution. In order to avoid computational difficulties we assume that the X_j has a doubly exponential extreme value distribution with a mode equal to α'_j.

$$P(X_j \leq x) = exp(exp{-}(x - \alpha'_j)) \tag{2.14}$$

This distribution ensures that the extreme values are taken into consideration, and variance is directly affected by the values but is not dependent on the mean of the distribution. Hence according to [Huang *et al.*, 2006]:

$$P(C_j \text{ is more expert than } C_k) = \frac{exp(\alpha'_j)}{(exp(\alpha'_j) + exp(\alpha'_k))} \tag{2.15}$$

where C_j is the crowd labeler j and C_k is the crowd labeler k. We use β's and β's to calculate the average score of reliability of the labelers denoted by P_j.

$$\mathrm{P}_j = \frac{1}{M} \sum_{i=1}^{N} \sigma(c\alpha_j\beta_i) \tag{2.16}$$

The average score is calculated to make sure that, while doing a pairwise comparison of labelers, their average performance on the whole dataset is taken into consideration. We assume that the probability of one labeler being better than another labeler is estimated by the ratio of the average score of the labelers [Huang *et al.*, 2006]. This can be expressed in the form of an equation by using equation 2.15 and the ratio of P_j and $\mathrm{P}_j + \mathrm{P}_k$.

$$\begin{aligned} \frac{exp(\alpha'_j)}{(exp(\alpha'_j) + exp(\alpha'_k))} &\approx \frac{\mathrm{P}_j}{\mathrm{P}_j + \mathrm{P}_k} \\ \implies \frac{1}{(1 + exp(-(\alpha'_j - \alpha'_k)))} &\approx \frac{1}{1 + \frac{\mathrm{P}_k}{\mathrm{P}_j}} \end{aligned} \tag{2.17}$$

$$\implies (\alpha'_j - \alpha'_k) \approx log(\frac{\mathrm{P}_j}{\mathrm{P}_k}) \tag{2.18}$$

This can be formulated as the least square model:

$$\implies \min_{\boldsymbol{\alpha'}} \sum_{j=1,k=j+1}^{M} [(\alpha'_j - \alpha'_k) - log(\frac{\mathrm{P}_j}{\mathrm{P}_k}))]^2 \tag{2.19}$$

Which can be written in the matrix form as

$$\min_{\boldsymbol{\alpha'}} \quad (\mathbf{G}\boldsymbol{\alpha'} - \mathbf{d})^T(\mathbf{G}\boldsymbol{\alpha'} - \mathbf{d}) \tag{2.20}$$

$\boldsymbol{G}$ is a matrix of order M' x M. The rows represent comparisons and columns represent the labelers. The matrix is defined as

$$G_{lj} = \begin{cases} 1 & j \text{ is the first labeler in the } l^{th} \text{ comparison} \\ -1 & j \text{ is the second labeler in the } l^{th} \text{ comparison} \\ 0 & \text{labeler } j \text{ is not in the } l^{th} \text{ comparison} \end{cases} \tag{2.21}$$

where $\quad j = 1, 2, \ldots, M; \quad l = 1, 2, \ldots, M'$

Also,

$$d_{(j,k)} = log(\frac{\mathrm{P}_j}{\mathrm{P}_k}) \tag{2.22}$$

where $\quad j = 1, 2, \ldots, M; \;\; k = j+1, j+2, j+3, \ldots, M.$

We can derive the following expression:

$$\boldsymbol{\alpha'} = (\mathbf{G^T G})^{-1}\mathbf{G^T d} \tag{2.23}$$

In order to avoid the difficulties when the matrix G^TG is not invertible, we add a regularized term $\mu\boldsymbol{\alpha'}^T\boldsymbol{\alpha'}$ where μ is a very small real number which can be learned heuristically.

$$\min_{\boldsymbol{\alpha'}} \quad (\mathbf{G}\boldsymbol{\alpha'} - \mathbf{d})^T(\mathbf{G}\boldsymbol{\alpha'} - \mathbf{d}) + \mu\boldsymbol{\alpha'}^T\boldsymbol{\alpha'} \tag{2.24}$$

The resulting expression for $\boldsymbol{\alpha'}$ we get is,

$$\boldsymbol{\alpha'} = (\mathbf{G^T G} + \mu\mathbf{I})^{-1}\mathbf{G^T d} \tag{2.25}$$

where $\boldsymbol{I}$ is the identity matrix.

This procedure can be repeated to find an expression for $\boldsymbol{\beta'}$s . First we find the average score of the difficulty of each instance:

$$\mathrm{Q}_i = \frac{1}{N}\sum_{j=1}^{M}\sigma(c\alpha_j\beta_i) \tag{2.26}$$

Then repeating the above mentioned steps and adding 1s to make β's positive, we get

$$\boldsymbol{\beta'} = (\mathbf{H^T H} + \nu\mathbf{I})^{-1}\mathbf{H^T d'} + \mathbf{1} \tag{2.27}$$

where $d'_{(i,p)} = log(\frac{\mathrm{Q}_i}{\mathrm{Q}_p})$ and $i = 1, 2, \ldots, N; \quad p = i+1, i+2, \ldots, N.$

Also,

$$H_{ri} = \begin{cases} 1 & i \text{ is the first instance in the } r^{th} \text{ comparison} \\ -1 & i \text{ is the second instance in the } r^{th} \text{ comparison} \\ 0 & \text{instance } i \text{ is not in the } r^{th} \text{comparison} \end{cases} \tag{2.28}$$

such that $i = 1, 2, \ldots, N; r = 1, 2, 3...N^{'}$.

After finding the ${\alpha'}_j$s and ${\beta'}_i$s we use them to infer the labels.

$$A_i = sign(\sum_{j=1}^{M}\sigma(|c\alpha_j\beta_i|) * L_{ij} * sign(\alpha_j\beta_i)) \tag{2.29}$$

As in the previous version of ELICE, we multiply $\alpha'_j\beta'_i$ by a scaling factor c to make sure that the range of the values is mapped to the whole range of the logistic function i.e., [0,1] and not just on its subinterval. This also serves to make the difference between the expertise of workers on different instances more pronounced. Since in this case the value of the product $|\alpha_j\beta_i| << 1$, the value of c has to be large. We used $c = 100$, chosen heuristically through experiments.

2.5.2 ELICE 3 with Circular Comparison

ELICE with circular comparison is a variant of ELICE with pairwise comparison. Instead of making comparison of every two labelers, it compares labelers to labelers and instances to instances in a circular fashion, for example, 1 to 2, 2 to 3, ..., i to $i+1$, ..., $M-1$ to M, M to 1. Our empirical results show that this produces results as good as ELICE with pairwise comparison but substantially reduces the computational cost.

2.5.3 Test case

Figure 2.3 presents the performance of majority voting, GLAD [Whitehill *et al.*, 2009], Dawid and Skene's method [Dawid and Skene, 1979], Belief Propagation [Liu *et al.*, 2012], Karger's iterative method [Karger *et al.*, 2014], ELICE 1, ELICE 1 with clustering, ELICE 2, ELICE 2 with clustering, ELICE 3 and ELICE 3 with clustering on the University of California Irvine (UCI) Machine Learning Repository Chess dataset. We start with all good labelers and keep on increasing the percentage of random and oppositional labelers. Number of expert labels used for ELICE and all its versions is 20.

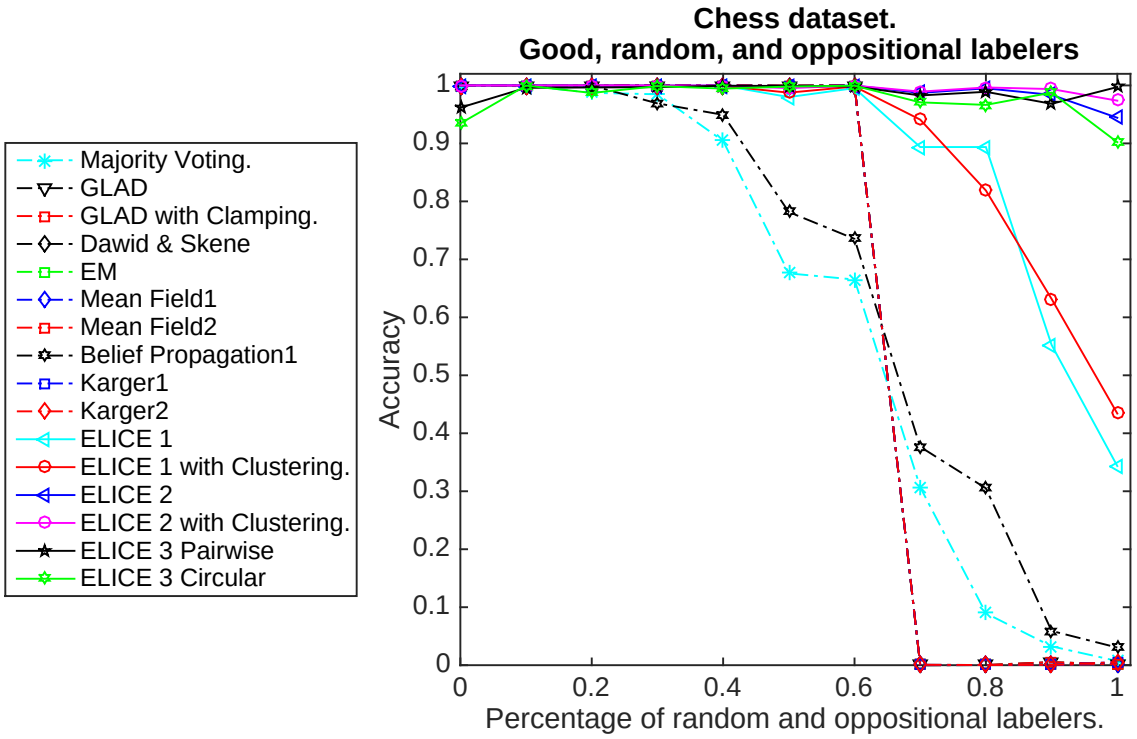

Figure 2.3: Accuracy of state-of-the-art methods along with ELICE 1, 2 and 3 on UCI chess dataset.

2.5.4 Summary

This version of ELICE is based on the idea of incorporating more information by comparison of labeler to labeler and instance to instance when ground truth is not known with certitude. This version has a higher computational cost than our previous approaches, especially in the case of

large dataset but can produce good results by using most of the available information. The test case results can be seen in Figure 2.3, more detailed experiments are presented in the next section.

Chapter 3

Empirical Evaluation

We implemented ELICE and its variants in Matlab. We compare our method to Majority voting (different variants), GLAD and GLAD with clamping [Whitehill *et al.*, 2009], Dawid and Skene method [Dawid and Skene, 1979], EM (Expectation Maximization), Karger's iterative method [Karger *et al.*, 2014], Mean Field algorithm and Belief Propagation [Liu *et al.*, 2012]. Please note that Karger's iterative method, Mean Field method and Belief Propagation have two versions each due to different parameter setting.

We also compared our results to a variant of majority voting that is majority voting with gold (expert labels) testing. In this variant, votes are aggregated after discarding the labels provided by the labelers who are below the specified performance threshold. The gold testing is done on randomly picked instances from the dataset, which are labeled by the expert. The number of expert-labeled instances used were same as the number used for ELICE and its versions.

All of these methods were also implemented in MATLAB and in most cases, the code was obtained from authors of the methods. We conducted the experiments using simulated and real crowd labels on the different datasets as follows:

- Five datasets from the University of California Irvine (UCI) Machine Learning Repository repository [Asuncion and Newman, 2007]: IRIS, Breast Cancer, Tic-Tac-Toe, Chess, Mushroom (Section 3.1). Crowd labels are simulated for different percentage of random and oppositional crowd-labelers in the pool of labelers.

- Two real applications Tumor Identification dataset and Race Recognition dataset (Section 3.3

and 3.2) for which we use Amazon Mechanical Turk to acquire labels from the crowd.

3.1 UCI Machine Learning Repository Datasets

We selected above-mentioned five UCI datasets. Classification tasks for these datasets are as follows:

- **IRIS:** Flower type, restricted to 2 classes only.
- **Breast cancer:** Malignant/Non-malignant tumor.
- **Tic-Tac-Toe:** x-can-win/x-cannot-win.
- **Chess:** White-can-win/White-cannot-win.
- **Mushroom:** Edible/Non-edible.

3.1.1 Experimental Design

In these experiments, we simulate crowd labels for each instance. The labels are generated so that a good crowd labeler makes less than 35% mistakes, a random crowd labeler makes 35% to 65% mistakes and an oppositional crowd labeler makes 65% to 100% mistakes. These are created by inverting $x\%$ of the original labels in the dataset, where x is a random number between 0 and 35 for good labeler, 35 to 65 for random labeler and 65 to 100 for oppositional labeler. We simulate the labels using MATLAB pseudo random number generator to ensure randomness. We randomly select n number of instances to play the role of the expert-labeled instances. In the cluster-based methods, built-in MATLAB k-means function is used for clustering the instances.

Simulated data is used to cover all possible labeler types, categorized as good, random and oppositional. Simulated labels helps us understand the performance of different methods with different combinations of labeler types. We observed similar patterns in the real data that we published on Amazon Mechanical Turk as well as in the other labeled data available at the website http://ir.ischool.utexas.edu/square/data.html. While a large percentage of real labelers was from the good or random category, there are cases where labelers want to attain a specific purpose and behave maliciously (discussed in more detail in Section 3.5). Therefore, we assume that our simulated data represents all real-life cases and gives us a broader picture.

3.1.2 Results

Table 3.1 shows a comparison of the accuracy of different methods along with ELICE across the five datasets. We use different percentages of random and oppositional labelers while the rest of the labelers are good. Table 3.1 has three sections, showing results for 30% or less, $30\% - 70\%$ and 70% or more, random or oppositional labelers in the crowd. In the first section, it is evident that when most of the labelers are good, all the methods even majority voting, perform really well. For the case when there are $30\% - 70\%$ random and oppositional labelers, the performance of all the methods drops except for ELICE and majority voting with gold testing.

However, it should be noted that majority voting with gold testing is not always reliable. The reason is that majority voting with gold testing, in many cases, results into "NaN" due to all labelers being discarded based on gold testing. The reported results in this table were averaged only over the cases when the experiments produced a number. Although majority voting with gold testing outperforms other methods in some cases but its performance remains unpredictable due to resulting in "NaN" many times. On the other hand, ELICE is able to produce highly accurate results without any factor of uncertainty. In the third section of Table 3.1 where there are more than 70% random and oppositional labelers, the performance of almost all the methods except ELICE drops to zero. All versions of ELICE substantially outperform all other methods, especially ELICE 2 which has the mechanism to flip the labels of oppositional labelers. Even majority voting with gold testing is unable to beat ELICE.

In Figures 3.1, 3.2, and 3.3, we show the accuracy of the methodologies for the IRIS and UCI breast cancer dataset for (a) good & oppositional, (b) random & oppositional and (c) good & random labelers respectively. All these graphs show the superiority of ELICE on other state-of-the-art methods. In Figure 3.1, we can see that ELICE 1 has a good performance even in the presence of all oppositional labelers and phase transition is delayed. ELICE 2 and ELICE 3 are able to perform exceptionally well due to the ability to flip the labels of the oppositional labelers. In Figure 3.2 we see that the performance of all the methods is around 50% when all the labelers are random but as the number of oppositional labelers is increased the performance of ELICE 2 and 3 improves, the accuracy of ELICE 1 drops slowly, and the accuracy of rest of the methods immediately drops to zero. In Figure 3.3, we see that all the methods have the similar performance when there is a combination of good and random labelers.

	Dataset ($\mathcal{D}$)	Mushroom	Chess	Tic-Tac-Toe	Breast Cancer	IRIS
Random/Oppositional	Total instances (N)	8124	3196	958	569	100
Labelers	+ve/-ve instances	3916/ 4208	1669/1527	626/332	357/212	50/50
	Expert labels(n)	20	8	8	8	4
Less than 30%	Maj. Voting	0.9918	0.9822	0.9718	0.9842	0.9925
	Maj. Voting (25%[1])	0.9995[2]	NaN[3]	0.9997[2]	0.9969[2]	0.9900[2]
	Maj. Voting (35%[1])	0.9888[2]	0.9888[2]	0.9995[2]	0.9956[2]	0.9950[2]
	Maj. Voting (45%[1])	0.9994[2]	1.0000[2]	0.9995[2]	0.9956[2]	1.0000[2]
	Maj. Voting (55%[1])	NaN[3]	1.0000[2]	0.9982[2]	0.9974[2]	NaN[3]
	GLAD	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	GLAD with clamping	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Dawid Skene	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	EM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Belief Propagation 1	——[4]	0.9918	**1.0000**	**1.0000**	**1.0000**
	Belief Propagation 2	——[4]	——[4]	**1.0000**	**1.0000**	**1.0000**
	Mean Field 1	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Mean Field 2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Karger 1	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Karger 2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	ELICE 1	0.9988	0.9994	0.9989	0.9993	**1.0000**
	ELICE 1 with clustering	0.9993	0.9994	0.9989	0.9991	**1.0000**
	ELICE 2	0.9997	0.9999	**1.0000**	0.9989	**1.0000**
	ELICE 2 with clustering	0.9998	**1.0000**	**1.0000**	0.9991	**1.0000**
	ELICE 3 Pairwise	——[5]	0.9768	0.9925	0.9701	0.9959
	ELICE 3 Circular	0.9567	0.9800	0.9842	0.9635	0.9891

Table continued on next page

[1]Using the labels provided by the labelers with performance above the given threshold. Performance was checked based on expert labeled instances.

[2]In many cases, majority voting with gold testing resulted into NaN (Not a number) due to all labelers being discarded in the testing phase. The reported results were averaged, only over the cases when the experiments produced a number, ignoring the case when the results were NaN.

[3] No result was produced as all labelers were discarded when tested, in all the runs of the experiment.

[4]Code for Belief propagation did not converge.

[5]Code for ELICE pairwise was parallelized for datasets with more than 3000 instances. For Mushroom dataset due to high time and space complexity as well as hardware availability constraints, it was not feasible to calculate the results.

Table 3.1 – continued from previous page

	Dataset ($\mathcal{D}$)	Mushroom	Chess	Tic-Tac-Toe	Breast Cancer	IRIS
Random/Oppositional	Total instances (N)	8124	3196	958	569	100
Labelers	+ve/-ve instances	3916/ 4208	1669/1527	626/332	357/212	50/50
	Expert labels(n)	20	8	8	8	4
30% to 70%	Maj. Voting	0.5509	0.6541	0.7116	0.5874	0.6825
	Maj. Voting (25%[1])	0.9712[2]	0.7858[2]	0.9582[2]	0.9284[2]	0.9467[2]
	Maj. Voting (35%[1])	0.9805[2]	0.9626[2]	0.9776[2]	0.9328[2]	0.9733[2]
	Maj. Voting (45%[1])	0.9723[2]	0.9219[2]	0.9945 [2]	0.9818[2]	0.9800[2]
	Maj. Voting (55%[1])	0.9480[2]	NaN[3]	0.9903[2]	0.9605[2]	0.9700[2]
	GLAD	0.7494	0.7502	0.7503	0.7504	0.7473
	GLAD with clamping	0.7494	0.7501	0.7505	0.7504	0.7473
	Dawid Skene	0.5001	0.7498	0.5003	0.7504	0.7475
	EM	0.5001	0.7498	0.5003	0.7504	0.7475
	Belief Propagation 1	——[4]	0.7107	0.5003	0.5004	0.7500
	Belief Propagation 2	——[4]	——[4]	0.5003	0.7504	0.7525
	Mean Field 1	0.5002	0.7498	0.5003	0.7504	0.7500
	Mean Field 2	0.5001	0.7498	0.5003	0.7504	0.7525
	Karger 1	0.5002	0.7498	0.5005	0.6254	0.7525
	Karger 2	0.5003	0.7498	0.5005	0.7504	0.7525
	ELICE 1	0.9779	**0.9981**	0.9915	0.9701	0.9837
	ELICE 1 with clustering	0.9731	0.9677	0.9839	0.9650	0.9715
	ELICE 2	0.9975	0.9964	**0.9991**	0.9973	0.9932
	ELICE 2 with clustering	**0.9985**	0.9973	0.9987	**0.9987**	**0.9960**
	ELICE 3 Pairwise	——[5]	0.9948	**0.9991**	0.9951	0.9905
	ELICE 3 Circular	0.9978	0.9907	**0.9991**	0.9949	0.9878

Table continued on next page

Table 3.1 – continued from previous page

	Dataset ($\mathcal{D}$)	Mushroom	Chess	Tic-Tac-Toe	Breast Cancer	IRIS
Random/Oppositional	Total instances (N)	8124	3196	958	569	100
Labelers	+ve/-ve instances	3916/ 4208	1669/1527	626/332	357/212	50/50
	Expert labels(n)	20	8	8	8	4
More than 70%	Maj. Voting	0.0842	0.0824	0.0832	0.0773	0.0900
	Maj. Voting (25%[1])	0.6505[2]	0.5810[2]	0.6002[2]	0.5841[2]	NaN[3]
	Maj. Voting (35%[1])	0.8541[2]	0.7113[2]	0.7112 [2]	0.6257[2]	0.6450[2]
	Maj. Voting (45%[1])	0.9351[2]	NaN[3]	0.7797[2]	0.8049[2]	0.7200[2]
	Maj. Voting (55% [1])	NaN[3]	0.9581[2]	0.8768[2]	0.6757[2]	0.5700[2]
	GLAD	4.1071e-04	0.0031	0.0011	0.0024	0.0145
	GLAD with clamping	4.1071e-04	0.0031	0.0014	0.0018	0.0145
	Dawid Skene	1.6412e-04	9.3867e-04	0.0045	0.0023	0.0133
	EM	1.6412e-04	8.3438e-04	0.0049	0.0023	0.0133
	Belief Propagation 1	——[4]	0.1315	0.0049	0.0023	0.0133
	Belief Propagation 2	——[4]	——[4]	0.0045	0.0023	0.0133
	Mean Field 1	1.6412e-04	8.3438e-04	0.0049	0.0023	0.0133
	Mean Field 2	1.6412e-04	9.3867e-04	0.0045	0.0023	0.0133
	Karger 1	3.6928e-04	0.0021	0.0042	0.0035	0.0100
	Karger 2	3.6928e-04	0.0021	0.0042	0.0035	0.0100
	ELICE 1	0.7451	0.6332	0.7441	0.6869	0.7065
	ELICE 1 with clustering	0.7228	0.6003	0.7346	0.7020	0.6993
	ELICE 2	0.9900	0.9847	0.9934	0.9872	0.9783
	ELICE 2 with clustering	**0.9942**	**0.9869**	**0.9956**	**0.9881**	**0.9801**
	ELICE 3 Pairwise	——[5]	0.9848	0.9605	0.9629	0.9656
	ELICE 3 Circular	0.9680	0.9521	0.9590	0.9635	0.9601

Table 3.1: Accuracy of state-of-the-art methods and ELICE (all versions and variants) for different datasets averaged over 50 runs. Good labelers: 0-35% mistakes, Random labelers: 35-65% mistakes, Oppositional labelers: 65-100% mistakes.

