Probabilistic Models for Aggregating Crowdsourced Annotations

Yuan Li

Submitted in total fulfilment of the requirements of the degree of
Doctor of Philosophy

School of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

February 2019
Abstract

This thesis explores aggregation methods for crowdsourced annotations. Crowdsourcing is a popular means of creating training and evaluation datasets for machine learning, e.g. used for computer vision, natural language processing, and information retrieval, at a low cost and in a timely manner. However, due to low-quality annotations, individual workers cannot be wholly trusted to provide reliable annotations and consequently items are typically redundantly labelled by different workers, with labels aggregated subsequently. Although many aggregation methods have been proposed to jointly learn non-uniform weights to workers and infer the truth, the simplest aggregation method, majority voting (MV) which grants workers equal votes towards consensus, still predominates in practice.

To find explanations to the predominance of MV, we conduct extensive experiments of evaluation on 19 datasets and identify two shortcomings that prevent existing methods from being applied in practice over the simple MV. A key finding is that most methods don’t significantly outperform MV across all datasets. These methods may achieve higher mean accuracy than MV does but are also outperformed by MV on several datasets. A secondary shortcoming is that several methods require slow and cumbersome inference, which doesn’t scale to large datasets that are common in practice.

To address the identified shortcomings, we propose two novel aggregation methods both of which significantly outperform MV. The first is a Bayesian version of a weighted average model. It learns unknown voting weights of workers in a principled way by estimating their posterior, unlike existing weighted average models that rely on heuristic update rules or optimising handcrafted objectives. The second approach, complementary to the above, is another Bayesian model that captures the correlations between worker labels which most existing models completely ignore or assume don’t exist. Learning the correlations also helps the method achieve the highest mean
accuracy among all methods compared in our experiments.

When applying aggregation methods in practice, it’s typically assumed that the only information we have is worker labels, but in many situations more information is available. For the setting where item content is available, e.g. feature vectors, we propose a novel model for aggregating binary labels using a Boltzmann machine prior to bias similar instances towards sharing the same label. We also show further gains by integrating a proposed active learning heuristic. We also consider a second, related, setting where instances are sentences, the task is annotating which words in the sentence denote a named entity, structural outputs from a few classifiers are given, and the goal is ensembling those classifiers. We discuss the strategy of adapting aggregation methods for crowdsourcing into this setting. We also discuss the effect of a few gold labels on truth inference and approaches for effectively leveraging gold labels.
Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Yuan Li, February 2019
I am extremely grateful to my supervisors, Trevor Cohn and Benjamin Rubinstein, for their guidance, knowledge, feedback, and patience throughout my studies. This thesis would have been possible without their support. I also appreciate the opportunity that they offered to let me tutor in their subjects, which provides me a lot of fun and also improves my communication skills. I would like to thank my colleagues in the Natural Language Processing group, particularly Afshin Rahimi, Moe Fang, Felix Liu, Yitong Li, and Elly Liang. The group has been a source of friendships as well as collaborations that contribute a lot to both life and work of mine over my Ph.D. candidature. I would like to acknowledge the financial assistance from a Melbourne International Research Scholarship and Trevor’s research projects, allowing me to explore more directions which also contribute to the thesis. Lastly, I would like to thank my parents and my girlfriend Riella Li, for their love and support to me.
Preface

• Chapter 3 is based on a paper accepted to the Web Conference 2019 (formerly known as WWW):
  – Truth inference at scale: A Bayesian model for adjudicating highly redundant crowd annotations. Yuan Li, Benjamin I. P. Rubinstein, Trevor Cohn.

• Chapter 4 is based on a paper accepted to ICML 2019:
  – Exploiting Worker Correlation for Label Aggregation in Crowdsourcing. Yuan Li, Benjamin I. P. Rubinstein, Trevor Cohn.

• Chapter 6 is derived from a paper accepted to ACL 2019:
  – Multilingual NER Transfer for Low-resource Languages. Afshin Rahimi*, Yuan Li*, Trevor Cohn.

This was a joint work where Afshin Rahimi and I contributed equally. Accordingly, the thesis reports the work that was my own contribution, namely the unsupervised zero-shot methods (“BEA”), but omits the contributions of my co-author on the supervised techniques (“RaRe”). Note that my collaborators performed the data preparation and evaluation work, and their contribution is noted in the relevant sections of the chapter.
# Contents

1 Introduction 1

2 Related Work and an Empirical Comparison of Existing Models 7
   2.1 Related Work ................................................................. 7
      2.1.1 Probabilistic Models ................................................. 9
      2.1.2 Non-probabilistic Models ........................................... 14
   2.2 Experiment Settings ....................................................... 16
      2.2.1 Benchmarks .............................................................. 16
      2.2.2 Real-world Datasets .................................................. 16
   2.3 Results ........................................................................ 18

3 A Bayesian Weighted Average Model 25
   3.1 Introduction ................................................................. 26
   3.2 Proposed Models ............................................................ 27
      3.2.1 The Bayesian Weighted Average Model ......................... 27
      3.2.2 Multi-class Extension .................................................. 32
      3.2.3 Hyper-parameter Settings ........................................... 32
   3.3 Discussion .................................................................... 33
      3.3.1 An Objective Intractable for Other Models ..................... 33
      3.3.2 A Novel Perspective on the Wisdom of Crowds ................. 35
      3.3.3 A Novel Constraint on Worker Reliability ....................... 36
   3.4 Experiments .................................................................. 37
      3.4.1 Choosing $a_v$ and $b_v$ .............................................. 37
      3.4.2 Results .................................................................. 38
   3.5 Conclusion .................................................................... 41

4 A Model Capturing Worker Correlations 43
   4.1 Introduction ................................................................. 43
   4.2 Preliminaries ................................................................. 44
      4.2.1 Bayesian Classifier Combination (BCC) Models ................ 44
   4.3 The Proposed Model ....................................................... 47
      4.3.1 Fitting the Joint Distribution over Worker Labels ............. 47
      4.3.2 Integrating with Tensor Decomposition .......................... 48
      4.3.3 The Generative Process and Joint Distribution ................. 49
      4.3.4 The Inference Algorithm ............................................ 50
List of Figures

2.1 Categories of 10 existing aggregation models used in our experiments. ........................................ 10
2.2 Mean accuracy of 10 benchmarks on 19 real-world datasets. ......................................................... 19
2.3 Accuracy of 10 benchmarks on 19 real-world datasets. ................................................................. 20

3.1 Plate diagram for our proposed BWA model (binary case). ............................................................ 29
3.2 The performance of our proposed BWA model with $b_v$ set by two different strategies. .................. 38
3.3 Running time (seconds) of all methods on 19 real-world datasets. ................................................. 39
3.4 Accuracies of all methods on 19 real-world datasets. ................................................................. 40

3.5 A toy example for item clustering on a cube. .................................................................................... 53
3.6 MV, iBCC, and EBCC on synthetic datasets. .................................................................................... 55
3.7 Mean accuracy of 10 benchmarks and EBCC models on 19 real-world datasets. ......................... 57
3.8 Accuracies of all methods on 19 real-world datasets. ..................................................................... 59

4.1 The plate diagram for the iBCC model. ............................................................................................ 45
4.2 The plate notation for our proposed model. .................................................................................... 49
4.3 MV, iBCC, and EBCC on synthetic datasets. .................................................................................... 55
4.4 MV, iBCC, and EBCC on synthetic datasets. .................................................................................... 55
4.5 Mean accuracy of 10 benchmarks and EBCC models on 19 real-world datasets. ......................... 57
4.6 Accuracies of all methods on 19 real-world datasets. ..................................................................... 59

5.1 Example confusion matrices for different worker types. .................................................................. 66
5.2 Two possible label assignments to points. ....................................................................................... 67
5.3 Accuracy comparisons between MV, GPC-MA, BMMA, and the active learner BMMA-BW on partial TREC dataset (50 simulations). ................................................................. 75
5.4 Accuracy comparisons between MV, BMMA, and two active learners BMMA-BW and BMMA-LV on partial sentiment polarity training sets (50 simulations). .............................. 77
5.5 Accuracy comparisons between two active learners BMMA-BW and BMMA-LV on partial sentiment polarity training sets (original & synthetic, 50 simulations). .............................. 78
5.6 Recall estimates for 4 simulated annotators on partial synthetic dataset (50 simulations). ................ 79
5.7 Specificity estimates for 4 simulated annotators on partial synthetic dataset (50 simulations). ............ 79
5.8 Number of true worker labels (real) and synthetic labels from 4 simulated annotators (c,d,e,f) requested by the active learner (50 simulations). ................................................................. 80

6.1 Transfer $F_1$ scores between different source and target languages. ............................................. 83
6.2 Plate diagram for the aggregation model. ......................................................................................... 85
6.3 The mean $F_1$ score of the proposed BEA models over 41 languages. ........................................... 91
6.4 The mean $F_1$ performance over the 41 languages categorised by the number of source languages used for transfer. ........................................................................................................ 93
List of Tables

1. Statistics about 19 real-world datasets used in our experiments . . . . . . . . . . . 17
2. Comparison between GLAD and MV . . . . . . . . . . . . . . . . . . . . . . . . 21
3. Critical values for one-sided Wilcoxon signed rank test . . . . . . . . . . . . . . 21
4. One-sided Wilcoxon signed rank test results (9 benchmarks against MV) . . . . . 22
5. Mathematical notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
6. One-sided Wilcoxon signed rank test results (BWA against MV) . . . . . . . . . 41
7. A toy example of two highly correlated workers A and B . . . . . . . . . . . . . . 47
8. Accuracy of 5 workers on different subtypes . . . . . . . . . . . . . . . . . . . . . 55
9. One-sided Wilcoxon signed rank test results (EBCC against MV) . . . . . . . . . 57
10. Notation used in this chapter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
11. Results on the full TREC dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
12. Results on the full sentiment polarity dataset . . . . . . . . . . . . . . . . . . . . 76
13. An example sentence with its aggregated labels in both token view and entity view. 87
14. Data requirements for different models . . . . . . . . . . . . . . . . . . . . . . . . 90
15. F1 scores of our proposed models on the test set . . . . . . . . . . . . . . . . . . . 92
This thesis presents new methods for inferring true labels for a dataset based on its crowdsourced annotations. Large volumes of labelled data are crucial for training supervised machine learning models or evaluating systems. Nowadays, almost all large-scale labelled datasets are collected by crowdsourcing, for example ImageNet (Russakovsky et al., 2015), SQuAD 2.0 (Rajpurkar et al., 2018), SemEval Sentiment Analysis in Twitter dataset (Rosenthal et al., 2017), and many more.

Ideally, ground truth labels of a dataset are collected from experts. However, for a very large dataset, this method is often too slow due to limited numbers of available experts or is prohibitively costly. In recent years, crowdsourcing (Howe, 2008; Callison-Burch and Dredze, 2010) provides an alternative way for collecting large amounts of labels at a low cost and in a timely manner. One well-known platform for crowdsourcing is Amazon Mechanical Turk. There requesters pay for workers to perform tasks, which are difficult for machines to automate but relatively easy for humans, such as image annotation, question answering, and sentiment analysis. Snow et al. (2008) and Novotney and Callison-Burch (2010) show that using non-expert labels for training machine learning algorithms can be as effective as using gold standard annotations from experts, despite being much less expensive to obtain.

Unfortunately crowdsourced labels are often noisy due to workers’ lack of domain expertise, or the presence of malicious bots labelling for short term financial gain (Difallah et al., 2012). To address this issue, for every single item, practitioners usually collect several labels from different workers, and then aggregate those worker labels by some algorithmic means to infer a more accurate consensus annotation. This problem motivates the objective of this thesis: inferring the hidden truth from crowdsourced annotations.

Formally, we assume that a crowdsourcing labelling task contains a number of homogeneous
instances or items, where each item has its own true label. The true labels are drawn from a label set. All instances are uploaded to a crowdsourcing platform by requesters. Generally the requesters don’t know the true labels, so they ask workers to label them. Workers need to choose a label from the label set per item. Typically a worker only labels a small fraction of all items in a task. The labels collected from workers are called worker labels. Finally, aggregation methods or truth inference algorithms are used to infer the true annotations based on all worker labels.

Aggregation methods are naturally used in creating classification task datasets, but also play an important role in creating other kinds of datasets. For example, in the creation of SQuAD 2.0, a reading comprehension dataset, workers are employed to write both answerable and unanswerable questions for paragraphs of text, after which additional workers are hired to collect multiple human answers for each question, which are then aggregated to produce the final answer.

Furthermore, aggregation methods for crowdsourcing are closely related to a more general problem of resolving the conflicts between information from different sources and inferring the unknown truth. A well-known aggregation method, the Dawid-Skene (DS) model (Dawid and Skene [1979]), was originally proposed for compiling patient records based on their history taken by different clinicians where the patient’s true response is unavailable. Another successful model, independent Bayesian Classifier Combination (iBCC) (Kim and Ghahramani [2012]), was originally proposed for ensembling classifiers based on their outputs on an unlabelled dataset. Moreover, there is another line of work from the database community on resolving the conflicts between different websites or structured datasets and inferring the truth (Zhao et al. [2012], Li et al. [2014]). Due to the close relatedness between the above areas, an improvement in aggregation methods for crowdsourcing is likely to benefit other areas as well.

Although aggregation methods have been extensively studied, the simplest aggregation method, majority voting (MV), still predominates in practice. MV is an ensemble where every worker is given equal vote, based on the assumption that with increasing redundancy, the most common label will converge to the true label. For better understanding of existing methods, we conduct extensive experiments of 10 representative aggregation methods on 19 datasets over a wide range of tasks (Chapter 2). Comparing to previous comparison studies (Sheshadri and Lease [2013], Zheng et al. [2017]) where methods are only compared on a few datasets, we not only compare methods on more datasets, but also include more recent large-scale datasets in the experiments. From the
results, we identify the following shortcomings in existing methods:

- Although several methods achieve higher mean accuracy across all datasets than MV does, a Wilcoxon signed rank test shows only one method, called Generative model of Labels, Abilities, and Difficulties (GLAD) ([Whitehill et al. 2009](#)) significantly outperforms MV at one sided significance level 0.05.

- However, GLAD has its own problems: it is the slowest among all methods compared and doesn’t scale well to large datasets. For example, on the [senti](#) dataset used in our experiments with 569k crowdsourced annotations, GLAD takes 3 hours to train, while MV runs in 3.5 seconds.

This thesis largely focuses on addressing the first problem, with the second shortcoming given comparatively less attention. The first problem is an important issue because not significantly outperforming MV indicates that a method either is slightly worse than MV on several datasets or fails terribly on one or more datasets, both of which will prevent the method from being applied in practice over simple MV.

We propose two novel aggregation methods for crowdsourcing.

- The first one is a Bayesian version of the Weighted Average model (BWA), where every worker has a voting weight and all weights are learned by a probabilistic model, unlike other weighted average models that either estimate the weights by heuristic update rules or learn them by optimizing a hand-crafted objective function. BWA is based on a Bayesian graphical model with conjugate priors, and simple iterative expectation-maximisation inference. It significantly outperforms MV across all datasets. Moreover, BWA is simple, implemented in only 50 lines of code, and trains in seconds.

- The second approach, enhanced Bayesian classifier combination (EBCC) is an enhanced version of independent Bayesian classifier combination (iBCC) which captures the correlations between worker labels that other models assume don’t exist. EBCC is also a Bayesian graphical model with inference based on a mean-field variational approach. An introduced mixture of intra-class reliabilities—connected to tensor decomposition and item clustering—induces inter-worker correlation. Results show that EBCC not only significantly outperforms MV but also achieves the highest mean accuracy across all datasets.
Apart from the above two models, we also study the aggregation problem in two related settings:

- We propose a novel model for aggregating binary labels in the case that item content is available. The model infers the ground truth and worker reliabilities jointly, making use of a Boltzmann machine prior to bias similar instances towards sharing the same label, with a per-worker confusion matrix to model the worker reliability. Experimental results on two datasets show that our proposed approach can outperform both MV and a Gaussian process classification model. Moreover, we show further gains when integrating a proposed active learning heuristic.

- We apply the iBCC model in the setting of combining a number of named entity recognition (NER) models transferred from different source languages to a target language. The entire process including transferring models and inferring the truth can be completely unsupervised, which is suitable for low-resource target languages. When some specific languages are the targets, the proposed unsupervised method even achieves competitive accuracy compared to its supervised counterpart which has access to a small set of sentences with gold labels for fine-tuning. Several practical issues are also discussed including the granularity of aggregation, spammer removal, and the effect of using a small number of gold sentences to help aggregation.

The remaining parts of the thesis are organized as follows:

- Chapter 2 presents the experimental results of 10 representative aggregation methods on 19 datasets over a wide range of tasks. Two main shortcomings of existing methods are identified and motivate the development of novel aggregation methods in the following chapters.

- Chapter 3 is based on a paper accepted to The Web Conference 2019 (formerly known as WWW), where a Bayesian weighted average model is proposed:

  - Truth inference at scale: A Bayesian model for adjudicating highly redundant crowd annotations. Yuan Li, Benjamin I. P. Rubinstein, Trevor Cohn.

- Chapter 4 is based on a paper accepted to ICML 2019, where a model capturing worker correlations is proposed:
Chapter 5 studies the setting where the content of items is available. A probabilistic model integrated with a Boltzmann machine is presented to make use of the item content. Active learning heuristics are also proposed to further reduce the requirement of labelled data.

Chapter 6 considers aggregating the outputs from NER models transferred from different source languages to one target language. Practical issues are discussed including the granularity of aggregation, spammer removal, as well as the effect of using a few gold sentences to help aggregation.

Chapter 7 summarises the thesis contributions and concludes with suggested future research directions.
Chapter 2

Related Work and an Empirical Comparison of Existing Models

This thesis deals with aggregating crowdsourced annotations. Although numerous models have been proposed to improve aggregation accuracy, the simplest technique, majority voting (MV), still predominates in practice.

In this chapter, we present an empirical comparison between MV and 9 representative benchmarks on 19 real-world datasets, in order to uncover explanations for MV’s prevalence. Surprisingly, the Wilcoxon signed rank significance test shows only one of the 9 benchmarks, GLAD proposed by Whitehill et al. (2009), significantly outperforms MV at one-sided significance level 0.05. However, GLAD still suffers from being the slowest one among the 9 benchmarks and doesn’t scale well to large datasets, for example, on the senti dataset used in our experiments with 569k crowdsourced annotations, GLAD takes 3 hours to train, while MV runs in a mere 3.5 seconds.

The findings in this chapter motivate the new aggregation methods proposed in following chapters.

2.1 Related Work

Ever since Dawid and Skene (1979) proposed a model to aggregate the clinical diagnoses of doctors, various methods have been proposed to aggregate worker labels and infer true annotations. They mainly come from the machine learning and database communities. Regardless of the origin, almost all aggregation methods except MV not only infer the truth but also estimate the reliability of every worker. This is because, no matter how labels are aggregated, we believe that the more
reliable a worker is, the more influence that worker should have on the truth. To implement this idea, we must consider two main challenges: how worker reliability is parameterised, and how the knowledge of worker reliability is transferred between items.

The first challenge largely depends on the task a worker performs: commonly accuracy is used for classification tasks and variance of errors is used for regression tasks. The second challenge motivates us to think about whether the parameter (accuracy/variance) of a worker applies on all items.

Let $y_{ij}$ denote a worker $j$'s response to item $i$ and $z_i$ the latent truth.

For regression tasks, it’s natural to assume $y_{ij}$ follows a normal distribution with mean $z_i$ and variance $\sigma_j^2$, i.e. $\mathcal{N}(z_i, \sigma_j^2)$ (Raykar et al., 2010). Then we can expect that $y_{ij}$ is within $3\sigma_j$ from $z_i$ with high probability. Given $\sigma_j$, when $z_i$ and $\sigma_j$ are of the same order of magnitude, this implies that the worker often makes large errors, but when $z_i$ is much larger than $\sigma_j$, this suggests that the worker is very reliable. The apparent inconsistency is the use of the same $\sigma_j$ for both large and small $z_i$’s. A possible solution could be letting $\sigma_j$ depend on $z_i$, e.g. $\sigma_j = \lambda_j z_i$ where $\lambda_j$ is a constant for modelling the relative error instead of absolute error.

Similarly, for classification tasks, a worker’s performance could depend on the true label of items, so that a worker only performs consistently on items belonging to the same class. This leads to adopt a confusion matrix to parameterise worker reliability: learning a $K \times K$ confusion matrix $V_j$ for worker $j$ where $K$ is the number of classes and $v_{jkl} = P(y_{ij} = l|z_i = k)$ (Dawid and Skene, 1979; Raykar et al., 2010; Kim and Ghahramani, 2012). In Chapter 4, we further assume that there are subtypes within each class and workers only perform consistently on subtypes and propose an enhanced model of Kim and Ghahramani (2012).

Furthermore, when the content of items is available, we can assume the worker reliability depends on items themselves. An example is due to Yan et al. (2010) where worker $j$’s accuracy is given by a logistic function $\sigma(\bar{w}_j \cdot \bar{x}_i + \gamma_j)$ with $\bar{x}_i$ being the feature vector of item $i$, $\bar{w}_j$ and $\gamma_j$ the weight vector and bias term learnt for worker $j$.

We call the models using only one parameter to capture worker reliability 1-coin models. Their relation to confusion matrix based models is shown in Figure 2.1. 10 representative models are sorted in to different categories. DS (Dawid and Skene, 1979), iBCC (Kim and Ghahramani, 2012), CBCC (Venanzi et al., 2014), LFC (Raykar et al., 2010), and Minimax (Zhou et al., 2012).
use confusion matrices to model worker reliability, so all of them are confusion matrix based models. ZC (Demartini et al., 2012) and GLAD (Whitehill et al., 2009) are classified as both confusion matrix based models and 1-coin models. They are 1-coin models because they only model accuracy of workers. They are classified as confusion matrix based models because they are generative models and have to define confusion matrices to specify how worker annotations are generated even if confusion matrices are not used for modelling worker reliability. Suppose there are 3 classes (\{0, 1, 2\}). A worker is assumed to have an accuracy \(q\) on all items then 
\[
q = P(y = k|z = k),
\]
but as a generative model, it has to define other probabilities such as 
\[
P(y = k'|z = k)
\]
not just 
\[
P(y \neq k|z = k),
\]
therefore it equivalently defines a confusion matrix based on a single parameter \(q\):

\[
\begin{bmatrix}
q & 1 - q/2 & 1 - q/2 \\
1 - q/2 & q & 1 - q/2 \\
1 - q/2 & 1 - q/2 & q
\end{bmatrix}.
\]

Finally, there is a category of weighted average models that completely belongs to 1-coin models. Weighted average models estimate the truth as 
\[
z_i = \frac{\sum_j v_j y_{ij}}{\sum_j v_j}
\]
where \(v_j\) is the voting weight for worker \(j\). In fact, CATD (Li et al., 2014) and CRH (Aydin et al., 2014) are originally proposed for regression tasks, but could also be used for classification tasks by replacing discrete \(y_{ij}\) with its one-hot vector representation so that the truth is estimated as the weighted average of vectors.

We now describe some representative models in more detail. Section 2.1.1 covers probabilistic models, and Section 2.1.2 presents non-probabilistic models.

### 2.1.1 Probabilistic Models

Most probabilistic models for inferring true annotations define the joint distribution of observed labels \(Y\) and latent true annotations \(Z\) as 
\[
P(Y, Z) = \prod_i P(z_i) \prod_j P(y_{ij}|z_i)
\]
where \(y_{ij}\) is worker \(j\)'s label to item \(i\), \(z_i\) is the true annotation for item \(i\).
Dawid and Skene (1979) proposed a model (DS) that parameterises $P(y_{ij}|z_i)$ by a confusion matrix, i.e.

$$v_{jkl} = P(y_{ij} = l|z_i = k),$$

and $P(z_i)$ by a categorical distribution $\tau_k = P(z_i = k)$. Then finds the maximum likelihood estimation (MLE) for both $\tau_k$’s and $v_{jkl}$’s:

$$\bar{\tau}^*, V^* = \arg \max_{\bar{\tau}, V} P(Y|\bar{\tau}, V) = \arg \max_{\bar{\tau}, V} \sum_Z P(Y, Z|\bar{\tau}, V).$$

Since conditioned on $\bar{\tau}$ and $V$, all $z_i$’s are independent, it’s easy to find $Z^*$:

$$Z^* = \arg \max_Z P(Y, Z|\bar{\tau}^*, V^*).$$

Then $z_i^*$ is the inferred true label of item $i$. 

**Figure 2.1:** Categories of 10 existing aggregation models used in our experiments, namely, MV (majority voting), ZC (Demartini et al., 2012), GLAD (Whitehill et al., 2009), DS (Dawid and Skene, 1979), Minimax (Zhou et al., 2012), iBCC (Kim and Ghahramani, 2012), CBCC (Venanzi et al., 2014), LFC (Raykar et al., 2010), CATD (Li et al., 2014), and CRH (Aydin et al., 2014).
Raykar et al. (2010) proposed a model (LFC) that adds a Dirichlet prior for $\vec{v}_{jk}$ and $\vec{\tau}$, i.e. $\text{Dir}(\vec{v}_{jk}|\vec{\beta}_k)$ and $\text{Dir}(\vec{\tau}|\vec{\alpha})$, then finds maximum a posteriori (MAP) estimation for $\tau$ and $V$:

$$\vec{\tau}^*, V^* = \arg \max_{\vec{\tau}, V} P(\vec{\tau}, V|Y, \alpha, \beta) = \arg \max_{\vec{\tau}, V} P(Y|\vec{\tau}, V) \cdot P(\vec{\tau}|\alpha) P(V|\beta)$$

$$= \arg \max_{\vec{\tau}, V} \sum_Z P(Y, Z|\vec{\tau}, V) \cdot P(\vec{\tau}|\alpha) P(V|\beta).$$

LFC estimates $Z$ in the same way as DS does. This model has been independently proposed by Carpenter (2008) and further generalised to allow workers to have different reliabilities in related sub-tasks. For example, a worker is asked to judge whether a document is relevant to a query, then instead of modelling the worker’s reliability on all query-document pairs as one confusion matrix, the worker could be allowed to have one confusion matrix per query, and all confusion matrices are drawn from the same distribution.

