PROBABILISTIC GRAPHICAL MODELS FOR CROWDSOURCING AND TURBULENCE

by

Zhaorui Luo

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF MATHEMATICS

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2020

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: Zhaorui (Jerry) Luo
titled:

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Junming Yin
Junming Yin — Date: Dec 16, 2020

JOE WATKINS
JOE WATKINS — Date: Jan 12, 2021

Hao Zhang
Hao Zhang — Date: Jan 5, 2021

Michael Chertkov
Michael Chertkov — Date: Dec 17, 2020

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Junming Yin
Junming Yin — Date: Dec 16, 2020
Department of Management Information Systems

ACKNOWLEDGMENTS

I would like to express my deepest thanks and gratitude to my advisor Junming Yin, without your support and guidance this dissertation would not be remotely possible. I want to thank you for having endless patience and eagerness to share knowledge with me.

I would also like to thank Misha Chertkov and my mentors at Los Alamos National Laboratory: Andrey Lokhov, Arvind Mohan, Sidhant Misra, and Marc Vuffray who helped guide me in the turbulence chapter. I very much enjoyed and learned a lot from each and every one of you in the span of our collaboration. Thank you for helping me out in a field I had very little knowledge or experience in.

I would like to thank my professors at The University of Arizona who always give full effort and attention to their students: Ning Hao, Tom Kennedy, Sunder Sethuraman, Hao Helen Zhang to name a few.

I would also like to thank my fellow classmates in the math and applied math programs, including Brian Bell, Hannah Biegel, Kyung Mi Chung, Victoria Gershuny, Cody Gunton, Philip Hoskins, Anthony Kling, Rachel Knak, Sam McLaren, Jason Quinones, Daniel Rossi, Yuan Tao, Jun Wang, Ken Yamamoto. I'd also like to thank some friends I met at Los Alamos: Rusty Davis, Alexandra DeLucia, Anya Katsevich, David Li, Ryan Maki. You are all amazing people and I hope you all accomplish your goals and dreams.

DEDICATION

Dedicated to my parents, Haobin Luo and Xuemei Li.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

Graphical models provide a useful framework and formalism from which to model and solve problems involving random processes. We demonstrate the versatility and usefulness of graphical models on two problems, one involving crowdsourcing and one involving turbulence.

In crowdsourcing, we consider the problem of inferring true labels from a set of crowdsourced annotations. We design generative models for the crowdsourced annotations involving as latent variables the worker reliability, the structure of the labels, and the ground truth labels. Furthermore, we design an effective inference algorithm to infer the latent variables.

In turbulence, we consider the problem of modeling the mixing distribution of homogeneous isotropic passive scalar turbulence. We consider models specifying the conditional distribution of a coarse grained node given its adjacent coarse grained nodes. In particular, we demonstrate the effectiveness of a higher order moments based extension of the Gaussian distribution.

CHAPTER 1

INTRODUCTION

Probabilistic graphical models[85, 47] provide an appealing and useful framework for solving scientific problems involving randomness and structure. Graphical model formalism provides a language and a suite of problem solving frameworks to apply.

In this thesis, we examine two different applications of graphical models. The first involves Bayesian modeling of the problem of labeling unlabeled data that is multi-label in nature, given a set of annotations from crowdsourced workers. The true labels are obtained by solving an inference problem. The second involves modeling the scalar field of a turbulent flow and constraining the model to satisfy the known physics thereby allowing for effective inference and learning.

In this section, we will first introduce the background of graphical models. We will then give a brief overview of the nature of the two problems. In addition, we will motivate the usage of graphical models in each of these two contexts. Baseline models will be discussed in this introduction. Novel extensions to the baseline models will be alluded to in this introduction and will be the main focus of the following two chapters. In the final chapter, we will discuss the advances made in the body chapters and connect them to the general development of graphical model methodology.

1.1 Graphical Models

In a graphical model, a set of random variables $\mathbf{x} = \{x_1, \ldots, x_p\}$ is associated with a graph $\mathcal{G} = (V, E)$ where each vertex $v \in V$ is associated with a random variable in $\mathbf{x}$. The set of edges E will be used to dictate the dependencies between the random variables in $\mathbf{x}$. These dependencies are different depending on if the graph is *directed* or *undirected*.