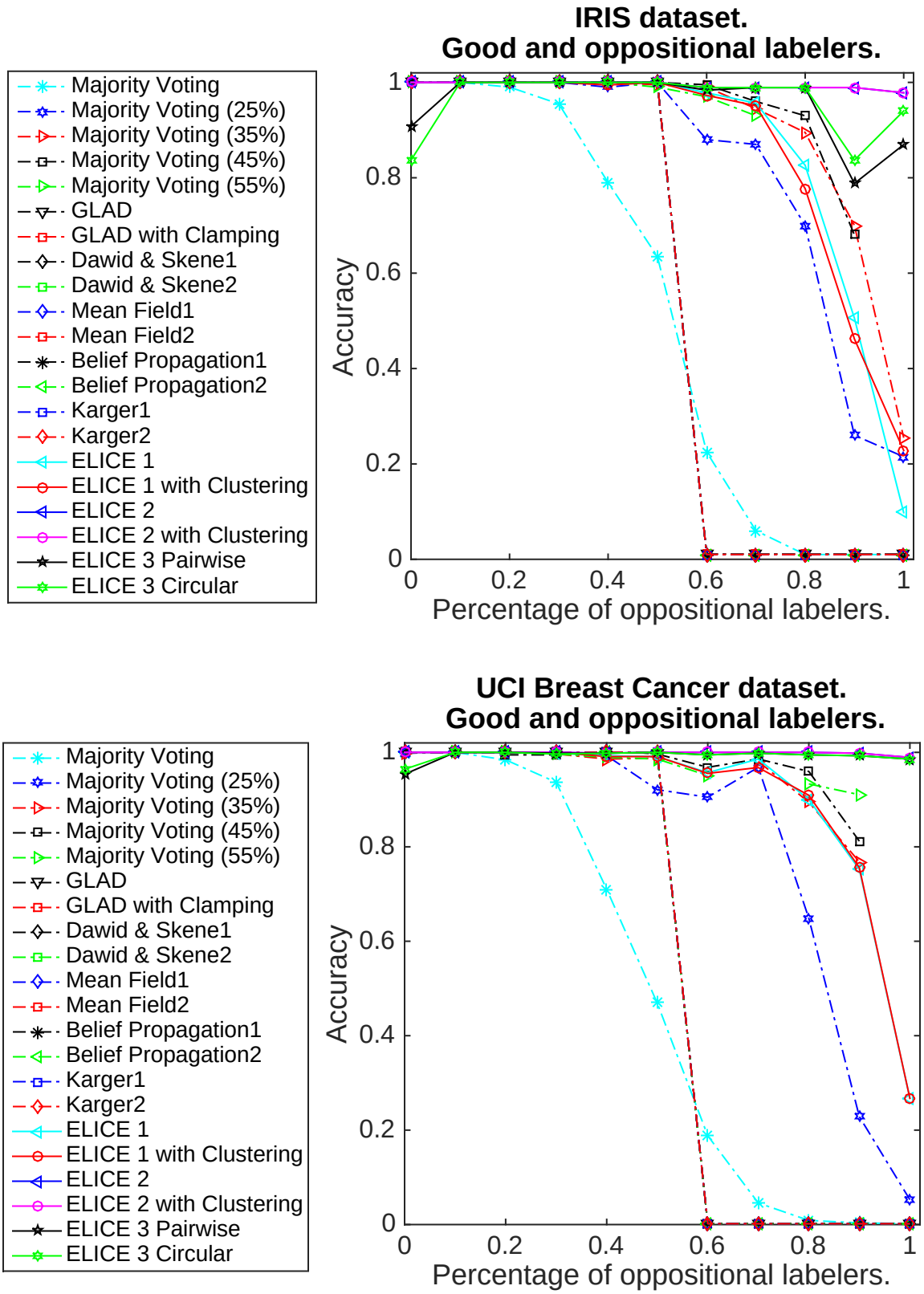

Figure 3.1: (Top) IRIS dataset. (Bottom) UCI Breast Cancer dataset. Simulated labels represent good and oppositional labelers.

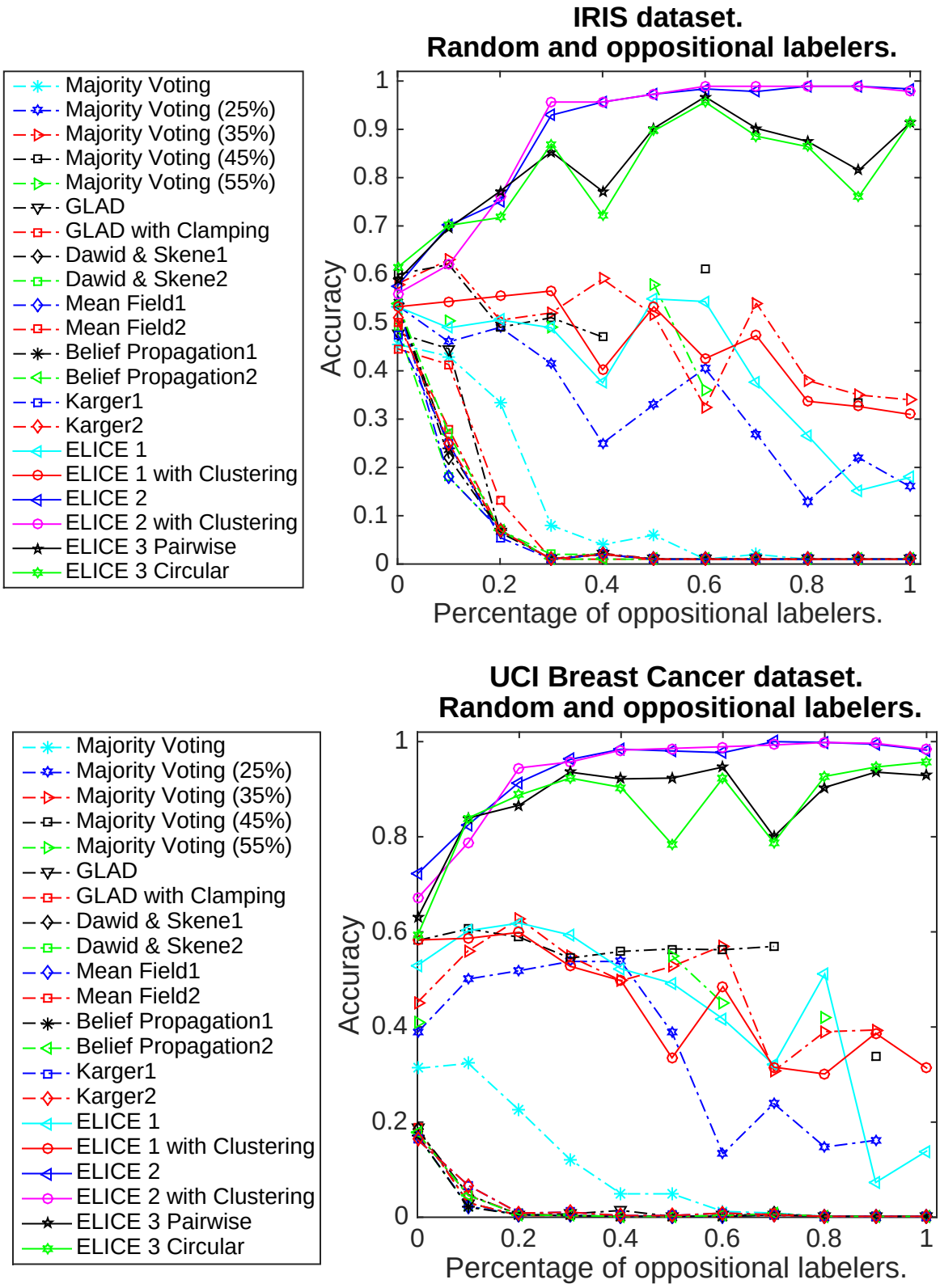

Figure 3.2: (Top) IRIS dataset. (Bottom) UCI Breast Cancer dataset. Simulated labels represent random and oppositional labelers.

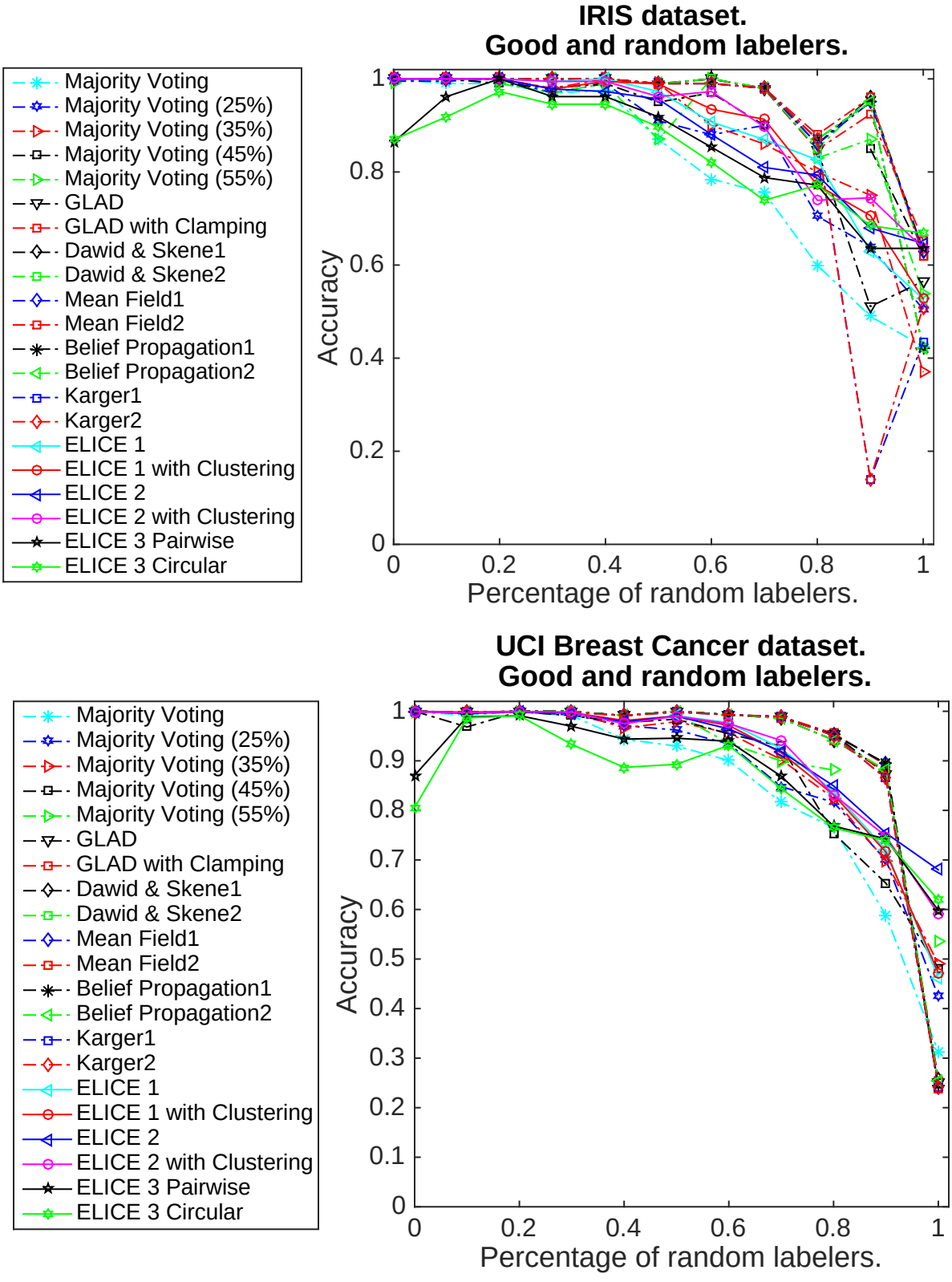

Figure 3.3: (Top) IRIS dataset. (Bottom) UCI Breast Cancer dataset. Simulated labels represent good and random labelers.

Efficiency: The experiments also reveal that ELICE is efficient as compared to the other methods. Figure 3.4 shows the runtime for Mushroom for all the methods as we increase the number of instances. It should be noted that for big datasets such as Chess (3196 instances), we used MATLAB's Parallel Computing Toolbox to run ELICE pairwise.

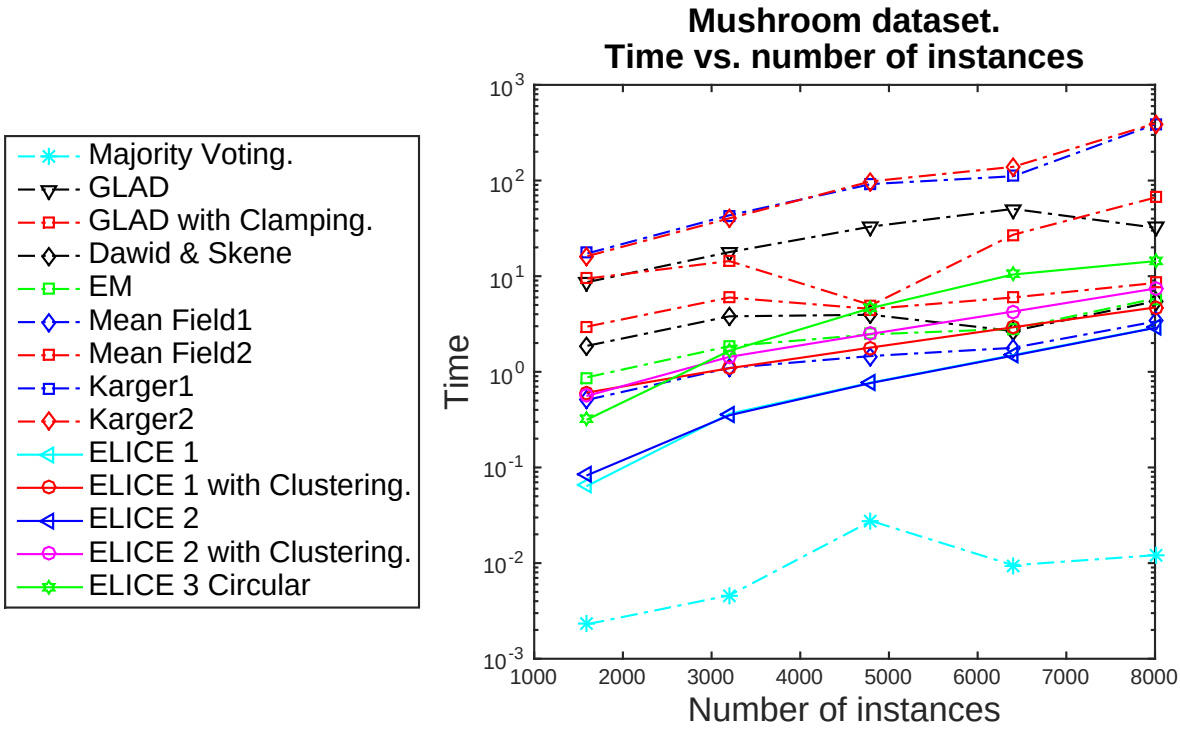

Figure 3.4: Time vs. Number of instances. Number of expert labels used for ELICE (all versions and variants) is 20.

Note: Code for Belief Propagation did not converge even after a long time. Code for ELICE pairwise was parallelized for datasets with more than 3000 instances therefore, we do not report its time as it is not comparable to the non-parallelized code.

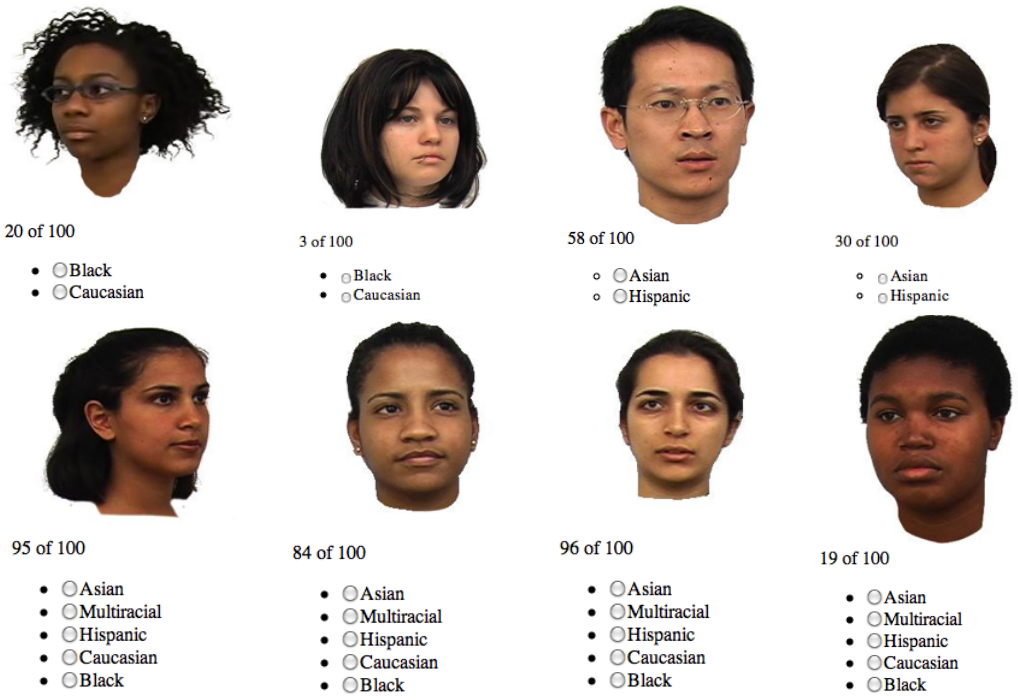

Figure 3.5: Example images from the Race recognition task posted on Amazon Mechanical Turk (Left to right): (Top) Black, Caucasian, Asian, Hispanic. (Bottom) Multiracial, Hispanic, Asian, Multiracial.

3.2 Race Recognition Dataset

Another real-life dataset we considered is *race recognition* dataset[6] containing images of people from different races. We found this dataset to be interesting due to variability in the difficulty of the task.

3.2.1 Experimental Design

We took three samples of 100 instances each and posted them as a race recognition task on Amazon Mechanical Turk. The samples were chosen to guarantee different levels of difficulty. The tasks were to identify: (1) Black versus Caucasian with 50 instances of each class, (2) Hispanic versus Asian with 50 instances of each class, (3) Multiracial versus other races with 40 instances of Multiracial and 60 instances of the other races i.e. Asian, Black, Caucasian and Hispanic. Some

[6]Available on Stimulus Images; Courtesy of Michael J. Tarr, Center for the Neural Basis of Cognition, Carnegie Mellon University `http://tarrlab.cnbc.cmu.edu/face-place`.

snapshots of the experiment as posted on AMT are shown in Figure 3.5.

For each task, we acquired six crowd labels for all 100 instances. The three tasks were chosen to guarantee easy to moderate difficulty level.

3.2.2 Results

For all variants of ELICE, we used 8 random instances as expert-labeled instances. The results are shown in Table 2. Black versus Caucasian was the easiest of the tasks. Therefore, most of the labelers performed really well with only 0% to 25% of mistakes. As all the labelers had a good performance, the accuracy of all the methods was approximately perfect including the most naive method majority voting.

Identifying Hispanic versus Asian was relatively more difficult. In this case, some labelers made less than 15% mistakes and the rest made over 48% mistakes. In this case ELICE 2 performed best because of its ability to flip the labels.

The most confusing and challenging of all race recognition tasks was identifying multiracial from the other races. While most of the labelers did equally bad, surprisingly it was not as bad as we expected as the percentage of mistakes ranged between 30% and 50%. In this case almost all the labelers were falling in the random labeler category probably due to guessing rather than intelligently thinking the answer. In this case ELICE 1 was the winner but many other methods had approximately close results. The reason is that the random labelers do not provide much information.

3.3 Tumor Identification Dataset

To test our approach on a real-life dataset, we considered a *tumor identification dataset*.[7] Early identification of cancer tumor can help in preventing thousands of deaths but identifying cancer is not an easy task for untrained eyes.

[7] Available on `http://marathon.csee.usf.edu/Mammography/Database.html`

3.3.1 Experimental Design

We posted 100 mammograms on Amazon Mechanical Turk. The task was to identify Malignant versus others (Normal, Benign, Benign without call back.) The following instruction for appropriate identification was provided to the labelers: "A breast tumor is a dense mass and will appear whiter than any tissue around it. Benign masses usually are round or oval in shape, but a tumor may be partially round, with a spiked or irregular outline as part of its circumference."

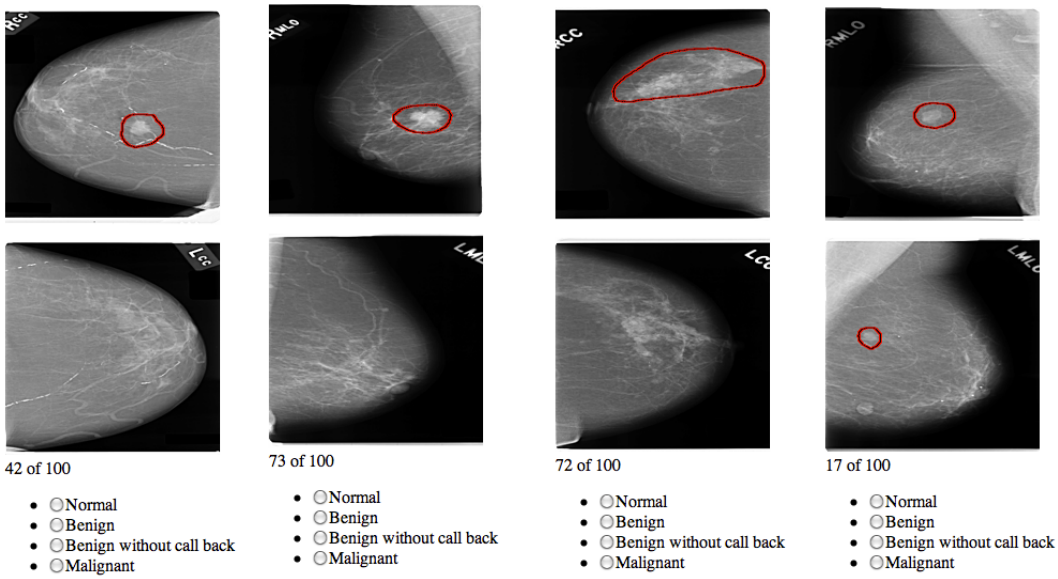

Figure 3.6: Example images of the Tumor Identification dataset. From left to right: First three are Malignant and fourth is benign.

3.3.2 Results

The task of tumor identification clearly requires expertise and is very difficult for an untrained person. On the other hand, an expert person can do really well. In this AMT experiment, we had two labelers with 6% and 32% mistakes and four labelers with more than 55% mistakes. The results are shown in Table 3.2 and demonstrate the superiority of ELICE as compared to the other methods except majority voting with gold testing, which performed best excluding the cases when the result was "NaN". The reason was that it was choosing only one labeler with 6% mistakes and relying on its labels but it was unpredictable.