Hovy et al. (2013) proposed the MACE model which assumes that every worker is a mixture of an expert who always provides the correct label and a spammer who randomly labels everything, where workers have their own spamming behaviors. Formally, worker $j$’s spamming behavior is captured by a categorical distribution parameterised by $\vec{\epsilon}_j$. Then worker $j$’s behavior can be captured by a confusion matrix

$$\theta_j I_K + (1 - \theta_j) 1_K \otimes \vec{\epsilon}_j,$$

where $I_K$ is a $K \times K$ identity matrix, $1_K$ a vector of $K$ ones, $\otimes$ the outer product for vectors, and $\theta_j$ the probability that a worker being an expert. Then for item $i$, $P(y_{ij}|z_i)$ is

$$P(y_{ij} = l|z_i = k) = \begin{cases} \theta_j + (1 - \theta_j) \epsilon_{jl} & \text{if } k = l \\ (1 - \theta_j) \epsilon_{jl} & \text{if } k \neq l. \end{cases}$$

Kim and Ghahramani (2012) proposed the iBCC model which is a Bayesian version of LFC. Gibbs sampling is used to sample $Z$, $\vec{\tau}$, and $V$ from their joint distribution $P(Z, \vec{\tau}, V|Y, \alpha, \beta)$. The marginal distribution for $z_i$ is approximated by samples of it, i.e. $\bar{P}(z_i|Y, \alpha, \beta)$, then the latent truth is estimated as

$$z_i^* = \arg \max_{z_i} \bar{P}(z_i|Y, \alpha, \beta).$$
Simpson et al. (2013) then developed variational Bayesian inference for the iBCC model, which is computationally efficient and outperforms the Gibbs sampling method. Furthermore, they analyzed the clusters naturally formed by the inferred workers’ confusion matrices. Venanzi et al. (2014) proposed CBCC which assumes there are a fixed number of clusters of workers, with workers in the same cluster possessing similar confusion matrices instead of modeling individual confusion matrices as done in previous works. In this way worker clusters are formed at inference-time instead of post-inference. Adding a Dirichlet process to the model to generate the Dirichlet priors on worker confusion matrices, Moreno et al. (2015) also allowed workers to generate their own confusion matrices from the cluster’s confusion matrix in order to increase model flexibility.

We refer the reader to Section 4.2.1 for more details on iBCC models.

Demartini et al. (2012) proposed the ZenCrowd (ZC) model which is a 1-coin model that only learns an accuracy \( v_j \) for worker \( j \). ZC then defines that

\[
P(y_{ij} = l | z_i = k) = \begin{cases} v_j & \text{if } k = l \\ \frac{1 - v_j}{K-1} & \text{if } k \neq l \end{cases},
\]

(2.1)

and fixes \( \tau_k \) to \( \frac{1}{K} \). Finally, ZC uses the MLE estimates of \( v_j \)’s to find the latent truth \( Z \) similarly to DS.

Whitehill et al. (2009) proposed a model called GLAD (Generative model of Labels, Abilities, and Difficulties). GLAD defines the probability that worker \( j \)’s label to item \( i \) being correct as

\[
\rho_{ij} = P(y_{ij} = z_i | \alpha_j, \beta_i) = \frac{1}{1 + e^{-\alpha_j \beta_i} },
\]

where the difficulty of the instance \( i \) is modeled by \( 1/\beta_i \in (0, +\infty) \) and the expertise of worker \( j \) is modeled by \( \alpha_i \in \mathbb{R} \). Then \( P(y_{ij} | z_i) \) can be calculated as in a way similar to Equation (2.1):

\[
P(y_{ij} = l | z_i = k) = \begin{cases} \rho_{ij} & \text{if } k = l \\ \frac{1 - \rho_{ij}}{K-1} & \text{if } k \neq l \end{cases}.
\]
The definition of $\rho_{ij}$ can also be written as
\[
\log \frac{P(y_{ij} = z_i | \alpha_j, \beta_i)}{1 - P(y_{ij} = z_i | \alpha_j, \beta_i)} = \alpha_j \beta_i,
\]
where the LHS is the log odds for the observed label $y_{ij}$ being correct. If $\beta_i > 0$ is fixed, then $\alpha_j = +\infty$ means that worker $j$ is a perfect worker, $\alpha_j = -\infty$ means that worker $j$ is an adversarial worker, and $\alpha_j = 0$ means worker $j$ cannot provide any useful information for us. If $\alpha_j$ is fixed, then as $1/\beta_i$ grows from 0 to $+\infty$, the probability of $y_{ij}$ being correct gradually decreases to 0.5, which indicates that instance $i$ becomes more and more difficult. In this way, GLAD models both the worker expertise $\alpha_j$ and instance difficulty $\beta_i$.

Regarding inference, an EM algorithm is derived for MLE of $\alpha$ and $\beta$. Furthermore, Whitehill et al. (2009) also added priors for $\alpha$ and $\beta$. $\alpha_j$ is assumed to follow a standard Gaussian distribution, while $\beta_i$ is nonnegative, $\beta'_i$ is defined as $\log \beta_i$ and is assumed to follow a standard Gaussian distribution as well. The inference algorithm for finding the MAP remains highly similar to that for the MLE.

Welinder et al. (2010a) is a bit special, because their method learns both a latent representation for every instance and a classifier for every worker. Since their method doesn’t make use of the content of instances, the latent representations are purely learned in an unsupervised manner.

They consider binary labelling tasks and define $\prod_j P(y_{ij}|z_i)$ as
\[
\prod_j P(y_{ij}|z_i) = P(x_i|z_i) \prod_j P(y_{ij}|x_i),
\]
where $x_i$ is the latent feature vector for instance $i$. Suppose $x_i$ has $K$ dimensions and $h_{ij}$ the same as $x_i$, then $P(x_i|z_i)$ is defined as
\[
P(x_i|z_i) = \prod_{k=1}^{K} P(x_i^{(k)}|z_i) = \mathcal{N}(x_i^{(k)}; \mu_z, \theta_z^2),
\]
where $x_i^{(k)}$ is the $k$-th element in $x_i$, $\mu_z = -1$ if $z_i = 0$ and $\mu_z = 1$ if $z_i = 1$, and $\theta_z = 0.8$. If we consider $P(x_i) = \sum_{z_i \in \{0,1\}} P(x_i|z_i)P(z_i)$, then it is a mixture of two Gaussian distributions, and
one center is $(1, 1, \ldots, 1)$, the other is $(-1, -1, \ldots, -1)$. Furthermore, $P(y_{ij} | x_i)$ is defined as

$$P(y_{ij} | x_i) = \Phi(w_j^T x_i - \tau_j)$$

where $\Phi(\cdot)$ is the cumulative distribution function of Gaussian distribution, $w_j$ and $\tau_j$ define a decision plane for worker $j$ similar to logistic regression.

Priors are also added over $w_j$ and $\tau_j$, with $P(z_i = 0)$ and $P(z_i = 1)$ fixed to be 0.5, then MAP inference is used to find optimal $x_i$, $w_j$, and $\tau_j$.

### 2.1.2 Non-probabilistic Models

[Zhou et al. (2012)] developed an algorithm estimating the true annotations by using a minimax entropy principle, which assumes there are categorical distributions per worker-item pair and the worker labels are generated from those categorical distributions. The objective there is in finding the true annotations that minimize the maximum entropy of those categorical distributions. Such categorical distributions are more concentrated on worker labels and can be roughly considered as having higher likelihood of generating labels.

[Li et al. (2014)] adopted the Gaussian to model error between the truth and worker label, which is similar to our model but not a probabilistic model. They developed an iterative approach which alternatively estimated the truth based on variances, and set the variances to be the upper-bounds of their 95% confidence interval based on the estimated truth.

[Aydin et al. (2014)] adopted a two-step procedure that iteratively updates estimated truth and worker weights so that the sum of worker weight times the distance between worker label and estimated truth is minimized.

[Dalvi et al. (2013)] proposed an MF-based (Matrix Factorization) method which addresses binary labeling problems under the one-coin model. Let $m$ be the number of instances and $n$ be the number of workers. Let $z_i \in \{-1, +1\}$ be the true annotation of instance $i$. Let $G \in \{0, 1\}^{m \times n}$ be the instance-worker assignment matrix, that is $G_{ij} = 1$ if instance $i$ is labelled by worker $j$. 


2.1 Related Work

A stochastic matrix $U \in \{-1, 0, +1\}^{m \times n}$ stores the labels where

$$U_{ij} = \begin{cases} 
  z_i & \text{if } G_{ij} = 1, \text{ with probability } p_j \\
  -z_i & \text{if } G_{ij} = 1, \text{ with probability } 1 - p_j \\
  0 & \text{if } G_{ij} = 0
\end{cases}$$

The assumed generation process of $U$ is that for worker $j$ if $G_{ij} = 1$ then they will toss a coin with bias $p_j$ and label instance $i$ (correctly) as $z_i$ with probability $p_j$ or label instance $i$ (wrongly) as $-z_i$ with probability $1 - p_j$. They further define $w_j = 2p_j - 1$ to be the worker reliability. Since $p_j \in [0, 1]$, $w_j \in [-1, +1]$. Let $\mathbf{w}$ be the vector of $w_j$’s and $\mathbf{z}$ be the vector of $z_i$’s.

Let $E$ denote the matrix containing the expected values of the stochastic matrix $U$, that

$$E_{ij} = \mathbb{E} U_{ij} = (p_jz_i + (1 - p_j)(-z_i))G_{ij} = z_i G_{ij} w_j.$$ Furthermore, 

$$(E^T E)_{jk} = \sum_i (z_i G_{ij} w_j)(z_i G_{ik} w_k) = (\sum_i G_{ij} G_{ik}) w_j w_k = (G^T G)_{jk} (\mathbf{w}^T \mathbf{w})_{jk}$$

By using Hadamard product symbol $\otimes$, the above equation can be written as 

$$E^T E = (G^T G) \otimes (\mathbf{w}^T \mathbf{w})$$

If $E$ is known, then $\mathbf{w}$ can be found by solving

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \|E^T E - (G^T G) \otimes (\mathbf{w}^T \mathbf{w})\|_F$$

s.t. $\forall j, \ w_j^2 \leq 1$

where $\|M\|_F = (\sum_{ij} M_{ij}^2)^{1/2}$ is the Frobenius norm.

However, in practice, there are two problems: (1) $E$ is unknown; (2) even if $E$ were known, the optimization problem cannot be efficiently solved. To address problem (1), we can make an approximation that $E \approx U$. To address problem (2), we assume that $G$ can be approximated by a rank one matrix $\mathbf{v}^T \mathbf{v}$, then $(G^T G) \otimes (\mathbf{w}^T \mathbf{w})$ is also rank one, so we can first approximate $E^T E$ by a rank one matrix $\mathbf{u}^T \mathbf{u}$, then find $\mathbf{w}$ with $\mathbf{u}^T \mathbf{u} = (\mathbf{v}^T \mathbf{v})(\mathbf{w}^T \mathbf{w})$. Therefore the optimization can be solved in the following steps:
Related Work and an Empirical Comparison of Existing Models

- solve \( \mathbf{u} = \arg \max \| \mathbf{U}^T \mathbf{U} - \mathbf{u}^T \mathbf{u} \|_F \)
- solve \( \mathbf{v} = \arg \max \| \mathbf{G}^T \mathbf{G} - \mathbf{v}^T \mathbf{v} \|_F \)
- \( \tilde{w}_j = \frac{u_j}{v_j} \) if \( v_j \neq 0 \) otherwise \( \tilde{w}_j = 0 \)
- \( \hat{w}_j = \text{sgn}(\tilde{w}_j) \min\{||\tilde{w}_j||, 1\} \)

After obtaining \( \hat{\mathbf{w}} \), \( z_i \) can estimated as \( z_i = \text{sgn}(\sum_j \mathbf{U}_{ij} \hat{w}_j) \)

2.2 Experiment Settings

2.2.1 Benchmarks

Zheng et al.\(^1\) compared 17 existing aggregation methods and released their implementations, and 10 of them supporting the multi-class setting are used in our experiment including MV, ZC (Demartini et al.\(^2\) 2012), GLAD (Whitehill et al.\(^3\) 2009), DS (Dawid and Skene\(^4\) 1979), Minimax (Zhou et al.\(^5\) 2012), iBCC (Kim and Ghahramani\(^6\) 2012), CBCC (Venanzi et al.\(^7\) 2014), LFC (Raykar et al.\(^8\) 2010), CATD (Li et al.\(^9\) 2014), and CRH (Aydin et al.\(^10\) 2014).

2.2.2 Real-world Datasets

There are 19 real-world datasets used in the experiments originating with four crowdsourcing dataset collections, namely the union of the CrowdScale 2013’s shared task challenge (Josephy et al.\(^11\) 2014\(^1\) (2 datasets), the Active Crowd Toolkit project (Venanzi et al.\(^12\) 2015\(^1\) (8 datasets), the Truth Inference in Crowdsourcing project (Zheng et al.\(^13\) 2017\(^1\) (7 datasets), and the GitHub repository for SpectralMethodsMeetEM paper (Zhang et al.\(^14\) 2014\(^1\) (5 datasets), noting that 3 datasets are in common between the last two collections.

Table 2.1 lists the 19 datasets used in our experiments. The first two, senti and fact, are from the CrowdScale 2013’s shared task challenge, and are the largest two in our experiment, with

---

\(^1\)https://zhydhkcws.github.io/crowd_truth_inference/index.html
\(^2\)https://sites.google.com/site/crowdscale2013/shared-task
\(^3\)https://github.com/orchidproject/active-crowd-toolkit
\(^4\)See footnote 1.
\(^5\)https://github.com/zhangyuc/SpectralMethodsMeetEM
2.2 Experiment Settings

Table 2.1: Statistics about 19 real-world datasets used in our experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#item(N)</th>
<th>#worker(W)</th>
<th>#class</th>
<th>#label</th>
<th>#truth</th>
<th>#label/N</th>
<th>#label/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>senti</td>
<td>98980</td>
<td>1960</td>
<td>5</td>
<td>569282</td>
<td>1000</td>
<td>5.75</td>
<td>290.45</td>
</tr>
<tr>
<td>fact</td>
<td>42624</td>
<td>57</td>
<td>3</td>
<td>214960</td>
<td>576</td>
<td>5.04</td>
<td>3771.23</td>
</tr>
<tr>
<td>CF</td>
<td>300</td>
<td>461</td>
<td>5</td>
<td>1720</td>
<td>300</td>
<td>5.73</td>
<td>3.73</td>
</tr>
<tr>
<td>CF*</td>
<td>300</td>
<td>110</td>
<td>10</td>
<td>6025</td>
<td>300</td>
<td>20.08</td>
<td>54.77</td>
</tr>
<tr>
<td>MS</td>
<td>700</td>
<td>44</td>
<td>3</td>
<td>2945</td>
<td>700</td>
<td>4.21</td>
<td>66.93</td>
</tr>
<tr>
<td>SP</td>
<td>4999</td>
<td>203</td>
<td>2</td>
<td>27746</td>
<td>4999</td>
<td>5.55</td>
<td>136.68</td>
</tr>
<tr>
<td>SP*</td>
<td>500</td>
<td>143</td>
<td>2</td>
<td>10000</td>
<td>500</td>
<td>20</td>
<td>69.93</td>
</tr>
<tr>
<td>ZCall</td>
<td>2040</td>
<td>78</td>
<td>2</td>
<td>20372</td>
<td>2040</td>
<td>9.99</td>
<td>261.18</td>
</tr>
<tr>
<td>ZCin</td>
<td>2040</td>
<td>25</td>
<td>2</td>
<td>10626</td>
<td>2040</td>
<td>5.21</td>
<td>425.04</td>
</tr>
<tr>
<td>ZCus</td>
<td>2040</td>
<td>74</td>
<td>2</td>
<td>11271</td>
<td>2040</td>
<td>5.53</td>
<td>152.31</td>
</tr>
<tr>
<td>prod</td>
<td>8315</td>
<td>176</td>
<td>2</td>
<td>24945</td>
<td>8315</td>
<td>3</td>
<td>141.73</td>
</tr>
<tr>
<td>tweet</td>
<td>1000</td>
<td>85</td>
<td>2</td>
<td>20000</td>
<td>1000</td>
<td>20</td>
<td>235.29</td>
</tr>
<tr>
<td>dog</td>
<td>807</td>
<td>109</td>
<td>4</td>
<td>8070</td>
<td>807</td>
<td>10</td>
<td>74.04</td>
</tr>
<tr>
<td>face</td>
<td>584</td>
<td>27</td>
<td>4</td>
<td>5242</td>
<td>584</td>
<td>8.98</td>
<td>194.15</td>
</tr>
<tr>
<td>adult</td>
<td>11040</td>
<td>825</td>
<td>4</td>
<td>89948</td>
<td>333</td>
<td>8.15</td>
<td>109.03</td>
</tr>
<tr>
<td>bird</td>
<td>108</td>
<td>39</td>
<td>2</td>
<td>4212</td>
<td>108</td>
<td>39</td>
<td>108</td>
</tr>
<tr>
<td>trec</td>
<td>19033</td>
<td>762</td>
<td>2</td>
<td>88385</td>
<td>2275</td>
<td>4.64</td>
<td>115.99</td>
</tr>
<tr>
<td>web</td>
<td>2665</td>
<td>177</td>
<td>2</td>
<td>15567</td>
<td>2653</td>
<td>5.84</td>
<td>87.95</td>
</tr>
<tr>
<td>rte</td>
<td>800</td>
<td>164</td>
<td>2</td>
<td>8000</td>
<td>800</td>
<td>10</td>
<td>48.78</td>
</tr>
</tbody>
</table>

569k and 217k labels respectively. The senti data asks the workers to judge the sentiment of a tweet discussing weather and label it within {Negative, Neutral, Positive, Unrelated, Can’t tell}. The fact data requests workers to judge whether relational statements about public figures are correct, e.g. “Stephen Hawking graduated from Oxford”. Workers may choose from {Yes, No, Skip}. Note that “Can’t tell” or “Skip” may also appear as the true labels of some items due to the inherent ambiguity of some questions. None of the methods compared in our experiments explicitly model the unsure options, instead treating these as a standard label in a 5 and 3-class setting, respectively.

The next 8 datasets are all from the Active Crowd Toolkit project. MS, collected by Rodrigues et al. (2013), asks workers to listen to music samples of duration 30s and to classify the music into 10 music genres, {country, jazz, disco, pop, reggae, rock, blues, classical, mental, hip-hop}. ZC_in and ZC_us, provided by Demartini et al. (2012), ask workers to judge if a provided Uniform Resource Identifier (URI) is relevant to a named entity extracted from news, where every
Related Work and an Empirical Comparison of Existing Models

URI describes an entity. ZC_all is a combination of the above two ZCs, and all ZCs share the same set of items. The two SPs are movie review datasets with binary sentiment labels. CF is just a small part of the full senti. CF* and SP* have the same items as CF and SP, but were re-annotated.

The next 5 datasets are from the Truth Inference project Zheng et al. (2017). prod, collected by Wang et al. (2012) for entity resolution, asks workers whether two product descriptions refer to the same product or not. tweet, collected by Zheng et al. (2017), contains tweets where every tweet is related to a company. Workers are asked to identify if the tweet has positive sentiment towards the company in the tweet. dog contains dog images from ImageNet (Krizhevsky et al., 2012) and the task is to recognize a breed (out of Norfolk Terrier, Norwich Terrier, Irish Wolfhound, and Scottish Deerhound) for a given dog. face, provided by Mozafari et al. (2014), contains face images, and workers are asked to judge whether the person in the image is happy, sad, angry, or neutral. adult, released by Mason and Suri (2012), asks workers to determine the age appropriateness (P, PG, R, X) of a website given its link.

The last five datasets are used in Zhang et al. (2014). bird, provided by Welinder et al. (2010b), is a dataset of bird images. The task is to determine whether an image contains at least one duck. The remaining three cover: recognising textual entailment with rte, Snow et al. (2008), assessing the quality of retrieved documents with trec (TREC 2011 crowdsourcing track), judging the relevance of web search results with web, Zhou et al. (2012).

2.3 Results

In this section, we report the results of 10 benchmarks on 19 real-world datasets. Figure 2.2 shows the mean accuracy of every benchmark across all datasets. More details are shown in Figure 2.3. As shown in Figure 2.2, iBCC achieves the highest mean accuracy across all 19 real-world datasets, followed by LFC and CATD.

We also report the results of one-sided Wilcoxon signed rank test performed on every method against MV. An example procedure for comparing GLAD and MV is shown in Table 2.2.

First, the null hypothesis $H_0$ is chosen as the performance difference between GLAD and MV.
follows a symmetric distribution around zero. The difference between their accuracies on every
dataset is calculated (δ) and all entries with δ = 0 are discarded. Since the accuracies of GLAD
and MV are not exactly the same on all datasets, no entry is discarded, which leaves an effective
sample of size \( N_r = 19 \). Then all entries in Table 2.2 is sorted in ascending order by the absolute
value of \( \delta \) (i.e. |\( \delta \)|) and a signed rank is assigned to every entry where the sign comes from \( \delta \). In
this case, GLAD performs worse than MV on 6 datasets with the accuracy difference < 1% on
3 on them. Finally, \( W_+ \) and \( W_- \) are calculated as the absolute value of the sum of all positive
values and negative values in signed ranks, respectively. Intuitively, the smaller \( W_- \) is, the more
significantly GLAD is better than MV. To determine the significance level, one needs the critical
values listed in Table 2.3.

In the case that GLAD and MV are compared, \( N_r = 19 \), then we learn from Table 2.3 that,
given that the null hypothesis \( H_0 \) is true, the probability of \( W_- \leq 53 \) is smaller than or equal to
0.05, i.e. \( P(W_- \leq 53|H_0) \leq 0.05 \). Due to that \( W_- = 47 \), the null hypothesis is rejected at
one-sided level 0.05, which could be interpreted as GLAD being significantly better than MV at
one-sided level 0.05.

Usually, the significance test level is set ahead of running the test. But in our evaluation, we
are more interested in the significance level that every method can pass, because we interpret it
as the robustness level of every method. Our intuition is that if a method is slightly worse than
MV on several datasets or fails terribly on one or more datasets then it is not considered as robust,

\[ \text{http://users.stat.ufl.edu/~athienit/Tables/tables} \]
Figure 2.3: Accuracy of 10 benchmarks on 19 real-world datasets.
2.3 Results

Table 2.2: Comparison between GLAD and MV

| Dataset | GLAD | MV | $\delta$ | $|\delta|$ | Signed rank |
|---------|------|----|----------|-------------|-------------|
| fact    | 0.9010 | 0.9019 | -0.0009 | 0.0009 | -1 |
| CF      | 0.8800 | 0.8822 | -0.0022 | 0.0022 | -2 |
| CF*     | 0.8567 | 0.8533 | 0.0033  | 0.0033 | 3  |
| adult   | 0.7628 | 0.7583 | 0.0045  | 0.0045 | 4  |
| SP*     | 0.9480 | 0.9430 | 0.0050  | 0.0050 | 5  |
| face    | 0.6284 | 0.6367 | -0.0083 | 0.0083 | -6 |
| bird    | 0.7685 | 0.7593 | 0.0093  | 0.0093 | 7  |
| senti   | 0.8940 | 0.8835 | 0.0105  | 0.0105 | 8  |
| ZCuS    | 0.8495 | 0.8615 | -0.0120 | 0.0120 | -9 |
| dog     | 0.8352 | 0.8222 | 0.0130  | 0.0130 | 10 |
| ZCall   | 0.8118 | 0.8301 | -0.0184 | 0.0184 | -11|
| tweet   | 0.9520 | 0.9335 | 0.0185  | 0.0185 | 12 |
| prod    | 0.9205 | 0.8966 | 0.0239  | 0.0239 | 13 |
| ZCin    | 0.7691 | 0.7404 | 0.0287  | 0.0287 | 14 |
| rte     | 0.9263 | 0.8969 | 0.0294  | 0.0294 | 15 |
| SP      | 0.9182 | 0.8859 | 0.0323  | 0.0323 | 16 |
| web     | 0.7946 | 0.7307 | 0.0639  | 0.0639 | 17 |
| trec    | 0.5754 | 0.6514 | -0.0760 | 0.0760 | -18|
| MS      | 0.7871 | 0.7042 | 0.0829  | 0.0829 | 19 |

Summary: $N_r = 19$, $W_+ = 143$, $W_- = 47$

Table 2.3: Critical values for one-sided Wilcoxon signed rank test.

<table>
<thead>
<tr>
<th>Significance level</th>
<th>0.1</th>
<th>0.05</th>
<th>0.025</th>
<th>0.01</th>
<th>0.005</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_r = 18$</td>
<td>55</td>
<td>47</td>
<td>40</td>
<td>32</td>
<td>27</td>
</tr>
<tr>
<td>$N_r = 19$</td>
<td>62</td>
<td>53</td>
<td>46</td>
<td>37</td>
<td>32</td>
</tr>
</tbody>
</table>

and probationers may not have enough confidence in it. We need a way to measure robustness that focuses on the cases where a method is outperformed by MV, and gives more attention to cases where the difference is larger. Fortunately, the statistic $W_-$ calculated in Wilcoxon signed rank test meets our need, and we treat the significance test as a heuristic to measure robustness. Therefore, we report the significance level that every method can pass in the test against MV as well as their approximated p-values.