For a directed graphical model, let $\mathcal{P}(x_i)$ denote the parents of x_i, i.e. the set of vertices v such that there exists a directed edge from v to x_i. A directed graphical

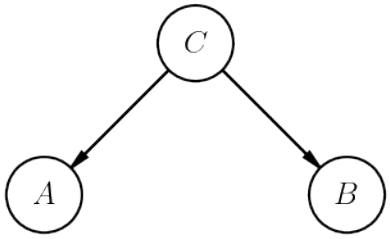


Figure 1.1: Directed Graphical Model Example.

model represents the joint distribution of $\mathbf{x}$ if the distribution $p(\mathbf{x})$ can be factored in the following manner:

$$p(\mathbf{x}) = \prod_{i=1}^{p} p(x_i|\mathcal{P}(x_i)). \tag{1.1}$$

We demonstrate the utility by considering the joint probability distribution in Figure 1.1. By the chain rule of probability, we can decompose in $3! = 6$ ways including:

$$p(A, B, C) = p(A)p(B|A)p(C|A, B) \tag{1.2}$$

However, by using the factorization prescribed by the graphical model, we have:

$$p(A, B, C) = p(C)p(A|C)p(B|C). \tag{1.3}$$

This factorization avoids the computational issue of describing a conditional distribution conditioned on 2 variables. In a simple example where the variables are all discrete with n possible outcomes, $p(C|A, B)$ would require n^3 different parameters to fully describe, thus Equation 1.2 would require n^3+n^2+n parameters. In contrast, by using Equation 1.3, we only have to use $n + 2n^2$ parameters to fully describe the joint distribution. Thus probabilistic graphical models allow us to more succinctly and compactly express complex joint probability distributions.

Many canonical statistical models can be expressed in terms of directed graphical models, including Hidden Markov Models, linear dynamical systems, and the Kalman filter [71]. Informally, an edge from a node A to a node B can be interpreted as

variations in A causing variations in B, thereby making directed graphs a useful tool in the field of causal inference [66].

In *undirected* graphical models, rather than specifying the joint distribution using conditional probability distributions, we specify a factorization of the joint distribution using *potential functions* associated with all of the cliques $\mathcal{C}$ of a graph. A clique is a subset of the graph such that every two vertices in the clique are adjacent. A joint distribution can be factorized as

$$p(\mathbf{x}) \propto \prod_{\mathbf{c} \in \mathcal{C}} \psi_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}), \tag{1.4}$$

where $\mathbf{x}_{\mathbf{c}}$ denotes the set of vertices in the clique $\mathbf{c}$. In an undirected graphical model, conditional distributions can be easily expressed in terms of the neighbors in the graph, i.e.

$$p(x_i | x_{\neg i}) = p(x_i | \mathcal{N}(x_i)) \tag{1.5}$$

where $\neg i$ denotes all indices except for i and $\mathcal{N}(i)$ denotes the neighbors of i.

Statistical models that are commonly expressed as undirected graphical models include the Ising model and the Potts model. Undirected graphical models have been applied in the study of sensor networks [55], genomics [90], and power grids [19].

1.2 Crowdsourcing

We consider the problem of learning the true labels of a dataset. More precisely, we consider $z_{i,j} \in \{0, 1\}$ for $i = 1, \ldots, N$ and $j = 1, \ldots, C$. Here, i indexes the N instances of the dataset, while j indexes the C possible labels. As a concrete example, we may consider a dataset of N images, which have possibly up to C different labels, e.g. sun, sky, grass, etc. Each image may have multiple labels, thereby making the problem *multi-label* [92], i.e. the ground truth of an instance i is represented by $\mathbf{z}_i \in \{0, 1\}^C$.

We assume that we have data available in the form of *crowdsourced annotations* from a set of L workers. We assume that each instance i is *not* labeled by all L

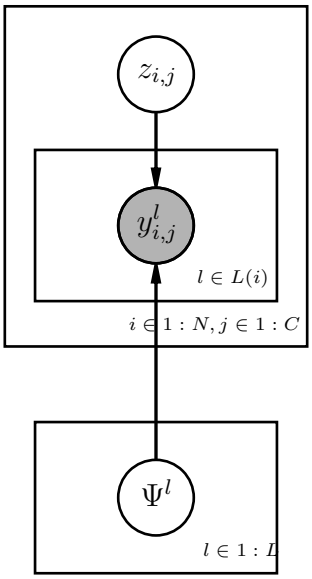


Figure 1.2: Graphical model for the crowdsourcing problem.

workers but rather a subset $N(i) \subset \{1, \ldots, L\}$ of workers. Then for each i, we have access to $y^l_{i,j} \in \{0, 1\}$ indicating the belief of worker $l \in N(i)$ as to whether label j is present in instance i. The problem we wish to solve is to ascertain the values $z_{i,j}$ given the set of crowdsourced annotations $y^l_{i,j}$.