Approach	Race Recognition			Tumor Identification
	Black/Caucasian	Hispanic/Asian	Multiracial/other	Malignant/Non-malignant
MajorityVoting	0.9900	0.5200	0.6500	0.5500
Majority Voting (25%[1])	0.9900[2]	0.5718[2]	0.6500[2]	0.7418[2]
Majority Voting (35%[1])	0.9900[2]	0.5600[2]	0.6400[2]	0.8660[2]
Majority Voting (45%[1])	0.9900[2]	0.5175[2]	0.5500[2]	0.8967[2]
Majority Voting (55%[1])	0.9900[2]	0.5150[2]	NaN[3]	0.8300[2]
GLAD	**1.0000**	0.5000	0.6630	0.3043
GLAD with Clamping	**1.0000**	0.5000	0.6630	0.3152
Dawid Skene	0.9900	0.4500	0.6100	0.7000
EM	0.9900	0.5000	0.6500	0.3700
Belief Propagation 1	0.9900	0.5000	0.6500	0.3600
Belief Propagation 2	0.9900	0.4500	0.5900	0.0600
Mean Field 1	0.9900	0.5000	0.6500	0.3600
Mean Field 2	0.9900	0.4500	0.6000	0.7000
Karger 1	0.9900	0.5000	0.6500	0.3600
Karger 2	0.9900	0.5000	0.6500	0.3600
ELICE 1	0.9906	0.6793	**0.6650**	0.7100
ELICE 1 with clustering*	-	-	-	-
ELICE 2	0.9896	**0.7648**	0.5746	0.7698
ELICE 2 with clustering*	-	-	-	-
ELICE Pairwise	0.9896	0.6729	0.5756	0.7648
ELICE Circular	0.9896	0.6887	0.5657	**0.7722**

Table 3.2: Accuracy of different methods on Amazon Mechanical Turk datasets. The given results are the average of 100 runs on 100 instances with 6 labels per instance. Randomly chosen 8 instances are used as expert labeled instances (the instances with ground truth.)

* Since the features for these datasets are not available therefore the results of ELICE with clustering could not be calculated.

3.4 Discussion

In the previous section, we showed through a wide range of experiments the superiority of ELICE. In this section, we will compare different versions of ELICE and discuss their appropriateness based on different situations. We will also discuss various aspects of our approach.

3.4.1 Comparison of All Versions and Variants of ELICE

- ◇ ELICE 1 is simple and easy to implement. As shown in the experimental section, it is not only efficient, it also provides better results than the slower state-of-the-art methods (Figure 3.4). The key factor of ELICE 1 good performance is relying on the judgment of the good labelers while minimizing the effect of the random or oppositional labelers. This can especially be helpful when at least one good labeler is available. When most of the labelers are average, it may not provide very high accuracy but can still perform as good as the other prevailing methods. The low computational cost and effectiveness of the approach as compared to state-of-the-art methods are the main advantages of this version.
  Best use: This method can be used when the labeling task is not very challenging and there is a high chance to get at least one good labeler.

- ◇ As compared to ELICE 1, the second version of ELICE provides even better accuracy because other than benefiting from good labelers, it takes also advantage of the oppositional labelers. This is done through a better aggregation of labels that leads to incorporating the information from the oppositional labelers.
  Best use: This version is helpful when there is a high chance of the task being misunderstood or difficult resulting into unintentional oppositional behavior. It can also take advantage of intentionally oppositional labeler getting as much information as possible. While it is likely that not many labelers are intentionally oppositional, whenever there is one, the information provided is not wasted.

- ◇ The third version of ELICE is based on the idea of incorporating most of the available information by comparison of labeler to labeler and instance to instance when ground truth is not known for certain.

Best use: ELICE 3 pairwise should only be used when expert labels are not gold standard. It has a higher computational cost as compared to the previous versions of ELICE, especially for large datasets. ELICE 3 circular has a relatively lower computational cost due to reduced number of comparisons involved. It can be observed in the experimental results that sometimes the performance of ELICE 3 circular is slightly lower than ELICE 3 pairwise. Therefore, we suggest that when the dataset consists of a few hundred instances, it is preferable to use ELICE 3 pairwise as the computational cost is not very high. But when the dataset consists of thousands of instances, switching to ELICE 3 circular could be a better option but the reduced computational cost comes with a little loss of information. On the other hand, different methods can also be used to reduce the computational cost, such as parallel programming that is used in our experiments presented in the previous section.

◇ The variants of ELICE with clustering can be used only when the features are available. Although it can increase the computational cost, it improves the results. There can also be the possibility of asking the experts to choose from the dataset such that instances from all the classes have equal representation.

Best use: The clustering variant of ELICE can especially be helpful when classes are highly imbalanced. This is because there is a high chance of missing the instances from the smaller class while randomly choosing the instances from the whole dataset. To avoid the non-representation of any class in the expert-labeled instances, clustering can be helpful.

3.4.2 Number of Expert-labels for Large Datasets

In this age of Big data, it is highly desirable to make all the methodologies scalable including crowdsourcing [Mozafari *et al.*, 2014]. ELICE has the advantage of being easily scalable. Once we have a few expert-labeled instances, we can use them no matter how big the labeled dataset is. We have shown empirically that all versions of ELICE always use very few expert-labeled instances that is 0.1% to 10% of the dataset to get high accuracy. It is evident from the results reported in the empirical section, as at most 20 expert-labeled instances are used while the size of the datasets varies from 100 to more than 8000 instances. To further strengthen our claim, we derive a theoretical lower bound on the number of expert-labeled instances needed to achieve highly accurate final labels in Chapter 4 and present more experiments to support it.

3.4.3 Expert-labels and Ground Truth

We assumed that expert-labels are ground truth for ELICE 1 and 2. This can be true for the simple and easy tasks such as language translation where experts of a language can provide the correct translation. In such cases, it is sufficient to get one expert-label per instance. It should, however, be noted that an expert-label may not always be acquired by a human expert. There can be alternate ways to get expert-labels or ground truth. Ground truth can be acquired by different means such as testing (e.g., doing a biopsy of a tumor) or investigating (e.g., direct questioning from subjects in the case of race recognition).

Sometimes, none of the above-mentioned methods give us ground truth. In such cases, we can still use expert-labels but do not consider them to be ground truth for certain. In some cases, expert-labels can be labels provided by the experienced, trained and reliable crowd labelers rather than a domain expert. In this situation, it is better to use all possible information available, as done in ELICE 3.

3.4.4 Cost-effectiveness of ELICE

Given M crowd-labelers, N total number of instances, and n expert-labeled instances, we can formulate the cost equation of the ELICE as follows:

$$\text{Cost of ELICE} = n.Cost_{expert} + M.N.Cost_{crowd} \tag{3.1}$$

where,

$Cost_{expert} =$ cost of one expert-label, $\qquad Cost_{crowd} =$ cost of one crowd-label,

$$Cost_{expert} >> Cost_{crowd}$$

Acquiring expert labels is expensive, however, if used effectively, can be rewarding. ELICE invests on a few expert-labeled instances but on the other hand, it is cost-effective in many other ways, listed as follows:

- a better accuracy with minimum infrastructure,
- no need to block the oppositional labelers and hire more labelers,

- no need to keep track of the history of each labeler,
- lesser time needed to get the results,
- ability to work offline and even for the datasets labeled in the past,
- easily scalable,
- can work with all kinds of labeling platforms,
- handling all kinds of labelers in an integrated manner with minimum wasted information.

3.4.5 Choice of Crowd Labeling Platform

We chose Amazon Mechanical Turk to acquire crowd labels for our experiments. AMT has had a strong impact on crowdsourcing research (see Figure 1.1). More recent crowdsourcing websites have learned from and improved AMT procedure [Vakharia and Lease, 2015] but AMT still remains one of the highly used crowd labeling websites. Many other crowdsourcing platforms use AMT including CrowdFlower[8] and Smartsheet[9]. Despite the fact that we used AMT for crowd labeling, our procedures can handle the labels provided by any other crowdsourcing platforms due to a minimal need for infrastructure, pre-processing, and blocking workers.

It should be noted that AMT has recently introduced more restrictions for requesters of the crowd labeling task. These restrictions include requesters must be living in USA and filing taxes. Although we acquired the crowd labels before these restrictions were applied but the performance of our methodology is unaffected in spite of the changed scenario.

3.4.6 Why not blocking the oppositional labelers?

ELICE framework is a one-shot method and does not block the labelers. Instead of keeping track of the labeler's history, we can simply estimate the ability of the labeler for one labeling task and improve the accuracy, utilizing the information provided by oppositional labelers. Although we do not completely disagree with the effectiveness of blocking the labelers, we believe that this technique may not be always helpful, mainly due to the following reasons:

[8] http://www.crowdflower.com

[9] http://www.smartsheet.com

- ◇ A labeler can always do well on the test and poorly afterwards.
- ◇ A labeler may have more than one account and have different strategies on each of them.
- ◇ It is also possible that one account be used by more than one labeler at different times resulting into different performance levels.

3.4.7 Do we always have many oppositional labelers?

Information provided by oppositional labelers can especially be helpful when such labelers are really knowledgeable and good at providing oppositional labels. Therefore, discarding the labels may result in a loss of information.

We know that oppositional labelers can be of two types, unintentionally oppositional and intentionally oppositional. Unintentionally oppositional labelers maybe fewer than other categories of labelers but do exist due to the following reasons.

- ◇ When the requester has not explained the task well enough and the labeler misunderstands the task resulting in all wrong (or flipped) labels.
- ◇ When the task is well-explained but the labeler is not familiar enough with English language to understand the task, this also results into wrong (or flipped) labels.

On the other hand, it is worth investigating the number of oppositional labelers that are involved in *real* malicious activities and the reason for such egregious behavior. We investigated the case of oppositional labeling where some labelers are polluting the data intentionally due to maliciousness. A lot of crowdsourcing literature has discussed it, a brief overview is presented in the next section.

3.5 Oppositional/Malicious Crowdsourcing

Internet is a collective venture of the people, by the people, for the people but sometimes can work against the people if not well managed. Therefore, it is crucial to prevent and eradicate malicious crowdsourcing activities such as crowdturfing.

"Crowdturfing" is a term coined by a team of researchers at UC Santa Barbara, led by Ben Zhao [Wang *et al.*, 2012]. Crowdturfing is a combination of the words "crowdsourcing" and "as-

troturfing". Wikipedia[10] defines astroturfing as *"Astroturfing is the use of fake grassroots efforts that primarily focus on influencing public opinion and typically are funded by corporations and governmental entities to form opinions."*

Crowdturfing refers to astroturfing campaigns run by crowd workers that is false crowd-support such as false labeling, bogus reviews, comments or followers. More specifically [Lee *et al.*, 2013] define it as *"Malicious crowdsourcing, also called crowdturfing, occurs when an attacker pays a group of Internet users to carry out malicious campaigns."*

With the increase in internet users and use of human intelligence online, crowdturfing or malicious crowdsourcing is getting more attention. There has been a lot of literature on malicious crowdsourcing including [Dalvi *et al.*, 2004; Tran *et al.*, 2009; Rubinstein *et al.*, 2009; Wang *et al.*, 2012; Lee *et al.*, 2013; Wang *et al.*, 2013; Wang *et al.*, 2014; Jagabathula *et al.*, 2014; Lee *et al.*, 2014; Sedhai and Sun, 2015; Liu *et al.*, 2016; Aggarwal, 2016; Liu *et al.*, 2016; Satya *et al.*, 2016] and [Choi *et al.*, 2016].

3.5.1 Common Types of Oppositional/Malicious Crowdsourcing

Common types of oppositional/malicious crowdsourcing include ([Lee *et al.*, 2013]) but are not limited to:

- Political and non-political campaigns on internet.
- Product/services promotions, advertisements and surveys.
- Spam dissemination.
- Fake blogs, social media accounts, and comments.
- Fake social media followers, friends or connections.
- Voluntary wrong labeling.

3.5.2 Oppositional/Malicious Crowdsourcing Structure

Oppositional/malicious crowdsourcing or crowdturfing structure usually consists of customers, agents and crowdworkers [Wang *et al.*, 2012]. Customers initiate the crowdturfing campaign and hire the

[10] http://www.wikipedia.com

agent services to fulfill their purpose. Agent plans and designs the campaign and makes it accessible to a pool of crowdworkers. The crowdworkers complete the malicious task and the agent submits it to the customer and receives his payment and pays the crowdworkers. This shows that the fake or malicious work is being conducted very systematically through the agents who are well-trained in doing so. It is speculated that the problem of malicious crowdsourcing will increase and become more organized in future. It is very important to understand, identify and mediate such activities.

3.5.3 Oppositional/Malicious Activities on Social Media

In recent years online social network (OSN) has become a way to gain attention, fame, and good reputation [Aggarwal, 2016]. Malicious crowdworkers can impact the OSN by fake likes on facebook, fake voluntary followers twitter (also called volowers [Liu *et al.*, 2016]) and false reviews/ratings on Yelp. This can lead to intentional biases in the online information such as the wrong recommendation by the recommender systems, wrong priority for the online search results, and misdirected advertising revenue.

3.5.4 Oppositional/Malicious Crowdsourcing Statistics

[Wang *et al.*, 2012] show that crowdturfing is very common. They conducted experiments on different crowdsourcing websites by crawling the data. They found 89% cases of crowdturfing on Microtasks, 83% on MyEasyTask, 70% on Minute Workers 95% on ShortTask and 12% on Amazon Turk. This shows that the problem of malicious crowdsourcing is not limited to a particular platform rather it has become a global problem.

3.5.5 Maliciousness in Buying and Selling Crowd Services

Similarly, [Lee *et al.*, 2013] present their findings about crowdturfing by conducting experiments on Fiverr[11] a microtask website. Users can buy and sell services on this website, the services are called gigs. The authors randomly selected 1550 gigs out of which 121 i.e., 6% were found to be crowdturfing tasks. Among these crowdturfing gigs, 55.3% were related to online marketing. Further categorization of the 121 crowdturfing gigs showed that:

[11] https://www.fiverr.com/

- 65 targeted social media including facebook and twitter to increase the number of friend/followers or to the popularity of posts.
- 47 targeted search engines by artificially creating backlinks for their (gig buyers) website. Top seller of this task earned $3 million, 100% positive ratings, and more than 47000 positive comments, which shows the high demand of such workers/sellers and lucrativeness of crowdturfing.
- 9 crowdturfing gigs were to increase the visitors of a particular website creating artificial popularity. Author also did experiments by creating 5 brand new twitter accounts with no followers or following. They used the gigs worth to get followers and they were able to get up to 5500 followers in one hour and increase the klout (website for social media analytics) score immediately.

3.5.6 Maliciousness in Binary Labeling

In particular, [Tran *et al.*, 2009] talk about the website called Digg[12] where crowd labelers label the written articles as digg (popular) or bury (unpopular). This website is a perfect example of binary crowd labeling. The authors experimented by accessing the data of this website and found that many crowd labelers intentionally label the articles of their interest as digg and the rest as bury that results in the popularity and advertisement of the article on the homepage of digg.com. This also includes the labels provided by the labelers who register on the same day as the publication of the article or the labelers who are active only around the time a particular article is submitted. This definitely proves that such malicious activities are ongoing. One way to stop such activities is to keep a record of the worker history, warn and then block the labeler. But this requires extra efforts and also it is always possible to make a new account if blocked.

3.5.7 Malicious Behavior in Online Surveys

[Gadiraju *et al.*, 2015] present the malicious behavior of the crowdworkers in taking online surveys, which is yet another aspect of using crowd for ill purposes. In their paper, authors present a detailed study of maliciousness in taking online surveys. They developed a survey to test 1000

[12]https://www.digg.com

crowdworkers. They classify untrustworthy labelers as: (a) *ineligible workers* who take the survey even if they are ineligible or do not meet the conditions to do the task, (b) *fast deceivers* who give invalid responses quickly to save time and deceive, (c) *rule breakers* who do not follow the provided instructions and rules for doing the task, (d) *smart deceivers* who follow the instructions but intentionally give misleading answers, and (e) *gold standard preys* who follow the rules but may make mistakes due to inattentiveness. It is interesting to note that the authors have also compared the time to complete the task for each type of untrustworthy workers. They found that fast deceivers had lowest response time while gold standard prey had the highest response time.

3.5.8 More Advanced Malicious Crowdworkers

The paper by [Wang *et al.*, 2012] shows the evidence of more systematic malicious crowdsourcing by intentionally polluting data for training machine learning classifiers. The authors refer to this technique of malicious crowdsourcing as "poisoning". They claim that it is done by the website administrators (such as ZhuBaJie (ZBJ)[13]and SanDaHa (SDH)[14]). The crowdturfing class is poisoned by adding non-malicious crowd accounts to the malicious crowd accounts. This collection is then used as ground truth to train the classifiers and leads to wrong results acquired by classifiers. The second method of poisoning is to inject turfing examples to the non-malicious accounts. Both of these cases show that crowdturfing can be done in a very careful way outsmart the machine learning classifiers and to nullify all the measures taken to prevent it.

3.5.9 How Can Our Methodology Help?

From the above-mentioned discussion, it is evident that malicious crowdsourcing is not only very common but also changing to a very organized and profitable business. Also as the crowdsourcing is getting popular so is crowdturfing. Online data is not being polluted by the lazy or careless crowdworkers but mostly by the malicious crowdworkers who intentionally are creating and propagating false information. If the information by the malicious crowdworkers is harnessed, it would save a lot of time, energy and cost.

[13]http://www.zhubajie.com/c-tuiguang/

[14]http://www.sandaha.com/

In this thesis, we focus on crowd labeling rather than crowdsourcing. We have presented the methodologies to identify and utilize labels provided by malicious crowdworkers. We have checked the robustness of our methodology on simulated and real datasets. Our methods have shown promising results and we believe that these can be used to get the truth out of the intentional wrong information provided online. For example, our methodology can help do more accurate rating of articles on digg.com by identifying the malicious labelers and using their information by adjusting it accordingly. Our methodology can also be helpful in accurate rating on social media by identifying fake likes and followers (volowers), which are also examples of binary labeling.

3.6 Conclusion

In the first part of the thesis, we have proposed a robust crowd labeling framework using both expert evaluation and pairwise comparison between crowd-labelers. The framework embeds a set of methodologies to advance the state-of-the-art in crowd labeling methods. Our methodologies are simple yet powerful and make use of a handful expert-labeled instances to squeeze the best out of the labeling efforts produced by a crowd of labelers.

We propose a variety of methodologies to choose from according to the crowd characteristics and labeling needs. We show through several experiments on real and synthetic datasets that unlike other state-of-the-art methods, our methods are robust even in the presence of a large number of bad labelers. The most important aspects of our method include overcoming the phase transition inherent in other approaches as well as utilizing the information provided by the malicious labelers.

Chapter 4

Lower Bound on the Number of Expert Labels

4.1 Motivation & Introduction

In ELICE framework, we use expert-labeled instances to learn labeler ability α and instance difficulty β. Therefore, it is important to have enough expert-labeled instances to be able to estimate these values accurately to make further estimations or decisions based on them. Given that expert-label acquisition can be expensive, it is desirable to find the lower bound on the number of expert-labeled instances needed, which can also provide a good estimate of α and β. In this chapter, we will derive this lower-bound. It should be noted here that the lower bound we derive here is only for the case when expert-labels are ground truth and does not cover the case when expert-labels can be wrong. This is to avoid the uncertainty that can be present when the expert-labels are not ground truth.

We believe that this scenario is similar to Probably Approximately Correct (PAC) learning where the learner has to learn the concept with the minimum possible examples with a given accuracy and confidence. Therefore, we use the PAC learning framework to derive a bound. As a prerequisite to this, we explain the following terms:

4.1.1 Quality of the Crowd (c)

Let p_j be the probability of getting a correct label from labeler j and f be the probability distribution of p_j. Then we define the quality of the crowd c as

$$c = E(P) = \sum_{j=1}^{M} p_j \; f(p_j) \tag{4.1}$$

where $0 \leq c \leq 1$ and large values of c represent better crowd.

4.1.2 Difficulty of the Dataset ($1 - d$)

We define,

$$d = E(Q) = \sum_{i=1}^{N} q_i \; h(q_i) \tag{4.2}$$

where $0 \leq d \leq 1$, q_i is the probability of getting the correct label for instance i and h is the probability distribution for q_i. Higher d represents easier dataset.

4.1.3 Judgment Error (e)

The judgment error is the error made in estimating/judging true labeler ability and true instance difficulty based on the expert-labeled (ground truth) instances. Instead of per labeler and per instance judgment error, we define the judgment error to be overall judgment error (e) based on crowd quality and dataset difficulty.

4.2 Judgment Error Relation with Crowd Quality & Dataset Difficulty

In general, c and d are unknown, we make a conjecture about the crowd quality and dataset difficulty based on the performance of crowd on a given dataset. So the judgment error depends on how much the conjecture deviates from the true values of c and d.

We can use c to categorize the crowd. When the crowd is below average $c < 1/2$ (or $(c-1/2) < 0$). When the crowd is above average $c > 1/2$ (or $(c - 1/2) > 0$). When the crowd quality is close to 0 or 1 it is easy to estimate the ability based on a few instances hence the error in the judgment is low. But when the crowd quality is around 1/2 the error can be high as analyzing the crowd is hard

and needs to have more instances to be able to decide. That can be thought that judgment error in judging the crowd is inversely proportional to the crowd quality.

Similarly the dataset can be categorized as easy and difficult using the same strategy. If $(1-d) < 1/2$ (or $(d-1/2) > 0$), this means a difficult dataset and $(1-d) > 1/2$ (or $(d-1/2) < 0$) shows an easy dataset. Error in judging the instances will be less when dataset difficulty is close 0 or 1 and it will be high when the dataset quality is around 1/2.

But since overall outcome of the error in judgment is based on both c and d, we need to look at them in the combined way. Therefore the judgment error of crowd and dataset is inversely proportional to $(c-1/2)(d-1/2)$ i.e.,

$$e \propto \frac{1}{(c-1/2)(d-1/2)}$$

We formalize the relationship between the crowd, the dataset quality, and the judgment error by the function:

$$e = \frac{1}{1+(c-1/2)(d-1/2)} \tag{4.3}$$

where using Laplace smoothing, 1 is added to avoid the undefined values.

When the values of c and d are close to 1/2 then $(c-1/2)(d-1/2)$ becomes small and hence e becomes high. When the values of c and d are close to 0 or 1, $(c-1/2)(d-1/2)$ is relatively larger so e is small. When one of the c or d is less than $1/2$ and the other is greater than $1/2$ then the value of e is average. The graph of the function (Eq. 4.3) is shown in Figure 4.1.

4.3 Intuitive Explanation

To be able to explain Eq. 4.3 intuitively, we define the following:

4.3.1 Judgment Error Categories

The judgment error is categorized as follows.

- ◇ **High**: When the crowd is good and we conjecture it as a bad crowd (or vice versa), the judgment error is high. This is also true when a dataset is easy and the conjecture is difficult (or vice versa).

- ◇ **Medium**: When the crowd is mediocre and we conjecture it as bad or good (or vice versa) the judgment error is considered to be medium. Same is true about the dataset.
- ◇ **Low**: The judgment error is deemed low when our judgment about the crowd and/or dataset is close to the true quality.

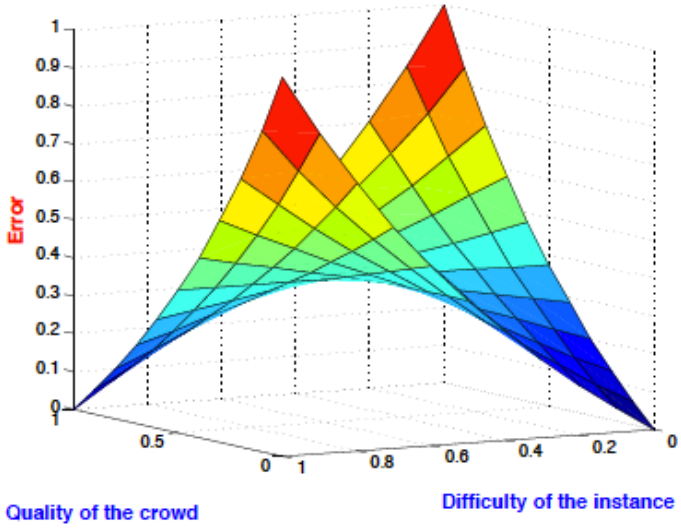

Figure 4.1: Graph of the normalized judgment error distribution. Quality of the crowd and difficulty of the dataset versus judgment error.