The same procedure has also been applied to compare other methods with MV. Table 2.4 summarizes the results showing whether every method is significantly better than MV and at which level. Zencrowd and iBCC achieve 85.33% accuracy on the CF* dataset which is exactly the same
Table 2.4: One-sided Wilcoxon signed rank test results (9 benchmarks against MV)

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean accuracy</th>
<th>$N_r$</th>
<th>$W_{-}$</th>
<th>Significance level</th>
<th>Approx. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV</td>
<td>0.819557</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ZC</td>
<td>0.829444</td>
<td>18</td>
<td>64</td>
<td>✓</td>
<td>0.174551</td>
</tr>
<tr>
<td>GLAD</td>
<td>0.830476</td>
<td>19</td>
<td>47</td>
<td>✓ ✓</td>
<td>0.026703</td>
</tr>
<tr>
<td>DS</td>
<td>0.830907</td>
<td>19</td>
<td>71</td>
<td></td>
<td>0.167069</td>
</tr>
<tr>
<td>Minimax</td>
<td>0.814085</td>
<td>19</td>
<td>81</td>
<td></td>
<td>0.286584</td>
</tr>
<tr>
<td>iBCC</td>
<td>0.835245</td>
<td>18</td>
<td>52</td>
<td>✓</td>
<td>0.072291</td>
</tr>
<tr>
<td>CBCC</td>
<td>0.829098</td>
<td>19</td>
<td>71</td>
<td></td>
<td>0.167069</td>
</tr>
<tr>
<td>LFC</td>
<td>0.833811</td>
<td>19</td>
<td>69</td>
<td></td>
<td>0.147712</td>
</tr>
<tr>
<td>CATD</td>
<td>0.833413</td>
<td>19</td>
<td>56</td>
<td>✓</td>
<td>0.058272</td>
</tr>
<tr>
<td>CRH</td>
<td>0.805379</td>
<td>19</td>
<td>91</td>
<td></td>
<td>0.436059</td>
</tr>
</tbody>
</table>

performance as MV resulting in $\delta = 0$, therefore the records on the $\text{CF}^*$ dataset are discarded for them and their $N_r$ is reduced to 18. A check mark means the method of that row is significantly better than MV at the one-sided level of that column. Based on the results in Table 2.4, we can claim that GLAD is the only method that significantly outperforms MV at any one-sided level $\alpha \geq 0.05$.

The rightmost column in Table 2.4 shows the p-value calculated based on approximating the cumulative density function (cdf) of $W_{-}$ given $H_0$ as a cdf of Gaussian. Due to the small $N_r$ in our experiment (18 or 19), the approximation is somehow inaccurate, but we keep them there as a reference, as they show ranking of the improvement achieved by all methods comparing to MV.

As shown in Table 2.4, although iBCC has the highest mean accuracy across all datasets, it obtains a significance level of 0.1 that it outperforms MV, which unfortunately is commonly not considered as significant enough to reject the null hypothesis. On the other hand, GLAD, whose mean accuracy is not outstanding, significantly outperforms MV at a significance level of 0.05, which may not be considered as evidence of strong significance, but it is the best among all 9 benchmarks.

iBCC achieves the highest mean accuracy but is not robust enough in terms of significantly outperforming MV. GLAD is robust enough but its mean accuracy is lower than iBCC’s. Interestingly they belong to two different categories, namely, confusion matrix based models and 1-coin models. We then seek to find better aggregation methods that not only achieve high mean accu-
racy but also strong significance of outperforming MV in both categories. We propose Bayesian Weighted Average (BWA), a 1-coin model, in chapter 3 and Enhanced Bayesian Classifier Combination (EBCC), a confusion matrix based model, in chapter 4, both of which outperform the state-of-the-art methods in their own categories.
Chapter 3

A Bayesian Weighted Average Model

This chapter is based on a paper that has been accepted to The Web Conference 2019:

Truth inference at scale: A Bayesian model for adjudicating highly redundant crowd annotations.

Yuan Li, Benjamin I. P. Rubinstein, Trevor Cohn.

Crowdsourcing is a cheap and popular means of creating training and evaluation datasets for machine learning, however it poses the problem of ‘truth inference’, as individual workers cannot be wholly trusted to provide reliable annotations. Research into models of annotation aggregation attempts to infer a latent ‘true’ annotation, which has been shown to improve the utility of crowdsourced data. However, as we saw in the previous chapter, existing sophisticated truth inference techniques beat simple baselines only in low redundancy settings, where the number of annotations per instance is low (≤ 3), or in situations where workers are unreliable and produce low quality annotations (e.g., through spamming, or random or adversarial behaviours.) As we show, datasets produced by crowdsourcing are often not of this type: the data is highly redundantly annotated (≥ 5 annotations per instance), and the vast majority of workers produce high quality outputs. In these settings, the majority vote heuristic performs extremely well, and most truth inference models can’t outperform this simple baseline significantly.

In this chapter, we propose a novel technique, based on a Bayesian graphical model with conjugate priors, and simple iterative expectation-maximisation inference. Our technique produces consistently strong results across several benchmark datasets, outperforming majority vote, and producing competitive performance against ten state-of-the art benchmark methods. Significance tests show that our technique is the only one significantly outperforming the majority vote heuristic at one-sided level 0.025. Moreover, our technique is simple, is implemented in only 50 lines of code, and trains in seconds.
3.1 Introduction

Both the average number of labels per item (#labels/#items), and the quality of worker labels, are important factors in the performance of aggregation methods. Low-redundancy (e.g. #labels/#items ≤ 3) and/or low-quality scenarios (e.g. there are adversarial workers) are challenging and have attracted much attention. Such scenarios require methods to make full use of a few available labels by being able to interpret labels from different workers. Models which incorporate per-worker confusion matrices (Dawid and Skene [1979]; Raykar et al. [2010]; Kim and Ghahramani [2012]; Venanzi et al. [2014]) are flexible enough to capture different types of worker behavior, and have been shown to be effective in low-redundancy and/or low-quality scenarios.

However, the simplest aggregation technique, majority voting (MV), still predominates in practice. MV is an ensemble where every worker is given equal vote, based on the assumption that with increasing redundancy, the most common label will converge to the true label. Despite its simplicity, on several real-world crowdsourced datasets, MV outperforms most models of truth inference, as we show in our experimental evaluation, and the best method from prior work doesn’t outperform MV significantly. To explain this observation, and the continued popularity of MV, we identify several misalignments between the experimental setup in previous papers and real-world scenarios:

Redundancy & worker quality. In practice #labels/#items is usually ≥ 5 to ensure sufficient redundancy, and crowdsourcing platforms provide mechanisms to select high quality workers (e.g. in the collection of question answering dataset SQuAD [Rajpurkar et al. 2016], workers were filtered based on the amount and acceptance rate of their prior work history). In such settings, MV is very competitive.

Scale. Several datasets used in prior work were small with respect to the number of labels collected from crowd (#labels). But on large-scale datasets, like senti used in our experiment with 569k labels, the scalability of inference becomes critical, for example GLAD [Whitehill et al. 2009], achieving the third highest mean accuracy among all existing methods in our experiment, takes more than 3 hours to train, while MV runs in 3.5 seconds.

This work is also motivated by qualitative considerations: ease of understanding, implementation, and hyper-parameter setting, for which MV is compelling being parameter free. However,
empirically the accuracy of MV is far from perfect, even for highly redundantly annotated datasets. We seek accurate, fast solutions to the high-resource high-quality scenario that are easy to implement. Our method is one such solution.

This chapter presents a novel method for aggregating crowdsourced labels for classification. We propose a generative Bayesian model, relaxing a discrete binary problem to a continuous regression task, which we model using a normal likelihood and conjugate inverse-Gamma prior. Our method allows for straightforward inference using expectation-maximisation (Dempster et al., 1977), which is implemented in 50 lines of code (see Appendix A). The resulting update rules reveal a novel criterion for learning from crowds—namely the idea of minimizing the likelihood that “every worker is making mistakes”. Our model has interpretable parameters and intuitive inference rules, which makes it easy to understand and use. We further extend it to operate in a multi-class setting by employing the one-versus-rest strategy.

Our experimental results show that our proposed method is the only one that significantly outperforms MV at significance level $0.025$ and performs competitively to the state-of-the-art method in terms of mean accuracy, across all datasets drawn from several different application areas, with different sizes and differing degrees of label redundancy. Our proposed method also has a total running time on the same datasets second only to MV.

### 3.2 Proposed Models

In this section, we describe our proposed model, Bayesian Weighted Average (BWA), for binary classification tasks, followed by an extension to multi-class classification tasks, and strategies to set its hyper-parameters.

#### 3.2.1 The Bayesian Weighted Average Model

We first consider binary classification tasks. Without loss of generality, let $y_{ij} \in \{0, 1\}$ be worker $j$’s label to item $i$ and let $z_i \in \mathbb{R}$ represent item $i$’s true label. We model $z_i$’s as continuous variables to make tractable the following optimization problem for inferring the truth, $Z$,

$$Z^* = \arg \max_Z P(Z|Y, \alpha),$$  

(3.1)
Table 3.1: Mathematical notation.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N)</td>
<td>number of items</td>
</tr>
<tr>
<td>(i)</td>
<td>item index, (i \in {1, \ldots, N})</td>
</tr>
<tr>
<td>(W)</td>
<td>number of workers</td>
</tr>
<tr>
<td>(j)</td>
<td>worker index, (j \in {1, \ldots, W})</td>
</tr>
<tr>
<td>(K)</td>
<td>number of classes</td>
</tr>
<tr>
<td>(k, l)</td>
<td>class/label index, (k, l \in {0, \ldots, K - 1})</td>
</tr>
<tr>
<td>(W_i)</td>
<td>set of workers who have labeled item (i)</td>
</tr>
<tr>
<td>(N_j)</td>
<td>set of items that worker (j) has labeled</td>
</tr>
</tbody>
</table>

**Binary setting**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(z_i)</td>
<td>item (i)'s latent true label (continuous), (z_i \in \mathbb{R})</td>
</tr>
<tr>
<td>(y_{ij})</td>
<td>worker (j)'s observed label to item (i) (discrete), (y_{ij} \in {0, 1})</td>
</tr>
</tbody>
</table>

**Multi-class setting**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(z_{ki})</td>
<td>score given to item (i) by model (k) which classifies items into class-(k) or not class-(k), (z_{ki} \in \mathbb{R})</td>
</tr>
<tr>
<td>(y_{kij})</td>
<td>binary indicator, equalling 1 if worker (j) labels item (i) as class (k) otherwise 0</td>
</tr>
</tbody>
</table>

where \(Z\) and \(Y\) are collections of their lowercase variables, \(\alpha\) denotes all hyper-parameters, \(P(Z | Y, \alpha)\) is the posterior distribution for \(Z\), and \(Z^*\) is the maximum a posteriori (MAP) estimate of \(Z\). Further discussion of our modelling choices is provided in Section 3.3.

After \(Z^*\) is computed, the discrete true label can be determined by rounding, i.e. the true label is 1 if \(z^*_i > 0.5\), otherwise 0. Frequently used symbols are defined in Table 3.1.

The Generative Process and Joint Distribution

Our proposed models are all probabilistic generative models. We assume \(z_i\) is drawn from a Gaussian prior with mean \(\mu\) and inverse variance \(\lambda\), and \(y_{ij}\) is drawn from a Gaussian distribution with mean \(z_i\) and inverse variance \(v_j\). Here \(v_j\) is a worker specific value for worker \(j\) and every \(v_j\) is drawn from \(\text{Gamma}(a_v, b_v)\), where \(a_v\) and \(b_v\) are hyper-parameters. The plate diagram for this model is shown in Figure 3.1.

The joint distribution \(P(Z, Y, V | \mu, \lambda, a_v, b_v)\) can be factorized as

\[
P(Z, Y, V | \mu, \lambda, a_v, b_v) = P(Z | \mu, \lambda) \cdot P(Y | Z, V) \cdot P(V | a_v, b_v)
\]
3.2 Proposed Models

\begin{align*}
= \prod_i \mathcal{N}(z_i | \mu, \lambda^{-1}) \cdot \prod_j \int \prod_i \mathcal{N}(y_{ij} | z_i, v_j^{-1}) \cdot \prod_j \text{Gamma}(v_j | a_v, b_v) \ dV_j,
\end{align*}

(3.2)

where \( V = \{v_j\}_{j=1}^W \), \( \lambda \) and \( v_j \) model the inverse variance of \( z_i - \mu \) and \( y_{ij} - z_i \). The larger \( \lambda \) and \( v_j \) are, the smaller the corresponding errors are likely to be. In this way they can be interpreted as encoding worker reliability, a perspective we explore further in Section 3.2.1.

Our proposed BWA model is deficient: data observations \( y_{ij} \) can only be binary, while the generative model has continuous support. We could force all \( y_{ij} \) to be binary variables (\( \{0, 1\} \)) then re-normalize the distributions accordingly to address the deficiency issue (Section 3.3.3). But the linking distribution between \( y_{ij}, z_i \) and \( v_j \) becomes a logistic function, which is much more difficult to work with as no conjugate prior for \( v_j \) exists.

Inference

In order to find the most likely \( Z \), we first integrate out \( V \) in Equation (3.2) to obtain \( P(Z, Y | \alpha) \) where \( \alpha = (\mu, \lambda, a_v, b_v) \) are the hyper-parameters.

\begin{align*}
P(Z, Y | \alpha) &= \int P(Z, Y, V | \mu, \lambda, a_v, b_v) \ dV \\
&= \prod_i \mathcal{N}(z_i | \mu, \lambda^{-1}) \cdot \prod_j \int \prod_i \mathcal{N}(y_{ij} | z_i, v_j^{-1}) \cdot \text{Gamma}(v_j | a_v, b_v) \ dV_j \\
&= \prod_i \mathcal{N}(z_i | \mu, \lambda^{-1}) \cdot \prod_j \left( \frac{b_v/2}{\Gamma(a_v/2)} \right)^{\frac{|N_j|}{2}} \left( \frac{1}{\sqrt{2\pi}} \right)^{|N_j|} \Gamma \left( \frac{a_v+|N_j|}{2} \right) \left( \frac{1}{2b_v + \frac{1}{2} \sum_{i \in N_j} (z_i - y_{ij})^2} \right)^{(a_v+|N_j|)/2}
\end{align*}

Figure 3.1: Plate diagram for our proposed BWA model (binary case).
\[\alpha \prod_i e^{-\frac{1}{2}(z_i - \mu)^2} \prod_j \left( \frac{1}{2} b_v + \frac{1}{2} \sum_{i \in N_j} (z_i - y_{ij})^2 \right)^{-\frac{a_v + |N_j|}{2}}.\]

Since \(P(Z, Y|\alpha) = P(Z|Y, \alpha) P(Y|\alpha)\) and \(P(Y|\alpha)\) is independent from \(Z\), maximizing \(P(Z, Y|\alpha)\) with respect to \(Z\) is equivalent to solving the program (3.1), i.e.

\[\arg \max_Z P(Z|Y, \alpha) = \arg \max_Z P(Z, Y|\alpha).\]

As we shortly show, \(\lambda, a_v, \text{ and } b_v\) have intuitive meanings, and accordingly can be set manually, while \(\mu\) can be optimized easily with \(Z\). Therefore, the goal becomes to maximize \(P(Z, Y|\alpha)\) with respect to \(Z\) and \(\mu\), which is equivalent to minimizing the negative log likelihood 

\[-\log P(Z, Y|\alpha)\]

with respect to \(Z\) and \(\mu\):

\[
\min_{Z,\mu} \sum_i \frac{\lambda}{2} (z_i - \mu)^2 + \sum_j \left( \frac{a_v + |N_j|}{2} \log \left( b_v + \sum_{i \in N_j} (z_i - y_{ij})^2 \right) \right) + \text{const.},
\]

(3.3)

where \(\text{const.}\) is the sum of values independent from \(Z\) and \(\mu\), and thus can be ignored in optimization. We then use the EM algorithm to minimize the above negative likelihood:

\(E\) step. \(v_j|Z, Y, a_v, b_v \sim \text{Gamma} \left( \frac{a_v + |N_j|}{2}, \frac{1}{2} \left( b_v + \sum_{i \in N_j} (z_i - y_{ij})^2 \right) \right)\).

\(M\) step. Let \(q(V) = P(V|Z^{\text{old}}, Y, a_v, b_v)\) and

\[Q(Z; Z^{\text{old}}) = \mathbb{E}_q \log P(Z, Y, V|\mu, \lambda, a_v, b_v),\]

then by setting the gradient of the \(Q\) function with respect to \(Z\) and \(\mu\) to be zero, we obtain

\[z_i = \frac{\lambda \mu + \sum_{j \in W_i} \mathbb{E}_q v_j y_{ij}}{\lambda + \sum_{j \in W_i} \mathbb{E}_q v_j},\]

(3.4)

where

\[\mathbb{E}_q v_j = \frac{a_v + |N_j|}{b_v + \sum_{i \in N_j} (z_i - y_{ij})^2},\]

(3.5)

\[\mu = \frac{1}{N} \sum_i z_i.\]
3.2 Proposed Models

Interpretations

\( z_i \) and its update rule. In Equation (3.4), \( z_i \) is updated by a weighted arithmetic average of \( \mu \) and \( \{y_{ij}\}_{j \in W_i} \), weighted by \( \lambda \) and \( \{E_{qv_j}\}_{j \in W_i} \), respectively. Higher weights lead to a larger impact on average, so these weights can be considered as encoding worker reliability. In particular, \( \lambda \) is the weight for a “default worker” who always labels every item as \( \mu \).

Because \( a_v \) and \( b_v \) are hyper-parameters for Gamma distributions and \( \lambda \) is the inverse variance for Gaussian distributions, they must be positive. It’s easy to see \( E_{qv_j} > 0 \). Then, initializing \( z_i \) in \([0, 1]\) results in \( \mu \in [0, 1] \) (averaging \( z_i \)), then \( z_i \in [0, 1] \) will be guaranteed in all following updates (weighted arithmetic average of values in \([0, 1]\)).

\( v_j \) and its estimation rule. In Equation (3.5), \( |N_j| \) is the number of items that worker \( j \) has labeled, \( \sum_{i \in N_j} (z_i - y_{ij})^2 \) is the sum of squared errors (SSE\(_j\)) that worker \( j \) has made. Note that \( y_{ij} \in \{0, 1\} \) and \( z_i \in [0, 1] \), thus \( 0 \leq \text{SSE}_j \leq |N_j| \). In the case that we know the true label, \( z_i \) is also binary just like \( y_{ij} \), then \( \text{SSE}_j \) is simply the number of wrong judgements that worker \( j \) has made. The estimation of \( E_{qv_j} \) can be considered as a smoothed inverse error rate with \( a_v \) and \( b_v \) serving as the smoothing parameters. The larger the \( E_{qv_j} \), the smaller the worker \( j \)’s smoothed error rate is.

Hyper-parameters. The values of \( a_v \) and \( b_v \) encode our belief that worker \( j \) has labeled \( a_v \) items before and \( b_v \) judgements were wrong, which naturally suggests \( b_v \leq a_v \) and further ensures \( E_{qv_j} \geq 1 \). So when \( b_v \leq a_v \), even if all labels from worker \( j \) in task \( k \) are wrong, we still have \( E_{qv_j} \geq 1 \). This mechanism guarantees everyone an unconditional minimum weight of 1 to vote for the truth. We may consider setting \( b_v > a_v \) so that \( E_{qv_j} \) is allowed to be smaller than 1, however, in this setting, a worker who gets 10/10 wrong will be penalized more than another worker who gets 1000/1000 wrong due to the weaker smoothing effect on the latter. Hence both estimations are smaller than 1, but the estimation for the latter is closer to 1. As this is not the desired effect, we recommend the natural setting of \( b_v \leq a_v \). As 1 is the lowerbound of \( E_{qv_j} \), we choose \( \lambda = 1 \) so that the “default worker” has limited ability to compete with other workers, but is most helpful in tie breaking.
3.2.2 Multi-class Extension

So far, our proposed models assume binary labels. However, it’s easy to extend them to the multi-class setting by employing a one-versus-rest strategy, a commonly used reduction to binary classifiers such as SVMs.

Let $K$ denote the number of classes, $k \in \{0, \ldots, K\}$ a class index, $y_{kij}$ a binary indicator that equals 1 if worker $j$ labels item $i$ as class $k$ otherwise 0, $Y_k$ the collection of $y_{kij}$’s. Then our models can work on $Y_k$ which is binary and output $Z_k$, a collection of $z_{ki}$’s, where $z_{ki}$ is the score that our model gives to item $i$. The higher $z_{ki}$ is, the more likely our model believes item $i$’s true label is $k$. For item $i$, our model generates a score $z_{ki}$ for each $k$ value given different $Y_k$. Finally, $k^* = \arg\max_k z_{ki}$ is picked as the estimated true label for item $i$.

3.2.3 Hyper-parameter Settings

We discuss how to set $b_v$ when $a_v$ is given in this section. As $a_v$ and $b_v$ translate to our prior belief that every worker has labelled $a_v$ items and made $b_v$ mistakes, $b_v / a_v = \epsilon$ is our assumed error rate. This error rate should depend on the quality of worker labels in a dataset, so we seek to estimate $\epsilon$ instead of setting it to be constant regardless of datasets. However, estimating $\epsilon$ requires true labels which we don’t have, so a natural way is to estimate them by a simple aggregation method such as majority voting, then calculate $\epsilon$ based on the estimates.

We first consider the binary case. Let $\hat{z}_i = \frac{1}{|W_i|} \sum_{j \in W_i} y_{ij}$ to be the majority voting results, then the overall error rate $\epsilon$ is

$$\epsilon = \frac{\sum_i \sum_{j \in W_i} (y_{ij} - \hat{z}_i)^2}{\sum_i |W_i|}.$$  

To simplify $\epsilon$, we define $n_{i0}$ and $n_{i1}$ to be the number of workers who have labelled item $i$ as 0 or 1, respectively. Then $|W_i| = n_{i0} + n_{i1}$, $\hat{z}_i = \frac{n_{i1}}{n_{i0} + n_{i1}}$ and we obtain

$$\epsilon = \frac{\sum_i n_{i0}(0 - \hat{z}_i)^2 + n_{i1}(1 - \hat{z}_i)^2}{\sum_i n_{i0} + n_{i1}} = \frac{\sum_i n_{i0}(\frac{n_{i0}}{n_{i0} + n_{i1}})^2 + n_{i1}(\frac{n_{i1}}{n_{i0} + n_{i1}})^2}{\sum_i n_{i0} + n_{i1}} = \frac{\sum_i \frac{n_{i0}n_{i1}}{n_{i0} + n_{i1}}}{\sum_i n_{i0} + n_{i1}}. \quad (3.6)$$

Then we can set $b_v = a_v \epsilon$ to encode that every worker has already annotated $a_v$ items, with $a_v \epsilon$ mistakes, i.e. an error rate $\epsilon$.

However, we find that the error rate is underestimated by Equation (3.6) in practice. We then
derive that the error rate given by Equation (3.6) has an upper bound $1/4$:

$$
\varepsilon = \sum_i \frac{1}{n_{i0} + n_{i1}} \leq \sum_i \frac{1}{n_{i0} + n_{i1}} \left( \frac{n_{i0} + n_{i1}}{2} \right)^2 = \frac{1}{4}.
$$

So given $\varepsilon \in [0, 1/4]$, setting $b_v = a_v \varepsilon$ means our prior belief is that every worker has an accuracy of 75% or higher even in the worst case, which is too optimistic and doesn’t reflect the reality. We then propose doubling $\varepsilon$ to extend its range to $[0, 1/2]$ to encode our belief that every worker is performing slightly better than random guessing.

For multi-class tasks, as there are $K$ classes, we expect random guessing can achieve an accuracy of $1/K$, i.e. an error rate of $1 - 1/K$. To encode the prior belief that every worker is better than random guessing, we extend the range of $\varepsilon$ to $[0, 1 - 1/K]$ by multiplying $\varepsilon$ with $4(1 - 1/K)$. This is consistent with the doubling strategy for binary tasks, as $4(1 - 1/2) = 2$ when $K = 2$.

3.3 Discussion

The primary motivation of modelling the truth $Z$ as continuous variables is to make the optimization of finding the most likely $Z$ tractable, i.e. $\arg \max_Z P(Z|Y, \alpha)$. We first describe the difference between this objective and other probabilistic models’ objectives, then discuss some interesting findings due to the continuous modelling of $Z$.

3.3.1 An Objective Intractable for Other Models

Consider a graphical model defining joint $P(Z, Y, V|\alpha)$. It is often easy to sum out $Z$ or integrate out $V$ in $P(Z, Y, V|\alpha)$ to get $P(Y, V|\alpha)$ or $P(Z, Y|\alpha)$. Ideally, the most likely $Z$ is computed by

$$
Z^* = \arg \max_Z P(Z|Y, \alpha), \text{ or equivalently } Z^* = \arg \max_Z P(Z, Y|\alpha),
$$

because $P(Z, Y|\alpha) \propto P(Z|Y, \alpha)$. However, when $Z$ is discrete, as in most existing probabilistic models, the above objective becomes an intractable discrete optimization task. Typical workarounds are:

1. Solve $V^* = \arg \max_V P(Y, V|\alpha)$ first, then set $V = V^*$ into the joint distribution which
naturally decouples \( z_i \)'s, so \( Z^* = \arg \max_Z P(Y, Z, V^* | \alpha) \) becomes tractable as \( z_i \)'s can be optimized independently.

\[
Z^* = \arg \max_Z P(Y, Z, V^* | \alpha)
\]

where \( V^* = \arg \max_V P(Y, V | \alpha) \). The compromise here is that although \( V^* \) is optimal, \( Z^* \) is only optimal when \( V = V^* \), which does not capture uncertainty. Examples of methods taking this approach include DS (Dawid and Skene, 1979) and LFC (Raykar et al., 2010).

2. Bayesian methods usually use the posterior to make inference. In \( P(Z, Y, V | \alpha) \), only \( Z \) is the variable of interest, so \( V \) is marginalized first, leaving \( P(Z, Y | \alpha) \). The posterior of \( Z \), \( P(Z | Y, \alpha) \) is proportional to \( P(Z, Y | \alpha) \), however, the normalizing constant \( P(Y | \alpha) \) is typically intractable to compute. There are two common ways to deal with this intractability.

*Variational approaches* approximate the posterior \( P(Z | Y, \alpha) \) by another distribution \( q(Z) \), which is then used to make inference. Mean-field variational inference is very popular and assumes \( q(Z) \) is factorized into \( \prod_i q(z_i) \), then the true label of item \( i \) is estimated as \( \arg \max_{z_i} q(z_i) \), which approximates \( \arg \max_{z_i} P(z_i | Y, \alpha) \) where \( P(z_i | Y, \alpha) \) is the marginal distribution of \( z_i \).

*Sampling techniques* generate samples from \( P(Z | Y, \alpha) \). Markov chain Monte Carlo methods are commonly used to generate samples as they can work without knowing the normalizing constant \( P(Y | \alpha) \). Then samples are used to approximate the marginal distribution of \( z_i \)'s, i.e. \( P(z_i | Y, \alpha) \). Finally, \( \arg \max_{z_i} P(z_i | Y, \alpha) \) is picked as the estimate of item \( i \)'s true label.