In order to solve this problem, we consider possible probabilistic generative models for random variables $z_{i,j}$ and $y^l_{i,j}$. Because we have access to the crowdsourced annotations $y^l_{i,j}$ but not the ground truth labels $z_{i,j}$, we treat $z_{i,j}$ as a latent variable in our model and $y^l_{i,j}$ as observed variables. The mechanism that links together these two variables is the worker annotation process, thus we can posit worker specific variables Φ_l for each worker l, which measures a worker's effectiveness and reliability. As such, we can propose the following factorization of the joint probability distribution:

$$p(\Phi_l, z_{i,j}, y^l_{i,j}) = p(\Phi_l)p(z_{i,j})p(y^l_{i,j}|\Phi_l, z_{i,j}) \tag{1.6}$$

which can be expressed succinctly using a directed graphical model (see Figure 1.2). Here, we gray out $y^l_{i,j}$ to denote that the variable is observed and the remaining white variables are latent unobserved variables. We also use panels/boxes to denote the index set each variable ranges over. We introduce this remedial model noting that it

is incomplete without further specification about the components in the right side of Equation 1.6. In Chapter 2, we complete and extend this model in a Bayesian manner. While the model accounts for varying worker reliability, we find that a model that additionally accounts for *label dependency* is able to improve results. Along the way, we are able to derive efficient computational algorithms.

1.3 Turbulence

Passive scalar turbulence attempts to describe the concentration of an injected pollutant in a turbulent fluid. Turbulence for both passive scalars and fluids is highly non-equilibrium in contrast to equilibrium mechanics where energy eventually dissipates evenly. It is this non-equilibrium behavior which we are interested in modelling in a probabilistic manner. Our goal is to provide a framework for characterizing the mixing distribution of passive scalar turbulence. As a proof of concept, we will be working with the example of homogeneous isotropic passive scalar turbulence[77]. In this context, homogeneous means that the distribution is translation/shift invariant, and isotropic means that the distribution is rotation invariant.

Our data comes from a direct numerical simulation generating a passive scalar[16] for a $128 \times 128 \times 128$ box (see Figure 1.3). We wish to be able to capture the (space) dynamics of the passive scalar turbulence through a grid-based graphical model, i.e. the nodes and edges of the grid for which the simulation was based on also corresponds to the nodes and edges of the graphical model.

However, this data may indeed be too *fine-grained* for the particular model we wish to consider. The graphical model we wish to use assumes that the value of the passive scalar at a node is conditionally independent of all non-adjacent nodes given the value of the passive scalar at the adjacent nodes. A perhaps more reasonable assumption is that this conditional independence doesn't hold given only immediately adjacent nodes, but does hold if given all nodes within a certain distance r.

One work around is to *coarse-grain* our data. Picking a scale r, we average $r \times r \times r$ boxes into single nodes, creating a $d \times d \times d$ passive scalar box (see Figure 1.4). Our graphical model keeps the same basic grid structure. More precisely, for each node