4.3.2 Analyzing Judgment Error

The intuitive explanation of the judgment error is summarized in Table 4.1 and described as follows:

a) **Good crowd & Difficult dataset**: When the crowd is good and the dataset is difficult the performance of the crowd may be average. The conjecture made is that the crowd is bad to average and/or the dataset is of medium to high difficulty. So the judgment error is high in this case.

b) **Bad crowd & Difficult dataset**: If the crowd is very bad and instances are very difficult, then the performance of the crowd will be poor. Hence the conjecture will be bad crowd and/or difficult dataset. Therefore, the judgment error is low.

			Dataset		
			Very Difficult	**Moderate**	**Very Easy**
Crowd	**Very Bad**	Conjecture about Crowd	Bad	Bad–Avg.	Avg.–Good
		Conjecture about Dataset	Diff.	Diff.–Mod.	Diff.– Mod.– Easy
		Judgment Error	Low	Medium	High
	Average	Conjecture about Crowd	Bad–Avg.	Bad–Avg.–Good	Good– Avg.
		Conjecture about Dataset	Diff.–Mod.	Diff.–Mod.–Easy	Mod.–Easy
		Judgment Error	Medium	Medium	Medium
	Very Good	Conjecture about Crowd	Bad– Avg.	Good–Avg.	Good
		Conjecture about Dataset	Diff.–Mod	Diff.–Mod.	Easy–Mod.
		Judgment Error	High	Medium	Low

Table 4.1: Judgment error distribution of the conjecture about the crowd and dataset. Crowd is categorized as very good, average, or very bad. Dataset is categorized as very easy, moderate, or very difficult. Judgment error can be high, medium, or low.

c) **Good crowd & Easy dataset**: When the crowd is very good and the instances are very easy our conjecture is good crowd and/or easy instances. Therefore, the judgment error is low.

d) **Bad crowd & Easy instances**: When the crowd is bad and dataset is very easy then the judgment can be biased and the judgment error can be high.

e) **Average crowd OR Moderate instances**: When the crowd is of average capability then for any kind of the instances the judgment may not be very far from the true value hence the judgment error is medium. This also holds for average difficulty dataset and any kind of crowd.

4.4 Theoretical Bound

For a given confidence $(1-\delta)$ and given values of c and d, the lower bound on the number of expert labels is given by

$$n_{LB} = [\frac{(b-a)(1+(c-1/2)(d-1/2))}{[1-a(1+(c-1/2)(d-1/2))]} log\ \frac{1}{\delta}] \tag{4.4}$$

where a and b are the minimum and maximum of the values of the judgment error e respectively and $[.]$ is the nearest integer function.

Proof: The proof of this theorem is straightforward. We know that the number of examples required by a PAC learning model is given by

$$n \geq \frac{1}{\epsilon} log\ \frac{1}{\delta} \tag{4.5}$$

where ϵ is the judgment error and δ is the level of confidence. In our case the judgment error is depending on c and d hence the judgment error e is here

$$e = \frac{1}{1+(c-1/2)(d-1/2)} \tag{4.6}$$

We normalize this judgment error as follows

$$\epsilon = \frac{(e-a)}{(b-a)} \tag{4.7}$$

where a = min(e) & b = max(e) for $0 \leq c \leq 1$ and $0 \leq d \leq 1$.

Therefore, we get

$$\epsilon = \frac{1}{(b-a)}[\frac{1}{1+(c-1/2)(d-1/2)} - a] \tag{4.8}$$

Plugging in the values into the PAC learning model (Eq. 4.5), we get the expression

$$n \geq \frac{(b-a)(1+(c-1/2)(d-1/2))}{[1-a(1+(c-1/2)(d-1/2))]} log\ \frac{1}{\delta}$$

More specifically, we have the lower bound

$$n_{LB} = [\frac{(b-a)(1+(c-1/2)(d-1/2))}{[1-a(1+(c-1/2)(d-1/2))]} log\ \frac{1}{\delta}]$$

where $[.]$ is the nearest integer function.

□

4.5 Empirical Evaluation of Theoretical Bound

We conducted experiments to evaluate the effectiveness of our theoretical results.

Experimental design: We simulated data with different levels of crowd quality c and dataset difficulty $(1-d)$. The number of crowd labels was 4-6 per instance while the size of the dataset varied between 200-500 instances. We checked the effect of the number of expert-labeled instances on the accuracy of the final label for different levels of confidence $(1-\delta)$. Some of the results are reported in the following graphs (Fig. 4.2 to 4.9). The vertical lines in these graphs show the calculated lower bound n_{LB} based on the parameters.

The results are shown for ELICE 1 & 2 only. Due to non-availability of the features of these datasets, the cluster-based versions of ELICE 1 & 2 are not available while the results for ELICE 3 were not reported due to the fact that the lower bound was derived for the case when expert-labeled instances are ground truth.

Results: The experiments show that nearly maximum possible accuracy is obtained at or around n_{LB} and in most cases increasing the number of expert-labeled instances beyond n_{LB} is not very helpful. This is especially evident when the confidence level $(1-\delta) = 0.99$. The accuracy of ELICE may vary depending on the quality of the crowd and difficulty of dataset but the theoretical lower bound n_{LB} gives us optimal way to achieve it,. The theoretical lower bound is usually a small number as compared to the cardinality of the dataset. Our experiments show that the lower bound is always less than 10% of the data.

4.6 Conclusion

In this chapter, we have derived the theoretical lower bound on the number of expert-labels needed to achieve a given accuracy. The idea is based on PAC learning. We have also demonstrated the utility of the lower bound through empirical evaluation. In the next part of this thesis, we extend our research to the Bayesian framework for learning the parameters, which lead to label aggregation.

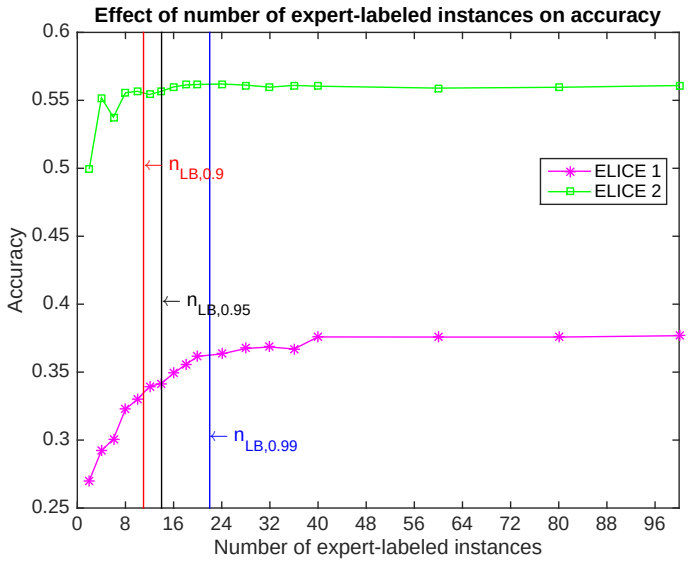

Figure 4.2: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.7842$ and dataset difficulty $(1 - d) = 0.1680$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 11, 14, 22$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

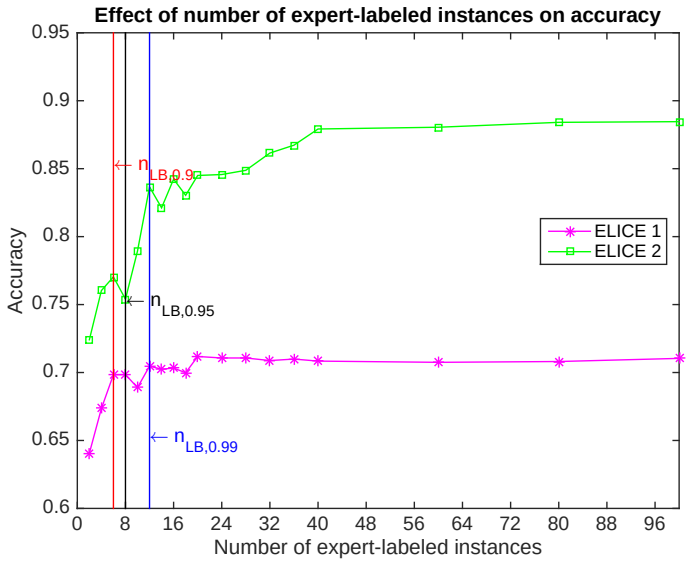

Figure 4.3: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.6019$ and dataset difficulty $(1 - d) = 0.4713$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 6, 8, 12$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

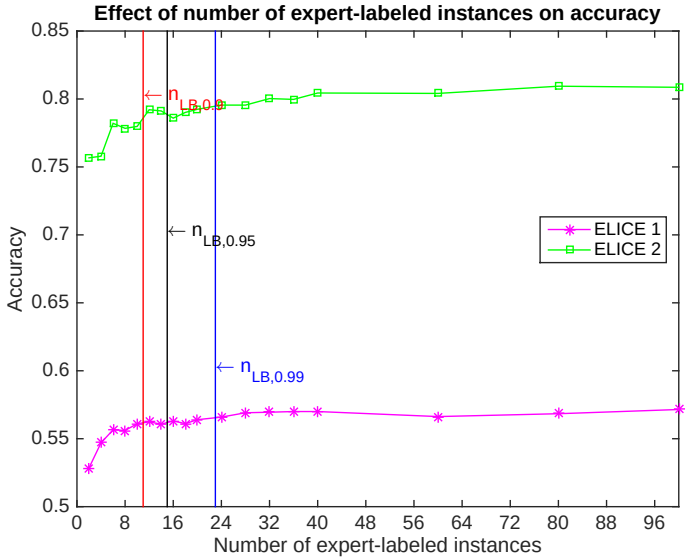

Figure 4.4: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.1894$ and dataset difficulty $(1 - d) = 0.8248$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 11, 15, 23$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

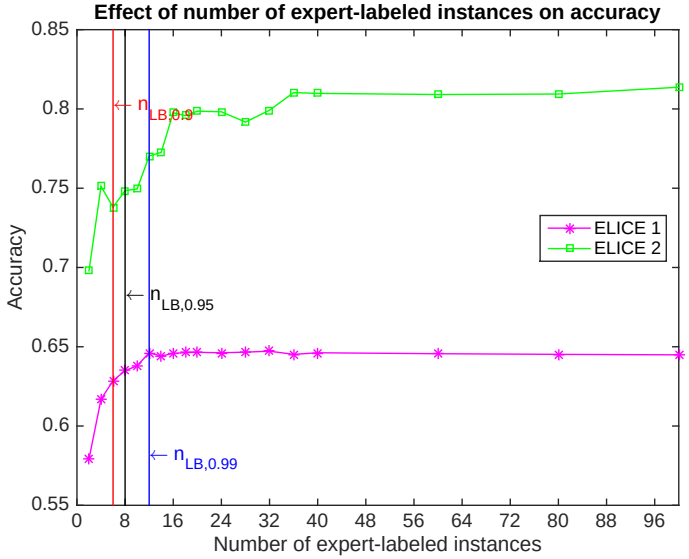

Figure 4.5: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.3617$ and dataset difficulty $(1 - d) = 0.4998$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 6, 8, 12$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

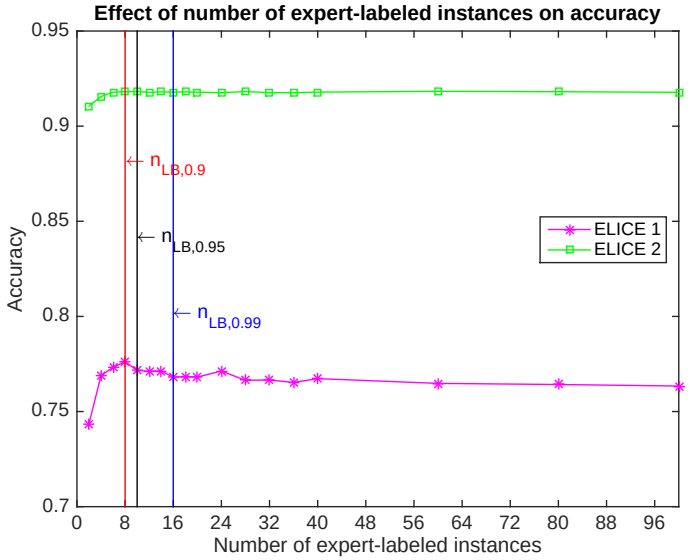

Figure 4.6: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.2175$ and dataset difficulty $(1 - d) = 0.6683$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 8, 10, 16$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

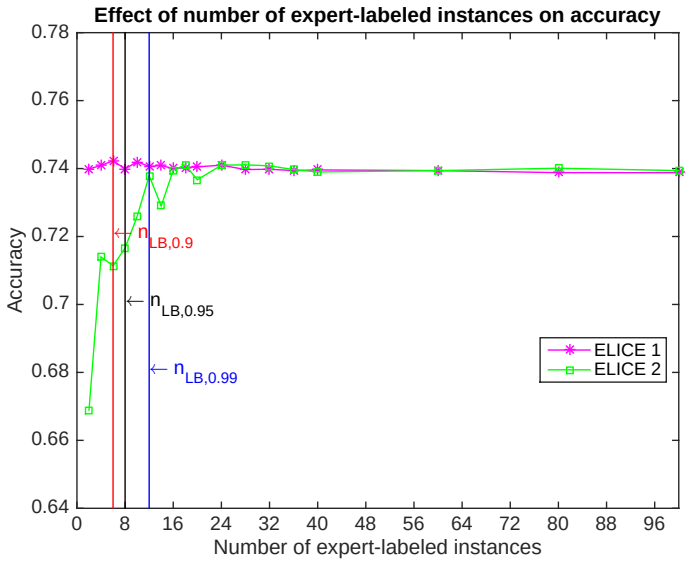

Figure 4.7: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.7896$ and dataset difficulty $(1 - d) = 0.4930$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 6, 8, 12$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

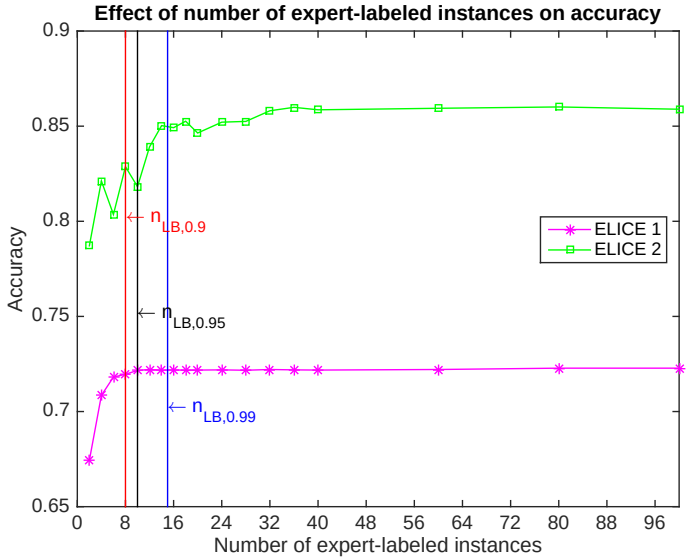

Figure 4.8: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.2706$ and dataset difficulty $(1 - d) = 0.6649$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 11, 15, 23$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

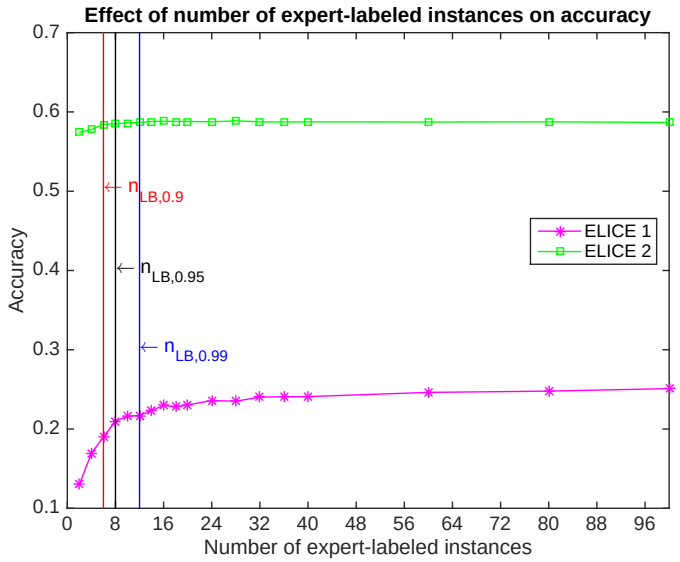

Figure 4.9: Number of instances $N = 500$, crowd labels $m = 6$, crowd quality $c = 0.1919$ and dataset difficulty $(1 - d) = 0.4796$. The theoretical bound is calculated using Eq. 4.4 is $n_{LB} = 6, 8, 12$ for confidence level $(1 - \delta) = 0.9, 0.95, 0.99$ respectively. It is shown by the vertical lines and is denoted by $n_{LB,(1-\delta)}$. The results are averaged over 100 runs.

Part II

The Bayesian Approach

Chapter 5

Crowd Labeling Using Bayesian Statistics (CLUBS)

In the second part of this thesis, we present a Bayesian approach to crowd labeling called Crowd Labeling Using Bayesian Statistics (CLUBS). Our approach is inspired by Item Response Theory (IRT) [Lord, 1952], that aims to design and analyze test scoring strategies. An IRT model is used to model parameters related to student and test questions as well as the probability of correctness of the answer. This makes IRT a compelling framework for crowd labeling.

We use a similar but more comprehensive approach for crowd labeling by introducing more parameters. The huge difference in an IRT model and our approach is that in an IRT model, the correct answers are known while in the CLUBS the answers are to be inferred. An IRT model is used to model student ability, test-question related parameters and probability of correctness of the answer to the question. Unlike an IRT model, our model not only learns the labeler and data-instance related parameters and probability of correctness of a label but also utilizes this information estimate the final labels.

This is made possible by incorporating expert labels (ground truth) for a small fraction of the dataset. Similar to our previous framework, expert-labeled instances are used here to help in the parameter estimation. Empirical evaluations on synthetic and real dataset show that our model produces more stable results as compared to the other state-of-the-art crowd labeling methods. We formally define our problem as follows:

Problem: A dataset $\mathcal{D}$ with N instances is labeled by M crowd labelers. Labels are chosen from predefined P number of classes. For $n(<< N)$ instances, one expert label (ground truth) per instance can be obtained. The expert labels are used to evaluate the parameters. The goal is to combine multiple labels to get one final label per instance with a maximum accuracy.

Our contribution in this chapter is summarized below:

- We present a new Bayesian model for crowd labeling that uses expert-labeled instances for a small fraction of a dataset.
- In our new methodology, we use a combination of parameters, namely per category labeler ability, instance difficult, prevalence of class, and question clarity.
- Our method is a one shot method and can work without the need of blocking the labelers and/or checking their previous history.
- We test our approach on synthetic and real datasets. We compare our approach to many other state-of-the-art methods.
- We present significance tests to evaluate the significance of the accuracy of our methods and other state-of-the-art methods.
- We present experiments showing the effect of using noisy labels as a training data.

5.1 Bayesian Versus Frequentist

It is well known that the frequentist approach is used when experiments can be easily repeated to estimate the parameters and their corresponding confidence intervals ([VanderPlas, 2014]). More specifically, in the frequentist approach, the underlying parameters are fixed while the data is variable and the results are based on the frequency of repeated events. Therefore, the frequentist analysis is based on the point estimates and maximum likelihood approaches.

In our frequentist approach ELICE, we have relied on few expert-labeled instances to learn the parameters. Although our approach presented good results, one limitation was that we could not repeat the experiments to learn the parameters an infinite number of times due to the fact that expert

labels are expensive and cannot be acquired frequently. This motivated us to explore the Bayesian approach in which, unlike the frequentist approach, the random sample is fixed.

In the Bayesian approach, parameters are unknown and are described using probabilities. The Bayesian approach can especially be used when repeating the experiments is not possible. The Bayesian methodology generally quantifies the properties of unknown model parameters in the light of observed data. The Bayesian approach considers probabilities to measure degrees of knowledge. For the Bayesian analysis, generally the posterior is computed, using analytical methods or through some version of MCMC sampling.

5.1.1 The Bayesian Approach Advantages

The advantages of Bayesian approach are as follows:

- Prior information can be easily incorporated. Posterior of the Bayesian can become prior for future observations.
- It relies on data without the need of asymptotic approximation like the frequentist approach.
- It has the flexibility of building hierarchical models.

Despite all the advantages, choice of the prior and in some cases high dependence on priors can be challenging. For large sample sizes, the Bayesian inference may provide results similar to the frequentist methods results. Also, the Bayesian inference may have high computational cost especially when the number of model parameters is large.

5.2 Our Approach

In an IRT model, the probability of getting a correct answer for a test question is assumed to be a mathematical function of student ability and question parameters. This model is used in many exams, including GRE and GMAT. It is considered to be a better approach than its other classical counterparts e.g., classical test theory ([Novick, 1966; Lord and Novick, 1968]), which consider the same level of difficulty for all questions on the test.

The IRT approach is used to model student ability, question difficulty, question clarity and probability of correctness of the answer for the question. The parameters are combined in the

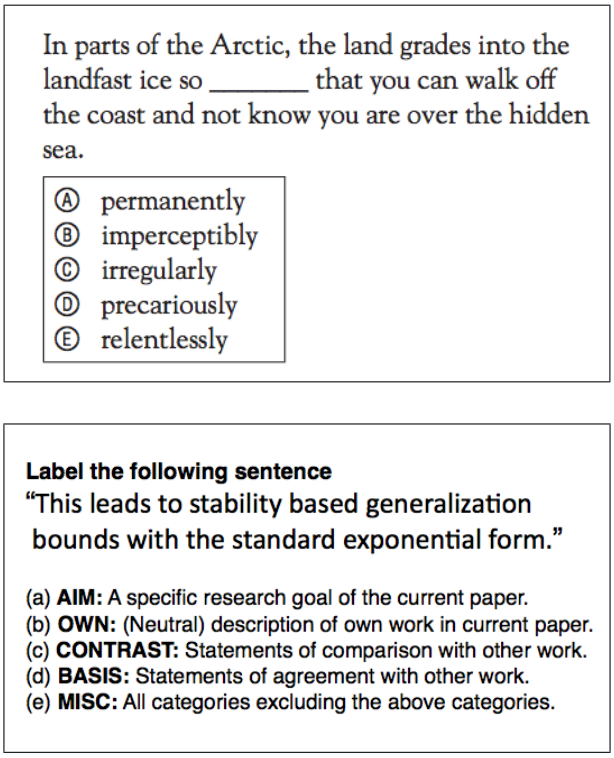

Figure 5.1: *(Top) An example of a typical GRE question (https://www.ets.org). Answer: B. IRT model is used to evaluate the students on Graduate Record Examination (GRE). (Bottom) An example from UCI Sentence Classification Dataset ([Asuncion and Newman, 2007]). Answer: D. This dataset consists of sentences from research articles to be classified as one of the given categories. This figure shows the similarity between test taking and crowd labeling scenarios.*

following formula, which uses the logistic function to estimate the probability of getting correct answer given all the parameters:

$$P[A_i|\alpha_j, \delta_i, \beta_i] = [\text{logit}^{-1}(\delta_i(\alpha_j - \beta_i))] \tag{5.1}$$

A_i : Correct answer to question i,

α_j : ability of student j,

β_i : difficulty of question i, δ_i : clarity of question i.