In summary, with either variational approaches or sampling, the goal of Bayesian methods is to solve

\[
z_i^* = \arg \max_{z_i} P(z_i | Y, \alpha)
\]

However they face the shortcoming that the variables of interest (\( z_i \)'s) must be optimized independently according to their marginal distributions, as joint optimization is intractable as \( Z \) is discrete. Examples include iBCC (Kim and Ghahramani, 2012) and CBCC (Venanzi et al., 2014).
3.3 Discussion

By contrast, our methods’ objectives are ideal:

\[ Z^* = \arg \max_{Z} P(Z|Y, \alpha). \]

The only compromise is that we have to model \( Z \) as continuous variables.

3.3.2 A Novel Perspective on the Wisdom of Crowds

For our proposed BWA model, the negative log likelihood \( -\log P(Z, Y|\alpha) \) defined in Equation (3.3) is reminiscent of a traditional sum-of-square-error objective as minimized in regression models, except that here the error terms are contained in separate log functions. Reorganizing the expression, and using the sum of squared errors defined as \( \text{SSE}_j = \sum_{i \in N_j} (z_i - y_{ij})^2 \), we obtain

\[ -\log P(Z, Y|\mu, \lambda, a_v, b_v) = \frac{\lambda}{2} \sum_i (z_i - \mu)^2 + \frac{1}{2} \sum_j \log \left( \frac{b_v + \text{SSE}_j}{a_v + |N_j|} \right)^{a_v + |N_j|} + \text{const}. \]

Reorganizing the expression, and using the sum of squared errors defined as \( \text{SSE}_j = \sum_{i \in N_j} (z_i - y_{ij})^2 \), we obtain

\[ -\log P(Z, Y|\mu, \lambda, a_v, b_v) = \frac{\lambda}{2} \sum_i (z_i - \mu)^2 + \frac{1}{2} \sum_j \log \left( \frac{b_v + \text{SSE}_j}{a_v + |N_j|} \right)^{a_v + |N_j|} + \text{const}. \]

The exponential of the RHS yields the equivalent objective

\[ \arg \min_{Z, \mu} -\log P(Z, Y|\mu, \lambda, a_v, b_v) = \arg \min_{Z, \mu} \prod_{i} e^{\lambda (z_i - \mu)} \prod_{j} \left( \frac{b_v + \text{SSE}_j}{a_v + |N_j|} \right)^{a_v + |N_j|}. \]

The first product over items serves as a regularization term for \( z_i \). For \( \mu \), it simply means the minimum can be reached when \( \mu \) is the mean of all \( z_i \). The product over workers is more interesting. Note that \( \frac{b_v + \text{SSE}_j}{a_v + |N_j|} \) also appears in \( \mathbb{E}_q \nu_j \)'s update rule, and can be considered as the smoothed error rate of worker \( j \). The exponent, \( a_v + |N_j| \), is just the smoothed number of items worker \( j \) has labelled. Accordingly, this factor measures the likelihood that all workers make all their annotations of the data incorrectly, with a uniform error rate per worker. The term \( \frac{b_v + \text{SSE}_j}{a_v + |N_j|} \) can be considered as an independent Bernoulli trial for each worker correctly or incorrectly labeling each instance, so the product of all of them over all labels and all workers is the probability of labeling everything wrong.

Therefore, the underlying principle behind our proposed BWA model is that it’s unlikely that all workers make mistakes for all their tasks. Our proposed BWA model infers a truth that minimizes the likelihood of everybody performing incorrectly. This principle makes our model differ-
A Bayesian Weighted Average Model

ent from other models which seek to explain worker labels through maximizing the likelihood of realizing these observations. Our principle is conservative, yet intuitively plausible, and a novel perspective on the wisdom of crowds.

3.3.3 A Novel Constraint on Worker Reliability

In the base model, if we were to force all $y_{ij}$ and $z_i$ to be binary variables ($\{0, 1\}$) and then re-normalize the distributions accordingly, we obtain a new link function between $y_{ij}$ and $z_i$,

$$P(y_{ij} = 1 | z_i, v_j) = \frac{\sqrt{v_j} \exp \left\{ -\frac{v_j}{2} (z_i - 1)^2 \right\}}{\sqrt{v_j} \exp \left\{ -\frac{v_j}{2} z_i^2 \right\} + \sqrt{v_j} \exp \left\{ -\frac{v_j}{2} (z_i - 1)^2 \right\}}$$

$$= \frac{1}{\exp \left\{ -v_j(z_i - \frac{1}{2}) \right\} + 1} \text{ for } z_i \in \{0, 1\},$$

or in an equivalent but more compact form,

$$P(y_{ij} | z_i, v_j) = \frac{\exp \left\{ \frac{1}{2} v_j \mathbf{1}[y_{ij} = z_i] \right\}}{1 + \exp \left\{ \frac{1}{2} v_j \right\}},$$

which means each worker has an accuracy of $\sqrt{e^{v_j}} / (1 + \sqrt{e^{v_j}})$, irrespective of the true label $z_i$. Such behavior can be captured by a symmetric confusion matrix

$$\begin{pmatrix} \sqrt{e^{v_j}} & 1 \\ 1 & \sqrt{e^{v_j}} \end{pmatrix}.$$

Due to $v_j$ having a Gamma prior, $v_j$ must be non-negative. Consequently the values on the diagonal must be larger than or equal to the values off diagonal. Accordingly our approach is capable of modelling: a perfect worker, $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, a typical worker, $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$, and a random worker, $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. However adversarial workers, e.g., $\begin{pmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{pmatrix}$, cannot be represented. This limitation arises from the choice of prior (Gamma), and does not feature in other confusion matrix based models.

This constraint means our models can’t capture adversarial workers, however this limitation fits with empirical observations of crowd-workers: in practice adversarial workers are rare, and are largely filtered out based on performance history, which is a consequence of the data acquisition
method, but it’s possible that less carefully collected data with less rigorous quality control may not be a good fit to our model. Typically in a confusion matrix based probabilistic model, worker labels are assumed to be drawn from categorical distributions parameterized by rows in confusion matrices, and every row in a confusion matrix is assumed to be drawn from a Beta or Dirichlet distribution. For binary tasks, the constraint that values on the diagonal are larger than or equal to the values off the diagonal could be achieved by using a truncated Beta prior with a support \([0.5, 1]\), but at a much higher computational cost because the distributions will no longer be conjugate.

### 3.4 Experiments

**Initialization.** Our techniques use the EM algorithm, and thus require initialization of either \(\mu\) and \(E_q V\), or \(Z\). We initialise \(Z\), the easier of the two, using majority voting which allows for fast convergence, as well as deterministic results. We did also experiment with random restarts, which did in some cases lead to improved results in terms of both training likelihood and held-out accuracy (results not reported). So we would suggest practitioners to run the algorithm with random initialization multiple times and find the best likelihood (including a majority voting run).

**Hyper-parameter settings.** Following the suggestions described in Section 3.2.1 we set \(\lambda = 1\). We choose \(a_v\) and \(b_v\) in Section 3.4.1. We set the tolerance in the stopping criteria as \(10^{-3}\), such that iteration of the algorithm stops when the relative difference between every \(z_i\) and its last value is within 0.1%.

**Methods and datasets.** We run the experiments following the setup described in Chapter 2 to compare our proposed methods with 10 benchmarks on 19 datasets. We apply the one-versus-rest strategy to our BWA model, so that it is able to work in multi-class settings in our experiments.

#### 3.4.1 Choosing \(a_v\) and \(b_v\)

Figure 3.2 shows the mean accuracy of our proposed BWA model across 19 datasets with \(b_v\) set by two different strategies described in Section 3.2.3, namely, using the original error rate \(\varepsilon\) and using the adjusted error rate \(\varepsilon \cdot 4(1 - \frac{1}{R})\). For both strategies, the performance increases steeply when
Figure 3.2: The performance of our proposed BWA model with $b_v$ set by two different strategies.

$a_v < 10$ as the prior becomes stronger and provides regularisation to workers who have labelled very few items. We also see both curves are flat around the optimum, so the performance is not very sensitive to the choice of $a_v$. Due to this reason, we pick $a_v$ from the flat region and report results in both settings: BWA($a_v = 30$, original $\varepsilon$) and BWA($a_v = 15$, adjusted $\varepsilon$).

3.4.2 Results

Scalability.

To measure the efficiency of all methods, we run all experiments on a modest desktop with a 7-th generation Intel i5 CPU and 8 GB memory. All methods including our proposed BWA model are implemented in Python, except for iBCC and CBCC which are implemented in C#, and Minimax is implemented in MATLAB.

As shown in Figure 3.3, our proposed methods outperform the other existing methods (except MV) in terms of running time. For datasets other than the largest two (senti and fact), our method converged within one second. GLAD and Minimax are the two slowest methods, as they both require gradient-based optimization algorithms, and the required gradients are expensive to compute.
3.4 Experiments

Table 3.2 demonstrates the Wilcoxon significance test for all methods against MV. Figure 3.4 presents the accuracies on all datasets. All experiments were run on a modest desktop with a 4-th generation Intel i7 CPU and 16 GB memory. All methods are implemented in Python except that iBCC and CBCC are implemented in C# and Minimax is implemented in MATLAB.

As shown in Table 3.2, our proposed BWA($a_o = 15$, adjusted $\varepsilon$) model and iBCC achieve the highest mean accuracy of 83.52%, and our BWA model using another hyper-parameter setting ($a_o = 30$, original $\varepsilon$) which achieves 83.42%, followed by LFC 83.38% and CATD 83.34%. Interestingly, both our proposed methods and CATD model the latent truth $z_i$’s as continuous variables, but in CATD the optimization objective is heuristic based, while in our models, optimization objectives is derived from explicit probabilistic models. Minimax is an interesting model in that it largely outperforms others on bird and web but performs the worst on MS, the three ZCs, and prod. This may be due to Minimax not being a probabilistic model thus its objective function is not well regularised and often too aggressive. This may also explain the results for CRH, another non-probabilistic model.

Besides accuracy, our proposed methods also outperform the other existing methods (except MV) in terms of total running time. For datasets other than the largest two (senti and fact), our method converged within 1 second. GLAD and Minimax are the two slowest methods, as they both require gradient-based optimization algorithms, and the required gradients are expensive to compute.
Figure 3.4: Accuracies of all methods on 19 real-world datasets.
### Table 3.2: One-sided Wilcoxon signed rank test results (BWA against MV)

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean accuracy</th>
<th>N_r</th>
<th>W_-</th>
<th>Significance level 0.1 0.05 0.025 0.01</th>
<th>Approx. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV</td>
<td>0.819557</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZC</td>
<td>0.829444</td>
<td>18</td>
<td>64</td>
<td>✓</td>
<td>0.174551</td>
</tr>
<tr>
<td>GLAD</td>
<td>0.830476</td>
<td>19</td>
<td>47</td>
<td>✓ ✓</td>
<td>0.026703</td>
</tr>
<tr>
<td>DS</td>
<td>0.830907</td>
<td>19</td>
<td>71</td>
<td></td>
<td>0.167069</td>
</tr>
<tr>
<td>Minimax</td>
<td>0.814085</td>
<td>19</td>
<td>81</td>
<td></td>
<td>0.286584</td>
</tr>
<tr>
<td>iBCC</td>
<td>0.835245</td>
<td>18</td>
<td>52</td>
<td>✓</td>
<td>0.072291</td>
</tr>
<tr>
<td>CBCC</td>
<td>0.829098</td>
<td>19</td>
<td>71</td>
<td></td>
<td>0.167069</td>
</tr>
<tr>
<td>LFC</td>
<td>0.833811</td>
<td>19</td>
<td>69</td>
<td></td>
<td>0.147712</td>
</tr>
<tr>
<td>CATD</td>
<td>0.833413</td>
<td>19</td>
<td>56</td>
<td>✓</td>
<td>0.058272</td>
</tr>
<tr>
<td>CRH</td>
<td>0.805379</td>
<td>19</td>
<td>91</td>
<td></td>
<td>0.436059</td>
</tr>
<tr>
<td>BWA($\alpha_v = 30$, original $\epsilon$)</td>
<td>0.834241</td>
<td>18</td>
<td>38</td>
<td>✓ ✓ ✓</td>
<td>0.019289</td>
</tr>
<tr>
<td>BWA($\alpha_v = 15$, adjusted $\epsilon$)</td>
<td>0.835206</td>
<td>18</td>
<td>39</td>
<td>✓ ✓ ✓</td>
<td>0.021429</td>
</tr>
</tbody>
</table>

#### 3.5 Conclusion

This chapter presents a novel method for aggregating crowdsourced labels for classification. We propose a generative Bayesian model, relaxing a discrete binary problem to a continuous regression task, which we model using a normal likelihood and conjugate inverse-Gamma prior, and can be extended to work in multi-class settings by employing the one-versus-rest strategy.

Our models allow for straightforward inference using expectation-maximisation, which is implemented in 50 lines of code. Our models have interpretable parameters and intuitive update rules, which make them easy to understand and use. We also provide detailed practical suggestions such as how to set the hyper-parameters.

Our experimental results show that our proposed method is the only one that significantly outperforms MV at significance level 0.025 and performs competitively to the state-of-the-art method in terms of mean accuracy, across all datasets drawn from several different application areas, with different sizes and differing degrees of label redundancy. Our proposed method also has a total running time on the same datasets second only to MV.
Chapter 4

A Model Capturing Worker Correlations

This chapter is based on a paper that has been accepted to ICML 2019:

Exploiting Worker Correlation for Label Aggregation in Crowdsourcing. Yuan Li, Benjamin I. P. Rubinstein, Trevor Cohn.

In this chapter, we argue that existing crowdsourcing approaches such as these in Chapter 2 and Chapter 3 do not adequately model worker correlations observed in practical settings. We propose in response an enhanced Bayesian classifier combination (EBCC) model, with inference based on a mean-field variational approach. This method is based on a mixture of intra-class reliabilities which induces inter-worker correlation. EBCC does not suffer the limitations of existing correlation models, i.e. intractable marginalisation of missing labels, and poor scaling to large worker cohorts. Extensive empirical comparison on 19 real-world datasets sees EBCC achieving the highest mean accuracy across 10 benchmark crowdsourcing methods.

4.1 Introduction

As argued in Chapter 1 and Chapter 2, parameterising worker reliability is a fundamental requirement for truth inference. In this chapter we argue by carefully worked example (Section 4.3.1) and experimentation on synthetic data (Section 4.4.1) that modeling correlation between worker labels has significant potential to improve truth inference. Correlations exist as workers don’t make errors independently due to various reasons such as workers are likely to make mistakes together on difficult items, or workers having different background tend to produce completely opposite labels on some items. We develop a model that captures worker correlation by modeling true classes as mixtures of subtypes, on which a worker’s performance may vary. We propose a model (Section 4.3.3) that captures worker correlation by modeling true classes as mixtures of subtypes, with
class-level correlation a consequence of worker behaviour varying by subtype. Extending a family of Bayesian classifier combination (BCC) models (Kim and Ghahramani, 2012), we term our model enhanced BCC (EBCC) and develop a variational approach for inference (Section 4.3.4).

While many relevant approaches exist for truth inference (Chapter 2), the only existing model incorporating worker correlation, dBCC (Kim and Ghahramani, 2012), has limitations that disqualify its use in crowdsourcing (Section 4.2.1): as the missing annotations cannot be tractably marginalised out, all workers must annotate all items; and because dBCC possesses parameters quadratic in the number of workers, it cannot scale to large worker cohorts. (In its original setting of classifier combination, where all classifier predictions are available on only moderately-many classifiers, dBCC’s shortcomings are unimportant.) Fortunately our proposed method EBCC suffers no such shortcomings (Section 4.3.2). We connect our proposed mixture model for classes to tensor decomposition (Sections 4.3.1, 4.3.2) and item clustering (Section 4.3.5), to help explain EBCC’s operation.

We conduct extensive experiments on 19 datasets with sources spanning music genre classification, news named entities labeling, movie review and tweet sentiment analysis. Compared to 10 state-of-the-art benchmark methods, EBCC achieves the highest mean accuracy (Section 4.4).

4.2 Preliminaries

In this section, we first define the notation, then discuss two representative Bayesian models for crowdsourcing aggregation.

Notation. Assume there are W workers who classify N items into K categories. Let $z_i$ be the latent true annotation of item $i$, $y_{ij}$ the label that worker $j$ assigns to item $i$, $W_i$ the set of workers who have labelled item $i$, and $N_j$ the set of items that worker $j$ has labelled. We use the capitalized letter of a variable to denote the collection of all such variables, for example, $Z \{z_1, z_2, \ldots, z_N\}$.

4.2.1 Bayesian Classifier Combination (BCC) Models

In Section 2.1.1 we briefly describe the BCC model. Here we elaborate on the model further. The BCC model (Kim and Ghahramani, 2012) was proposed for unsupervised ensembling of discrete outputs from several black-box classifiers. It has been successfully used in crowdsourcing
aggregation by making the analogue that workers are black-box classifiers and labels are their
discrete outputs (Simpson et al., 2013). The BCC model has several variants. Here we discuss two
representative ones, namely independent BCC (iBCC) and dependent BCC (dBCC).

Independent BCC

The iBCC model is a directed graphical model which assumes that given the true label $z_i$ of an
item, worker labels to item $i$ are generated independently by different workers,

$$p(y_{i1}, \ldots, y_{iW} | z_i) = \prod_{j=1}^{W} p(y_{ij} | z_i).$$

We refer to this as the \textit{worker conditional independence assumption}.

Furthermore, $p(y_{ij} = l | z_i = k) = v_{jkl}$ is assumed invariant to items. We denote the parametri-
sation of $p(y_{ij} | z_i = k)$ as $\vec{v}_{jk} = (v_{j,k}, v_{j,k^2}, \ldots, v_{j,k^K})$.

An important property of Equation (4.1) is that in the case that not all workers have labelled
item $i$, the likelihood of its observed labels $\{y_{ij} | j \in W_i\}$ can be calculated easily by marginalising
those unobserved labels $\{y_{ij} | j \notin W_i\}$ out,

$$p(\{y_{ij} | j \in W_i\} | z_i) = \sum_{(y_{ij})_{j \notin W_i}} p(y_{i1}, \ldots, y_{iW} | z_i) = \sum_{(y_{ij})_{j \notin W_i}} \prod_{j=1}^{W} p(y_{ij} | z_i) = \prod_{j \in W_i} p(y_{ij} | z_i).$$

The iBCC model is depicted in Figure 4.1. Apart from how $y_{ij}$’s are generated, it assumes that
$z_i \sim \text{Categorical}(\vec{\tau})$, then $\vec{\alpha}_{jk}$ and $\vec{\tau}$ are all random vectors that parameterise categorical distributions, and are drawn from Dirichlet($\vec{\beta}_k$) and Dirichlet($\vec{\alpha}$) respectively. The joint distribution
A Model Capturing Worker Correlations

\[ p(Y, Z, V, \tau | \alpha, \beta) = \prod_i p(z_i | \bar{\tau}) \prod_{j \in W_i} p(y_{ij} | z_i, V_j) \cdot \text{Dir}(\bar{\tau} | \bar{\alpha}) \prod_k \text{Dir}(\bar{\beta}_k | \bar{\beta}_k) \]

\[ \propto \prod_i \tau_i \prod_{j \in W_i} v_{jz_i} y_{ij} \prod_k \tau_k^{\alpha_k - 1} \cdot \prod_k \prod_l \tau_k^{\beta_{kl} - 1}. \]

The iBCC model is a popular extension to the DS model (Dawid and Skene, 1979) and has been independently re-discovered and implemented using Gibbs sampling (Kim and Ghahramani, 2012; Zhao et al., 2012), mean-field variational Bayes (Simpson et al., 2013; Felt et al., 2015), and expectation propagation (Venanzi et al., 2014). Despite its popularity, underlying independence assumptions prevent the model from capturing correlations between labels from different workers—a serious limitation as we will show.

Dependent BCC

dBCC is an undirected graphical model proposed to overcome the above limitation. In contrast to Equation (4.1), dBCC uses a Markov network to model the dependence between \( y_{ij} \)’s,

\[ p(\{y_{i1}, y_{i2}, \ldots, y_{iW}\} | z_i, V, U) = \frac{1}{C(V, U, z_i)} \exp \left\{ \sum_{1 \leq j < j' \leq W} u_{jj'} y_{ij} y_{ij'} + \sum_{j=1}^W v_{jz_i} y_{ij} \right\}, \quad (4.2) \]

where \( C(V, U, z_i) \) is a partition function that normalises the exponential part; \( U \) and \( V \) are two matrices of shape \((W, K, K)\) and \((W, K)\) respectively; \( u_{jj'} \) relates worker \( j \) and \( j' \), the larger it is the more likely worker \( j \) and \( j' \) assign \( l \) and \( l' \) to the same item; \( v_{jkl} \) relates \( y_{ij} \) and \( z_i \), the higher it is the more likely worker \( j \) labels a class-\( k \) item as \( l \). Since \( v_{jkl} \) appears in the exponent, it is similar to the log of the \( v_{jkl} \) in the iBCC model, consequently the constraint that \( \sum_l v_{jkl} = 1 \) is removed. The dBCC model further assumes \( u_{jj'} \) and \( v_{jkl} \) are drawn from Gaussian distributions \( \mathcal{N}(0, \sigma_u^2) \) and \( \mathcal{N}(0, \sigma_v^2) \) respectively. The generation of \( z_i \) is the same as in iBCC.

Note that Equation (4.2) is the full joint distribution over labels from all workers to item \( i \), but in practice we may only observe labels from a small set of workers, then marginalisation of the
4.3 The Proposed Model

4.3.1 Fitting the Joint Distribution over Worker Labels

We begin with a toy example to illustrate the relation between modelling the correlation between workers and tensor decomposition. Suppose there are two workers A and B who have labelled 10 class-0 items, the labels they generate are shown in Table 4.1.

<table>
<thead>
<tr>
<th></th>
<th>10 items (z = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker A</td>
<td>1 1 1 1 1 0 0 0 0 0</td>
</tr>
<tr>
<td>worker B</td>
<td>1 1 1 1 0 1 0 0 0 0</td>
</tr>
</tbody>
</table>

The joint distribution over their labels is

\[
p(y_A, y_B \mid z = 0) = \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{bmatrix} \begin{bmatrix} y_A = 0 \\ y_A = 1 \end{bmatrix} \begin{bmatrix} y_B = 0 \\ y_B = 1 \end{bmatrix}.
\]

For each worker, the marginal distribution is \([0.5, 0.5]\), then following Equation (4.1), we calculate the outer product of two marginal distributions and obtain a very poor approximation to the joint.
This is partly due to being constrained to rank-1 approximations owing to the worker conditional independence assumption. If using rank-2 approximation, we could obtain a far better approximation,

\[ p(y_A, y_B | z = 0) \approx \frac{1}{2} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \otimes \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0.41 & 0.09 \\ 0.09 & 0.41 \end{bmatrix}, \]

where \( \otimes \) is the tensor product. In general, the joint worker label distribution of more workers can be approximated by a linear combination of more rank-1 tensors, known also as tensor rank decomposition (Hitchcock, 1927), i.e.,

\[ p(y_1, \ldots, y_W | z) \approx \sum_{m=1}^{M} \pi_{km} \tilde{v}_{1km} \otimes \cdots \otimes \tilde{v}_{Wkm}. \]

This approach is more flexible than the Markov network in dBCC as the quality of approximation can be controlled by the number of components \( M \) instead of being constrained to a fixed capacity. Since the number of parameters is \( O(WK^2M) \) which is linear in \( W \) instead of quadratic, this approach scales to large cohort of workers unlike dBCC.