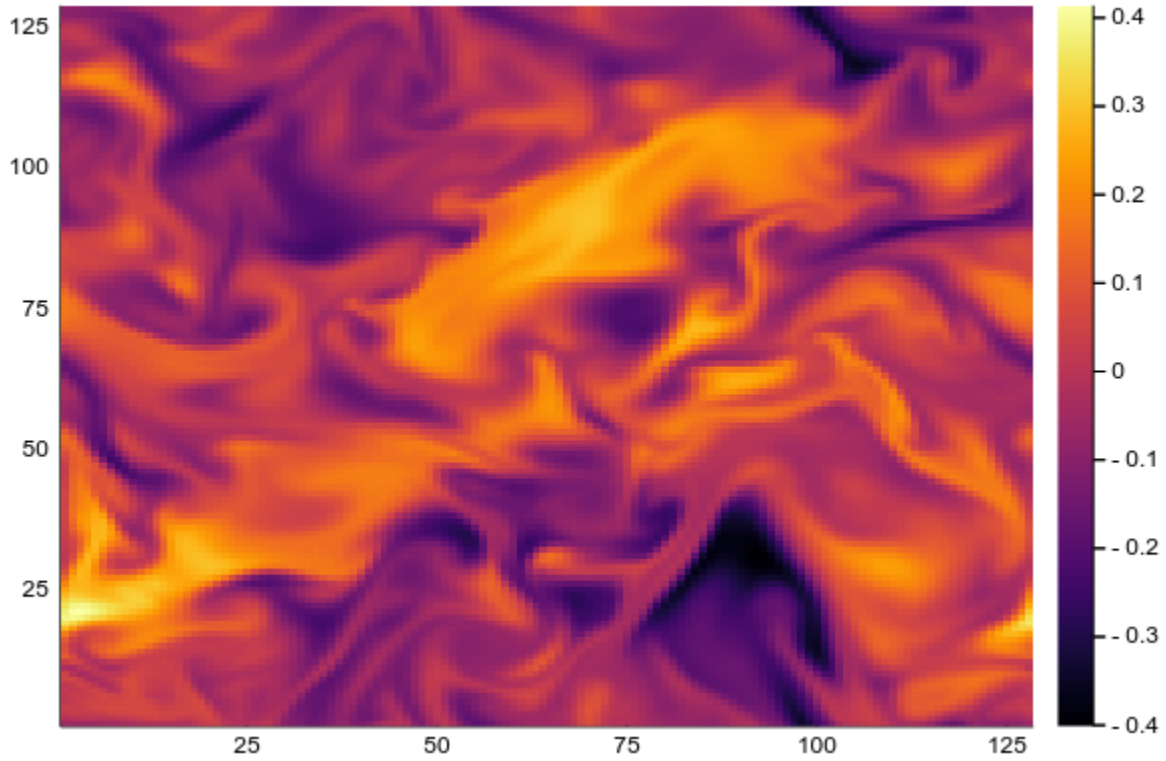


Figure 1.3: Heat map of cross section of passive scalar.