5.2.1 Crowd Labeling Using Bayesian Statistics (CLUBS)

We believe that crowd labeling is very similar to test taking [Carpenter, 2008], as can be seen in Figure 5.1. In both test taking and crowd labeling case, there is a set of predefined possible answers to choose from while the ability of the person making the choice is unknown.

We introduce new parameters and refine the existing IRT parameters to fit the crowd labeling scenario. Our new approach is called crowd labeling Using Bayesian Statistics (CLUBS). Despite the similarity of the test taking and crowd labeling scenarios, crowd labeling is more challenging. In the test taking scenario, the answers to the test questions are known and the goal is to only estimate the parameters. In contrast, for crowd labeling we need to estimate the parameters and get the final label based on the parameter estimates.

To deal with this challenge, we use expert-labeled instance (ground truth) for a small percentage of dataset instances (usually 0.1% -10%) to learn the parameters. Once the parameters are learned, they are used for aggregation of multiple crowd-labels for the rest of the dataset with no ground truth available.

We include the following parameters in our model that consist of modified IRT parameters as well as new ones.

5.2.1.1 Per-category Ability (π):

Human beings can be biased in their choices due to cultural differences, religious beliefs and personal preferences. Therefore, judging the labelers just in terms of correct or incorrect labels does not give us enough insights about the performance of the labeler. It can be more informative to estimate the labeler ability on a per-category basis leading to better labeling results.

Another reason for considering per-class ability is due to the fact that some labelers can intentionally label all the instances with the same label, in the hope of getting a portion of the labels right. This can help them avoid the mental effort and yet may result in high overall correctness score of the labeler. This can especially affect the labeling accuracy when one class is expected to be in majority. This is the case for imbalanced datasets with a skewed distribution such as malignant versus benign tumors identification task. We detected similar behavior of the labelers while experimenting on real datasets, explained in the next section.

We define a per-category ability parameter as π_c, the log odds for labeler j to correctly classify

an instance from class c_k.

$$\pi_{c_k}^{(j)} = log(\frac{\text{Number of correct labels for class } c_k \text{ by labeler } j}{\text{Number of incorrect labels for class } c_k \text{ by labeler } j})$$

5.2.1.2 Labeling Question Difficulty (β):

Many test taking ([Lord and Novick, 1968]) and crowd labeling methods ([Dawid and Skene, 1979]) consider the difficulty level of a question or data instance to be the same, which may not be always true. In any given dataset, instances can be of heterogeneous difficulty level. It is crucial to consider instance difficulty as it can affect the labeler ability in identifying the correct label. In our proposed model, we use a parameter β to quantify the level of difficulty of an instance.

5.2.1.3 Prevalence (γ):

Unlike the IRT model, in crowd labeling, we do not have any information about the class proportion of the dataset. Prior information can help in more accurate estimation of the model. This prior information can be incorporated in the form of prevalence of class. It is well known that the class proportion can vary for each dataset. Prevalence of the class can affect the results and not incorporating it can result in a loss of important information ([Byrt *et al.*, 1993]).

To make our model complete and to incorporate all possible information, we introduce a parameter to capture prevalence. Prevalence parameter is defined as $\gamma_{c_k} = P(i \in c_k)$, the probability that any instance i belonging to class c_k. Prevelence of class has been used by other researchers as well, for example, it is used in [Dawid and Skene, 1979] for binary classification tasks, although not referred to there as "prevalence". Similarly, prevalence is also used in the work based on [Dawid and Skene, 1979] model, such as [Carpenter, 2008] and [Passonneau and Carpenter, 2014]. It should be noted that prevalence has a direct effect on the per category ability of the labeler.

5.2.1.4 Clarity of Question (δ):

A clear explanation of the labeling question can improve the accuracy of the final label. In many cases, a misunderstood or poorly designed labeling task can produce flipped labels resulting into wasted efforts, work rejection for the labeler, extra cost and low accuracy for the requester of the task ([Kittur *et al.*, 2008]).

In our model, the clarity of the labeling question is quantified by the parameter δ. This parameter can be assumed to be unique for the whole dataset or can be considered individually for each instance, depending on the context of the labeling task. It should be noted that we use separate parameters for instance difficulty and question clarity because the former cannot be changed while the latter can be improved by providing better instructions.

5.2.2 Parameter Estimation

We use all the above mentioned parameters and formulate our new crowd labeling model is as follows.

$$P[c_k|l_{ij} = c_k, \gamma_{c_k}, \beta_i, \delta_i, \pi_{c_k}^{(j)}] = [\text{logit}^{-1}(\delta_i(\gamma_{c_k} + \pi_c^{(j)} - \beta_i))] \tag{5.2}$$

where

$$c_k : \text{class/category},$$

$$l_{ij} : \text{Label provided by labeler } j \text{ to instance } i,$$

$$\pi_{c_k}^{(j)} : \text{per-class ability of labeler } j,$$

$$\beta_i : \text{difficulty of instance } i,$$

$$\delta_i : \text{clarity of question asked about instance } i,$$

$$\gamma_{c_k} : \text{prevalence of class } c_k.$$

This model is run on the expert-labeled instances and the parameters are estimated. The graphical model for parameter estimation is given in Figure 5.2 (top). In this graphical model the shaded nodes show the observed values. The plate notation represents the variables that are repeated in the model i.e., instances, labelers and classes denoted by i, j and k respectively.

5.2.3 Label Aggregation

After the parameter estimation, the next step is to get the final label, that is F_i where i is an instance from the rest of the dataset, for which expert-labels (ground truth) are not available. The final label F_i is determined by the sign of the weighted sum of the labels, where the weight is the probability

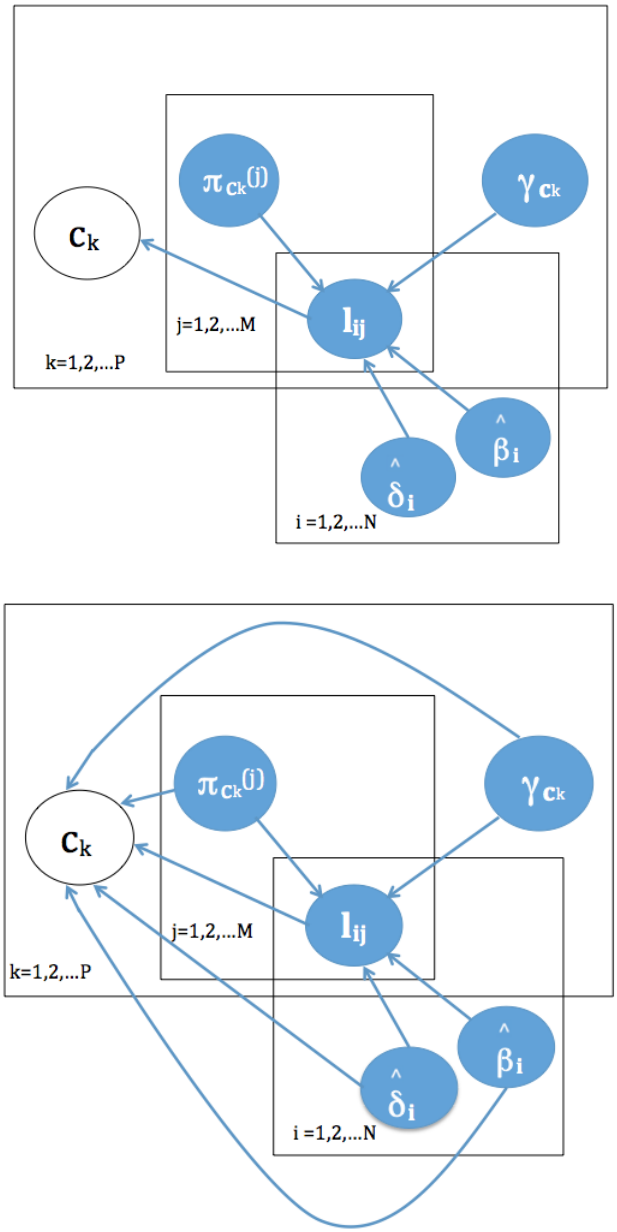

Figure 5.2: *Graphical model of CLUBS. (Top) Parameter estimation (Bottom) Label aggregation. Shaded nodes represent observed values.*

of correctness of label provided by a labeler. This probability is calculated by inputting estimated parameters in the model. The aggregation formula is as follows.

$$F_i = \text{sign}[\sum_j (P[c_k|l_{ij} = c_k, \gamma_{c_k}, \hat{\beta}_i, \hat{\delta}_i, \pi_{c_k}^{(j)}] * l_{ij})] \tag{5.3}$$

where

$$P[c_k|l_{ij} = c_k, \gamma_{c_k}, \hat{\beta}_i, \hat{\delta}_i, \pi_{c_k}^{(j)}] = [\text{logit}^{-1}(\hat{\delta}_i(\gamma_{c_k} + \pi_{c_k}^{(j)} - \hat{\beta}_i))]$$

In this aggregation formula, we use the expert-labeled instances based estimates of the prevalence of the class (γ_c) and per-category labeler ability ($\pi_{c_k}^{(j)}$), based on the assumption that these values remain unchanged for the rest of the dataset. But difficulty (β_i) and discrimination level (δ_i) of the instances without expert-labels are unknown.

To mediate this problem, we assume that both of these unknown parameters (β_i and δ_i) follow the same statistical distribution as the parameter estimates for the expert-labeled instances. We calculate the mean and standard deviation of the estimates of these parameters for expert-labeled instances and generate these parameters for the rest of the dataset, that is

$$\hat{\beta}_i \sim \text{normal(mean-of-estimated-beta, sd-of-estimated-beta)}$$

$$\hat{\delta}_i \sim \text{normal(mean-of-estimated-delta, sd-of-estimated-delta)}$$

The graphical model for label aggregation is given in Figure 5.2 (bottom).

In the next chapter we present empirical results of CLUBS on simulated and real labels.

Chapter 6

Empirical Evaluation

In this chapter, we present experimental results to check the performance and understand the behavior of our method as compared to the other state-of-the-art methods.

Implementation: We implemented our model in Stan ([Team, 2014]), a probabilistic programming language for Bayesian inference. Stan program computes a log posterior density while inference engine performs Hamiltonian Monte Carlo using no-U-turn sampler for sampling from posterior distributions. Using a Stan program, we can define a statistical model as a conditional probability function on unknown values including latent variables, unknown parameters, missing data and future predictions. The model is conditioned on the known values of data.

State-of-the-art methods: We compared our method to Majority voting (different versions), GLAD (Generative model of Labels, Abilities, and Difficulties) and GLAD with clamping ([Whitehill *et al.*, 2009]), [Dawid and Skene, 1979] method, Expectation Maximization (EM), iterative method by [Karger *et al.*, 2014] (KOS), Mean Field algorithm (MF), Belief Propagation (BP) by [Liu *et al.*, 2012] and ELICE all versions. However, it should be noted that ELICE with clustering results could not be calculated due to unavailability of the features for clustering. Moreover, the results for iterative method by [Karger *et al.*, 2014] (KOS), Mean Field algorithm (MF) and Belief Propagation (BP) by [Liu *et al.*, 2012] are reported for two different parameters setting. All other methods except ours were implemented in MATLAB and in most cases the code was obtained from the authors of the methods.

Datasets: We conducted several experiments on the following datasets:

1. Synthetically generated data.
2. Recognizing Textual Entailment (RTE) dataset.[1]
3. Temporal dataset.[1]

The two real datasets have labels available and were used as benchmarks to evaluate state-of-the-art methods (e.g., [Liu *et al.*, 2012]).

6.1 Synthetic Data

We conducted several experiments on synthetically generated data, using variation in the size of the dataset, number of expert-labeled instances and number of crowd-labelers. It allowed us to check the robustness of our method for a variety of data. We are reporting the summary of our findings on four simulated datasets.

Data Generation: To make the experiments complete, we generated data in two different ways, which are stated below:

- Dataset A and B are generated with probability obtained by assigning different values to the parameters in equation 5.2. For the sake of observing labeler ability effect on the accuracy of the methods, we use a fixed range of values for all the parameters except the labeler log-odds (π_j). We vary labeler log-odds for each dataset that is reported in Table 6.1. We generated the rest of the parameters as follows: instance difficulty $\beta_i \sim \mathcal{N}(0, 2)$, instance question clarity $\delta_i \sim \mathcal{N}(0, 0.75)$ ($\mathcal{N}(.\ ,.)$ denotes the normal distribution) and prevalence of class $\gamma_{c_k} = 0.5$, where $k = 2$. Each dataset consists of 5000 instances with four crowd-labels per instance. We took 20 ground truth instances as expert-labeled instances.
- We generated labels for datasets C and D using different ranges of per class correctness for each labeler ($x\% - y\%$), reported in Table 6.2 and 6.3. Dataset C and D each consist of 5000

[1] available at http://ir.ischool.utexas.edu/square/data.html

		Dataset A	Dataset B
Class 1	Per labeler % correctness	(0.89, 0.90, 0.89, 0.95)	(0.98, 0.98, 0.85, 0.80)
	True Log-odds	(2.07, 2.18, 2.12, 3.01)	(3.89, 3.89,1.74,1.39)
	Estimated Log-odds	(1.42, 0.84, 1.42, 2.43)	(1.11, 0.64, 0.20, 0.20)
Class 2	% Correctness	(0.96, 0.83 0.98, 0.76)	(0.97, 0.89, 0.73, 0.74)
	True Log-odd	(3.18, 1.58, 4.23, 1.17)	(3.48, 2.09, 0.99, 1.05)
	Estimated Log-odds	(2.42, 1.00, 2.00, 2.00)	(1.19,1.50, 0.86,1.51)

Table 6.1: Synthetic Data generation parameters and estimated parameters for the labelers. For the sake of presenting the labeler ability impact, the other parameters are kept fixed that instance difficulty $\beta \sim \mathcal{N}(0, 2)$, instance question clarity $\delta \sim \mathcal{N}(0, 0.75)$ and prevalence of class $\gamma = 0.5$.

	Labelers			
	L1	L2	L3	L4
Class1	0%-40%	0%-40%	0%-40%	70%-100%
Class 2	30%-70%	30%-60%	90%-100%	50%-60%

Table 6.2: Labeler correctness rate for Dataset C.

	Labelers			
	L1	L2	L3	L4
Class1	60%-80%	20%-40%	80%-90%	30%-50%
Class 2	50%-70%	30%-50%	90%-100%	50%-60%

Table 6.3: Labeler correctness rate for Dataset D.

and 3000 instances respectively. Crowd labels were simulated for 4 labelers and ground truth labels for 20 instances were used as expert-labeled instances.

It should be noted here that we conducted the experiments with 20-50 expert-labeled instances and 4-8 crowd-labels per instance, which produced similar results but here we only report the results based on 20 expert-labeled instances and 4 crowd-labels.

Results: The accuracy of the final labels for the simulated datasets is given in Table 6.1. The results were averaged over 20 runs. We can see that majority voting with gold testing has a good performance but since its results are not stable due to resulting in 'NaN' many times, it is not the best option. On the other hand all of our methods have good performance. CLUBS is a winner on dataset A and B while ELICE performs better on dataset C and D. It should also be noted that the performance of our methodologies is stable and does not end up in surprisingly different outcome.

6.2 Recognizing Textual Entailment Dataset

For this dataset the task was described as "whether the second sentence (the Hypothesis) is implied by the information in first sentence (the Text)." Labels provided were "Yes, No" (converted to "1, -1" for implementation.)

6.2.1 Experimental Design

We randomly selected 153 labeled instances from Recognizing Textual Entailment (RTE) dataset along with ground truth. The crowd labeling task was to judge the textual entailment for two sentences Text and Hypothesis. Each of the 153 instances was labeled by the same five labelers. For

[2]In many cases, majority voting with gold testing resulted into NaN (Not a number) due to all labelers being discarded in the testing phase. The reported results were averaged, only over the cases when the experiments produced a number, ignoring the case when the results were NaN.

[3]In many cases, majority voting with gold testing resulted into NaN (Not a number) due to all labelers being discarded in the testing phase. The reported results were averaged, only over the cases when the experiments produced a number, ignoring the case when the results were NaN.

[4] No result was produced as all labelers were discarded when tested, in all the runs of the experiment.

[5]Code for Belief propagation did not converge.

Method \ Dataset	**Dataset A**	**Dataset B**	**Dataset C**	**Dataset D**
Majority Voting	0.82	0.81	0.50	0.61
Majority Voting (25%[3])	0.90[2]	0.83[2]	0.70[2]	0.77[2]
Majority Voting (35%[3])	0.94[2]	0.84[2]	0.82[2]	0.78[2]
Majority Voting (45%[3])	0.98[2]	0.90[2]	0.93[2]	0.92[2]
Majority Voting (55%[3])	1.00[2]	0.92[2]	NaN[4]	1.00[2]
GLAD	0.75	0.76	0.18	0.46
GLAD with clamping	0.75	0.76	0.18	0.46
D &S	0.86	0.79	0.22	0.46
EM	0.77	0.76	0.17	0.46
BP (uniform prior)	0.82	0.83	0.58	0.69
BP (Beta(2,1) prior)	0.85	0.87	—[5]	0.57
MF (uniform prior)	0.77	0.76	0.17	0.46
MF (Beta(2,1) prior)	0.86	0.79	0.22	0.46
KOS	0.75	0.76	0.17	0.38
KOS2	0.75	0.76	0.17	0.46
ELICE 1	0.87	0.87	0.90	**0.84**
ELICE 1 with clustering⋆	-	-	-	-
ELICE 2	0.88	**0.89**	**0.93**	**0.84**
ELICE 2 with clustering ⋆	-	-	-	-
CLUBS	**0.89**	**0.89**	0.59	0.78

Table 6.4: Performance on Synthetic Data. Each dataset consists of 3000-5000 instances labeled by four labelers. Ground truth for 20 instances was taken as expert-labels. ELICE with clustering results could not be calculated due to unavailability of the features for clustering. ⋆ Since the features for these datasets are not available therefore the results of ELICE with clustering could not be calculated.

Labelers / % Correctness	L1	L2	L3	L4	L5
Overall	80.40%	47.72%	52.29%	49.68%	46.41%
Class1	82.90 %	34.22%	21.06%	18.43%	22.37%
Class 2	77.93 %	61.04%	83.12%	80.52%	70.13%

Table 6.5: Labeler performance for RTE Data.

Labelers / % Correctness	L1	L2	L3	L4	L5
Overall	Good	Random	Random	Random	Random
Class1	Good	Oppositional	Oppositional	Oppositional	Oppositional
Class 2	Good	Random	Good	Good	Good

Table 6.6: Labeler category for RTE Data.

our experiments, we use ground truth for 20 randomly selected instances as expert-labels. We have reported the overall and per-class error rate of all the labelers in Table 6.5. While Table 6.6 reports the overall and per-class category of these labelers.

6.2.2 Results

The accuracy of the final label is reported in Table 6.7. The accuracy presented in each column is based on different combination of the labelers (e.g., column L1-L5 shows the performance of each method based on the labels provided by labelers 1 to 5). This is done to evaluate the performance of the approaches for different labeler abilities. We discuss each column of the Table 6.7 as follows:

L1-L5: In the first column of this table, overall performance of the labelers is categorized as one good and four random labelers. We can see that in this column the performance of most of the methods MV to KOS2 is around 50%, while ELICE 1, ELICE 2, and CLUBS show a good performance. The reason is that one good labeler is helping to improve the overall performance of our methodologies.

L1-L4: This column consists of the results produced by the labels of one good and three random labelers. CLUBS has the highest accuracy with ELICE 1 and ELICE 2 having second best accuracy

Method \ Labelers	L1-L5	L1-L4	L1-L3	L1-L2	L2-L5	L2-L4	L2-L3
Majority Voting	0.55	0.52	0.61	0.57	0.47	0.50	0.50
Majority Voting (25%[3])	0.56[2]	0.59[2]	0.62[2]	0.67[2]	0.49[2]	0.51[2]	0.63[2]
Majority Voting (35%[3])	0.56[2]	0.58[2]	0.62[2]	0.75[2]	0.50[2]	0.51[2]	0.64[2]
Majority Voting (45%[3])	0.56[2]	0.55[2]	0.71[2]	0.80[2]	0.49[2]	0.51[2]	0.58[2]
Majority Voting (55%[3])	0.53[2]	0.58[2]	0.52[2]	NaN[4]	NaN[4]	0.46[2]	0.52[2]
GLAD	0.51	0.51	0.63	0.48	**0.51**	**0.53**	**0.54**
GLAD with clamping	0.51	0.51	0.63	**0.80**	**0.51**	**0.53**	0.48
D &S	0.41	0.46	0.47	**0.80**	0.46	0.47	0.48
EM	0.50	0.49	0.48	0.62	0.50	0.50	0.45
BP (uniform prior)	0.50	0.50	0.52	0.30	0.49	0.50	0.51
BP (Beta(2,1) prior)	0.46	0.34	0.51	**0.80**	0.46	0.49	0.48
MF (uniform prior)	0.50	0.50	0.48	0.59	0.50	0.50	0.51
MF (Beta(2,1) prior)	0.46	0.50	**0.80**	0.70	0.46	0.46	0.48
KOS	0.50	0.50	0.48	**0.80**	0.50	0.51	0.48
KOS2	0.50	0.50	0.38	0.38	0.50	0.51	0.50
ELICE 1	0.62	0.69	0.69	0.79	0.49	0.50	0.50
ELICE 1 with clustering⋆	-	-	-	-	-	-	-
ELICE 2	**0.67**	0.67	0.71	**0.80**	0.47	0.48	0.50
ELICE 2 with clustering⋆	-	-	-	-	-	-	-
ELICE 3 (Pairwise)	0.60	0.61	0.62	0.72	0.48	0.48	0.50
ELICE 3 (Circular)	0.57	0.61	0.60	0.52	0.48	0.48	0.50
CLUBS	0.65	**0.70**	0.73	0.74	0.48	0.51	**0.54**

Table 6.7: Accuracy of final label for RTE Data.

⋆ Since the features for these datasets are not available therefore the results of ELICE with clustering could not be calculated.

Labelers / % Correctness	L1	L2	L3	L4	L5	L6
Overall	91.02%	52.65%	46.53%	46.53%	53.47%	93.06%
Class 1	90.52%	0%	93.10%	97.41%	83.62 %	93.10%
Class 2	91.47%	100%	4.65%	0.78%	26.36%	93.02%

Table 6.8: Labeler performance for Temp Data.

Labelers / % Correctness	L1	L2	L3	L4	L5	L6
Overall	Good	Random	Random	Random	Random	Good
Class 1	Good	Oppositional	Good	Good	Good	Good
Class 2	Good	Good	Oppositional	Oppositional	Oppositional	Good

Table 6.9: Labeler category for Temp Data.

level.