### 4.3.2 Integrating with Tensor Decomposition

We interpret the tensor decomposition as a mixture model where \( \tilde{v}_{1km} \otimes \cdots \otimes \tilde{v}_{Wkm} \) are the mixture components and \( \pi_{km} \) the mixture weights, so that we have

\[ p(y_1, \ldots, y_W | z) = \sum_{m=1}^{M} p(g = m | z) \prod_{j=1}^{W} p(y_j | z, g = m). \]  

Here \( g \) is an auxiliary latent variable used for indexing mixture components. We treat the \( M \) components under class \( k \) as its \( M \) subtypes and use subtypes to explain the correlation between worker labels given class \( k \). For example, in Table 4.1, the first 4 items and the last 4 items could be two subtypes under class 0: a difficult subtype and an easy subtype. This can explain the fact that both workers misclassify the first 4 items and correctly classify the last 4. The remaining 2 items in the middle can be treated as half-difficult and half-easy, thus could belong to two subtypes with probability \([0.5, 0.5]\). The mixture components capture the worker’s different behaviours under
4.3 The Proposed Model

4.3.3 The Generative Process and Joint Distribution

Based on the iBCC model, we add the mixture weight and components index variables $\vec{\pi}_k$ and $g_i$ and enlarge $\vec{v}_{jkm}$ $M$ times to capture worker behaviour under different subtypes. There are $K \times M$ subtypes in total, and we assume item $i$ belongs to the $g_i$-th subtype of class $z_i$. The proposed model is shown in Figure 4.2 and its generative process is:

1. for $k$ in $1 \ldots K$
   
   (a) $\vec{\pi}_k|a_\pi \sim \text{Dir}(a_\pi 1_M)$

   (b) for $m$ in $1 \ldots M$

   
   $p((y_j)_{j \in W}|z) = \sum_{(y_j)_{j \in W}} p(y_1, \ldots, y_W|z) = \sum_{m=1}^{M} p(g = m|z) \prod_{j \in W} p(y_j|z, g = m)$,

   by using the fact that $\prod_{j \in W} \sum_{y_j} p(y_j|z, g) = 1$
for \( j \) in 1 \ldots W

\[
\bar{\pi}_{jkml} | \bar{\beta}_k \sim \text{Dir}(\bar{\beta}_k)
\]

2. \( \bar{\tau} | \bar{\alpha} \sim \text{Dir}(\bar{\alpha}) \)

3. for \( i \) in 1 \ldots N

(a) \( z_i | \bar{\tau} \sim \text{Cat}(\bar{\tau}) \)

(b) \( g_i | \bar{\tau}_{z_i} \sim \text{Cat}(\bar{\tau}_{z_i}) \)

(c) for \( j \) in \( \mathcal{W}_i \)

- \( y_{ij} | \bar{\tau}_{j;g_i} \sim \text{Cat}(\bar{\tau}_{j;g_i}) \)

Following the generative process, the joint distribution is

\[
p(\pi, V, \tau, Z, G, Y | a_\pi, a, \beta) = p(\tau | a_\pi) p(V | \beta) \cdot p(\pi | a) p(Z | \tau) p(G | \pi, Z) p(Y | Z, G, V)
\]

\[
\propto \prod_k \prod_m \pi^{\delta_{km}-1} \cdot \prod_j \prod_m \prod_l v^{'j;km}_{j;kl} \cdot \prod_k \tau^{a_k-1} \cdot \prod_i \tau_{z_i} \cdot \prod_i \pi_{z_i;g_i} \cdot \prod_i \prod_j \prod_{l \in \mathcal{W}_i} v^{j;g_i,y_{ij}}.
\]

\[
= \prod_k \prod_m \pi^{\delta_{km}-1} \cdot \prod_j \prod_m \prod_l \beta^{a_k-1+\sum_i 1[z_i=k,g_i=m]} \cdot \prod_k \tau^{a_k-1+\sum_i 1[z_i=k]} \cdot \prod_j \prod m \prod l v^{j;km}_{j;kl}.
\]

4.3.4 The Inference Algorithm

The goal of inference is to find the most likely \( Z \) given the worker labels \( Y \) and all hyperparameters, i.e. \( \arg\max_Z p(Z | Y, a_\pi, a, \beta) \). However, due to \( G \) being coupled with \( \pi \) and \( V \) given \( Z \), a direct expectation maximisation algorithm is not applicable. We adopt a mean-field variational approach that seeks to find a distribution \( q \) that approximates \( p(\tau, Z, G, \pi, V | Y, a_\pi, a, \beta) \) so that the following holds

\[
\arg\max_Z p(Z | Y, a_\pi, a, \beta)
\]

\[
= \arg\max_Z \sum_G \int p(\tau, Z, G, \pi, V | Y, a_\pi, a, \beta) \, d\tau d\pi dV
\]

\[
\approx \arg\max_Z \sum_G \int q(\tau, Z, G, \pi, V) \, d\tau d\pi dV
\]
4.3 The Proposed Model

\[ \text{arg max}_Z q(Z). \]

Where \( q \) is assumed to be factorised as

\[ q(\tau, Z, G, \pi, V) = \text{Dir}(\vec{\nu} | \vec{\nu}) \cdot \prod_i q(z_i, g_i) \cdot \prod_k \text{Dir}(\vec{\eta}_k | \vec{\eta}_k) \cdot \prod_k \prod_m \prod_j \text{Dir}(\vec{\mu}_{kmj} | \vec{\mu}_{kmj}). \]

Since the joint distribution is fully factorised in \( q \), it’s easy to solve \( \text{arg max}_Z q(Z) \) by finding \( k \) that maximises every individual \( q(z_i = k) \). In summary, inference follows

\[ \hat{z}_i = \text{arg max}_k q(z_i = k). \]

Let \( \rho_{ikm} = q(z_i = k, g_i = m) \) and \( \gamma_{ik} = q(z_i = k) \), then follow the standard mean-field variational Bayes steps, we can derive the update rules shown below

\[ \rho_{ikm} \propto \exp \left\{ \mathbb{E}_q \log \tau_k + \mathbb{E}_q \log \pi_{km} + \sum_{j \in V_i} \mathbb{E}_q \log v_{kmjy_{ij}} \right\} \]

\[ \gamma_{ik} = \sum_m \rho_{ikm} \]

\[ v_k = \alpha_k + \sum_i \gamma_{ik} \]

\[ \eta_{km} = \alpha_{\pi} + \sum_i \rho_{ikm} \]

\[ \mu_{jkm} = \beta_{kl} + \sum_{i \in N_j} \rho_{ikm} \mathbf{1}[y_{ij} = l]. \]

The expectations are calculated as follows

\[ \mathbb{E}_q \log \tau_k = \psi(v_k) - \psi(\sum_k v_k) \]

\[ \mathbb{E}_q \log \pi_{km} = \psi(\eta_{km}) - \psi(\sum_m \eta_{km}) \]

\[ \mathbb{E}_q \log v_{jkm} = \psi(\mu_{jkm}) - \psi(\sum_l \mu_{jkm}). \]
where Ψ(·) is the digamma function. The Evidence Lower BOund (ELBO) is

$$\mathbb{E}_q \log p(\tau, Z, G, Y, \pi, V | a, \alpha, \beta) - \log q(\tau, Z, G, \pi, V)$$

$$= \sum_k (v_k - 1)\mathbb{E}_q \log \tau_k + \sum_k \sum_m (\eta_{km} - 1)\mathbb{E}_q \log \pi_{km} + \sum_j \sum_k \sum_m (\mu_{jkm} - 1) \log v_{jkm}$$

$$+ H(\text{Dir}(\vec{\tau} | \vec{\nu})) + \sum_i H(q(z_i, g_i)) + \sum_k H(\text{Dir}(\vec{\pi}_k | \vec{\eta}_k)) + \sum_j \sum_k \sum_m H(\text{Dir}(\vec{v}_{jkm} | \vec{\mu}_{jkm})),$$

where $H(\cdot)$ denotes entropy. The ELBO lower bounds $p(Y | a, \alpha, \beta)$, so is considered a criterion of how well $q$ approximates $p$. Because of the factorisation assumption of $q$, it has one single mode; owing to properties of KL divergence, minimizing $KL(q || p)$ will see $q$ approximate one mode in $p$. However, $p$ is unlikely to have only one mode, therefore one has to run the algorithm many times with different initialisations and pick the best $q$ based on ELBO.

### 4.3.5 Comparison to Item Clustering

Both iBCC and our proposed EBCC models can be considered as clustering methods. For an item, although we have no information about its content, its worker labels serve as features. The only unusual thing is that the feature vector of an item may have missing values due to that not all workers have labelled all items. Fortunately, the generative distribution in iBCC and EBCC, as shown in Equation (4.1) and (4.3), can handle this by marginalising out missing values.

Figure 4.3 shows the distribution over worker labels from 3 workers on 1600 items where all workers have labelled all items, where •• denote 300 items and •● 100 items. Colors indicate the majority voting aggregation results. Note that this example is symmetric with respect to workers, so all workers are equally good and the majority voting aggregation is reasonable. We run both iBCC and EBCC ($M = 3$) on this toy dataset. iBCC fits two clusters with their centroids at 000 and 111,

$$\begin{bmatrix} .99 \\ .01 \end{bmatrix} \otimes \begin{bmatrix} .99 \\ .01 \end{bmatrix} \otimes \begin{bmatrix} .99 \\ .01 \end{bmatrix} + 88\% \begin{bmatrix} .43 \\ .57 \end{bmatrix} \otimes \begin{bmatrix} .43 \\ .57 \end{bmatrix} \otimes \begin{bmatrix} .43 \\ .57 \end{bmatrix}.$$
Figure 4.3: A toy example for item clustering on a cube.

flatness, which makes the latter much sharper. EBCC, on the other hand, is flexible enough to fit all four natural clusters,

$$
23\% \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} + 77\% \begin{cases} 
\frac{1}{3} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \\
+ \frac{1}{3} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \\
+ \frac{1}{3} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \otimes \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}
\end{cases},
$$

with three clusters grouped together under class 1. The shapes of four clusters are also similar which is reasonable due to the symmetry of the distribution.

Arguably, iBCC can also fit all four clusters if we relax the constraint that the number of learned clusters has to be the same as the number of classes. We call the relaxed iBCC the Item Clustering model (IC). IC can learn a rectangle $K' \times K$ confusion matrix for every worker, where $K'$ is the number of clusters ($K' > K$). We run IC on the same toy dataset and find it finds the same four clusters as EBCC does, with the portions being $[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$.

However, IC doesn’t model the latent true labels of items, so after obtaining the clustering results, post-processing is required to map $K'$ clusters to $K$ classes. Another drawback is that prior knowledge, such as workers are better than random guessing, can’t be encoded in IC because the true label of every cluster is unknown during inference.

Therefore, our proposed EBCC is superior to IC, as the hierarchical generative process of true labels $z_i$ and clusters (subtypes, $g_i$) defines the mapping from clusters to classes, so that it can directly encode prior knowledge for every class and doesn’t require any post-processing.
4.4 Experiments

Initialization. We first initialise $\gamma_{ik}$ by majority voting, i.e. $\gamma_{ik} = \frac{1}{|W_i|} \sum_{j \in W_i} 1[y_{ij} = k]$, then multiply it with a random vector drawn from $\text{Dir}(1_M)$ to initialise $\tilde{\rho}_{ik}$. For every dataset, we run the algorithm 10 times with different random initialisations, and pick the solution with the highest ELBO.

Hyper-parameter settings. We set $a_\pi = 0.1/M$ to encourage sparsity of clusters; and $\beta_{kk} = 4$, $\beta_{kk'} = 1, k \neq k'$ to encode that we believe workers are better than random guessing. This is equivalent to assuming that every worker has correctly labelled 4 items under every class, and has made all kinds of mistakes once, i.e. labelling a class-$k$ item as $k'$, $k \neq k'$. We explore two strategies to initialise $\alpha_k$: (1) set $\alpha_k = 1$ to make the Dirichlet prior for $\tilde{\tau}$ uninformative; (2) set $\alpha_k = \sum_i \gamma_{i(k)}^{(0)}$ where $\gamma_{i(k)}^{(0)}$ is the MV initialisation for $\gamma_{ik}$. The intuition is that MV can provide a reliable estimate of the class portion in the dataset. We use a superscript $\text{Emp.}$ to indicate that the second strategy is used.

Number of components. In real-world datasets, most workers only label a few items so the chance is low that two workers have labelled a sufficient number of items for reliably estimating their correlation. We find in practice that the number of learned components per class is usually smaller than 10. We run experiments on real-world datasets with $M = 5, 20$ and synthetic datasets with $M = 2, 5$.

4.4.1 Synthetic Datasets

We run MV, iBCC, and EBCC on a synthetic dataset to show that correlations between worker labels can be captured and exploited to assist truth inference. In the dataset, there are 5 workers classifying items into two categories. Every class has two subtypes and all subtypes are distributed evenly with exactly 25% items belonging to each. The first two workers’ performances vary on different subtypes with an average accuracy of 50% per class, while the last three workers perform consistently with an accuracy of 70% across subtypes. Table 4.2 summarises the settings.

The first two workers are likely to agree with each other on class-0 items, but generate different labels to class-1 items. This observation is helpful for inferring the truth, and we expect the EBCC
Table 4.2: Accuracy of 5 workers on different subtypes.

<table>
<thead>
<tr>
<th>true label</th>
<th>subtype g</th>
<th>worker1</th>
<th>worker2</th>
<th>worker3</th>
<th>worker4</th>
<th>worker5</th>
<th>portion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>0.9</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>25%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>25%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.9</td>
<td>0.1</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>25%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.9</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>25%</td>
</tr>
</tbody>
</table>

Figure 4.4: MV, iBCC, and EBCC on synthetic datasets.

The performance of MV is very stable at 72.0% while iBCC starts with 67.4% then surpasses MV when #items > 500 and finally converges to 78.4%. The first two workers are completely random on the class level with their confusion matrices being \([0.5 \ 0.5 \ \ 0.5 \ 0.5]\). iBCC can estimate their confusion matrices reliably as more data is available and then effectively ignore them during the inference. That’s why iBCC converges to the theoretic MV performance of the last three workers: \(0.7^3 + 3 \cdot 0.7^2 \cdot 0.3 = 0.784\). EBCC has the ability to capture the special correlation between the first two workers, therefore achieves the highest accuracy 85.9% and consistently outperforms MV and iBCC. There is little difference between \(M = 2\) and \(M = 5\) for EBCC, which suggests insensitivity to over-parametrisation.
Surprisingly, the performance of both EBCC models gets worse when \( \#\text{items} > 50k \) with their variance increasing and decreasing when \( 50k \leq \#\text{items} \leq 200k \). We have examined the estimates of parameters and found that EBCC actually converges to the same solution as iBCC does. But the ELBO of iBCC’s solution is lower than the ELBO of the correct solution, therefore we conclude that this is an optimisation problem and that our algorithm gets stuck on bad local optima. As is commonly known, this is a weakness for batch learning on large-scale datasets, therefore we believe a stochastic optimisation algorithm would likely mitigate the problem \cite{LeCun}. We run the experiments following the setup described in Chapter 2 to compare our proposed methods with 10 benchmarks on 19 datasets. We also include a mean-field variational inference implementation of iBCC (iBCC-MF) to compare with our proposed method implemented by the same technique. We run EBCC with \( M = 2, 5 \) and two different settings for \( \beta_k \) as discussed in hyper-parameter settings.

**Results.** We adopt the same experimental settings described in Section 2.2: 10 benchmarks and 19 datasets. Additionally, we also implement a mean-field variational inference algorithm for iBCC called iBCC-MF. Figure 4.5 shows the mean accuracy of every method on 19 datasets. Table 4.3 demonstrates the Wilcoxon significance test for all methods against MV. Figure 4.6 presents the accuracies on all datasets.

\( \text{EBCC}_{M=5} \) and \( \text{EBCC}^{\text{Emp.}}_{M=5} \) have the highest mean accuracy of 84.4\%, outperforming the best existing method iBCC-MF which achieves 83.6\%. Overall, confusion-matrix-based probabilistic models (DS, iBCC, CBCC, LFC) perform similarly with mean accuracy within range \([82.9\%, 83.5\%]\), followed by three “1-coin” models, namely, CATD (83.3\%), GLAD (83.0\%), ZC (83.0\%). This name arises from these models only learning a single parameter per worker to capture their accuracy. However, a worker may behave differently across classes or subtypes, which “1-coin” models cannot capture. Due to this reason, all “1-coin” models are worse than confusion matrix based methods on bird and trec. But confusion matrix based methods also fail when their parameters are now well regularised (DS and LFC when there are many workers who have annotated very few items) or the assumption doesn’t hold (CBCC when a few clusters of confusion matrices can’t cover different worker behaviours). The fact that iBCC works the best shows the benefit of integrating out variables instead of maximising over them, and the robustness of simple
4.4 Experiments

Figure 4.5: Mean accuracy of 10 benchmarks and EBCC models on 19 real-world datasets. The baseline of the bar plot is the average of all methods’ mean accuracies.

Table 4.3: One-sided Wilcoxon signed rank test results (EBCC against MV)

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean accuracy</th>
<th>N_r</th>
<th>W_</th>
<th>Significance level</th>
<th>Approx. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV</td>
<td>0.819557</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZC</td>
<td>0.829444</td>
<td>18</td>
<td>64</td>
<td></td>
<td>0.174551</td>
</tr>
<tr>
<td>GLAD</td>
<td>0.830476</td>
<td>19</td>
<td>47</td>
<td>✓</td>
<td>0.026703</td>
</tr>
<tr>
<td>DS</td>
<td>0.830907</td>
<td>19</td>
<td>71</td>
<td>✓</td>
<td>0.167069</td>
</tr>
<tr>
<td>Minimax</td>
<td>0.814085</td>
<td>19</td>
<td>81</td>
<td></td>
<td>0.286584</td>
</tr>
<tr>
<td>iBCC</td>
<td>0.835245</td>
<td>18</td>
<td>52</td>
<td>✓</td>
<td>0.072291</td>
</tr>
<tr>
<td>CBCC</td>
<td>0.829098</td>
<td>19</td>
<td>71</td>
<td></td>
<td>0.167069</td>
</tr>
<tr>
<td>LFC</td>
<td>0.833811</td>
<td>19</td>
<td>69</td>
<td></td>
<td>0.147712</td>
</tr>
<tr>
<td>CATD</td>
<td>0.833413</td>
<td>19</td>
<td>56</td>
<td>✓</td>
<td>0.058272</td>
</tr>
<tr>
<td>CRH</td>
<td>0.805379</td>
<td>19</td>
<td>91</td>
<td></td>
<td>0.436059</td>
</tr>
<tr>
<td>EBCC_M=5</td>
<td>0.843553</td>
<td>19</td>
<td>35</td>
<td>✓</td>
<td>0.007878</td>
</tr>
<tr>
<td>EBCC_M=20</td>
<td>0.835630</td>
<td>18</td>
<td>44</td>
<td>✓</td>
<td>0.035355</td>
</tr>
<tr>
<td>EBCC_Emp_M=5</td>
<td>0.843924</td>
<td>19</td>
<td>27</td>
<td>✓</td>
<td>0.003105</td>
</tr>
<tr>
<td>EBCC_Emp_M=20</td>
<td>0.842772</td>
<td>19</td>
<td>28</td>
<td>✓</td>
<td>0.003507</td>
</tr>
</tbody>
</table>
assumption that every worker has his own confusion matrix.

Apart from mean accuracy, we identify a failure case for EBCC\textsubscript{M=20}. Its accuracy on the web dataset is 55.4%, which is much lower than EBCC\textsubscript{M=5}’s on the same dataset. After examining the estimates of parameters, we found that it learns a very skewed class distribution $\vec{\tau}$ with the value of one class (A) being zero, consequently it doesn’t classify any items into class A. Further analysis shows that many workers confuse class A with another two classes, which absorb all clusters from class A in the learning process. Our solution is to set the prior of $\vec{\tau}$ to be the class distribution estimated by majority voting to encode our belief. As shown in Figure 4.5 and 4.6, EBCC\textsubscript{Emp. M=20} is better than EBCC\textsubscript{M=20} and is close to $M = 5$.

4.5 Conclusion

We have developed a Bayesian model for aggregating crowdsourced labels that is capable of capturing correlations between labels of different workers. Our model, enhanced Bayesian classifier combination (EBCC), achieves this by introducing a mixture of subtypes per true class, while worker performance varying per subtype induces inter-worker correlation. The efficacy of EBCC is demonstrated in extensive experiments on synthetic data, which confirms the importance of worker correlation, and over a suite of 19 crowdsourced datasets drawn from a wide variety of domains, where EBCC achieves highest mean accuracy. We intend to explore the application of stochastic optimisation to EBCC in future work, which should improve the method’s robustness.
Figure 4.6: Accuracies of all methods on 19 real-world datasets.
Chapter 5
A Source Conditioning Model

In previous chapters, we assume that the only information we have is worker labels, but in many situations more information is available. In this chapter, we propose a novel model for aggregating binary labels in the case that item content, e.g. vector representation, is available. The model infers the ground truth and worker reliabilities jointly, making use of a Boltzmann machine prior to bias similar instances towards sharing the same label, with a per-worker confusion matrix to model the worker reliability. Experimental results on two datasets show that our proposed approach can outperform both MV and a state-of-the-art Gaussian process classification model. Moreover, we show further gains when integrating a proposed active learning heuristic.

5.1 Introduction

In recent years crowdsourcing has become a promising technique for collecting large amounts of labelled data at a low cost. However, due to low quality annotations, items are typically redundantly labelled by several different workers, with labels aggregated subsequently. Unfortunately, naive methods such as majority voting are inefficient for uncovering true annotations. This approach assumes that all workers who pass the test are equally good, and all who fail are useless. However this is rarely true: individuals are observed to vary significantly from one another—and the ground truth—and there is often weak signal in the outputs of rejected workers. To truly capture the latent ground truth requires knowing the reliability of each worker, such that they can be non-uniformly weighted in estimating the truth. In this chapter, we propose an approach that infers the ground truth and worker reliabilities jointly, making use of a Boltzmann machine prior to bias similar instances towards sharing the same label, with a per-worker reliability likelihood to model
both the correlation between true annotations and the levels of expertise of annotators.

Keeping crowdsourcing cost as low as possible calls for an active learning strategy that orches-
trates the data selection and worker assignment so that resources are spent more efficiently. We
explore two active learning approaches which select the most uncertain item and then the worker
most likely to label the item correctly or the worker who would reduce uncertainty the most given
that worker’s label to the most uncertain instance at the moment, which we show greatly reduces
the requirement for labelled data. In our thorough experimental study, we reach the accuracy of
MV when our method selects about a third of the data, and substantially exceed the MV accuracy
when more data is available. Moreover we show that our technique is robust to adversarial workers
and those attempting to cheat, effectively learning to invest the annotation budget in workers with
informative outputs.

We make the following contributions:

• We propose a novel model that combines a Boltzmann machine with confusion matrices to
model both the correlation between true annotations and the levels of expertise of annotators.

• We explore two active learning heuristics for our model that actively select the item with
highest uncertainty and assign it to a suitable worker to label.

• We conduct extensive experiments on two real world crowdsourcing datasets on relevance
judgements and sentiment polarity, demonstrating that our proposed model consistently out-
performs baseline and state-of-the-art models. The improvement is extended with active
learning.

The remainder of this chapter is organized as follows: In Section 5.2 we review the works
of inferring ground truth from multiple annotator data and related active learning literature. Sec-
tion 5.3 states the problem. In Section 5.4 and Section 5.5 we describe our proposed BMMA
model and its inference algorithm. In Section 5.6 we present two active learning heuristics we de-
sign for BMMA. Experimental results are reported in Section 5.7. Finally, Section 5.8 concludes
the chapter.
5.2 Related Work

While most existing models infer the true annotations without using instance content, we focus on the scenario that the content of the instances is available for use within inference. Models designed for this scenario typically consist of a classifier to model the correlation between true annotations, and confusion matrices to model workers’ reliabilities.

Raykar et al. (2010) were the first to jointly infer the true annotations and learn a classifier in our setting. The proposed probabilistic model consists of a logistic regression classifier (LRC) and confusion matrices. The expectation maximization (EM) algorithm is used to fit confusion matrix parameters. Rodrigues et al. (2013) also propose a probabilistic model with LRC component, but instead of using a confusion matrix to model worker reliability, they take a simpler approach by assuming that every worker is a mixture between a LRC (providing correct answers) and a uniform random response, with these parameters learned using EM. In this way, every worker’s reliability is modeled by a single parameter and thus the number of parameters is reduced regardless of the number of classification categories used, further avoid the potential of overfitting.

Yan et al. (2010) assume that each worker’s reliability is dependent on the data instance, i.e., some instances may be more difficult for a worker than others. They use a sigmoid function to model the difficulty term for every worker. Since the model becomes complicated, they adopt a gradient descent algorithm for training.

In contrast to the above, Rodrigues et al. (2014) use a Gaussian process classifier with confusion matrices to model the instances, the true annotations, and the labels from workers. A Gaussian process is a non-parametric model endowing it with greater capacity to fit the data, in contrast to logistic regression which is parametric.

To achieve better performance with fewer worker labels, several of the above papers have included an active learning algorithm for iteratively selecting the instances to be labelled and also selecting the worker. Yan et al. (2011) find the optimal instance such that there exists an annotator that can label it with the maximal confidence. In contrast, Rodrigues et al. (2014) first select the instance closest to the decision boundary of the classifier, and then select the annotator who is more likely to label it correctly.
Table 5.1: Notation used in this chapter.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>the number of instances</td>
</tr>
<tr>
<td>$i$</td>
<td>the index of an instance, $i \in {1, 2, \ldots, N}$</td>
</tr>
<tr>
<td>$W$</td>
<td>the set of all workers</td>
</tr>
<tr>
<td>$j$</td>
<td>the index of a worker, $j \in W$</td>
</tr>
<tr>
<td>$W_i$</td>
<td>the set of workers who have labeled instance $i$</td>
</tr>
<tr>
<td>$D_j$</td>
<td>the set of instances labeled by worker $j$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>the feature vector for instance $i$</td>
</tr>
<tr>
<td>$z_i$</td>
<td>the true annotation for instance $i$, $z_i \in {-1, 1}$</td>
</tr>
<tr>
<td>$y_{ij}$</td>
<td>worker $j$’s label to instance $i$, $y_{ij} \in {-1, 1}$</td>
</tr>
<tr>
<td>$r$</td>
<td>true annotation, $r \in {-1, 1}$</td>
</tr>
<tr>
<td>$l$</td>
<td>worker label, $l \in {-1, 1}$</td>
</tr>
<tr>
<td>$\theta_{jlr}$</td>
<td>$P(x_{ij} = l</td>
</tr>
<tr>
<td>$n_{jlr}$</td>
<td>$#{i : x_{ij} = l, z_i = r}$</td>
</tr>
<tr>
<td>$n_{jr}$</td>
<td>$#{i : z_i = r}$ or defined as $\sum_l n_{jlr}$</td>
</tr>
<tr>
<td>$n^i_{jlr}$</td>
<td>$#{i' : x_{i'j} = l, z_{i'} = r}$</td>
</tr>
<tr>
<td>$k_{i,l'}$</td>
<td>weight between $z_i, z_{i'}$ in the Boltzmann prior</td>
</tr>
<tr>
<td>$\alpha_{lr}$</td>
<td>the parameter used in $\theta_{jlr}$’s prior</td>
</tr>
<tr>
<td>$\alpha_r$</td>
<td>$(\alpha_{-1,r}, \alpha_{+1,r})$</td>
</tr>
</tbody>
</table>

5.3 Problem Statement

Our goal is to infer the true annotations for all instances. In this paper, we restrict our focus to the scenario that (1) the content of all instances is available for use within inference; (2) true annotations and worker labels are binary\(^1\).

We evaluate inference algorithms according to two goals: (1) benchmark against expert annotators over data with worker labels (training set), which are assumed to be “gold” standard; (2) predictive accuracy on data without worker labels (test set).

The main variables of the problem are given in Table 5.1. The next sections describe our approach which involves the last few symbols defined.

\(^1\)However, our model extends easily for $k$-class outcomes, $k \geq 2$, by using $k \times k$ confusion matrices with Dirichlet priors and a multi-valued Boltzmann machine (Lin and Lee, 1995).
5.4 Proposed Model

In this section, we present our proposed model *Boltzmann Machine with Multiple Annotators* (BMMA). The inference algorithm for this model is described in Section 5.5.

5.4.1 Intuition

Our model is a combination of a Boltzmann machine for modelling the correlations between true annotations and confusion matrices for a generative model of worker reliability. We now describe each component and explain the intuition behind our modelling choices.

**Confusion Matrix**  A confusion matrix parameterizes the distributions over worker labels conditioned on true annotations. For binary worker labels and true annotations, the confusion matrix is a $2 \times 2$ matrix. Each worker has a different confusion matrix. See Figure 5.1 for examples. As we can see, a confusion matrix is flexible enough to model workers’ reliabilities (annotation noise and bias), and we further assume that a worker’s reliability is invariant to instance.

Suppose that worker $j$’s confusion matrix is (d) in Figure 5.1 then we know worker $j$ will label negative instances correctly with probability 0.6 (specificity), positive instances with 0.7 (recall or sensitivity).

\[
P(y_{ij} = -1 | z_i = -1) = 0.6, \quad \forall i,
\]
\[
P(y_{ij} = +1 | z_i = +1) = 0.7, \quad \forall i.
\]

We seek to learn these parameters for each worker from data.

**Boltzmann Machine**  A Boltzmann machine defines a distribution over a set of binary variables. In our case, we use a Boltzmann machine to model the correlation between true annotations, thus we define

\[
P(Z) = \frac{1}{C} \exp \left( \sum_{j \neq i} k_{i,j} z_i z_j \right) \tag{5.1}
\]
where $C$ is a normalizing constant. The values for $z_i$’s are $\{-1, +1\}$ which has symmetry in contrast to using $\{0, 1\}$. In fact, since $z_i^2 = 1$, Equation (5.1) can be written in a more compact form

$$P(Z) = \frac{1}{C'} \exp \left( Z^T K Z \right)$$

(5.2)

where the diagonal terms in $K$ are arbitrary and can be absorbed in the normalizing constant $C'$.

The Boltzmann machine (Ackley et al., 1985) is a compelling prior for two reasons. First, Boltzmann machines are generative models able to learn the relationship between binary random variables. Typically, given a dataset containing binary vectors $\{Z^{(s)}\}_{s=1}^S$, learning algorithms can find values of $\{k_{i,j}\}_{i \neq j}$ that maximize the likelihood of generating $\{Z^{(s)}\}_{s=1}^S$. A large magnitude $k_{i,j}$ indicates that $z_i$ and $z_j$ often have the same or opposite values: $k_{i,j}$ can be interpreted as a measure of the similarity between $x_i$ and $x_j$. However, instead of using a Boltzmann machine to learn the distribution of $Z$, we use it in reverse: we use a Boltzmann machine with fixed weights to encode our prior belief on the binary variables $Z$, where the prior belief is that the more similar $x_i$ and $x_j$ are, the more likely $z_i$ and $z_j$ have the same value, which is known as the Cluttering Hypothesis (Chapelle et al., 2006). Such a Boltzmann machine can quantify the probability of a given label assignment, and thus given two label assignments, it can tell which one is more likely. For example, homogeneous labelling of clusters can be preferred over heterogeneous label assignments.

\footnote{For example given a dataset $\{Z^{(1)} = (1, 1, -1), Z^{(2)} = (1, -1, 1), Z^{(3)} = (-1, -1, 1)\}$, when the likelihood $P(Z^{(1)}) P(Z^{(2)}) P(Z^{(3)})$ is maximized, we desire that the Boltzmann machine can learn from the data that $z_2$ and $z_3$ always have opposite signs, and a large negative $k_{2,3}$.}
5.4 Proposed Model

Figure 5.2: Two possible label assignments to points.

Figure 5.2 illustrates two possible label assignments to 14 points where the two colors, grey and black, represent two classes. By simply setting \( k_{i, i'} \) to the reciprocal of the distance between \( x_i \) and \( x_{i'} \), the probability of the left assignment defined by the Boltzmann machine will be larger.

The second motivation is that Boltzmann machines are the generative counterpart of Hopfield networks (Hopfield, 1982) which use energy minima (or probability maxima in Boltzmann machines) to represent memories. This idea makes a content-addressable memory possible: an item can be accessed by just knowing part of it. For example, suppose the true annotations of the points are just the label assignment on the left in Figure 5.2, then if we lose the label of the bottom-left point, a Boltzmann machine can use the information of other points to restore the missing label, and in this case, it will restore the label correctly with high confidence.

5.4.2 The Joint Likelihood and Priors

Combining the confusion matrices and the Boltzmann machine, we define the joint distribution of binary variables \( \{ -1, +1 \} \) \( Y \) and \( Z \) as

\[
P(Y, Z | \theta) = \prod_{i=1}^{N} \prod_{j \in W_i} P(y_{ij} | z_i, \theta_j) P(Z)
\]

where \( P(Z) \) is the Boltzmann machine of Equation (5.2) and \( P(y_{ij} | z_i, \theta_j) \) is the \( 2 \times 2 \) confusion matrix for worker \( j \) as

\[
P(y_{ij} = l | z_i = r, \theta_j) = \theta_{jl}, \forall i \in D_j.
\]

Since the number of the parameters in confusion matrices is proportional to the number of workers, in order to avoid overfitting from insufficient worker labels, we add priors to \( \theta_{jl} \). A
natural choice is the Beta prior which is conjugate to Bernoulli $P(y_{ij}|z_i, \theta_j)$.

$$P(\theta|\alpha) = \prod_{j \in W} P(\theta_j|\alpha) \propto \prod_{j \in W} \prod_{r=\pm 1} \prod_{l=\pm 1} \theta^{\alpha_l r}_{jlr}^{-1},$$

where $\alpha$ are Beta hyper-parameters.

In summary, the full generative process is

1. For each worker $j$ and each label class $r$
   - $(\theta_{j,-1,r}, \theta_{j,1,r}) \sim \text{Beta}(\alpha_{-1,r}, \alpha_{1,r})$

2. Generate $Z$ from the Boltzmann machine $P(Z)$

3. For each item $i$ and each worker $j \in D_i$
   - Choose $y_{ij}$ from $P(y_{ij} = l|z_i, \theta_j) = \theta_{jlz_i}$

5.4.3 Parameter Settings

In our model, $K$ in the Boltzmann machine and $\alpha$ in the prior of $\theta$ are fixed hyper-parameters. $\alpha$ is a $2 \times 2$ matrix containing 4 free parameters, which are known as pseudo counts. We set the values for true negative and true positive larger than those for false negatives and false positives to encode the prior belief that worker labels are better than random guesses.

For the kernel $K$, we borrow the popular setting in Gaussian processes (GP) for that the form of Boltzmann machine $P(Z) \propto \exp(Z^T K Z)$ resembles that of GP $P(f) \propto \exp(f^T \Sigma^{-1} f)$. We choose the radial basis function (RBF) kernel so that

$$\Sigma_{i,j'} = \exp\left(-\frac{\|x_i - x_{j'}\|^2}{\sigma^2}\right), \quad (5.3)$$

where $\sigma^2$ is a free parameter, and then let $K = \Sigma^{-1}$. In this way, the Boltzmann machine works as a module performing least-squares classification which reduces binary classification to regression [Rasmussen 2004].

Since the latent truth $Z$ is discrete, a Boltzmann machine prior directly models $P(Z)$, but a GP prior $P(f)$ is defined over continuous variables $f$, therefore an extra link function $P(z_i|f_i)$.
is required for modelling the latent truth, so that \( P(Z) = \int \prod_z P(z_i|f_i)P(f) \, df \). Therefore the Boltzmann machine prior simplifies the formulation and makes the model more tailored to the task of aggregating discrete labels. Empirical results in Section 5.7 also show the advantage of using a Boltzmann machine prior.

Another natural choice is to set \( K = \Sigma \), however, this can be problematic for the Boltzmann machine. Since all elements in \( K \) are positive, the configurations with \( z_i \) all 1s or all \(-1\)s maximize \( P(Z) \propto \exp (Z^T K Z) \), which is not desirable. Even worse, when the number of instances \( N \) is very large, the above two cases will have much larger probabilities than other configurations. The Boltzmann machine will then dominate the confusion matrices, and these cases will become the most probable two globally. Fortunately, setting \( K = \Sigma^{-1} \) does not cause this degeneracy, because the inverse of \( K \) must contain both negative and positive values.

### 5.5 Inference

We first derive the inference algorithm for our proposed model BMMA and then introduce two practical tricks helpful for reducing the estimate variance and encouraging faster mixing.

#### 5.5.1 Collapsed Sampling for BMMA

The joint distribution with Beta priors for \( \theta \) is

\[
P(Y, Z, \theta | \alpha) \propto \prod_j \prod_r \prod_l \theta_{jlr}^{n_{jlr} + \alpha_{lr} - 1} P(Z).
\]

To estimate \( P(z_i) \) and \( \theta \), we first integrate out \( \theta \) so as to perform a collapsed Gibbs sampling for \( Z \), then estimate \( \theta \) based on the samples of \( Z \). After marginalizing over \( \theta \), the joint distribution becomes

\[
P(Y, Z | \alpha) = \int P(Y, Z, \theta | \alpha) \, d\theta \propto \left( \prod_j \prod_r \frac{\Gamma(n_{jlr} + \alpha_{lr})}{\Gamma(n_{jlr} + \alpha_r)} \right) P(Z).
\]
Then collapsed Gibbs sampling is based on

\[ P(z_i = r | Y, Z^{-i}, \alpha) \propto \left( \prod_{j \in W_i} \frac{n_{j|r}^{-i} + \alpha_{y_{j|r}}}{n_{j|r} + \alpha_r} \right) \exp \left( \sum_{i' \neq i} k_i z_{i'} \right). \]  

(5.4)

After obtaining samples \( \{Z^{(s)}\}_{s=1}^{S} \) from the posterior, the posterior mean estimate for \( \theta_{jl} \) is

\[
E_{\theta_{jl}|Y,\alpha} \theta_{jl} = \int \theta_{jl} P(\theta|Y,\alpha) \, d\theta = \sum_Z \theta_{jl} P(\theta|Y, Z, \alpha) P(Z|Y, \alpha) \, d\theta
\]

\[
\approx \frac{1}{S} \sum_{s=1}^{S} \int \theta_{jl} P(\theta|Y, Z^{(s)}, \alpha) \, d\theta = \frac{1}{S} \sum_{s=1}^{S} \int_{0}^{1} \theta_{jl} \frac{\Gamma(n_{jl}^{(s)} + \alpha_r)}{\prod_{r'} \Gamma(n_{jl'}^{(s)} + \alpha_{r'})} \prod_{r'} n_{jl'}^{(s)} + \alpha_{r'} \, d\theta_{jl}.
\]

\[
= \frac{1}{S} \sum_{s=1}^{S} n_{jl}^{(s)} + \alpha_{jl}.
\]

This mean estimate for worker reliability will be used in the active learning heuristics to select workers (Section 5.6).

### 5.5.2 Practical Tricks

In standard inference, after we obtain samples \( \{Z^{(s)}\}_{s=1}^{S} \), \( P(z_i = r) \) is estimated by

\[ P(z_i = r) \approx \frac{1}{S} \sum_{s} 1(z_i^{(s)} = r). \]

Alternatively, since we use Gibbs sampling to obtain samples from the posterior, we can maintain a record of the distributions that we sample \( z_i \) from, and use the Rao-Blackwell estimator which also yields unbiased estimates but with lower variance than the standard estimator (Nikulin, 2001). Let \( q(s)_{z_i} \) be the distribution that we sample \( z_i^{(s)} \) from, then \( P(z_i = r) \) can be estimated as

\[ P(z_i = r) \approx \frac{1}{S} \sum_{s} q(s)_{z_i = r}. \]

Besides the estimation process, there is an alternative single update rule for Gibbs sampling. In Gibbs sampling, let \( q^{(k)}(z_i) \) denote the distribution that \( z_i^{(k+1)} \) is sampled from at the \( k \)th iteration.
Then the transition matrix from $z_i^{(k)}$ to $z_i^{(k+1)}$ is

$$
\begin{bmatrix}
p_{-1\rightarrow -1} = q_- & p_{-1\rightarrow +1} = q_+ \\
p_{+1\rightarrow -1} = q_- & p_{+1\rightarrow +1} = q_+
\end{bmatrix},
$$

where $q_+ = 1 - q_- = q^{(k)}(z_i = +1)$. This transition matrix demonstrates a property of Gibbs sampling: it ignores the current state when generating the next state.

It is obvious that this single update satisfies the detailed balance condition,

$$q_- \times p_{-1\rightarrow +1} = q_+ \times p_{+1\rightarrow -1}.$$

However, the above is not the only transition matrix that satisfies the detailed balance condition. Without any loss of generality, suppose $q_+ > q_-$, then the following transition matrix also satisfies detailed balance,

$$
\begin{bmatrix}
p_{-1\rightarrow -1} = 0 & p_{-1\rightarrow +1} = 1 \\
p_{+1\rightarrow -1} = \frac{q_-}{q_+} & p_{+1\rightarrow +1} = 1 - \frac{q_-}{q_+}.
\end{bmatrix}
$$

This corresponds to the update in Metropolis-Hastings (MH) sampling if we choose a proposal distribution that always proposes the opposite state against the current one.

The MH update may be helpful to accelerate the mixing of the Markov chain since the probability of making a transition to the opposite state is magnified by a $1/ \max\{q_+, q_-\}$ factor and thus the state changes more rapidly than in standard Gibbs sampling. Furthermore, MH updates often run faster than Gibbs updates because sampling can be avoided in some cases, for example when $q_+ > q_-$ and current state is $-1$ then the state will transit to $+1$ unconditionally.

## 5.6 Active Learning Heuristics

We propose two active learning approaches, based on selecting the item with the highest uncertainty, and then requests a label for the item from either: the worker most likely to label the item correctly (best worker, BW), based on Rodrigues et al. (2014); or the worker who will most reduce the uncertainty over the item’s true label (largest variance, LV).
To bootstrap active learning, we request one label for every item, then perform inference to obtain an estimate of the worker reliabilities $\theta_{jlr}$ and the distribution of $z_i$ denoted by $q(z_i)$. For the workers that we haven’t obtained any labels from, we set their recall and specificity to be the mean estimates according to the Beta prior. We then iteratively request labels. At every iteration, there are four steps:

**Step 1** We use the variance of $z_i$ as the measure of uncertainty. The variance is calculated as $\text{Var}(z_i) = 1 - (\mathbb{E}z_i)^2$ where $\mathbb{E}z_i = q(z_i = 1) - q(z_i = -1)$. We select the item with highest variance

$$i_s = \arg \max_i \text{Var}(z_i).$$

**Step 2 (BW)** We select the worker $j_s$ who has the highest probability of labelling the item $i_s$ correctly,

$$j_s = \arg \max_j \sum_r P(y_{is} = r | z_{is} = r) P(z_{is} = r),$$

**Step 2 (LV)** We calculate the expected conditional variance of $z_i$ based on obtaining label $y_{ij}$ from each worker $j$ as

$$\mathbb{E}(\text{Var}(z_{is} | y_{ij})) = \sum_l \text{Var}(z_{is} | y_{ij} = l) P(y_{ij} = l)$$

where

$$P(y_{ij} = l) = \sum_r P(y_{ij} = l | z_{is} = r) P(z_{is} = r) = \sum_r \theta_{jlr} q(z_{is} = r)$$

and

$$\text{Var}(z_{is} | y_{ij} = l) = 1 - \left( \sum_r r q(z_{is} = r | y_{ij} = l) \right)^2$$

where

$$q(z_{is} = r | y_{ij} = l) \propto P(y_{ij} = l | z_{is} = r) q(z_{is} = r) = \theta_{jlr} q(z_{is} = r).$$
The idea of using $E(\text{Var}(z_i|y_{ij}))$ is to estimate the outcome of requesting label from worker $j$ for document $i$, i.e. how will the variance change after obtaining a new label, and an expectation is calculated because we don’t know the new label before requesting it.

We then select the worker minimising the expected conditional variance,

$$j_s = \arg \min_j E(\text{Var}(z_i|y_{ij})) .$$

**Step 3** Suppose we obtain a new label for item $i_s$ from worker $j_s$, then we update $q(z_{is})$ to be

$$q(z_{is} = r) \leftarrow q(z_{is} = r|y_{is}) \propto \theta_{j_s|y_{is}}r \cdot q(z_{is} = r) .$$

**Step 4** After obtaining $M$ new labels\( ^3 \) since the last update by the inference algorithm, we will perform inference again to update the estimate for $\theta_{jlr}$ and $q(z_i)$.

### 5.7 Experiments

In this section, we evaluate the effectiveness of our proposed algorithm and the baseline algorithms. We first describe the measurement of effectiveness. Then we compare the effectiveness of our proposed method BMMA with Majority Voting (MV) and GPC-MA (Rodrigues et al., 2014) on the TREC dataset for relevance judgments and the sentiment polarity dataset. Two active learners BMMA-BW and BMMA-LV are also compared with the above methods.

**Metric** We measure algorithm effectiveness by accuracy and area under curve (AUC) calculated based on their predictions and the true annotations (labels from experts). Algorithms may be compared on partial/full training set or a held out test set. Here the training set simply means all the worker labels and the instances of these labels, while the held out test set contains instances with no worker label.

Since randomness is introduced when partial sets are sampled, we compare the average accuracy over 50 simulations with different random seeds. To generate different sizes of partial sets, we sample the worker labels. If the size of the partial set is larger than or equal to $N$, we guarantee

\(^3\)In our experiments, we set $M$ to be $N/2$.\n
that every instance has at least one worker label. If the size of the partial set is smaller than \( N \), we enforce that every instance has either one worker label or no worker label.

**Parameter Settings**  We set \( \alpha = [3, 1, 3, 3] \) for the TREC dataset, which encodes our prior belief that every worker has a 75% sensitivity and a 75% specificity when we have no information about that worker. For the sentiment polarity dataset, since on average every worker has labelled 137 items, we scale the \( \alpha \) by a factor of 5. The length scale \( \sigma^2 \) in the RBF kernel (Equation 5.3) was empirically chosen as 0.3 for both datasets. For the inference algorithm, we estimate parameters based on the first 50 samples after 10 burn-in iterations on both datasets.

### 5.7.1 TREC dataset

**Dataset**  We use the dataset for TREC Crowdsourcing Track 2011 Task 1 (Lease and Kazai, 2011). The whole dataset is comprised of 30 ‘topics’ which we treat as separate datasets. Each topic has a given query and the document relevance judgments for that topic by multiple annotators. On average, each small dataset contains about 100 documents, and about 1600 labels from about 160 annotators. In every small dataset, there are some documents that have the true annotations labeled by experts.

**Preprocessing**  We follow the same steps as described by (Davtyan et al., 2015): the documents are the ‘text only’ version provided by TREC, use whitespace tokenisation and strip non-ASCII characters. We calculate the \( \text{tfidf} \) value for a word as \( \log(\text{tf} + 1) \times \log(N/\text{df}) \), and use normalized tfidf vectors of instance \( i \) as the instance feature vector \( x_i \).

<table>
<thead>
<tr>
<th>MV</th>
<th>BMMA</th>
<th>GPC-MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.8478</td>
<td>0.8839*</td>
</tr>
</tbody>
</table>

* Number shown in **bold** is the best result.

---

4We found this relatively short sampling run to yield good performance while additional sampling did not improve results.
5.7 Experiments

**Result on Full Training Set** Table 5.2 reports the accuracy of all methods on the full training set. Note that BMMA performs the best and is slightly better than the GPC-MA, while both BMMA and GPC-MA outperform MV.

Figure 5.3: Accuracy comparisons between MV, GPC-MA, BMMA, and the active learner BMMA-BW on partial TREC dataset (50 simulations).

**Result on Partial Training Sets** Figure 5.3 shows the comparison between MV, GPC-MA, BMMA, and the active learner BMMA-BW. The accuracies of GPC-MA and BMMA are comparable and outperform MV when on average more than 4 labels are available for every document. BMMA outperforms MV consistently, however, GPC-MA suffers lower accuracy than MV when few labels are available. A reason may be the large number of parameters in the GPC-MA model, which consequently overfits the data. When the algorithm is permitted to request labels for documents actively, the performance further improves. Figure 5.3 shows BMMA-BW consistently outperforms the other methods.

5.7.2 Sentiment Polarity Dataset

**Dataset** We use the sentiment polarity dataset created by Rodrigues et al. (2013). The dataset contains 10429 sentences extracted from movie reviews labelled as to whether the review was positive or negative. Among them, 5000 sentences are randomly selected and published on AMT for Davtyan et al. (2015) also propose a related approach, and evaluate on the TREC crowdsourcing dataset. We do not include comparative results for their method as we were unable to replicate their results, and moreover, their method is equivalent to a baseline reported in Rodrigues et al. (2014), which was shown to be inferior to the GPC-MA approach.
annotation. In total 27747 worker labels are collected from 203 annotators. These 5000 sentences form the training set, while the remaining 5429 sentences without worker labels form the test set.

**Preprocessing** The dataset was preprocessed by stop word removal and stemming, after which Latent Semantic Analysis was applied over the bag-of-words representations to reduce the dimensionality of each sentence to 1200.

**Table 5.3:** Results on the full sentiment polarity dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>AUC</td>
</tr>
<tr>
<td>GPC-GOLD#†</td>
<td>0.987</td>
<td>0.999</td>
</tr>
<tr>
<td>LR-GOLD#*</td>
<td>0.822</td>
<td>0.822</td>
</tr>
<tr>
<td>GPC-MV†</td>
<td>0.886</td>
<td>0.923</td>
</tr>
<tr>
<td>GPC-MA†</td>
<td>0.900</td>
<td>0.944</td>
</tr>
<tr>
<td>MV†</td>
<td>0.8851</td>
<td>-</td>
</tr>
<tr>
<td>BMMA*</td>
<td>0.9140‡</td>
<td>0.9553‡</td>
</tr>
<tr>
<td>BMMA&amp;LR*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MVsoft‡</td>
<td>0.8070</td>
<td>-</td>
</tr>
<tr>
<td>MVhard‡</td>
<td>0.7968</td>
<td>-</td>
</tr>
<tr>
<td>Raykar‡</td>
<td>0.4991</td>
<td>-</td>
</tr>
<tr>
<td>Raykar (with prior)‡</td>
<td>0.8492</td>
<td>-</td>
</tr>
<tr>
<td>MA-LR‡</td>
<td>0.8540</td>
<td>-</td>
</tr>
</tbody>
</table>

† Numbers shown in **bold** are the best results.
‡ Methods trained on true annotations instead of worker labels. Their results are in *italic* and provide upper bounds for others.
§ Methods implemented by us.
†,‡ Results taken from Rodrigues et al. (2013, 2014).
‡ The results of MVsoft and MVhard on test set were obtained by a logistic regression trained on their inferred true labels on the training set. Raykar, Raykar (with prior) and MA-LR all have a logistic regression component in their models, which generated the results on test set.

**Result on Full Training Set and Test Set** Table 5.3 demonstrates the results of several methods on the same dataset from competing research papers. The BMMA&LR method first uses BMMA to infer the true annotations of instances in training set, then trains a logistic regression classifier (LRC) using the inferred annotations and makes predictions on the test set. The results show that our proposed BMMA method outperforms all other methods on the training set, however, is not the best ranked over the testing data which has no worker layers (although the Boltzmann machine could be augmented for this purpose). But with the help of LRC, it slightly outperforms others on
5.7 Experiments

<table>
<thead>
<tr>
<th>Average number of labels per sentence</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.78</td>
<td>MV</td>
</tr>
<tr>
<td>0.80</td>
<td>BMMA</td>
</tr>
<tr>
<td>0.82</td>
<td>BMMA-BW</td>
</tr>
<tr>
<td>0.84</td>
<td>BMMA-LV</td>
</tr>
</tbody>
</table>

Figure 5.4: Accuracy comparisons between MV, BMMA, and two active learners BMMA-BW and BMMA-LV on partial sentiment polarity training sets (50 simulations).

the test set accuracy.

**Result on Partial Training Sets** Figure 5.4 shows how the accuracies of MV, BMMA on the training set changes when given different number of worker labels. When every sentence has exactly one worker label, the accuracies of MV and BMMA are almost the same, however, as more worker labels are available, BMMA outperforms MV and the gap grows with larger training data. Figure 5.4 also shows how the accuracy of BMMA with the active learning heuristic on the training set evolves as more worker labels are obtained. We can see that the accuracies of both BMMA-BW and BMMA-LV grow quickly and are 3.4% and 4% larger than BMMA when on average every sentence has 3 labels. For BMMA to achieve the same accuracy, on average 5 labels are required for each sentence, while by using active learning only 60% is needed.

**Evaluating Worker Reliability** To show that our model can also estimate the worker reliability accurately, and moreover is robust to poor worker labels, we augment the sentiment polarity dataset with four simulated workers, corresponding to Figure 5.1(c)-(f), namely two useless, one reasonable and one adversarial worker. We generate synthetic labels randomly for each of the four workers for all the sentences using their confusion matrices conditional on the gold tags. This increases the number of labels per sentence from 5.55 to 9.55. We train using BMMA-BW and

---

6 GPC-MA was excluded due to its prohibitive computational cost, stemming from a $O(N^3)$ time complexity (here $N = 5000$).
Figure 5.5: Accuracy comparisons between two active learners BMMA-BW and BMMA-LV on partial sentiment polarity training sets (original & synthetic, 50 simulations).

BMMA-LV, starting as before with no synthetic labels, but now allow the algorithm to select between the true worker labels or synthetic labels. The result is shown in Figure 5.5, where both algorithms are shown to be a little slower initially, due in part to the need to estimate posterior worker reliabilities starting only with the prior. Both algorithms are robust, however, to these inputs, reaching the performance of them on the original dataset after about three labels per sentence. At this point the model has learned accurate estimates for the synthetic workers, with all recall and specificity estimates within 0.1 of their known value, as shown in Figure 5.6 and 5.7.

Figure 5.6 and 5.7 also show that the estimated recall and specificity gradually converge to the ground truth as more and more worker labels are used. The reason why all estimates start at 0.75 is that no labels from the 4 simulated workers is used at the beginning and the mean of prior is used as estimates.