L1-L3: In this column, one good labeler and two random labelers are used. It should be noted that while all the other methods show similar results as the previous two columns but surprisingly BP with beta prior gives the highest accuracy. On the other hand BP with uniform prior has a below average performance.

L1-L2: In this case, we have one good and one random labeler. Many methods have a better performance as the percentage of good labelers has increased. Highest accuracy is obtained by Dawid and Skene method, BP (Beta(2,1) prior), KOS, and ELICE 2.

L2-L5: All four labelers are random so none of the methods performs exceptionally well but GLAD (both versions) seem to be slightly better.

L2-L4: None of the results is exceptional but GLAD has the best accuracy once again.

L2-L3: Here we have two random labelers who are mostly correct on instances from the class 2. Here majority voting with gold testing has the highest accuracy while the second best methods are GLAD and CLUBS.

By looking at the results, we can see that our methods consistently have a good performance

Method \ Labelers	L1-L6	L1-L5	L1-L4	L1-L3	L1-L2	L2-L6	L2-L5	L2-L4	L2-L3
Majority Voting	0.75	0.60	0.64	0.89	0.73	0.60	0.50	0.46	0.46
Majority Voting (25%[3])	0.92[2]	0.73[2]	0.71[2]	0.73[2]	0.72[2]	0.96[2]	0.74[2]	0.53[2]	0.53[2]
Majority Voting (35%[3])	0.90[2]	0.71[2]	0.72[2]	0.72[2]	0.71[2]	0.97[2]	0.72[2]	0.53[2]	0.53[2]
Majority Voting (45%[3])	0.84[2]	0.70[2]	0.66[2]	0.69[2]	0.67[2]	0.91[2]	0.69[2]	0.53[2]	0.53[2]
Majority Voting (55%[3])	0.82[2]	0.69[2]	0.63[2]	0.64[2]	0.69[2]	0.94[2]	0.67[2]	0.53[2]	0.53[4]
GLAD	0.47	0.47	0.47	**0.91**	**0.91**	0.47	0.47	0.47	**0.53**
GLAD with clamping	0.47	0.47	0.47	**0.91**	**0.91**	0.47	0.47	0.47	**0.53**
D &S	**0.92**	**0.91**	0.88	0.89	**0.91**	0.86	0.46	0.46	0.46
EM	0.47	0.47	0.47	0.53	0.57	0.47	0.47	0.47	0.47
BP (uniform prior)	0.47	0.47	0.46	0.10	0.30	0.47	0.47	0.46	0.46
BP (Beta(2,1) prior)	**0.92**	**0.91**	0.89	**0.91**	**0.91**	**0.87**	**0.52**	0.46	0.46
MF (uniform prior)	0.47	0.47	0.47	0.53	0.62	0.47	0.47	0.47	0.47
MF (Beta(2,1) prior)	**0.92**	0.90	0.88	0.89	**0.91**	**0.87**	**0.52**	0.46	0.46
KOS	0.47	0.47	0.47	0.53	**0.91**	0.47	0.47	0.47	0.47
KOS2	0.47	0.47	0.47	0.53	0.30	0.47	0.47	0.47	0.47
ELICE 1	0.83	0.73	0.76	0.90	**0.91**	0.79	0.50	**0.49**	0.51
ELICE 1 with clustering⋆	-	-	-	-	-	-	-	-	-
ELICE 2	0.78	0.62	0.62	0.60	**0.91**	0.62	0.50	**0.49**	0.50
ELICE 2 with clustering⋆	-	-	-	-	-	-	-	-	-
ELICE 3 (Pairwise)	0.83	0.66	0.64	0.75	**0.91**	0.63	0.50	**0.49**	0.50
ELICE 3 (Circular)	0.81	0.65	0.70	0.88	0.47	0.63	0.50	**0.49**	0.51
CLUBS	0.91	0.88	**0.90**	**0.91**	**0.91**	0.85	**0.52**	0.46	0.46

Table 6.10: Accuracy of final label for Temp Data.

⋆ Since the features for these datasets are not available therefore the results of ELICE with clustering could not be calculated.

and even if they are not the best in some cases they are not unpredictably off. On the other hand, we have seen that many other methods have a very low performance on some datasets and very high on others. While the reason for the mysterious behavior of these methods remains unknown but it makes them unreliable and unpredictable. We believe that our methods can serve the purpose of accurate crowd labeling in more reliable manner.

6.3 Temporal Dataset

The Temporal (Temp) dataset consists of the labels given to the temporal sequence of the events in a given text. We describe the experimental design for this dataset in the next section.

6.3.1 Experimental Design

We randomly selected 245 instances labeled by 6 labelers. Number of expert-labeled instance was 20. The error rates of the labelers are given in Table 6.8 while the categories of the labelers are given in Table 6.9. From the Table 6.8 it is evident that using the per-category ability of the labelers gives a better insight into labeler performance as the labelers may perform very well on one class and do poorly for the other e.g., labeler 2 performs 100% on one class while 0% on the other. Similarly labeler 3 and 4 have a nearly perfect score on one class and nearly zero performance on the other class.

6.3.2 Results

The accuracy of the final label in Table 6.10 shows the stability of CLUBS. In this set of labelers, L1 and L6 are good labelers and the rest of labelers are random. It should be noted here that L2, L3, and L4 are extreme cases of being almost perfect on one class and totally oppositional on the other class. This created different results as compared to the previous set of labelers where we had one good and rest random labelers. Analysis of each column of the Table 6.10 is as follows:

L1-L6: In this case, Dawid and Skene method, BP (Beta(2,1) prior) and MF (Beta(2,1) prior) produced excellent results although CLUBS was also doing nearly as good. On the hand despite the good performance of versions of ELICE it lagged behind.

L1-L5: Removing one good labeler L6 produced the same pattern of the results but with lower

accuracy.

L1-L4: This set of labelers consists of one good labeler and three highly skewed random labelers. CLUBS is the winner in this case but Dawid and Skene method, BP (Beta(2,1) prior) and MF (Beta(2,1) prior) produced are also producing good results.

L1-L3: In this case CLUBS, GLAD (both versions), and BP (Beta(2,1) prior) have a tie and produce the best results. While ELICE1 and MF (Beta(2,1) prior) are nearly as good.

L1-L2: One good and one random labeler produce very good results almost for all the methods.

L2-L6: In this case, majority voting with gold testing is the winner but we know that results are not always reliable. Dawid and Skene method and BP (Beta(2,1) prior) are second, while MF (Beta(2,1) prior) and CLUBS are runner ups.

L2-L5: All labelers are random labelers in this case. Majority voting with gold testing is producing best results. It is because labeler L3, L4 and L5 are doing excellent on one class and L2 is perfect on the other class.

L2-L4: Again majority voting with gold testing is the winner.

L2-L3: Majority voting and GLAD (both versions) are the winner while ELCIE all versions have nearly same level of accuracy.

From the results reported in this table, we can see that our methods have good, stable and consistent performance. Some methods do perform well in some case but have unpredictable outcomes.

6.4 Technical Details

Our experiments are coded in RStan. Data generation/loading was done in R. Stan code is called from the R platform. Like most Stan programs our Stan code consists of three main blocks data (imported from R), parameter block consisting of parameter definition and the model block. The model block consists of the our CLUBS model as well as prior for the parameters. We use hierarchical priors and assume normal distribution of the priors. Number of iteration and chains are predefined. We tried different number of iterations and number of chains but we found the most optimal choice for our experiments was 1000-2000 iterations and 4-8 chains.

6.5 Discussion

The purpose of devising CLUBS is to explore and understand the Bayesian approach for parameter estimation using expert-labeled instances. Since we have already ventured the frequentist approach for parameter estimation, Bayesian is a natural choice to make our research comprehensive. Especially, due to the dependence of Bayesian approach on data without the need of asymptotic approximation like frequentist approach as well as the flexibility of building hierarchical models makes it a compelling idea to explore.

We believe that CLUBS has the advantages of estimating the per-category ability, having more variety of parameters, incorporating prior information, and easy extension to multi-class. Despite the fact that CLUBS does not always outperform ELICE but we believe that it has the capacity to enhance. The new parameters that are introduced in this approach can be something worth investigating further and can give us insight into the intricacies of the labeling scenario. In future, these parameters can be helpful in designing and conducting the labeling task.

6.6 Significance Tests

To check the significance of accuracy of the different methods, we perform the t-test between the accuracy levels of all different methods. We conduct a one-tailed paired t-test with significance level $\alpha = 0.01$ and $\alpha = 0.05$ on different UCI datasets. As all the results are similar, we only report the t-test results for the UCI breast cancer dataset with significance level $\alpha = 0.01$.

We used the MATLAB function *ttest2* for the experiments. Results are given in the Tables 6.11, 6.12 and 6.13 for different levels of labeler ability. In these tables, the methods in the leftmost column are compared to the methods in the top row and the outcomes 0, 1 and NaN are reported. The null and alternative hypothesis along with the meaning of the outcome is described as follows:

$H_o : \mu_A = \mu_B$ i.e., Accuracy of method A is as good as the accuracy of method B.

$H_1 : \mu_A < \mu_B$ i.e., Accuracy of method A is worse than the accuracy of method B.

where method A refers to the methods in the leftmost column and method B refers to the methods in topmost row.

If outcome is 0 $\implies$ fail to reject H_o. If outcome is 1 $\implies$ reject H_o.

If outcome is NaN $\implies$ the test is inconclusive.

Abbreviations used in the table are as follows,

MV = Majority voting,

MV1 = Majority voting (25%), **MV2** = Majority voting (35%),

MV3 = Majority voting (45%), **MV4** = Majority voting (55%),

G = GLAD, **GW** = GLAD with Clamping,

DS1 = Dawid & Skene 1, **DS2** = Dawid & Skene 2,

BP1 = Belief Propagation 1, **BP2** = Belief Propagation 2,

MF1 = Mean Field 1, **MF2** = Mean Field 2,

KOS1 = Karger's Iterative methods 1, **KOS2** = Karger's Iterative methods 2,

E1 = ELICE 1, **E1-C** = ELICE 1 with clustering,

E2 = ELICE 2, **E2-C** = ELICE 2 with clustering,

E3-P = ELICE 3 Pairwise & **E3-C** = ELICE 3 Circular.

Table 6.11 shows the significance test results for the crowd labelers who are correct 65% of time. As we can see that most methods in this case are performing equally good. The reason is that if the labeler are good all the methods perform well, even most naive ones like majority voting.

In Table 6.12 when the 50% of the crowd is making 35% mistakes and the rest is making 65% mistakes all versions of ELICE are showing higher significance. In this table, majority voting with gold testing (MV1, MV2, MV3 and MV4) show higher significance due to reliance in the good labelers but it should be noted that the results are unpredictable and only the cases are considered when the results are not NaN. CLUBS on the other hand, is not always the winner.

Table 6.13 shows similar results as the Table 6.12. More tables with different levels of labeler ability can be found in the appendix.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	NaN	0	0	0	NaN	NaN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV1	NaN	-	0	0	0	NaN	NaN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV 3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV 4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	NaN	NaN	0	0	0	-	NaN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GW	NaN	NaN	0	0	0	NaN	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DS1	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DS2	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0
BP1	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
BP2	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0
MF1	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0
MF2	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
KOS1	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0
KOS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0
E3-C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-	0
CLUBS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	-

Table 6.11: Paired t-tests results for the accuracy level of different methods: All labelers are making less than 35% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
G	1	1	1	1	1	-	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
GW	0	1	1	1	1	0	-	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
DS1	1	1	1	1	1	0	0	-	0	0	0	0	0	0	0	1	1	1	1	1	1	0
DS2	0	1	1	1	1	0	0	0	-	0	0	0	0	0	0	1	1	1	1	1	1	0
BP1	1	1	1	1	1	0	0	0	0	-	0	0	0	0	0	1	1	1	1	1	1	0
BP2	1	1	1	1	1	0	0	0	0	0	-	0	0	0	0	1	1	1	1	1	1	0
MF1	0	1	1	1	1	0	0	0	0	0	0	-	0	0	0	1	1	1	1	1	1	0
MF2	1	1	1	1	1	0	0	0	0	0	0	0	-	0	0	1	1	1	1	1	1	0
KOS1	0	1	1	1	1	0	0	0	0	0	0	0	0	-	0	1	1	1	1	1	1	0
KOS2	0	1	1	1	1	0	0	0	0	0	0	0	0	0	-	1	1	1	1	1	1	0
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	NaN	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
CLUBS	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	-

Table 6.12: Paired t-tests results for the accuracy level of different methods: 50% labelers are making less than 35% mistakes and 50% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
MV2	0	0	-	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
G	0	1	1	1	1	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
GW	0	1	1	1	1	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS1	0	1	1	1	1	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS2	0	1	1	1	1	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP1	0	1	1	1	1	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP2	0	1	1	1	1	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
MF1	0	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	1	1	1	1	1	1	1
MF2	0	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	1	1	1	1	1	1	1
KOS1	0	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	1	1	1	1	1	1	1
KOS2	0	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	1	1	1	1	1	1	1
E1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	-	0
CLUBS	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	-

Table 6.13: Paired t-tests results for the accuracy level of different methods: All labelers are making more than 65% mistakes.

	Error			
Noise level	Decision Trees	Random Forest	KNN	SVM
0	0.08	0.04	0.1	0.04
0.1	0.12	0.05	0.18	0.06
0.2	0.18	0.09	0.26	0.07
0.3	0.31	0.19	0.35	0.15
0.4	0.40	0.32	0.42	0.27
0.5	0.49	0.49	0.50	0.50
0.6	0.59	0.68	0.58	0.71
0.7	0.71	0.83	0.68	0.85
0.8	0.79	0.90	0.75	0.90
0.9	0.88	0.94	0.83	0.94
1.0	0.90	0.94	0.88	0.96

Table 6.14: The effect of noisy crowd-labeled data for Breast Cancer dataset. Results using decision trees, random forest, K-nearest neighbor and support vector machine for different noise levels. Results were averaged over 100 runs.

6.7 The Effect of Noisy Crowd-labeled Data

It is well known that one of the main purposes of crowd labeling is to train machine learning classifiers. If there is noise in the training data, it can lead to misclassification of the test data. Despite the fact that many researchers have investigated the methods to produce good results with noisy data, still no method beats [Frénay and Verleysen, 2014] the availability of the noise-free or at least less noisy data. In this section, we have conducted a few experiments to see how much noisy data can effect the some simple classifier.

We conducted experiments on UCI datasets, which had features available, we are reporting the results only for the UCI breast cancer dataset. This dataset has 569 instances and 30 features. We used 100 instances for training and the rest were used for testing. We used decision trees, random

forest and k-nearest neighbors. As expected, the results shown in Table 6.14 demonstrate that the lower the noise level in the training set, the better the accuracy.

6.8 Conclusion

We developed a new framework for crowd labeling, which incorporates more parameters than most crowd labeling frameworks. The main idea of our approach is to have a better understanding of the impact of different factors in the crowd labeling scenario. The summary of our contribution is as follows:

1. We have provided a better approach to get high-quality results even in the presence of heterogenous quality labels. The results show that our model has a better and stable performance as compared to the other state-of-the-art methods.

2. We have proposed a better way to evaluate the labelers by considering most underlying parameters (instance difficulty, prevalence, question clarity) that can affect the labeler ability.

3. We introduced fine grained labeler ability, which captures the bias of labeler. It also identifies the lazy labelers who label the dataset with only one class to avoid mental effort, hoping to produce good results due to the skewness of the data.

4. The prevalence of the class is introduced to incorporate as much information available about the dataset.

5. The clarity of the question is introduced to quantify the possibility that labeler mistakes could be due to vague or incomplete description of the task.

We have presented a new methodology with empirical evaluation showing good results. Next, we plan to explore theoretical aspects and guarantees of our approach. We also plan to make our approach more fine-grained by adding more parameters to make the model more comprehensive. These parameters include the variability in labeler ability ([Csathó *et al.*, 2012], [Boksem *et al.*, 2005], [Topi *et al.*, 2005]) and pseudo guessing parameter.

Part III

Related Work & Conclusion

Chapter 7

Research On Crowd Labeling

In this chapter, we summarize crowd labeling research. In Section 7.1, we describe the research on improving crowd labeling design and in Section 7.2 the research for quality assurance. It can however be noted that the underlying aim of both types of research is to improve the accuracy of final labels.

7.1 Crowd Labeling Design Related Research

Paper by [Quinn and Bederson, 2011] summarizes human computation. The authors also give an overview of the closely related fields of human computation including data mining, social computing, crowdsourcing and collective intelligence. They discuss various schemes for classification, design and quality control of the human computation task. It is a good introductory paper about human computation but the discussion in this paper gives a high-level picture of human computation and relevant fields but lacks detail for a more curious reader.

7.1.1 Crowd Labeling Workflow

Work by [Little *et al.*, 2010a] discusses possible workflows for crowd. The authors classify the crowdsourcing task into two categories: decision task and creation task. The examples of decision task include labeling of images and annotation of words while the examples of creation task include writing an essay and designing a logo. The authors suggest that parallel workflow is more suitable for decision tasks while iterative workflow is more appropriate for the creation task. They also

mention that a combination of the two workflows can be used for most of the tasks. This paper presents a comparison of the different workflows for crowdsourcing tasks but lacks the details about the number of required workers in each kind of workflow.

7.1.2 Effects of Clarity of Instructions

Similarly, [Kittur *et al.*, 2008] show the importance of a clear instructions through a case study on Amazon Mechanical Turk (AMT). For this purpose, they repeat an experiment on AMT a second time with more detailed instructions. Their results show an improved outcome in the second experiment. They argue that good design and clear instructions can improve the accuracy of the crowdsourcing tasks. This paper brings up a very good point for the requesters to consider. This is especially helpful because many crowd workers may not know the language of the instruction very well. This is the reason why we have introduced a question clarity parameter to quantify the effect of clear instructions. Although the paper is very interesting but lacks extensive experiments for covering different types of crowdsourcing as well as concrete guidelines about the instruction design.

7.1.3 Task Division Strategy

The strategy for task division for crowd work is discussed by [Kulkarni *et al.*, 2011]. They describe that the requester can post the undivided task on AMT through *Turkomatic*. The workers are instructed to do the task in the given amount of time and price or divide the task into smaller parts. These subtasks are automatically posted again with similar instructions and this iterative process goes on until workers complete the task. At the end, workers combine the solutions to make one final solution. The proposed strategy can alleviate complaints of unfairness by the crowd. It can also lead to less work for the requester. On the other hand, this strategy gives control to the workers who can exploit the requester by maliciously dividing the tasks into undesirably small subtasks hence reducing the benefits of using the crowd. Also, it is more difficult to keep a check on the workers and know their quality. Moreover, it may not always be possible to use this strategy for task division.

7.1.4 Task Designing Toolkit

In their paper, [Little *et al.*, 2009; Little *et al.*, 2010b] propose a toolkit for deploying crowdsourcing tasks. This toolkit is an extension of Javascript. It is an appealing idea for the requesters with programing skills since the task can be controlled easily by changing the script as needed. Also it provides a mechanism for storing the results that can safeguard from loss of data in the case of crash. This toolkit provides an easy way to control the crowd labeling task but requesters with no or little programming skills need to hire a programmer. Moreover, this procedure may not be suitable for all kind of tasks e.g., designing a logo.

7.1.5 Task Assignment According to Worker Expertise

The authors in [Ho *et al.*, 2013] propose a method for assigning the tasks according to the worker expertise. They test the workers on a few gold standard instances to calculate the task value (i.e., quality) of each worker, for each type of instance separately. Instances from the unlabeled dataset are assigned to the workers with the highest task value. The method proposes an intelligent way of improving accuracy by using the suitable of worker for each type of the instance but the method requires using extra workers for exploration purpose. This results in extra cost, which may reduce the advantage obtained by the improvement in accuracy.

7.1.6 Solving Worker's Problems

Work by [Silberman *et al.*, 2010b] presents the problems faced by workers and enlist some open questions in this regard. Worker problems described include low pay, long pay delays, unaccountable and seemingly arbitrary rejections, prohibitive time limits, uncommunicative requesters and administrators, cost of requesters error borne by the workers and fraudulent tasks. Although this paper presents the problems faced by the workers to improve the crowd work process, no suggestion about solution is provided.

Works by [Bederson and Quinn, 2011] and [Silberman *et al.*, 2010a] propose some solutions to worker problems. The solutions include defining hourly pay, disclosing and following payment terms, valuing workers time, immediate quality feedback, long-term feedback, providing grievance process, providing task context and limiting anonymity of requesters. They provide a good initiative

to propose solution to worker problems. While these solutions can help to mediate worker problems, they need to be enforced by making laws about crowdsourcing e.g., defining worker pay and limiting anonymity of the requesters.

A more practical solution to worker problems is suggested by [Irani and Silberman, 2013] named *Turkopticon*, which is an extension of chrome and firefox used to get workers' reviews about requesters. This method helps the workers to know about the requesters beforehand. Moreover, it can help the requesters to get feedback about themselves and improve accordingly. This solution is easy to use and helpful for both requesters and workers. But problem can be caused by the workers giving wrong feedback.

7.1.7 Crowd Labeling Surveys

Work by [Ross *et al.*, 2010] gives an overview of the crowd demographics. The information is useful and should be kept in mind while designing the tasks. Since the crowd workers come from different parts of the world with different cultural and social background, their perception about the same problem can be quite different. Although being an outdated paper it presents a good example of demographics summary. Such surveys need to be done yearly.

7.1.8 Standardizing Crowd Labeling

The authors in [Ipeirotis and Horton, 2011] suggest to standardize crowd labeling by introducing design templates, fixed prices for similar tasks, pricing the smaller units, deciding the complex unit prices accordingly and optimizing the workflow. They also suggest improvement in the role of platforms to avoid fake or malicious tasks. In general, idea of standardization is good to make the rules uniform across the platforms and minimize the exploitation of workers. The strategies proposed by the authors can be applied to certain extent and can improve the overall structure of crowd labeling.

7.1.9 Crowd Labeling Career Ladder

In their paper [Kittur *et al.*, 2013] propose possible future directions for crowd work design, crowd computation and crowd workers. While most of the ideas discussed in this paper are not totally new, one of the novel suggestions is the career ladder. They describe career ladder as different ranks

assigned to workers according to their experience e.g., entry level worker, trusted worker, hourly contractor and employees. Other suggestions include task recommender systems and improving task design through a better communication between the crowd and the requesters. Career ladder suggested in this paper is a nice way to envision future crowd structure. It can motivate the crowd to take crowdsourcing more seriously, portraying crowd work as a reasonable and a real job opportunity. Some of the suggestions are not easy to implement e.g., since the crowd workers are usually not permanent, the identification of crowd workers based on their credentials and previous work history, is not easy and requires link across all the platforms. Even in the case of one platform much effort is required to check and verify the identity of the workers, while preserving their privacy.

7.1.10 Our Task Design

Although the above mentioned suggestions by different researchers provide good solutions for designing a better crowd labeling task, in our frameworks we have approached the problem in a different way, described as follows:

- We have designed the task to be able to get high accuracy with less preprocessing and minimum infrastructure.
- Our strategy is also useful for labels acquired in the past.
- We do acknowledge the importance of clear instructions and that is why we have included clarity of the question parameter in our latest methodology.
- We do not block oppositional/malicious workers rather use the information provided by them.

7.2 Crowd Labeling Research about Quality Assurance

Many recent works have addressed the topic of *learning from crowd* along with quality assurance techniques (e.g., [Raykar *et al.*, 2010; Le *et al.*, 2010]). In this section, we present some research with respective pros and cons.

7.2.1 Effects of Acquiring Multiple Labels

In their paper, [Sheng *et al.*, 2008] show different traits of multiple labels and majority voting through a set of experiments. They show that for a uniform quality labelers with quality $p > 0.5$ multiple labels improve accuracy, when $p < 0.5$ accuracy deteriorates, while no improvement is observed when $p = 0.5$ or $p = 1$. Similarly, for the crowd with variable quality it is shown that increasing the number of labels may not always be helpful and similar results may be obtained by a single label. They also propose *Selective repeated-labeling*, which refers to the procedure of getting more labels for the instances with mixed multi-set of labels. Since the mixed multi-set can be due to the labeler quality or model, they also introduce a score called *Labeler and Model Uncertainty score.* This score is used to decide whether acquiring more labels for the instance is helpful or not. They present a nice comparison of single labeling, multiple labeling and selective labeling but they only experiment with a naive majority voting method while more intelligent methods are available with a better accuracy.

7.2.2 Natural Language Processing (NLP) tasks

The paper by [Passonneau *et al.*, 2012] experiments on Natural Language Processing (NLP) tasks. They show through experiments that fine grained sense inventory produces better results. Same labeling accuracy can be achieved by many untrained labelers, few trained workers and one expert labeler. They also experiment to identify the instances with high agreement, average agreement and split agreement. They present extensive experiments with different level of labeler expertise but they focus on few words. In their paper [Snow *et al.*, 2008] experiment on five different NLP tasks using experts and non-experts. The method used is inter annotator agreement (ITA). They show that on average four non-experts can do as good as one expert for these tasks. They also introduce a bias recognition technique, which automatically adjusts the biased labels. They use different variety of NLP tasks to experiment. They do not compare ITA with other methods.

Work proposed by [Passonneau and Carpenter, 2013] presents a case study to show deficiencies of inter annotator agreement (ITA). They argue that ITA is based on pairwise comparison while comparing one annotator to the average of the rest is a better option. Moreover, the difficulty of the instance may increase the agreement on the wrong label. Also some annotators can be biased, which can increase the wrong label agreement. Their conclusion is that learning a model can be

a better option. They also propose a model. A comprehensive overview is presented about inter annotator agreement shortcomings but no general guidelines are provided about forming a model, only a specific model is presented.

7.2.3 Classifier Based Methods

Support Vector Machines (SVM) to learn a classifier based on a few labels provided by the crowd is presented by [Dekel and Shamir, 2009]. In this paper no repeated labeling is used, instead good labelers are identified and their labels are used to learn the classifier ignoring the labels by the bad labelers. Multiple labels are not needed. They make the assumption that good labelers are always available who can provide correct labels irrespective of the variable quality of the instances but this method may not do well if no good labeler is available.

Another approach aims to identify adversarial labelers (e.g., [Paolacci *et al.*, 2010]). This is tackled through an a priori identification of those labelers before the labeling task starts. However, an oppositional/malicious labeler can perform well initially and then adversarially behave during the labeling process.

7.2.4 EM Based Method

In their paper, [Dawid and Skene, 1979] use Expectation Maximization (EM) for learning the underlying parameters and latent variables of the crowd labeling task. They propose methods that apply, whether a few ground truth instances are available or not available. The authors introduce for the first time the use of EM for crowd labeling but they do not consider the difficulty of the instance and handle all types of labelers in the same way.

A probabilistic model called Generative model of Labels, Abilities, and Difficulties (GLAD) is proposed by [Whitehill *et al.*, 2009]. In their model, EM is used to obtain maximum likelihood estimates of the unobserved variables, which outperforms majority voting. The authors also propose a variation of GLAD that *clamps* some known labels into the EM algorithm. More precisely, clamping is achieved by choosing the prior probability of the true labels very high for one class and very low for the other. The idea used for aggregation of labels based on the expertise of the labeler and difficulty of the instance is valuable. Their method is shown to outperform majority voting (which is the case for most state-of-art methods) but they do not compare to other methods, except for Dawid

& Skene [Dawid and Skene, 1979]. Also, the assumptions they make about accuracy level of good and bad workers are too general and do not cover the extreme cases where the workers are really biased or oppositional/malicious.

A probabilistic framework is also proposed by [Yan *et al.*, 2010] as an approach to model annotator expertise and build classification models in a multiple label setting. They do not explicitly model the difficulty of the instance, instead they use variable expertise of the labeler i.e., labeler expertise varies according to instances. The notion of variable expertise of the labeler is a realistic approach since labeler's performance changes according to the instance. M-step of the EM does not have a closed form so Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) Quasi Newton method is used. Calculations are complex and the method is difficult to implement.

A generative model is proposed by [Hovy *et al.*, 2013b] that can detect the spammers. The spamming behavior of the labeler is modeled by a binary variable and EM is used to learn the underlying parameters. While their method focuses on identifying spammers from non-spammers, they make a very strong assumption i.e., when a labeler is not spamming, he can produce the correct label. This is not true in general. Moreover, labelers do not change their strategy for each instance. Mostly their behavior is consistent at least during one set of task. The methods used in this paper are EM and Variational Bayes (VB). The model presented in this paper is simple but the authors make wrong assumption about the labeler's behavior.

Paper by [Raykar and Yu, 2012] proposes a strategy based on (a) using the good labelers, (b) identifying the biased and oppositional/malicious labeler and adjusting their labels and (c) pruning the random labelers. Their algorithm updates the sensitivity and specificity using the MAP estimator given the hyper parameter. Hyper parameter is also updated iteratively and penalizes the spammers more than the other labelers. A labeler for which the value of hyper parameter is higher than a predefined threshold is pruned. They also extend their method to multi-class and categorical data. The main strength of their paper is that they try to maximize the use of all kinds of workers and automatically prune the noisy/random workers but the method complexity makes it difficult to implement.

7.2.5 Iterative Methods

An iterative method is proposed by [Karger *et al.*, 2014; Karger *et al.*, 2011], which is similar to Belief Propagation. Iterative algorithm improves the worker estimates by comparing the workers contribution to the other workers. This algorithm is simple to implement and requires no prior knowledge. But this method works well if most of the workers in the crowd are good but in the presence of more biased or oppositional/malicious workers, the accuracy goes down.

Belief Propagation (BP) and Mean Field method (MF) is used in [Liu *et al.*, 2012] for estimating the true labels and expertise of the labelers. Posterior distribution is marginalized over the expertise of the labeler. Priors used are Beta prior, discrete prior, Haldane prior, and deterministic prior. They show that Karger's method [Karger *et al.*, 2011], EM and majority voting are special case of their method i.e., their method is more general. On the other hand method is complex and involves lots of calculations.

7.2.6 Ground Truth Based Methods

The approach adopted in CrowdFlower [Le *et al.*, 2010] suggests the use of the gold standard to train and test the workers before the actual labeling task and blocking the workers who do not fulfill a predefined standard. It is also suggested that gold units be embedded in the labeling task without the knowledge of workers to keep a check on their performance. This method works well in most cases if ground truth instances are available. Although this method seems promising, it can discourage the workers and hinder the new workers from learning through experience. Moreover, the need of large number of gold units is a big challenges. Similarly, the idea of using ground truth labels has been used by Crowdflower ([Le *et al.*, 2010]) where crowd ability is tested based on a few ground truth instances. This proposed approach tests the crowd labelers during the training phase (before the actual labeling starts) and blocks the labelers who do not pass the training. Subsequent tests are also used to block bad crowd labelers after giving warnings. This is done by injecting instances for which ground truth is available during the actual labeling task. This approach can be helpful when a large number of ground truth instances are available. To handle this problem [Oleson *et al.*, 2011] propose "Programmatic gold" that generates gold units automatically which may not be possible for many datasets.

Another method called *Programmatic Gold* is proposed by [Oleson *et al.*, 2011], which gener-

ates gold units. This is done either by creating gold units through injecting known type of errors into instances or using the data, which has been labeled by the crowd with high confidence. This approach can be a successful in some areas e.g., event temporal ordering. But it should be noted that this approach cannot be applied to all types of crowd labeling tasks e.g., data for cancer diagnosis cannot be created by injecting errors.

Paper by [Wang *et al.*, 2011] claims to identify the types of labelers and adjust their labels accordingly. They do not rely on the exact labels given by the labelers but convert the hard labels into soft labels, using the underlying information about the true labels. Moreover, they also propose an active learning strategy, which compares the utility of testing the worker with gold truth instance and the utility of assigning him an unlabeled instance. Decision is made based on whichever has a higher utility. Converting the hard labels to soft labels gives a better insight to labelers strategy or inclination. In this method prevalence of class and confusion matrices of labelers are learned by comparing each worker's labels to the majority voting of the rest of the crowd. When the majority of labelers is bad, there is a high chance of wrong perception of class prevalences and confusion matrices. This problem can be alleviated by using a few ground truth instances.

Another paper [Welinder *et al.*, 2010b] devises a method to identify the class of the image, using multiple labels. They use different attributes of the image to model the difficulty of the instance. Further, they add noise to these attributes, which represents the image as seen by each labeler due to the quality of the image and the expertise of the labeler. After forming the probabilistic model, alternating optimization is done using gradient ascent method to learn the model. Finally a classifier is learned for each worker. The idea used for determining the difficulty of the instance using the attributes is realistic and covers different aspects of the instance instead of just the notion of 'difficulty'. Moreover the classifier learned for each labeler depends on each of these attributes. Since different workers tend to focus on different attributes of the instances, the evaluation of workers can be more accurate. The method could benefit from a few ground truth instances, which are not used here.

7.2.7 Active Learning Based Methods

A second line of research (e.g., [Donmez *et al.*, 2009]) uses active learning to increase labeling accuracy by choosing the most informative labels. This is done by constructing a confidence interval

called "Interval Estimate Threshold" for the reliability of each labeler. Also, [Yan *et al.*, 2011] develop a probabilistic method based on the idea of active learning, to use the best labels from the crowd.

Wallace et al. [Wallace *et al.*, 2011] propose a method called MEAL (Multiple Expert Active Learning). Labelers are grouped together based on some given information, each group has its rank. The lowest ranked group of labelers label the instances as {1, -1, 'difficult'} . The instances labeled as 'difficult' are passed on to the next level of workers. The top ranked labelers are not allowed to label any instance as 'difficult'. This procedure is repeated until the budget is exhausted. Labelers can explicitly identify the instances about which they are doubtful, instead of labeling them randomly. Only the instances which are labeled as 'difficult' are labeled by expert and expensive labelers from a higher ranked group, reducing the cost. Although having information about all the labelers beforehand to be able to rank them, is unrealistic.

Some researchers [Donmez and Carbonell, 2008] claim to remove the wrong assumptions made by active learning that is oracles are never wrong, always answer, are free of cost or have uniform cost and there is only one oracle. They present three different scenarios with two oracles each. These oracles have different qualities e.g., reliable, reluctant, uniform cost, variable cost etc. For each scenario an algorithm is proposed to choose the best oracle depending on the cost and probability of getting the correct answer. The method suggested to calculate the utility of each oracle is helpful but extra work is required to decide about the oracle.

7.2.8 Classifier Based Methods

A Bayesian framework is proposed by [Raykar *et al.*, 2009] to estimate the ground truth and learn a classifier. The main novelty of their work is the extension of the approach from binary to categorical and continuous labels. [Sheng *et al.*, 2008; Sorokin and Forsyth, 2008; Snow *et al.*, 2008] show that using multiple, noisy labelers is as good as using fewer expert labelers.

More recent works have proposed approval voting and incentivizing the crowd ([Shah *et al.*, 2015; Shah and Zhou, 2015; Shah *et al.*, 2013]). This approach is good but requires longer time and infrastructure to get good results. Similarly another recent work [Zhang and Chaudhuri, 2015] relies on active learning and [Menon *et al.*, 2015] use class-probability estimation to study learning from corrupted binary labels.

Another proposed model by [Ipeirotis *et al.*, 2010], and [Ipeirotis and Paritosh, 2011] identifies biased or adversarial labelers and corrects their assigned labels. This is done by replacing *hard labels* by a *soft labels*. Class priors and the probability of a labeler assigning an instance from a particular class to some other class is used to calculate the soft labels. [Ipeirotis and Horton, 2011] suggest to standardize crowd labeling by introducing design templates, fixed prices for similar tasks, pricing the smaller units, deciding the complex unit prices accordingly and optimizing the workflow. They also suggest improvement in the role of platforms to avoid fake or oppositional/malicious tasks. In general, the idea of standardization is good to make the rules uniform across the platforms and minimize the exploitation of workers but do not help with the increasing variety of crowd labeling tasks.

7.2.9 Our Approaches

We have presented ELICE and CLUBS. Both methods are based on parameter estimation using a few expert-labeled instances. ELICE uses frequentist approach while CLUBS uses the Bayesian approach. The empirical evaluation shows that both of the approaches perform better than many state-of-the-art methods.

Chapter 8

Conclusion & Future Directions

In this chapter, we conclude the thesis by summarizing our efforts and presenting future directions.

8.1 Conclusion

In this thesis, we have presented a set of methods for improving crowd labeling. We have focused on devising methods for estimating parameters and using them in label aggregation. Parameters estimation is done using expert-labeled instances. We have explored frequentist and Bayesian approaches for parameter estimation.

Our frequentist approach called ELICE consists of three versions along with their variants. Each version of ELICE has its own advantages. ELICE 1 is simple to implement, fast to get results and delays phase transition. It is a very good option when the task is easy and we expect to have lots of good labelers. ELICE 1 has a clustered-based variant in which before the random selection of instances for getting expert-labels, clusters are formed and an equal number of instances are selected randomly from each cluster separately. This is done to possibly get an equal representation of instances from both classes.

ELICE 2 is a more sophisticated version of ELICE, which is based on entropy. Its ability to identify the labeler type and deal with it accordingly results into good performance. In the label aggregation step, for a given instance, low weight is assigned to the label provided by a random labeler and high weight is assigned to the label provided by a good or oppositional labeler but at the same time, the label provided by the oppositional labeler is automatically flipped. This technique

is especially helpful when there are many oppositional labelers in the crowd. In this method, the information provided by the oppositional labelers is not wasted, rather it is decoded and harnessed intelligently, resulting in high accuracy of the final labels. It should be noted that when the set of crowd labelers is good it is easy to get good results but the challenge arises when we have oppositional labelers. ELICE 2 has this ability to deal with oppositional labelers and works best as compared to many state-of-the-art methods even when all labelers are oppositional. With the increasing trend of maliciousness on internet, this is a highly desirable property and makes ELICE 2 unique. ELICE 2 also has a cluster-based variant.

Although ELICE 2 is very effective, it is conditioned on the availability of ground truth, while it is a known fact that expert-labels may not always be ground truth, either because of the task being so tricky or because the experts disagree on one label. ELICE 3 with pairwise comparison mediates this problem by comparing labeler to labeler and instance to instance, other than relying on expert labels. Therefore, ELICE 3 squeezes all available information from all sources including experts, crowd, and instances. ELICE 3 with circular comparison is also presented, which is similar to ELICE 3 with pairwise comparison but with lower computational cost.

Our empirical evaluation has shown that ELICE is a robust framework as compared to the state-of-the-art. It also has a universal application as it covers different labeling scenarios including the presence of oppositional/malicious labelers and unavailability of ground truth from an expert. Other than being effective, it is very efficient and can be used for large dataset. It is also cost efficient because of minimum preprocessing of data, minimum infrastructure for labeling and no history tracking or blocking of the labelers.

We also have derived a theoretical lower bound for the number of expert-labels needed to achieve good accuracy. Our derivation is based on PAC learning framework. We have shown the utility of our theoretical lower bound through experiments.

After exploring the frequentist approach, we developed a Bayesian approach for parameter estimation called CLUBS. Our Bayesian approach is called CLUBS. An important characteristic of CLUBS is covering more aspects of crowd labeling scenario by introducing new parameters, such as clarity of the question, prevalence of class and per-category ability of the labeler. CLUBS in most cases has shown good results compared to state-of-the-art. Although sometimes ELICE outperformed CLUBS, we believe that the true potential of CLUBS can further be explored.

We hope our contribution will prove to be useful to the crowd labeling community. In the first chapter of this thesis, we had initiated a few unresolved questions. We conclude the thesis by analyzing how many of these questions have been resolved.

8.1.1 Unresolved Questions Revisited

1. What are the best ways to evaluate labeler ability and instance difficulty?

   Discussion: In this thesis, we have used two different approaches to evaluate the labelers and instances. In the first part of this thesis, we used a frequentist approach to estimate these parameters. In the second part of the thesis, we explored the Bayesian approach to estimate not only labeler ability and instance difficulty but we also estimated more advanced parameters. While it cannot be easily determined what is the best way to evaluate the parameters but the empirical evaluation shows that our procedures are very helpful in attaining higher accuracy as compared to the other state-of-the-art methods.

2. Can phase transition be handled in a more effective way?

   Discussion: Our experiments show that ELICE successfully has been able to delay phase transition, something which many state-of-the-art methods were unable to do.

3. It is common to use expert-labeled instances or ground truth to evaluate labelers and instances [Le *et al.*, 2010; Khattak and Salleb-Aouissi, 2011; Khattak and Salleb-Aouissi, 2012; Khattak and Salleb-Aouissi, 2013]. The question is, how many expert-labeled instances should be used in order to obtain an accurate evaluation?

   Discussion: We have used expert-labeled instances to evaluate the labelers and instances. Chapter 4 provides the theoretical work regarding the number of expert-labeled instances needed to attain a certain final-label accuracy.

4. How can labelers and instances be evaluated if ground truth is not known with certitude?

   Discussion: To be able to address this problem, we introduced ELICE 3 with pairwise and circular comparison. In this method not only the expert labeled instances (which are not necessarily ground truth) are used to evaluate the parameters but also maximum information is extracted using instance to instance and labeler to labeler comparison.

5. Is there any optimal way to combine multiple labels to get the best labeling accuracy?
   Discussion: In our methodologies, multiple labels are combined using weighted majority voting while weights are calculated using logistic function of labeler and instance related parameters. The purpose of calculating the weights in this way is to combine all the characteristics of the labeling scenario and give according validity to each crowd label. The experiments presented in this thesis show that our label aggregation method is able to produce more accurate results.

6. Should the labels provided by oppositional labelers be discarded and blocked? Or is there a way to use the "information" provided by oppositional labelers?
   Discussion: Despite the fact that blocking the oppositional labelers has been one of the recommended solutions to avoid polluted results, ELICE has shown to work really well by using the information provided by the oppositional labelers. This shows that such labels can have high value if handled intelligently hence diminishing the need for blocking oppositional labelers and searching for better labelers.

8.2 Future Directions

We plan to advance our Bayesian approach by making it exhaustive in terms of parameters. Our future plans for extending CLUBS are described in the Section 8.2. We are currently working to explore, how the labeler performance can vary due to the amount of time spent on a task causing fatigue, boredom and disinterest [Csathó *et al.*, 2012]. The goal is to incorporate this idea in the CLUBS to get better results.

8.2.1 Variability in Labeler Productivity

Most of the crowd labeling literature assumes that labeler performance remains constant (at least) during one labeling session. But in fact humans unlike computers and other machines have variable performance even during a short period of time ([Boksem *et al.*, 2005; Topi *et al.*, 2005]). The variation in performance can be due to fatigue, boredom and/or other external reasons [Albert, 2002; Jensen *et al.*, 2009].

Common reasons for variability in human performance are as follows [Loukidou, 2008]:

- **Time of the day:** The time of the day can have an important effect on human performance. Generally, work done at late hours will exhibit low performance, which will soon become worse.

- **Complexity of the work:** The complexity of the task can have a direct effect on the performance of the human. More complex task result in early deterioration in the worker's performance.

- **Lack of challenge:** If the task has no challenge for the human then it can cause boredom and as a result can lower the performance level.

- **Stress level:** According to the Yerkes-Dodson law [Yerkes and Dodson, 1908] of psychology the amount of stress also affects the performance level.

- **Time-on-Task (ToT):** The time spent on the task also causes variability in performance. Generally, human performance can be divided into three phases [Csathó *et al.*, 2012]: warmup phase, peak phase, and decline phase.

We are currently working with the data with some ground truth from different temporal frames to be able to understand and predict the labeler performance pattern.

8.3 Crowd Labeling Future: The Broader Picture

In the past decade crowd labeling research has made significant progress. Following is a glimpse of a broader picture of crowd labeling future as envisioned by different researchers.

- **Standardizing the Crowd Work:** To bring uniformity in crowd labeling, tasks can be standardized [Ipeirotis and Horton, 2011] by introducing design templates, fixed prices for similar tasks, pricing the smaller units, deciding the complex unit prices accordingly and optimizing the workflow.

- **Improving Role of the Platform:** Improvement in the role of the platforms [Ipeirotis and Horton, 2011] to avoid fake or oppositional/malicious tasks is of crucial importance. Plat-

forms should be designed in a way to be able to check and verify the identity of the workers, while preserving their privacy.

- **Task recommender system:** Task recommender systems should be built to suggest work according to the work history of the workers [Kittur *et al.*, 2013].

- **Worker Career Ladder:** Career ladder [Kittur *et al.*, 2013] is a nice way to envision future crowd structure. Career ladder consists of ranks assigned to workers according to their experience e.g., entry-level worker, trusted worker, hourly contractor and employees.

- **Collaboration and Monitoring:** Collaboration and real time interaction with the requesters and other workers can help them both feel more active and engaged in the work. In this regard survey from workers can also be helpful.

- **Educating Requesters and Workers:** This may include platforms educating requesters about task design, job assignment, training-assessment cycle for better learning, and online crowd work tutoring system. Workers can be encouraged by giving rewards for good performance.

Further investigation needs to be done to explore more opportunities for reaching out to the crowd. We conclude this thesis with the following thoughts.

8.4 Final Thoughts

- **What else a crowd can do?** The crowd may not be kept limited to traditional uses but also can be utilized for non-traditional and more challenging tasks. Further investigation needs to be done to explore more opportunities for the crowd usage.

- **Will there always be a crowd?** It is debatable whether a crowd will be available for crowd work after 50 years. More people can be attracted to crowd labeling if it is recognized as a profession with more incentives and opportunities.

- **Can crowd be replaced?** A last worth-considering question is whether advances in technology will ever replace the crowd. For example, computers might be able to interpret images

better, or may provide more reliable tools for language translation. Hence, it is debatable whether one will still reach out to the crowd in 50 years from now.

Part IV

Bibliography

Bibliography

[Aggarwal, 2016] Anupama Aggarwal. Detecting and mitigating the effect of manipulated reputation on online social networks. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 293–297, 2016.

[Ahn *et al.*, 2008] Luis von Ahn, Benjamin Maurer, Colin Mcmillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[Albert, 2002] Jim. Albert. *Smoothing Career Trajectories of Baseball Hitters*. Technical report., 2002.

[Arthur and Vassilvitskii, 2007] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[Asuncion and Newman, 2007] A. Asuncion and D.J Newman. University of california irvine (UCI) Machine Learning Repository. In *School of Information and Computer Sciences*, 2007.