Figure 5.8 shows how the number of labels from each simulated worker picked by the active learner changes. At the beginning, every worker’s reliability estimate is the same, so is picked evenly by the active learner. Before every sentence has 3 worker labels on average, the accuracy of inferred labels grows quickly, however almost no label from the simulated workers is picked, simply because there are better workers to pick. When more worker labels are requested, simulated workers are more often picked, especially the type (d) worker. The reason may be that as the performance converges, the uncertainty of inferred labels are greatly reduced, so type (d) worker who always labels 1 will be preferred by the sentences with a high confidence being 1 to further
Figure 5.6: Recall estimates for 4 simulated annotators on partial synthetic dataset (50 simulations).

Figure 5.7: Specificity estimates for 4 simulated annotators on partial synthetic dataset (50 simulations).
reduce the variances. We defer designing a more robust heuristic for future work.

5.8 Conclusion

In this chapter, we present a novel model for aggregating binary labels that combines a Boltzmann machine with latent confusion matrices to model both the correlation between the latent true annotations and the reliability of workers. We empirically demonstrate that under most conditions, our proposed method achieves comparable or superior results comparing to a state-of-the-art model \cite{Rodrigues2014} on two real-world datasets.

In future work we intend to learn all model parameters from data, to make the model more flexible, and explore ways to jointly select instances and workers in the active learning algorithm as well as make the algorithm more robust.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.8.png}
\caption{Number of true worker labels (real) and synthetic labels from 4 simulated annotators (c, d, e, f) requested by the active learner (50 simulations).}
\end{figure}
Chapter 6

An Application to Multilingual Transfer for Named Entity Recognition

This chapter is derived from a paper accepted to ACL 2019:

*Multilingual NER Transfer for Low-resource Languages*. Afshin Rahimi*, Yuan Li*, Trevor Cohn, where Afshin and I contributed equally.

iBCC is a state-of-the-art aggregation method in classification settings as discussed in Chapter 2. We now move to a sequence labelling setting, where we need to combine a number of named entity recognition (NER) models transferred from different source languages to a target language, and seek to adapt iBCC for this purpose. The entire process including transferring models and inferring the truth can be completely unsupervised, which is suitable for low-resource target languages. When some specific languages are the targets, the proposed unsupervised method even achieves competitive accuracy compared to its supervised counterpart which has access to a small set of sentences with gold labels for fine-tuning. Several practical issues are also discussed including the granularity of aggregation, spammer removal, and the effect of using a small number of gold sentences to help aggregation.

6.1 Introduction

Supervised learning remains king in natural language processing, with most tasks requiring large quantities of annotated corpora. The majority of the world’s 6,000+ languages however have limited or no annotated text, and therefore much of the progress in NLP has not been realised for the majority. Cross-lingual transfer learning is a technique which can compensate in part for the dearth of data, by transferring knowledge from high- to low-resource languages, which has typically taken the form of projection of annotation over aligned parallel corpora or other multilingual
resources [Yarowsky et al. (2001); Hwa et al. (2005)], or making use of transferable representations, such as phonetic transcriptions Bharadwaj et al. (2016), closely related languages Cotterell and Duh (2017) or bilingual dictionaries Mayhew et al. (2017); Xie et al. (2018).

Most proposed models for cross-lingual transfer rely on a single source language, which limits the transferable knowledge to only one source. The target language might be similar to many source languages, on the grounds of the script, word order, loan words etc, and transfer would benefit from these diverse sources of information. There are few exceptions, which use transfer from several languages, ranging from multi-task learning Duong et al. (2015); Ammar et al. (2016); Fang and Cohn (2017), and annotation projection from several languages Täckström (2012); Fang and Cohn (2016); Plank and Agić (2018). However, to the best of our knowledge, none of these approaches adequately account for the quality of transfer, but rather “weight” the contribution of each language uniformly.

In this chapter, we propose a novel method for multilingual transfer, inspired by research in truth inference in crowdsourcing, a related problem, in which the ‘ground truth’ must be inferred from the outputs of several unreliable annotators Dawid and Skene (1979). In this problem, the best approaches include means of individuals’ reliability, and their patterns of mistakes Kim and Ghahramani (2012). Our proposed model adapts these ideas to a multilingual transfer setting, whereby we learn the quality of transfer, and language-specific transfer errors, in order to infer the best labelling in the target language, as part of a Bayesian graphical model. The key insight is that while the majority of poor models make lots of mistakes, these mistakes are diverse, while the few good models consistently provide reliable input. This allows the model to infer which are the reliable models in an unsupervised manner, i.e., without explicit supervision in the target language, and thereby make accurate inferences despite the substantial noise. We also consider a supervised setting, where a tiny annotated corpus is available in the target language which could be used to estimate reliability parameters of the Bayesian model.

We evaluate on named entity recognition (NER), over a collection of 41 language corpora, using a leave-one-language-out evaluation setting, i.e., in each case using 40 languages as input. We show that single model transfer has very variable performance, with F1 scores ranging from 0 to 80, and in many cases, the closest related language does not result in the best transfer. In contrast, our unsupervised approaches do much better, exceeding the performance of the single
6.2 Approach

We frame the problem of multilingual transfer as follows. We assume a collection of $H$ models, all trained in a high resource setting, denoted $M^h = \{ M^h_i, i \in (1, H) \}$. Each of these models are not well matched to our target data setting, for instance these may be trained on data from different domains, or on different languages, as we evaluate in our experiments, where we use cross-lingual embeddings for model transfer. This is a problem of transfer learning, namely, how best we can use the $H$ models for best results in the target language.

Simple approaches in this setting include a) choosing a single model $M \in M^h$, on the grounds of practicality, or the similarity between the model’s native data condition and the target, and this model is used to label the target data; or b) allowing all models to ‘vote’ in an classifier ensemble.

---

1We limit our attention to transfer in a ‘black-box’ setting, that is, given predictive models, but not assuming access to their data, nor their implementation. This is the most flexible scenario, as it allows for application to settings with closed APIs, and private datasets. However, it precludes multi-task learning, as the source models are assumed to be static.
such that the most frequent outcome is selected as the ensemble output. Unfortunately neither of these approaches are very accurate in a cross-lingual transfer setting, as Figure 6.1 illustrates, showing that using a single model (English: en) is often a terrible choice, and this is also often true for majority voting. The oracle best language does much better, however it is not always a closely related language (e.g., Italian: it does best for Indonesian: id, despite the target being closer to Malay: ms). Note the collection of Cyrillic languages (bg, mk, uk) where the oracle is substantially better than the majority vote, which is likely due to script differences. The transfer relationship is not symmetric e.g., Persian: fa does best for Arabic: ar, but German: de does best for Persian.

Figure 6.1 also shows that ensemble voting is well below the oracle best language, which is likely to be a result of error correlation: groups of models all make mistakes on the same instances, and this coupled with high error rates means the slated error reduction benefits from ensembling are not realised.

Motivated by these findings, we propose novel methods for learning. Where no labelled data is available in the target, we propose the $\text{BEA}_{\text{uns}}$ method inspired by work in truth inference from crowdsourced datasets (Section 6.2.1). Section 6.2.2 presents a supervised variant of it, $\text{BEA}_{\text{sup}}$, based on using very limited annotations in the target language for estimating the reliability parameters in $\text{BEA}_{\text{uns}}$.

### 6.2.1 Model Transfer as Truth Inference

One way to improve the performance of the ensemble system is to select a subset of component models carefully, or more generally, learn a non-uniform weighting function. Clearly some models do much better than others, on their own, so it stands to reason that identifying these handful of models will give rise to better ensemble performance. How might we proceed to learn the relative quality of models in the setting where no annotations are available in the target language? This is a classic unsupervised inference problem, for which we derive variational inference rules for $\text{iBCC}$ [Kim and Ghahramani (2012)].

We define $\text{iBCC}$, a generative model, illustrated in Figure 6.2 of the transfer models’ predictions, $y_{ij}$, where $i \in [1, N]$ is an instance (a token or an entity span), and $j \in [1, H]$ indexes a
6.2 Approach

The generative process assumes a ‘true’ label, $z_i \in [1, K]$, which is corrupted by each worker, in producing the observation, $y_{ij}$. The corruption process is described by

$$P(y_{ij} = l | z_i = k, V^{(j)}) = V^{(j)}_{kl}$$

where $V^{(j)} \in \mathcal{R}^{K \times K}$ is a worker-specific confusion matrix.

To complete the story, the confusion matrices are drawn from vague row-wise independent Dirichlet priors, with a parameter $\alpha = 1$ and the true labels are governed by a Dirichlet prior, $\pi$, which is drawn from an uninformative Dirichlet distribution with a parameter $\beta = 1$.

Inference under this model involves explaining the observed $Y$ in the most efficient way. Where several workers have identical outputs, $k$, on an instance, this can be explained by letting $z_i = k$ and the workers’ confusion matrices assigning high probability to $V^{(j)}_{kk}$. Other, less reliable, workers will have divergent labels, which are less likely to be in agreement, or else are heavily biased towards a particular class. Accordingly, the model can better explain these outputs through label confusion, using the off-diagonal elements of the confusion matrix. Aggregated over a corpus of instances, the model can learn to differentiate between those reliable workers, with high $V^{(j)}_{kk}$ and those less reliable ones, with high $V^{(j)}_{kl}$, $l \neq k$. This procedure applies per-label, and thus worker ‘reliability’ is with respect to a specific label, and may differ between classes. This helps in the NER setting where many poor classifiers have excellent accuracy for the outside label, but

---

2For simplicity, hereinafter we refer to $y_{ij}$ as ‘annotations’ and the transfer models as ‘workers’. This reflects the inspiration of our approach from models of truth inference from crowdsourcing [Dawid and Skene (1979)].

3Using a non-uniform prior to encourage a dominant diagonal did not have a noticeable effect. We consider a supervised setting in Section 6.2.2 where this prior is modified to encode empirical confusion statistics.

4Although there is no explicit breaking of the symmetry of the model, we initialise inference using the majority vote, which results in a bias towards this solution.
considerably worse performance for all entity labels.

For inference, we use mean-field variational Bayes [Jordan (1998)], which learns a variational distribution, \( q(Z, V, \pi) \) to optimise the evidence lower bound (ELBO),

\[
\log P(Y|\alpha, \beta) \geq \mathbb{E}_{q(Z, V, \pi)} \log \frac{P(Y, Z, V, \pi|\alpha, \beta)}{q(Z, V, \pi)}
\]

assuming a fully factorised variational distribution, \( q(Z, V, \pi) = q(Z)q(V)q(\pi) \). This gives rise to an iterative learning algorithm with update rules:

\[
E_q \log \pi_k = \psi \left( \beta + \sum_i q(z_i = k) \right) - \psi \left( K\beta + N \right) \tag{6.1a}
\]

\[
E_q \log V_{kl}^{(j)} = \psi \left( \alpha + \sum_i q(z_i = k)1[y_{ij} = l] \right) - \psi \left( K\alpha + \sum_i q(z_i = k) \right) \tag{6.1b}
\]

\[
q(z_i = k) \propto \exp \left\{ E_q \log \pi_k + \sum_j E_q \log V_{kj}^{(j)} \right\} \tag{6.2}
\]

where \( \psi \) is the digamma function, defined as the logarithmic derivative of the gamma function. The sets of rules (6.1) and (6.2) are applied alternately, to update the values of \( E_q \log \pi_k \), \( E_q \log V_{kl}^{(j)} \), and \( q(z_i = k) \) respectively. This repeats until convergence, when the difference in the ELBO between two iterations is smaller than a threshold.

The final prediction of the model is based on \( q(Z) \), using the maximum a posteriori label \( \hat{z}_i = \arg \max_k q(z_i = k) \). This method is referred to as \( \text{BEA}_{\text{un}} \).

**Token versus Entity Granularity**

Our proposed aggregation algorithm in Section 6.2.1 is based on an assumption that the true annotations are independent from each other, which simplifies the model but may generate undesired results. That is, entities predicted by different models could be mixed, resulting in labels inconsistent with the BIO scheme. Table 6.1 shows an example, where a sentence of length 4 is annotated differently by 5 models and among their predictions at most one is correct as the predicted entities overlap. However, the aggregated result in the token view is a mixture of two predictions, which
Table 6.1: An example sentence with its aggregated labels in both token view and entity view.
Aggregation in token view may generate results inconsistent with the BIO scheme.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>[1,4]</th>
<th>[2,4]</th>
<th>[3,4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1^h$</td>
<td>B-ORG</td>
<td>I-ORG</td>
<td>I-ORG</td>
<td>I-ORG</td>
<td>ORG</td>
<td>O</td>
</tr>
<tr>
<td>$M_2^h$</td>
<td>O</td>
<td>B-ORG</td>
<td>I-ORG</td>
<td>I-ORG</td>
<td>O</td>
<td>ORG</td>
</tr>
<tr>
<td>$M_3^h$</td>
<td>O</td>
<td>O</td>
<td>B-ORG</td>
<td>I-ORG</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>$M_4^h$</td>
<td>O</td>
<td>B-PER</td>
<td>I-PER</td>
<td>I-PER</td>
<td>O</td>
<td>PER</td>
</tr>
<tr>
<td>$M_5^h$</td>
<td>O</td>
<td>B-PER</td>
<td>I-PER</td>
<td>I-PER</td>
<td>O</td>
<td>PER</td>
</tr>
<tr>
<td>Agg.</td>
<td>O</td>
<td>B-PER</td>
<td>I-ORG</td>
<td>I-ORG</td>
<td>O</td>
<td>PER</td>
</tr>
</tbody>
</table>

is supported by no classifiers.

To deal with this problem, we consider aggregating the predictions in the entity view. As shown in Table 6.1, we convert the predictions for tokens to predictions for ranges, aggregate labels for every range, and then resolve remaining conflicts. A prediction is ignored if it conflicts with another one with higher probability. By using this greedy strategy, we can solve the conflicts raised in entity-level aggregation. We use superscripts $tok$ and $ent$ to denote token-level and entity-level aggregations, i.e. $BEA_{uns}^{tok}$ and $BEA_{uns}^{ent}$.

Spammer Removal

Raykar and Yu (2012) show that iteratively removing spammers and re-estimating the true labels based on remaining workers can improve the accuracy of estimation. Here spammers are defined as workers whose confusion matrix can be well approximated by an outer product of two vectors, i.e. a rank-1 matrix. Such matrices have identical rows, i.e. $p(y_{ij} = l | z_i = k) = p(y_{ij} = l | z_i = k')$, thus their labels are independent from the true labels and not informative for estimating the truth. Even worse, due to the goal of maximizing the likelihood of all observed labels, the fact that labels from spammers can’t be well explained by the truth can make the model alter the truth to better explain spammers’ labels. This effect degrades the quality of truth inference. One can also consider workers as features in supervised learning, where noisy features can lead to worse performance and had better be removed via feature selection. In our task, it’s assumed that no gold labels are available, therefore spammer removal is essentially unsupervised feature selection.

In our NER transfer task, classifiers are diverse in their F1 scores ranging from almost 0 to
around 80, so spammer removal is necessary. We adopt a simple strategy that first estimates the confusion matrices for all classifiers on all labels, then ranks classifiers based on their mean recall on different entity categories (elements on the diagonals of their confusion matrices), and then runs the model again on labels from the top k classifiers only. We call this method $BEA_{uns} \times 2$ and its results are reported in Section 6.4.

6.2.2 Using Target Supervision

Until now, we have assumed no access to annotations in the target language. However, when some labelled text is available, how might this best be used? In our experimental setting, we assume a modest set of 100 labelled sentences, in keeping with a low-resource setting [Garrette and Baldridge (2013)]. We propose a variant of $BEA_{uns}$ model for this setting.

Supervising $BEA_{sup}$  We use the labelled data to find the posterior for the parameters $V^{(j)}$ and $\pi$ of the Bayesian model described in Section 6.2.1. Let $n_k$ be the number of instances in the labelled data whose true label is $k$, and $n_{jkl}$ the number of instances whose true label is $k$ and classifier $j$ labels them as $l$. Then the quantities in Equation (6.1) can be calculated as

$$E \log \pi_k = \psi(n_k) - \psi(N)$$

$$E \log v_{jkl} = \psi(n_{jkl}) - \psi\left(\sum_l n_{jkl}\right).$$

These are used in Equation (6.2) for inference on the test set.

$$q(z_{test}^i = k) \propto \exp\left\{E_{\pi|\beta} \log \pi_k + \sum_j E_{V_{kl}} \log V^{(j)}_{kly}\right\}.$$

Note the close similarity to (6.2), with the difference being the expected log terms are replaced with the log MLE parameter estimates. We refer to this setting as $BEA_{sup}$.

---

[Garrette and Baldridge (2013)] showed that about 100 sentences can be annotated with POS tags in two hours by non-native annotators.
6.3 Experiments

We evaluate using the Wiki-NER dataset (Pan et al., 2017), which comprises 282 languages automatically extracted from Wikipedia, based on using the cross-lingual link structure of Wikipedia to identify entity spans, alongside grounding in an English knowledge base, and several other techniques such as self-training. Although the annotations in these datasets are automatically generated, and of varying quality, throughout this chapter we treat them as gold labels.

We chose 41 of those languages based on the overlap between the NER datasets and accessible bilingual dictionaries from Conneau et al. (2017). We omitted Chinese, Japanese, and Thai, because of difficulties with tokenisation. The distribution of entity annotations was often highly skewed so we created a balanced subset for each language, and split these into training, test, and development sets. We set the development and test set sizes to be either 1k or 10k (depending on how much data was available for the language), and used the remained sentences for training with sizes between 5k to 20k.

We use fastText 300 dimensional monolingual embeddings trained on Wikipedia Bojanowski et al. (2017). The monolingual embeddings are in different vector spaces, to create cross-lingual word embeddings we use the bilingual dictionaries and supervised cross-lingual Procrustes rotation method from project MUSE Conneau et al. (2017), such that all language embeddings are mapped to the English vector space. We included embeddings for words in fastText, using a special UNK tag for others. Note that accurate unsupervised methods exist for creating cross-lingual word embeddings (Conneau et al., 2017; Artetxe et al., 2018), which would allow our technique to be applied without the need for bilingual dictionaries.

We experiment in four different data requirement scenarios, relating to the amount of annotated target language data, and the use of unannotated target data, as shown in Table 6.2. The bottom rows simulate low resource settings, while HSup provides an empirical comparison against a high

---

6 The data preparation described in this section was done by my co-author Afshin Rahimi.
7 With ISO 639-1 codes: af, al, ar, bg, bn, bs, ca, cs, da, de, el, en, es, et, fa, fi, fr, he, hi, hr, hu, id, it, lt, lv, mk, ms, nl, no, pl, pt, ro, ru, sk, sl, sq, sv, ta, tl, tr, uk, and vi.
8 The dataset details omitted, and we refer the reader to our paper on arXiv for more details.
9 https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
10 https://github.com/facebookresearch/MUSE
11 One might expect better results using embeddings from sub-words for words not covered in fastText, however, our preliminary results for LSup on English showed a reduction in performance when doing so, versus using UNK token. This might be due to OOVs being a good indicator of named entities captured by UNK.
Table 6.2: Data requirements for each of the models, in terms of the number of annotated sentences in each target language. Entries with “+” denote variable sizes, depending on the availability of data in each language.

<table>
<thead>
<tr>
<th>Training</th>
<th>Development</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>5k+</td>
<td>1k+</td>
<td>HSup</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>LSup</td>
</tr>
<tr>
<td>100</td>
<td>-</td>
<td>BEA\text{sup}</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>MV, BEA\text{uns}, BEA\text{uns} \times 2</td>
</tr>
</tbody>
</table>

resource setting. \text{LSup}, shows a naive low resource setting without using transfer learning. In both these cases, a development set was used for early stopping.

Other models compared include:

**MV** uniform ensemble, a.k.a. “majority vote”;

**BEA\text{uns} \times 2, BEA\text{uns}** unsupervised aggregation models, applied to entities or tokens (see Section 6.2.1);

**BEA\text{sup}** supervised estimation of BEA prior (Section 6.2.2);

**Oracle** selecting the best performing single transfer model, based on test performance.

As the sequential tagger, we use a BiLSTM-CRF \cite{Lample2016}, which has been shown to result in state-of-the-art results in high resource settings. This model includes both word embeddings (for which we used fixed cross-lingual embeddings) and character embeddings, to form a parameterised potential function in a linear chain conditional random field. We trained models on all 41 languages, which were then used for transfer in a leave-one-out setting, i.e., taking the predictions of 40 models into a single target language.

The parameters of the model and optimiser were taken from \cite{Lample2016}, with the exception of batch size and learning rate. We tuned the batch size and the learning rate using development sets in four languages\footnote{Afrikaans, Arabic, Bulgarian and Bengali.} and then fixed these hyper-parameters for all other languages in each model. The batch size was 1 sentence in low-resource scenarios (in baseline \text{LSup}), and to 100 sentences, in high-resource settings (\text{HSup}). The learning rate was set to 0.001 and 0.01
6.4 Results

We report the results for the proposed low-resource supervised model (BEA\textsuperscript{sup}), and unsupervised models (BEA\textsubscript{uns} and BEA\textsubscript{uns}$\times2$) for each of the 41 languages in Table 6.3 and compare them with high- and low-resource supervised baselines (HSup and LSup, respectively). Figure 6.3 shows summary statistics for these models, alongside their annotation requirement. The best achievable performance of $F_1 = 89.4$ is achieved with a high supervision (HSup), while very limited supervision (LSup) results in a considerably lower $F_1$ of 63.6. The results for MV\textsuperscript{tok} show that uniform ensembling of multiple source models is even worse, by about 3 points.

Unsupervised truth inference improves dramatically over MV\textsuperscript{tok}, and BEA\textsubscript{ent} outperforms BEA\textsuperscript{tok}, and also showing the effectiveness of inference over entities (BEA\textsubscript{ent}) rather than tokens in BEA\textsubscript{uns}. It is clear that having access to limited annotation in the target language makes a substantial difference in BEA\textsubscript{sup} with $F_1$ of 77.0.
<table>
<thead>
<tr>
<th></th>
<th>Supervised</th>
<th>Unsupervised</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HSup</td>
<td>LSup</td>
<td>BEA\textsuperscript{ent} t10</td>
<td>BEA\textsuperscript{ent}\textsubscript{uns} t10</td>
<td>BEA\textsuperscript{tok}\textsubscript{uns}</td>
<td>BEA\textsuperscript{tok}</td>
<td></td>
</tr>
<tr>
<td>af</td>
<td>85</td>
<td>56</td>
<td>79</td>
<td>78</td>
<td>78</td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td>ar</td>
<td>88</td>
<td>66</td>
<td>69</td>
<td>65</td>
<td>56</td>
<td>56</td>
<td>24</td>
</tr>
<tr>
<td>bg</td>
<td>91</td>
<td>70</td>
<td>83</td>
<td>84</td>
<td>79</td>
<td>77</td>
<td>24</td>
</tr>
<tr>
<td>bn</td>
<td>95</td>
<td>69</td>
<td>72</td>
<td>67</td>
<td>66</td>
<td>61</td>
<td>59</td>
</tr>
<tr>
<td>bs</td>
<td>89</td>
<td>65</td>
<td>82</td>
<td>79</td>
<td>77</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>ca</td>
<td>91</td>
<td>62</td>
<td>86</td>
<td>85</td>
<td>80</td>
<td>79</td>
<td>72</td>
</tr>
<tr>
<td>cs</td>
<td>90</td>
<td>63</td>
<td>79</td>
<td>77</td>
<td>76</td>
<td>73</td>
<td>72</td>
</tr>
<tr>
<td>da</td>
<td>90</td>
<td>63</td>
<td>83</td>
<td>82</td>
<td>82</td>
<td>80</td>
<td>79</td>
</tr>
<tr>
<td>de</td>
<td>86</td>
<td>57</td>
<td>73</td>
<td>73</td>
<td>71</td>
<td>65</td>
<td>69</td>
</tr>
<tr>
<td>el</td>
<td>91</td>
<td>68</td>
<td>77</td>
<td>76</td>
<td>74</td>
<td>73</td>
<td>47</td>
</tr>
<tr>
<td>en</td>
<td>81</td>
<td>50</td>
<td>66</td>
<td>64</td>
<td>63</td>
<td>59</td>
<td>58</td>
</tr>
<tr>
<td>es</td>
<td>90</td>
<td>66</td>
<td>85</td>
<td>85</td>
<td>81</td>
<td>77</td>
<td>74</td>
</tr>
<tr>
<td>et</td>
<td>89</td>
<td>63</td>
<td>78</td>
<td>78</td>
<td>77</td>
<td>71</td>
<td>73</td>
</tr>
<tr>
<td>fa</td>
<td>93</td>
<td>76</td>
<td>76</td>
<td>73</td>
<td>70</td>
<td>67</td>
<td>30</td>
</tr>
<tr>
<td>fi</td>
<td>89</td>
<td>66</td>
<td>81</td>
<td>81</td>
<td>80</td>
<td>68</td>
<td>78</td>
</tr>
<tr>
<td>fr</td>
<td>88</td>
<td>60</td>
<td>84</td>
<td>84</td>
<td>80</td>
<td>75</td>
<td>71</td>
</tr>
<tr>
<td>he</td>
<td>85</td>
<td>53</td>
<td>56</td>
<td>54</td>
<td>56</td>
<td>51</td>
<td>39</td>
</tr>
<tr>
<td>hi</td>
<td>84</td>
<td>68</td>
<td>70</td>
<td>70</td>
<td>67</td>
<td>62</td>
<td>52</td>
</tr>
<tr>
<td>hr</td>
<td>89</td>
<td>61</td>
<td>81</td>
<td>81</td>
<td>79</td>
<td>75</td>
<td>74</td>
</tr>
<tr>
<td>hu</td>
<td>90</td>
<td>61</td>
<td>80</td>
<td>81</td>
<td>79</td>
<td>72</td>
<td>74</td>
</tr>
<tr>
<td>id</td>
<td>91</td>
<td>66</td>
<td>74</td>
<td>60</td>
<td>58</td>
<td>61</td>
<td>54</td>
</tr>
<tr>
<td>it</td>
<td>89</td>
<td>61</td>
<td>81</td>
<td>79</td>
<td>78</td>
<td>75</td>
<td>73</td>
</tr>
<tr>
<td>lt</td>
<td>87</td>
<td>64</td>
<td>79</td>
<td>79</td>
<td>77</td>
<td>73</td>
<td>76</td>
</tr>
<tr>
<td>lv</td>
<td>91</td>
<td>69</td>
<td>72</td>
<td>69</td>
<td>69</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td>mk</td>
<td>93</td>
<td>73</td>
<td>85</td>
<td>84</td>
<td>80</td>
<td>72</td>
<td>24</td>
</tr>
<tr>
<td>ms</td>
<td>90</td>
<td>66</td>
<td>82</td>
<td>70</td>
<td>69</td>
<td>66</td>
<td>68</td>
</tr>
<tr>
<td>nl</td>
<td>89</td>
<td>55</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>78</td>
<td>76</td>
</tr>
<tr>
<td>no</td>
<td>90</td>
<td>67</td>
<td>83</td>
<td>84</td>
<td>83</td>
<td>76</td>
<td>79</td>
</tr>
<tr>
<td>pl</td>
<td>89</td>
<td>62</td>
<td>81</td>
<td>82</td>
<td>81</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>pt</td>
<td>90</td>
<td>59</td>
<td>83</td>
<td>82</td>
<td>78</td>
<td>74</td>
<td>71</td>
</tr>
<tr>
<td>ro</td>
<td>91</td>
<td>66</td>
<td>81</td>
<td>77</td>
<td>74</td>
<td>67</td>
<td>65</td>
</tr>
<tr>
<td>ru</td>
<td>87</td>
<td>57</td>
<td>67</td>
<td>56</td>
<td>44</td>
<td>32</td>
<td>19</td>
</tr>
<tr>
<td>sk</td>
<td>91</td>
<td>64</td>
<td>80</td>
<td>78</td>
<td>78</td>
<td>77</td>
<td>73</td>
</tr>
<tr>
<td>sl</td>
<td>92</td>
<td>67</td>
<td>78</td>
<td>78</td>
<td>77</td>
<td>74</td>
<td>72</td>
</tr>
<tr>
<td>sq</td>
<td>88</td>
<td>71</td>
<td>84</td>
<td>84</td>
<td>82</td>
<td>77</td>
<td>78</td>
</tr>
<tr>
<td>sv</td>
<td>93</td>
<td>69</td>
<td>81</td>
<td>80</td>
<td>79</td>
<td>67</td>
<td>76</td>
</tr>
<tr>
<td>ta</td>
<td>86</td>
<td>54</td>
<td>57</td>
<td>53</td>
<td>54</td>
<td>42</td>
<td>46</td>
</tr>
<tr>
<td>tl</td>
<td>93</td>
<td>67</td>
<td>76</td>
<td>68</td>
<td>64</td>
<td>60</td>
<td>53</td>
</tr>
<tr>
<td>tr</td>
<td>90</td>
<td>62</td>
<td>75</td>
<td>75</td>
<td>72</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>uk</td>
<td>91</td>
<td>64</td>
<td>75</td>
<td>70</td>
<td>65</td>
<td>60</td>
<td>18</td>
</tr>
<tr>
<td>vi</td>
<td>89</td>
<td>60</td>
<td>63</td>
<td>56</td>
<td>55</td>
<td>48</td>
<td>46</td>
</tr>
<tr>
<td>$\mu$</td>
<td>89.4</td>
<td>63.6</td>
<td>77.0</td>
<td>74.7</td>
<td>72.4</td>
<td>67.6</td>
<td>60.7</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>2.7</td>
<td>5.5</td>
<td>7.2</td>
<td>9.1</td>
<td>9.6</td>
<td>10.5</td>
<td>18.8</td>
</tr>
</tbody>
</table>

Table 6.3: $F_1$ scores on the test set, comparing baseline supervised models (HSup, LSup (5 runs)), aggregation methods: majority voting ($\text{MV}^\text{tok}$), BEA\textsuperscript{tok} and BEA\textsuperscript{ent} (Bayesian aggregation in token- and entity-level), and the oracle single best annotation (Oracle). The mean and standard deviation over all 41 languages, $\mu, \sigma$, are also reported.
The detailed results in Table 6.3 show that majority voting works reasonably well for Roman and Germanic languages, which are well represented in the dataset, but fails miserably compared to single best for Slavic languages (e.g., ru, uk, bg) where there are only a few related languages.

An important question is how the unsupervised variants, are affected by the number and choice of source languages. Figure 6.4 charts the performance of MV and BEA against the number of source language classifiers, comparing the use of ideal or realistic selection methods to attempt to find the best input classifiers. Even when coupled with supervised data for ordering the source classifiers, MV does poorly, and does not show any benefit from using more than 3 classifiers. In contrast, BEA continues to improve with substantially more inputs: in a “cheating” setting where the oracle ranking is used, but with BEA\textsuperscript{ent} uns × 2, performance peaks at 7 inputs. However, BEA\textsuperscript{ent} sup outperforms further, by about 1 \( F_1 \) point, showing that the model is highly effective at discriminating between good and bad input models, and accordingly filtering out the bad inputs gives the best results. The BEA\textsuperscript{ent} uns × 2 curve shows the effect of filtering using purely unsupervised signal, which has a positive, albeit more mild, effect on performance. Note also that neither of the curves for the two realistic BEA scenarios (unsupervised: BEA\textsuperscript{ent} uns × 2, and supervised: BEA\textsuperscript{ent} sup) show evidence of the sawtooth pattern, i.e., they largely benefit from more inputs, irrespective of

\[ \text{Figure 6.4: The mean } F_1 \text{ performance of } MV \text{ent}, \ BEA \text{ent} \textsuperscript{sup}, \ BEA \text{ent} \textsuperscript{uns} \times 2, \ BEA \text{ent} \textsuperscript{uns cheat} \text{ (cheating selection) over the 41 languages categorised by the number of source languages used for transfer.} \]
their parity.

6.5 Related Work

Cross-lingual transfer approaches can broadly be classed into annotation or representation projection, based on the manner in which information from high resource languages are transferring into the target language.

**Annotation Projection:** In annotation projection, the annotations of tokens in a source sentence are projected to their aligned tokens in the target language through a parallel corpus. Annotation projection has been applied to POS tagging [Yarowsky et al. (2001); Das and Petrov (2011); Duong et al. (2014); Fang and Cohn (2016)], NER [Zitouni and Florian (2008); Ehrmann et al. (2011)], and parsing [Hwa et al. (2005); Rasooli and Collins (2015)]. The Bible, Europarl, and recently the Watchtower has been used as parallel corpora, which are limited in genre, size, and language coverage, motivating the use of Wikipedia to create weak annotation for multilingual tasks such as NER [Nothman et al. (2013)].

Annotations can also be projected through cheap translation models from gold bilingual dictionaries [Mayhew et al. (2017)]. Recent advances in (un)supervised bilingual dictionary induction [Gouws and Søgaard (2015); Duong et al. (2016); Conneau et al. (2017); Artetxe et al. (2018)] has enabled cross-lingual lexical alignment without parallel corpora, which can be used to project annotations [Xie et al. (2018)]. Most annotation projection methods with few exceptions [Täckström (2012); Plank and Agić (2018)] use only one language (often English) as the source language. In multi-source language setting, majority voting is often used to aggregate noisy annotations (e.g. [Plank and Agić (2018)]). Fang and Cohn (2016) show the importance of modelling the annotation biases that the source language(s) might project to the target language.

**Representation Projection:** Representation projection learns a model in a high-resource source language using representations that are cross-linguistically transferable, and then directly applies the model to data in the target language. This can include the use of cross-lingual word clusters [Täckström et al. (2012)] and word embeddings [Ammar et al. (2016); Ni et al. (2017)], multi-task learning with a closely related high-resource language (e.g. Spanish for Galician) [Cotterell and
6.6 Conclusion

Duh (2017), or bridging the source and target languages through phonemic transcription Bharadwaj et al. (2016) or Wikification Tsai et al. (2016).

Transfer from multiple source languages: Previous work has shown the improvements of multi-source transfer in NER Täckström (2012); Fang et al. (2017), POS tagging Snyder et al. (2009); Plank and Agić (2018), and parsing Ammar et al. (2016) compared to single source transfer, however, multi-source transfer might be noisy as a result of divergence in script, phonology, morphology, syntax, and semantics between the source languages, and the target language. To capture such differences, various methods have been proposed: latent variable models Snyder et al. (2009), majority voting Plank and Agić (2018), utilising typological features Ammar et al. (2016), or explicitly learning annotation bias Fang and Cohn (2017). In this work, we use truth inference to model the transfer annotation bias from diverse source models.

6.6 Conclusion

Cross-lingual transfer does not work out of the box, especially when using large numbers of source languages, and distantly related target languages. We proposed a multilingual transfer model $\text{BEA}_{\text{uns}}$, based on unsupervised transfer, and $\text{BEA}_{\text{sup}}$ based on a supervised transfer setting with a small 100 sentence labelled dataset in the target language.

Our unsupervised method, $\text{BEA}_{\text{uns}}$, provides a fast and simple way of annotating data in the target language, which is capable of reasoning under noisy annotations, and outperforms several competitive baselines, including the majority voting ensemble, a low-resource supervised baseline, and the oracle single best transfer model. We also show that light supervision $\text{BEA}_{\text{sup}}$ improves performance further.
Chapter 7

Conclusion

This thesis investigates methods for aggregating crowdsourced annotations. Production systems for machine learning, natural language processing, computer vision, and information retrieval are regularly trained and evaluated on vast annotated datasets collected by crowdsourcing services ([Howe] [2008] [Callison-Burch and Dredze] [2010]). While crowdsourced annotations are low cost per label, they can be highly noisy, with labels procured from large cohorts of worker annotators ([Difallah et al.] [2012]). Consequently, inferring consensus aggregation of collected annotations is a core crowdsourcing task.

7.1 Contributions

Although aggregation methods have been studied extensively in several related areas and applied on crowdsourcing, the simplest technique, majority voting (MV) that grants workers equal votes towards consensus, still predominates in practice. Our main contributions include conducting extensive experiments to find explanations to the predominance of MV, identifying the shortcomings of existing methods that prevent them from being applied in practice, proposing two novel aggregation methods to address the identified problems. Other contributions include introducing a novel aggregation method that can exploit the content of items if available to help aggregation, and applying aggregation methods to zero-shot learning for named entity recognition.

In Chapter 2 we demonstrate an empirical comparison of 10 representative aggregation methods on 19 datasets over a wide range of tasks, including recent large-scale datasets, unlike previous studies where datasets used are fewer and smaller. Based on the results, we identify two main shortcomings of existing methods:
Conclusion

- A Wilcoxon signed rank test shows only one method (GLAD) \cite{Whitehill2009} significantly outperforms MV at one sided significance level 0.05.

- Some methods don’t scale well to large datasets, including GLAD being the slowest among all methods compared. On the senti dataset used in our experiments with 569k crowdsourced annotations, GLAD takes 3 hours to train, while MV runs in 3.5 seconds.

Methods that don’t significantly outperform MV either have clear failure cases or are outperformed by MV on several datasets, despite that their mean accuracy across all datasets could be higher than that of MV. We believe the above shortcomings prevent existing methods being used in practice, which motivates us to propose novel aggregation methods.

In Chapter 3 we present a Bayesian version of the Weighted Average model (BWA). In the BWA model, every worker has a voting weight and all weights are learned by approximating the posterior, unlike other weighted average models that either estimate the weights by heuristic update rules or learn them by optimizing a hand-crafted objective function. BWA is based on a Bayesian graphical model with conjugate priors, and simple iterative expectation-maximisation inference. It significantly outperforms MV across all datasets. Moreover, BWA is simple, implemented in only 50 lines of code, and trains in seconds.

In Chapter 4 we present enhanced Bayesian classifier combination (EBCC) which is an enhanced version of independent Bayesian classifier combination (iBCC). EBCC captures the correlations between worker labels that other models assume are independent. EBCC is also a Bayesian graphical model with inference based on a mean-field variational approach. An introduced mixture of intra-class reliabilities–connected to tensor decomposition and item clustering–induces inter-worker correlation. Results show that EBCC not only significantly outperforms MV but also achieves the highest mean accuracy across all datasets.

When applying aggregation methods in practice, the settings may be different from the experimental settings reported in Chapter 2. In Chapter 5 the content of items is considered available, while in Chapter 6 the task considered is aggregating spans in a sentence instead of labels of an item.

In Chapter 5 we propose a novel model for aggregating binary labels in the case that item content is available. The model infers the ground truth and worker reliabilities jointly, making use of a Boltzmann machine prior in order to bias similar instances towards sharing the same
7.1 Contributions

label, with a per-worker confusion matrix to model the worker reliability. Experimental results on two datasets show that our proposed approach can outperform both MV and a Gaussian process classification model. Moreover, we show further gains when integrating a proposed active learning heuristic.

In Chapter 6, we apply the iBCC model for ensembling a number of named entity recognition (NER) models transferred from different source languages to a target language by aggregating the text spans generated by different models. The entire process including transferring models and inferring the truth can be completely unsupervised, which is suitable for low-resource target languages. When some specific languages are the targets, the proposed unsupervised method even achieves competitive accuracy compared to its supervised counterpart which has access to a small set of sentences with gold labels for fine-tuning. Several practical issues are also discussed including the granularity of aggregation, spammer removal, and the effect of using a small number of gold sentences to help aggregation.

In general, when item content is available, source conditioning models exploiting it such as BMMA are expected to perform better than aggregation methods that don’t make use of it such as iBCC, BWA, and EBCC. However, source conditioning models require there exist appropriate vector representations for item content where items close in the vector space are likely to have the same label. Existing source conditioning models mainly focus on images and documents, as there are well-known vector representations for them, but for tasks of labelling other kinds of items, finding an appropriate vector representation is very challenging, therefore source conditioning models have not been widely used for general crowdsourcing tasks.

On the other hand, aggregation methods can infer true labels for any kind of items as the they only need crowdsourced labels. Existing aggregation methods mainly differ in how worker reliability is modelled, in particular, how many parameters are used for modelling worker reliability. To summarize their properties, we consider three representatives, namely BWA, iBCC, and EBCC, whose number of parameters used for worker modelling are $W$, $WK^2$, and $MWK^2$, respectively. As shown in Chapter 3 and 4, EBCC achieves the highest mean accuracy across 19 real-world datasets, followed by iBCC and BWA whose mean accuracies are very close. Both BWA and EBCC significantly outperform MV, but iBCC doesn’t. BWA is a 1-coin model which uses only one parameter to model worker reliability, while iBCC and EBCC are confusion-matrix-based
models which at least models each worker by a $K \times K$ matrix.

Comparing to iBCC, BWA has fewer parameters to estimate, so it’s fast and less likely to overfits the data when crowdsourced labels are insufficient or very noisy. But BWA may underfit the data especially when workers have different accuracies on different classes as its capacity is limited by one parameter per worker. EBCC, on the other hand, models the correlation between workers and can even learn worker reliability on subtypes instead of classes, so is flexible enough to capture information that is ignored by iBCC and BWA. But EBCC is likely to overfit the noise more so than other methods, which follows as it has many more parameters. In addition, EBCC is slow and often gets stuck on local optima due to its variational inference algorithm, therefore one has to run EBCC with random initialisation for many times in order to obtain good solutions, which makes it cost even more time. In general, we recommend using EBCC due to its stable performance, but suggest using lower capacity models for very noisy datasets, e.g. iBCC or even 1-coin models such as BWA.

Furthermore, aggregation methods for crowdsourcing are closely related to a more general problem of resolving the conflicts between information from different sources and inferring the unknown truth. A well-known aggregation method, the Dawid-Skene (DS) model (Dawid and Skene, 1979), was originally proposed for compiling patient records based on their history taken by different clinicians where the patient’s true response is unavailable. Another successful model, independent Bayesian Classifier Combination (iBCC) (Kim and Ghahramani, 2012), was originally proposed for ensembling classifiers based on their outputs on an unlabelled dataset. Moreover, there is another line of work from the database community on resolving the conflicts between different websites or structured datasets and inferring the truth (Zhao et al., 2012; Li et al., 2014). Due to the close relatedness between the above areas, an improvement in aggregation methods for crowdsourcing is likely to benefit other areas as well.

### 7.2 Future Research

The work in this thesis also provides grounds for many directions of future research.

1. For the iBCC model, the goal of truth inference is to find the truth $Z$ that maximises its posterior distribution with other random variables integrated out: $\arg\max_Z P(Z|Y, \alpha, \beta)$. This
is a discrete optimisation problem and was considered to be intractable, therefore several approximate algorithms have been proposed for iBCC. However, we find that this discrete optimisation can be solved by a straight-forward expectation maximisation algorithm. Preliminary results show that the EM algorithm that directly optimises $Z$ outperforms other implementations of iBCC such as Gibbs sampling, variational inference, and expectation propagation. We plan to further investigate this new algorithm and possibly generalize it to other models.

2. MV has well-understood performance guarantees with the number of labels per item and the quality of worker labels being two important factors. We believe this is a reason why MV predominates in practice. Although we empirically show that our proposed models significantly outperform MV, there are no theoretic guarantees. And since our proposed models don’t outperform MV on all datasets, a criterion for selecting the most appropriate aggregation method for different datasets would be of great value.

3. As discussed in Section 4.4, our EBCC model may get stuck at local optima on large-scale datasets, which is a commonly known issue for batch learning. Stochastic optimization would likely mitigate the problem and help the method to scale better. This also applies to existing methods such as GLAD which is also based on batch learning.

4. When a few gold labels are available, it seems straight-forward to plug gold labels into the probabilistic model. However, in the iterative updating process, the model doesn’t distinguish whether a label is gold or estimated, therefore the few gold labels will be dominated by the many more estimated labels and have very little impact to the learning process. As discussed in Chapter 6, estimating the worker reliability purely based on the gold labels works better than the straight-forward plug-in approach. However, this ignores all information on the unlabelled items. How to combine the information from two sources is still an open problem.

5. The 19 real-world datasets used in our experiments have been highly redundantly labelled, where most of them on average have more than 5 labels per item. This setting aligns with offline aggregation tasks in practice which is the focus of this thesis. However, it’s interesting to consider the online scenario where the number of labels per item starts from 0. In such
cases, methods like EBCC are too flexible and will have severe overfitting problems, but 1-coin models like BWA are expected to perform better as they have much fewer parameters. It’s possible to switch between models of different capacity as more labels are collected, but it’s better if a model is non-parametric so that its capacity grows with the amount of data so that the transition is more smooth.

6. This thesis mainly studies how crowdsourced labels should be aggregated when they are given, but there are other factors playing important roles in practice, such as how much to pay workers, whether the user interface is friendly and intuitive, how to instruct workers before they perform the task, etc. Some factors haven’t been studied as extensively as aggregation methods, and could be further explored in future research.
Appendix A

Implementations

All the code with example iPython notebooks to run them can be found at our GitHub repository[1]

A.1 BWA

```python
import numpy as np
import scipy.sparse as ssp

def bwa_binary(y_exists_ij, y_is_one_ij, W_i, lambda_, a_v, adj_coef):
    N_j = y_exists_ij.sum(axis=0)
    z_i = y_is_one_ij.sum(axis=-1) / y_exists_ij.sum(axis=-1)
    b_v = a_v * W_i.dot(np.multiply(z_i, 1-z_i)) / y_exists_ij.sum()* adj_coef
    for _ in range(500):
        last_z_i = z_i.copy()
        mu = z_i.mean()
        v_j = (a_v+N_j) / (b_v + (y_exists_ij.multiply(z_i)-y_is_one_ij).power(2).sum(0))
        z_i = (lambda_*mu + y_is_one_ij.dot(v_j.T)) / (lambda_ + y_exists_ij.dot(v_j.T))
        if np.allclose(last_z_i, z_i, rtol=1e-3): break
    return z_i

def bwa(tuples, a_v=15, lambda_=1, prior_correction=True):
    num_items, num_workers, num_classes = tuples.max(axis=0) + 1
    num_labels = tuples.shape[0]
    W_i = np.bincount(tuples[:, 0], minlength=num_workers)

    adj_coef = 4 * (1 - 1 / num_classes) if prior_correction else 1

    y_exists_ij = ssp.coo_matrix((np.ones(num_labels), tuples[:, :2].T),
        shape=(num_items, num_workers), dtype=np.bool).tocsr()
    y_is_one_kij = []
    for k in range(num_classes):
        selected = (tuples[:, 2] == k)
        y_is_one_kij.append(ssp.coo_matrix((np.ones(selected.sum()), tuples[selected, :2].T),
            shape=(num_items, num_workers), dtype=np.bool).tocsr())
    z_ik = np.empty((num_items, num_classes))
    for k in range(num_classes):
        z_ik[:, k] = bwa_binary(y_exists_ij, y_is_one_kij[k], W_i, lambda_, a_v, adj_coef)
    return z_ik
```

Implementations

A.2 EBCC

```python
import numpy as np
import scipy as sp
from scipy import import digamma, gamma
from scipy import import entropy, dirichlet

def ebcc_vb(tuples, num_groups=10, a_pi=0.1, alpha=1, a_v=4, b_v=1, seed=1234, max_iter=500, empirical_prior=False):
    num_items, num_workers, num_classes = tuples.max(axis=0) + 1
    num_labels = tuples.shape[0]
    y_is_one_lij = []
    y_is_one_lji = []
    for k in range(num_classes):
        selected = (tuples[:, 2] == k)
        coo_ij = sp.coo_matrix((np.ones(selected.sum()), tuples[selected, :2].T),
                               shape=(num_items, num_workers), dtype=np.bool)
        y_is_one_lij.append(coo_ij.tocsr())
        y_is_one_lji.append(coo_ij.T.tocsr())

    beta_0 = np.eye(num_classes)*(a_v-b_v) + b_v

    # initialize z_ik, zg_ikm
    z_ik = np.zeros((num_items, num_classes))
    for l in range(num_classes):
        z_ik[:, [l]] += y_is_one_lij[l].sum(axis=-1)
    z_ik /= z_ik.sum(axis=-1, keepdims=True)

    if empirical_prior:
        alpha = z_ik.sum(axis=0)

    np.random.seed(seed)
    zg_ikm = np.random.dirichlet(np.ones(num_groups), z_ik.shape)* z_ik[:, :, None]
    for it in range(max_iter):
        eta_km = a_pi/num_groups + zg_ikm.sum(axis=0)
        nu_k = alpha + z_ik.sum(axis=0)

        mu_jkml = np.zeros((num_workers, num_classes, num_groups, num_classes))
        for l in range(num_classes):
            for k in range(num_classes):
                mu_jkml[:, k, :, l] += y_is_one_lji[l].dot(zg_ikm[:, k, :])

        Eq_log_pi_km = digamma(eta_km) - digamma(eta_km.sum(axis=-1, keepdims=True))
        Eq_log_tau_k = digamma(nu_k) - digamma(nu_k.sum())
        Eq_log_v_jkml = digamma(mu_jkml) - digamma(mu_jkml.sum(axis=-1, keepdims=True))

        zg_ikm[:, :] = Eq_log_pi_km[None, :, :] + Eq_log_tau_k[None, None, :]
        zg_ikm[:, :, :] += y_is_one_lij[l].dot(Eq_log_v_jkml[:, :, :, l])

        zg_ikm = np.exp(zg_ikm)
        zg_ikm /= zg_ikm.reshape(num_items, -1, axis=-1)[1, None, None]

        last_z_ik = z_ik
        z_ik = zg_ikm.sum(axis=-1)
        if np.allclose(last_z_ik, z_ik, atol=1e-3):
            break

    ELBO = ((eta_km-1)*Eq_log_pi_km).sum() + ((nu_k-1)*Eq_log_tau_k).sum() + ((mu_jkml-1)*Eq_log_v_jkml).sum() + dirichlet.entropy(nu_k)
    for k in range(num_classes):
        ELBO += dirichlet.entropy(eta_km[k])
        ELBO += (gamma(log(sum(mu_jkml - (mu_jkml-1)*digamma(mu_jkml)).sum()))
        ELBO += (gamma(log(sum(alpha0_jkm)) - gamma(log(alpha0_jkm)).sum())
        ELBO += entropy(zg_ikm.reshape(num_items, -1, 1).sum())
    return z_ik, ELBO
```
A.3 iBCC

```python
import numpy as np
import scipy.sparse as ssp
from scipy.special import digamma

def ibcc(tuples, a_v=4, b_v=1, alpha=1):
    num_items, num_workers, num_classes = tuples.max(axis=0) + 1
    num_labels = tuples.shape[0]
    y_is_one_kij = []
    y_is_one_kji = []
    for k in range(num_classes):
        selected = (tuples[:, 2] == k)
        coo_ij = ssp.coo_matrix((np.ones(selected.sum()), tuples[selected, :2].T),
                                  shape=(num_items, num_workers), dtype=np.bool)
        y_is_one_kij.append(coo_ij.tocsr())
        y_is_one_kji.append(coo_ij.T.tocsr())

    # initialization
    prior_kl = np.eye(num_classes)*(a_v-b_v) + b_v
    n_jkl = np.empty((num_workers, num_classes, num_classes))

    # MV initialize Z
    z_ik = np.zeros((num_items, num_classes))
    for l in range(num_classes):
        z_ik[:, [l]] += y_is_one_kij[l].sum(axis=-1)
    z_ik /= z_ik.sum(axis=-1, keepdims=True)
    last_z_ik = z_ik.copy()

    for iteration in range(500):
        # E step
        Eq_log_pi_k = digamma(z_ik.sum(axis=0) + alpha) # - digamma(num_items + num_classes + alpha)
        for l in range(num_classes):
            n_jkl[:, :, l] = y_is_one_kji[l].dot(z_ik)
            Eq_log_v_jkl = digamma(n_jkl + prior_kl[None, :, :]) - digamma(n_jkl.sum(axis=-1) + prior_kl.sum(axis=-1))[None, :, None]

        # M step
        last_z_ik[:] = z_ik
        z_ik[:] = Eq_log_pi_k
        for l in range(num_classes):
            z_ik += y_is_one_kij[l].dot(Eq_log_v_jkl[:, :, l])
        z_ik = np.exp(z_ik)
        z_ik /= z_ik.sum(axis=-1, keepdims=True)
        if np.allclose(last_z_ik, z_ik, atol=1e-3):
            break
    return z_ik
```
Bibliography


Author/s: Li, Yuan

Title: Probabilistic models for aggregating crowdsourced annotations

Date: 2019

Persistent Link: http://hdl.handle.net/11343/227106

File Description: Probabilistic Models for Aggregating Crowdsourced Annotations

Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.