[Bederson and Quinn, 2011] Benjamin B. Bederson and Alexander J. Quinn. Web workers unite! addressing challenges of online laborers. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, pages 97–106, New York, NY, USA, 2011. ACM.

[Boksem *et al.*, 2005] Maarten A. S. Boksem, Theo F. Meijman, and Monicque M. Lorist. Effects of Mental Fatigue on Attention: An ERP study. *Cognitive Brain Research*, 25(1):107–116, Sep 2005.

[Bradley and Terry, 1952] Ralph Allan Bradley and Milton Terry. Ranking analysis of incomplete block design: I. the method of paired comparisons. In *Biometrika*, pages 324–345, 1952.

[Byrt *et al.*, 1993] Ted Byrt, Janet Bishop, and John B. Carlin. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46:423–429, 1993.

[Carpenter, 2008] Bob Carpenter. Multilevel bayesian models of categorical data annotation. Available at http://lingpipe-blog.com/ lingpipe-white-papers, 2008.

[Choi *et al.*, 2016] Hongkyu Choi, Kyumin Lee, and Steve Webb. Detecting malicious campaigns in crowdsourcing platforms. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, pages 197–202, 2016.

[Cooper *et al.*, 2010] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting Protein Structures with a Multiplayer Online Game. *Nature*, 466(7307):756–760, 2010.

[Csathó *et al.*, 2012] Árpád Csathó, Dimitri van der Linden, István Hernádi, Péter Buzás, and Ágnes Kalmár. Effects of mental fatigue on the capacity limits of visual attention. *Journal of Cognitive Psychology*, 24(5):511–524, Aug 2012.

[Dalvi *et al.*, 2004] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 99–108, New York, NY, USA, 2004. ACM.

[Dawid and Skene, 1979] A. P. Dawid and A. M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28:20–28, 1979.

[Dekel and Shamir, 2009] Ofer Dekel and Ohad Shamir. Good learners for evil teachers. In *International Conference on Machine Learning (ICML)*, page 30, 2009.

[Doan *et al.*, 2011] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, 54(4):86–96, April 2011.

[Donmez and Carbonell, 2008] Pinar Donmez and Jaime G. Carbonell. Proactive learning: Cost-sensitive Active Learning with Multiple Imperfect Oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 619–628, New York, NY, USA, 2008. ACM.

[Donmez *et al.*, 2009] Pinar Donmez, J. G. Carbonell, and J. Schneider. Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling. In *Knowledge Discovery and Data Mining (KDD)*, pages 259–268, 2009.

[Frénay and Verleysen, 2014] Benoît Frénay and Michel Verleysen. Classification in the Presence of Label Noise: A Survey. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 25(5):845–869, May 2014.

[Gadiraju *et al.*, 2015] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015*, pages 1631–1640, 2015.

[Hellerstein and Tennenhouse, 2010] Joseph M. Hellerstein and David L. Tennenhouse. Searching for Jim Gray: A Technical Overview. Technical Report UCB/EECS-2010-142, EECS Department, University of California, Berkeley, Dec 2010.

[Ho *et al.*, 2013] Chien-ju Ho, Shahin Jabbari, and Jennifer W. Vaughan. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 534–542. JMLR Workshop and Conference Proceedings, 2013.

[Hovy *et al.*, 2013a] Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning Whom to Trust with MACE. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), Atlanta, Georgia*, 2013.

[Hovy *et al.*, 2013b] Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. Learning whom to trust with MACE. In *HLT-NAACL*, pages 1120–1130. The Association for Computational Linguistics, 2013.

[Huang *et al.*, 2006] Tzu-Kuo Huang, Chih-Jen Lin, and Ruby C. Weng. Ranking individuals by group comparisons. In *International Conference on Machine Learning (ICML)*, ICML '06, pages 425–432, New York, NY, USA, 2006.

[Ipeirotis and Horton, 2011] Panagiotis G Ipeirotis and John J Horton. The Need for Standardization in Crowdsourcing. In *CHI 2011 Workshop on Crowdsourcing and Human Computation*, 2011.

[Ipeirotis and Paritosh, 2011] Panagiotis G. Ipeirotis and Praveen K. Paritosh. Managing Crowdsourced Human Computation: A tutorial. In *WWW (Companion Volume)*, pages 287–288, 2011.

[Ipeirotis *et al.*, 2010] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality Management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2010.

[Irani and Silberman, 2013] Lilly C. Irani and M. Six Silberman. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 611–620, New York, NY, USA, 2013. ACM.

[Jagabathula *et al.*, 2014] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2492–2500, 2014.

[Jensen *et al.*, 2009] Shane T. Jensen, Blakeley McShane, and Abraham J. Wyner. *Hierarchical Bayesian modeling of hitting performance in baseball.* Number 191–212. Bayesian Analysis, 4, 2009.

[Karger *et al.*, 2011] David Karger, Seewong Oh, and Devavrat Shah. Iterative Learning for Reliable Crowdsourcing Systems. In *Neural Information Processing Systems (NIPS), Granada, Spain*, 2011.

[Karger *et al.*, 2014] David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research*, 62(1):1–24, 2014.

[Khattak and Salleb-Aouissi, 2011] Faiza Khan Khattak and Ansaf Salleb-Aouissi. Quality Control of Crowd Labeling through Expert Evaluation. In *Neural Information Processing Systems (NIPS) 2nd Workshop on Computational Social Science and the Wisdom of Crowds, Granada, Spain.*, 2011.

[Khattak and Salleb-Aouissi, 2012] Faiza Khan Khattak and Ansaf Salleb-Aouissi. Improving Crowd Labeling through Expert Evaluation. In *2012 AAAI Symposium on the Wisdom of the Crowd*, 2012.

[Khattak and Salleb-Aouissi, 2013] Faiza Khan Khattak and Ansaf Salleb-Aouissi. Robust Crowd Labeling using Little Expertise. In *Sixteenth International Conference on Discovery Science, Singapore*, 2013.

[Khattak and Salleb-Aouissi, 2015] Faiza K. Khattak and Ansaf Salleb-Aouissi. An Item Response Theory (IRT) Like Approach to Crowd-labeling. In *Workshop for Women in Machine Learning (WiML 2015) held in conjunction with Neural Information Processing Systems (NIPS), Montreal, Canada. 2015.*, 2015.

[Khattak and Salleb-Aouissi, 2016] Faiza K. Khattak and Ansaf Salleb-Aouissi. *Robust Crowd Labeling using Expert Evaluation and Pairwise Comparison*. Submitted to Journal of Artificial Intelligence (JAIR). Special Track on Human Computation and AI, 2016.

[Khattak *et al.*, 2016a] Faiza K. Khattak, Ansaf Salleb-Aouissi, and Anita Raja. Crowd labelers as students: An item response theory (IRT) like approach to crowd-labeling. In *Double-blind submission*, 2016.

[Khattak *et al.*, 2016b] Faiza K. Khattak, Ansaf Salleb-Aouissi, and Anita Raja. An item response theory (IRT) like approach to crowd-labeling. In *Submitted to Collective Intelligence Conference*, 2016.

[Kittur *et al.*, 2008] Aniket Kittur, H. Chi, and Bongwon Suh. Crowdsourcing user studies with Mechanical Turk. In *Proc. CHI 2008, ACM Pres*, pages 453–456, 2008.

[Kittur *et al.*, 2013] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The Future of Crowd Work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, CSCW '13, pages 1301–1318, New York, NY, USA, 2013. ACM.

[Kulkarni *et al.*, 2011] Anand Pramod Kulkarni, Matthew Can, and Bjórn Hartmann. Turkomatic: Automatic, Recursive Task and Workflow Design for Mechanical Turk. In *Human Computation*, 2011.

[Le *et al.*, 2010] John Le, Andy Edmonds, Vaughn Hester, Lukas Biewald, V. Street, and S. Francisco. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Evaluation*, pages 17–20, 2010.

[Lee *et al.*, 2013] Kyumin Lee, Prithivi Tamilarasan, and James Caverlee. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.*, 2013.

[Lee *et al.*, 2014] Kyumin Lee, Steve Webb, and Hancheng Ge. The dark side of micro-task marketplaces: Characterizing fiverr and automatically detecting crowdturfing. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*, 2014.

[Little *et al.*, 2009] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. Turkit: Tools for iterative tasks on Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 29–30, New York, NY, USA, 2009. ACM.

[Little *et al.*, 2010a] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 68–76, New York, NY, USA, 2010. ACM.

[Little *et al.*, 2010b] 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.

[Liu *et al.*, 2012] Qiang Liu, Jian Peng, and Alex Ihler. Variational Inference for Crowdsourcing. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q.Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 701–709, 2012.

[Liu *et al.*, 2016] Yuli Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. Pay me and i'll follow you: Detection of crowdturfing following activities in microblog environment. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3789–3796, 2016.

[Lord and Novick, 1968] Frederic Lord and Melvin R. Novick. *Statistical Theories of Mental Test Scores.* Reading, Mass., Addison-Wesley Pub. Co., 1968.

[Lord, 1952] Frederic Lord. *A Theory of Test Scores*. Psychometric Monograph No. 7, 1952.

[Loukidou, 2008] Evangelia Loukidou. *Boredom in the workplace: A qualitative study of psychiatric nurses in Greece.* Doctoral Thesis, 2008.

[Lynch, 2012] A Lynch. Crowdsourcing is not new-The History of Crowdsourcing (1714 to 2010). In *http://blog.designcrowd.com/article/202/crowdsourcing-is-not-new-thehistory-of-crowdsourcing-1714-to-2010*, 2012.

[Menon *et al.*, 2015] Aditya Krishna Menon, Brendan van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from Corrupted Binary Labels via Class-Probability Estimation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015.

[Mozafari *et al.*, 2014] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proc. VLDB Endow.*, 8(2):125–136, October 2014.

[Nelson, 2008] Sue Nelson. Big data: The Harvard computers. In *Nature*, volume 455, pages 455(7209), 36, 2008.

[Novick, 1966] Melvin R. Novick. The Axioms and Principal Results of Classical Test Theory. *Journal of Mathematical Psychology Volume 3, Issue 1, February 1966*, pages 1–18, 1966.

[Oleson *et al.*, 2011] David Oleson, Alexander Sorokin, Greg P. Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. In *Human Computation*, 2011.

[Paolacci *et al.*, 2010] Gabriele Paolacci, Jesse Chandler, and Panagiotis Ipeirotis. Running Experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, Vol. 5, No. 5:411–419, 2010.

[Passonneau and Carpenter, 2013] Rebecca J. Passonneau and Bob Carpenter. The Benefits of a Model of Annotation. In *The 7th Linguistic Annotation Workshop & Interoperability with Discourse, ACL Workshop, August 8-9*, 2013.

[Passonneau and Carpenter, 2014] Rebecca J. Passonneau and Bob Carpenter. "The Benefits of a Model of Annotation". In *Transactions of the Association for Computational Linguistics.*, 2014.

[Passonneau *et al.*, 2012] Rebecca Passonneau, Vikas Bhardwaj, Ansaf Salleb-Aouissi, and Nancy Ide. Multiplicity and Word Sense: Evaluating and Learning from Multiply Labeled Word Sense Annotations. In *Language Resources and Evaluation, 2012*, 2012.

[Quinn and Bederson, 2011] Alexander J. Quinn and Benjamin B. Bederson. Human computation: A survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 1403–1412. ACM, 2011.

[Raykar and Yu, 2012] Vikas C. Raykar and Shipeng Yu. Eliminating Spammers and Ranking Annotators for Crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012.

[Raykar *et al.*, 2009] Vikas Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to Trust When Everyone Lies a Bit. In *International Conference on Machine Learning (ICML)*, pages 889–896, 2009.

[Raykar *et al.*, 2010] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo H. Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, April 2010.

[Ross *et al.*, 2010] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: Shifting demographics in mechanical turk. In *Proceedings of CHI 2010, Atlanta GA, ACM*, 2010.

[Rubinstein *et al.*, 2009] Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 1–14, 2009.

[Satya *et al.*, 2016] Prudhvi Ratna Badri Satya, Kyumin Lee, Dongwon Lee, Thanh Tran, and Jason (Jiasheng) Zhang. Uncovering fake likers in online social networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 2365–2370, 2016.

[Sedhai and Sun, 2015] Surendra Sedhai and Aixin Sun. Hspam14: A collection of 14 million tweets for hashtag-oriented spam research. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 223–232, 2015.

[Shah and Zhou, 2015] Nihar Shah and Dengyong Zhou. Double or Nothing: Multiplicative Incentive Mechanisms for Crowdsourcing. In *Neural Information Processing Systems (NIPS), Montreal, Canada. 2015.*, 2015.

[Shah *et al.*, 2013] Nihar Shah, Bradley Zhou, Parekh Perses, Yuval, Wainwright Abhay, Ramchandran Martin J., and Kannan. A case for ordinal peer-evaluation in MOOCs. In *Neural Information Processing Systems (NIPS): Workshop on Data Driven Education, Lake Tahoe*, 2013.

[Shah *et al.*, 2015] Nihar Shah, Dengyong Zhou, and Yuval Perses. Approval Voting and Incentives in Crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-15).*, 2015.

[Sheng *et al.*, 2008] V. Sheng, F. Provost, and P. Ipeirotis. Get Another Label? Improving Data Quality and Data Mining using Multiple, Noisy Labelers. In *Knowledge Discovery and Data Mining (KDD)*, pages 614–622, 2008.

[Silberman *et al.*, 2010a] M. Six Silberman, Lilly Irani, and Joel Ross. Ethics and tactics of professional crowdwork. *XRDS*, 17(2):39–43, December 2010.

[Silberman *et al.*, 2010b] M.Six Silberman, Joel Ross, Lily Irani, and Bill Tomlinson. Sellers' problems in human computation markets. In *Human Computation Workshop (HComp 2010)*, 2010.

[Snow *et al.*, 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast—but is it good?: Evaluating Non-expert Annotations for Natural Language Tasks. In *EMNLP '08*, pages 254–263, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[Sorokin and Forsyth, 2008] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops*, Jan 2008.

[Team, 2014] Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual, Version 2.5.0*, 2014.

[Topi *et al.*, 2005] Heikki Topi, Joseph S. Valacich, and Jeffrey A. Hoffer. The Effects of Task Complexity and Time Availability Limitations on Human Performance in Database Query Tasks. *International Journal of Human-Computer Studies*, 62(3):349–379, 2005.

[Tran *et al.*, 2009] Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'09, pages 15–28, Berkeley, CA, USA, 2009.

[Vakharia and Lease, 2015] Donna Vakharia and Matt Lease. Beyond Mechanical Turk: An Analysis of Paid Crowd Work Platforms. In *Proceedings of iConference*, 2015.

[VanderPlas, 2014] Jake VanderPlas. Frequentism and bayesianism: A python-driven primer. In *SciPy 2014, Austin, TX*, 2014.

[von Ahn and Dabbish, 2004] Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.

[Wallace *et al.*, 2011] Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. Who should label what? Instance Allocation in Multiple Expert Active Learning. In *In Proc. of the SIAM International Conference on Data Mining (SDM*, 2011.

[Wang *et al.*, 2011] Jing Wang, Panagiotis G. Ipeirotis, and Foster Provost. Managing Crowdsourcing Workers. In *Winter Conference on Business Intelligence, Utah*, 2011.

[Wang *et al.*, 2012] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 679–688, New York, NY, USA, 2012. ACM.

[Wang *et al.*, 2013] Tianyi Wang, Gang Wang, Xing Li, Haitao Zheng, and Ben Y. Zhao. Characterizing and detecting malicious crowdsourcing. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, SIGCOMM '13, pages 537–538, New York, NY, USA, 2013. ACM.

[Wang *et al.*, 2014] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 239–254, San Diego, CA, August 2014. USENIX Association.

[Welinder *et al.*, 2010a] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The Multidimensional Wisdom of Crowds. In *Neural Information Processing Systems (NIPS)*, 2010.

[Welinder *et al.*, 2010b] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The Multidimensional Wisdom of Crowds. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 2424–2432, 2010.

[Whitehill *et al.*, 2009] Jacob Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Neural Information Processing Systems (NIPS)*, pages 2035–2043, 2009.

[Yan *et al.*, 2010] Yan Yan, R. Rómer, F. Glenn, S. Mark, H. Gerardo, B. Luca, M. Linda, and G. Jennifer. Modeling Annotator Expertise: Learning When Everybody Knows a Bit of Something. In *AISTAT*, 2010.

[Yan *et al.*, 2011] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G Dy. Active Learning from Crowds. In *International Conference on Machine Learning (ICML)*, pages 1161–1168, 2011.

[Yerkes and Dodson, 1908] RM Yerkes and JD Dodson. The relation of strength of stimulus to rapidity of habit-formation. *Journal of Comparative Neurology and Psychology*, pages 459–482, 1908.

[Zhang and Chaudhuri, 2015] Chicheng Zhang and Kamalika Chaudhuri. Active Learning from Weak and Strong Labelers. In *Neural Information Processing Systems (NIPS), Montreal, Canada. 2015.*, 2015.

Part V

Appendices

Appendix

Abbreviations used in the table are as follows,

MV = Majority voting,

MV1 = Majority voting (25%), **MV2** = Majority voting (35%),

MV3 = Majority voting (45%), **MV4** = Majority voting (55%),

G = GLAD, **GW** = GLAD with Clamping,

DS1 = Dawid & Skene 1, **DS2** = Dawid & Skene 2,

BP1 = Belief Propagation 1, **BP2** = Belief Propagation 2,

MF1 = Mean Field 1, **MF2** = Mean Field 2,

KOS1 = Karger's Iterative methods 1, **KOS2** = Karger's Iterative methods 2,

E1 = ELICE 1, **E1-C** = ELICE 1 with clustering,

E2 = ELICE 2, **E2-C** = ELICE 2 with clustering,

E3-P = ELICE 3 Pairwise & **E3-C** = ELICE 3 Circular.

This appendix contains the results for the significance tests.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
GW	0	0	0	0	0	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
DS1	0DS	0	0	0	0	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
DS2	0	0	0	0	0	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
BP1	0	0	0	0	0	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
BP2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
MF1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
MF2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
KOS1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	0	0	0
KOS2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	0	0	0
E1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	0	0	0
E1-C	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	0	0	0
E2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	0	0	0
E2-C	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
CLUBS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-

Table 1: Paired t-tests results for the accuracy level of different methods: 90% labelers are making less than 35% mistakes and 10% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV2	0	0	-	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
GW	0	0	0	0	0	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
DS1	0	0	0	0	0	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
DS2	0	0	0	0	0	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
BP1	0	0	0	0	0	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
BP2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
MF1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	0	0	NaN	NaN	0	0	0
MF2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	0	0	NaN	NaN	0	0	0
KOS1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	0	0	NaN	NaN	0	0	0
KOS2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	0	0	NaN	NaN	0	0	0
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0
E2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	-	NaN	0	0	0
E2-C	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
CLUBS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-

Table 2: Paired t-tests results for the accuracy level of different methods: 80% labelers are making less than 35% mistakes and 20% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
GW	0	0	0	0	0	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
DS1	0	0	0	0	0	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
DS2	0	0	0	0	0	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
BP1	0	0	0	0	0	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
BP2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
MF1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	0	0	NaN	0	NaN	0	0
MF2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	0	0	NaN	0	NaN	0	0
KOS1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	0	0	NaN	0	NaN	0	0
KOS2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	0	0	NaN	0	NaN	0	0
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0
E2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	-	0	NaN	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	0	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
CLUBS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-

Table 3: Paired t-tests results for the accuracy level of different methods: 70% labelers are making less than 35% mistakes and 30% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
MV1	0	-	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0
MV2	0	0	-	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0
MV3	0	0	0	-	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
GW	0	0	0	0	0	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
DS1	0	0	0	0	0	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
DS2	0	0	0	0	0	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
BP1	0	0	0	0	0	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
BP2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
MF1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
MF2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	0	0	NaN	NaN	NaN	NaN	0
KOS1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	0	0	NaN	NaN	NaN	NaN	0
KOS2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	0	0	NaN	NaN	NaN	NaN	0
E1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	-	1	1	1	1	0
E2	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	-	NaN	NaN	NaN	0
E2-C	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	-	NaN	NaN	0
E3-P	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	-	NaN	0
E3-C	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	NaN	NaN	NaN	-	0
CLUBS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-

Table 4: Paired t-tests results for the accuracy level of different methods: 60% labelers are making less than 35% mistakes and 40% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV1	0	-	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV2	0	0	-	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	1	1	1	1	1	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
GW	1	1	1	1	1	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS1	1	1	1	1	1	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS2	1	1	1	1	1	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP1	1	1	1	1	1	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
MF1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	1	1	1	1	1	1	1
MF2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	1	1	1	1	1	1	1
KOS1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	1	1	1	1	1	1	1
KOS2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	1	1	1	1	1	1	1
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	-	0
CLUBS	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	-

Table 5: Paired t-tests results for the accuracy level of different methods: 40% labelers are making less than 35% mistakes and 60% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
MV1	0	-	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV2	0	0	-	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
G	1	1	1	1	1	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
GW	1	1	1	1	1	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS1	1	1	1	1	1	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS2	1	1	1	1	1	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP1	1	1	1	1	1	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
MF1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	1	1	1	1	1	1	1
MF2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	1	1	1	1	1	1	1
KOS1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	1	1	1	1	1	1	1
KOS2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	1	1	1	1	1	1	1
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	-	0
CLUBS	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	-

Table 6: Paired t-tests results for the accuracy level of different methods: 30% labelers are making less than 35% mistakes and 70% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
MV1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV2	0	0	-	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
G	1	1	1	1	1	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	1	1	1	1	1	1	1
GW	1	1	1	1	1	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	0	0	1	1	1	1	1	1	1
DS1	1	1	1	1	1	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	0	0	1	1	1	1	1	1	1
DS2	1	1	1	1	1	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	0	0	1	1	1	1	1	1	1
BP1	1	1	1	1	1	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	0	0	1	1	1	1	1	1	1
BP2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	0	0	1	1	1	1	1	1	1
MF1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	0	0	1	1	1	1	1	1	1
MF2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	0	0	1	1	1	1	1	1	1
KOS1	1	1	1	1	1	0	0	0	0	0	0	0	0	-	0	1	1	1	1	1	1	1
KOS2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	-	1	1	1	1	1	1	1
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	-	0
CLUBS	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	-

Table 7: Paired t-tests results for the accuracy level of different methods: 20% labelers are making less than 35% mistakes and 80% are making more than 65% mistakes.

	MV	MV1	MV2	MV3	MV4	G	GW	DS1	DS2	BP1	BP2	MF1	MF2	KOS1	KOS2	E1	E1-C	E2	E2-C	E3-P	E3-C	CLUBS
MV	-	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
MV1	0	-	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV3	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
MV4	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
G	1	1	1	1	1	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
GW	1	1	1	1	1	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS1	1	1	1	1	1	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
DS2	1	1	1	1	1	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP1	1	1	1	1	1	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
BP2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1
MF1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	NaN	1	1	1	1	1	1	1
MF2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	NaN	1	1	1	1	1	1	1
KOS1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	NaN	1	1	1	1	1	1	1
KOS2	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-	1	1	1	1	1	1	1
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	1	1	1	0
E1-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	1	1	1	1	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
E2-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
E3-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	0	0
E3-C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	-	0
CLUBS	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	-

Table 8: Paired t-tests results for the accuracy level of different methods: 10% labelers are making less than 35% mistakes and 90% are making more than 65% mistakes.