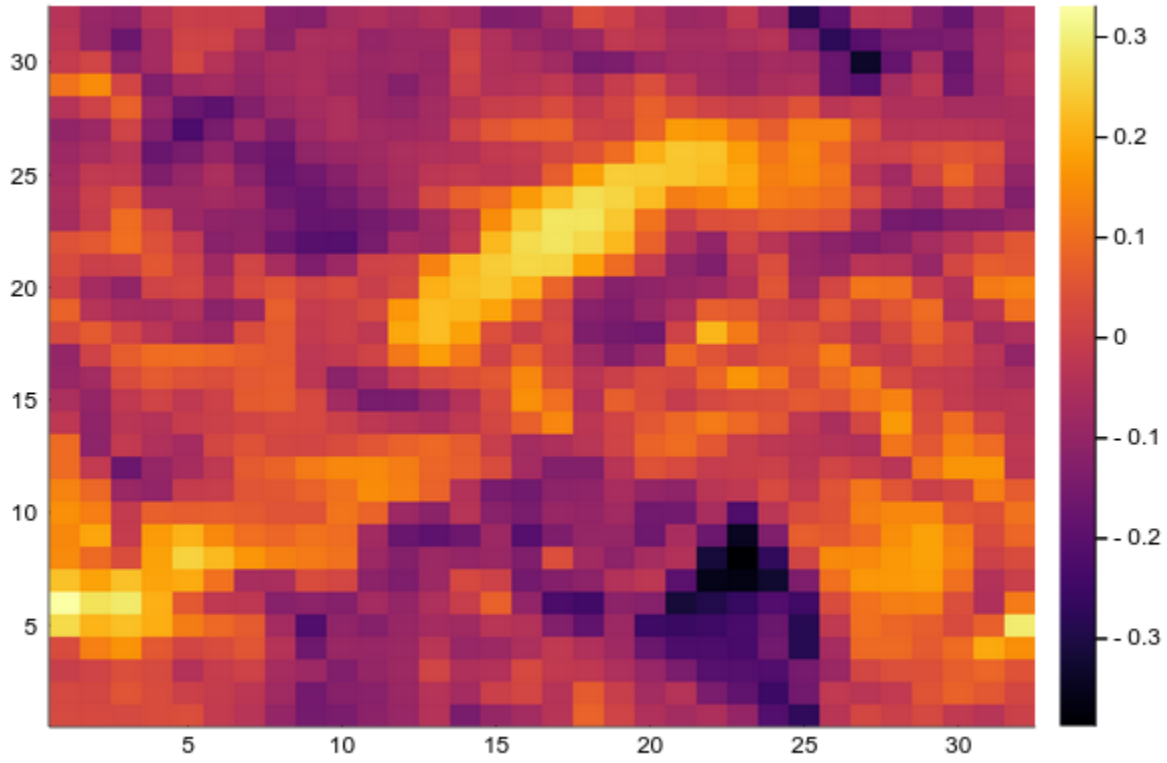


Figure 1.4: Heat map of cross section of coarse grained passive scalar with $r = 4$.

of $i = (x, y, z)$ of the $d \times d \times d$ box, we let ϕ_i be the passive scalar at node i and let $\backslash i$ denote all nodes except for i. Then the conditional probability of ϕ_i obeys the following principle:

$$p(\phi_i|\phi_{\backslash i}) = p(\phi_i|\phi_{i'}, i' \in N(i)) \tag{1.7}$$

where $N(i)$ refers to the 6 nodes adjacent to node i (2 in each of the 3 dimensions).

In Chapter 3, we first use this conditional distribution stipulation to arrive at a distribution that is ultimately multivariate Gaussian. We then extend this to a richer model that is able to outperform the multivariate Gaussian model at inference tasks.

CHAPTER 2

CROWDSOURCED MULTI-LABELLING: A VARIATIONAL BAYESIAN APPROACH

2.1 Introduction

Recent technological advances have generated a growing interest in the development of hybrid systems capable of combining human and computational intelligence to perform tasks that are difficult to solve by humans or computers alone. One compelling example of such human-in-the-loop systems is crowdsourcing, which refers to a distributed problem-solving strategy that leverages the cognitive and computational power of large crowds to achieve a cumulative goal [22]. Microtask crowdsourcing is particularly valuable when a task can be divided into many smaller and independent units, each of which requires a certain amount of human perception or common-sense knowledge that is otherwise difficult for a computer to complete. Crowdsourcing platforms such as MTurk[1] and CrowdFlower (rebranded as Figure Eight[2]) provide an online marketplace where task requesters can post a batch of microtasks for a crowd of workers to complete for a small monetary compensation. However, as the information provided by crowd workers can be prone to errors, it is necessary and critical to design additional algorithmic techniques to improve the quality of crowdsourced results. As such, microtask crowdsourcing is a well-suited and prominent mechanism for humans and computers to collaborate to solve large-scale data processing and annotation problems.

While there is a great economic opportunity in this rapidly growing industry, using microtask crowdsourcing for label acquisition also presents several challenges. The major problem is that the labels collected from a crowd are often of varying quality and accuracy. Empirical evidence from MTurk shows that crowd workers can be

[1] https://www.mturk.com/

[2] https://www.figure-eight.com/

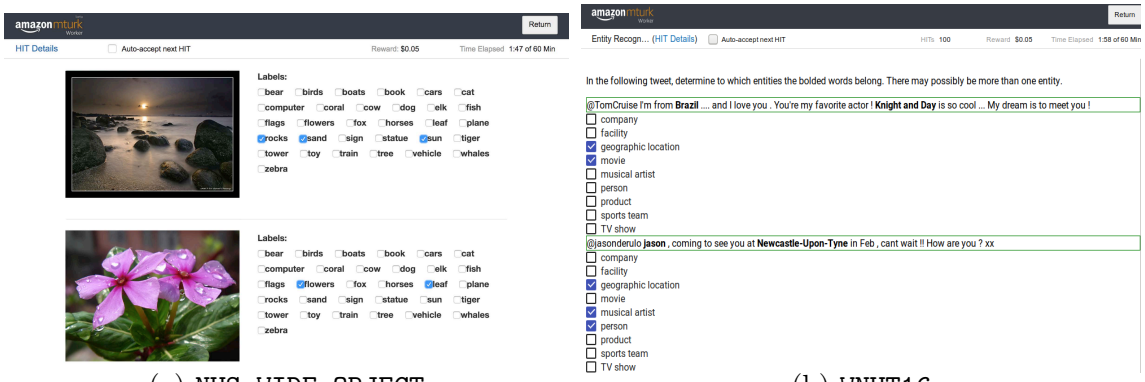
imperfect or unreliable in various different ways. They may be incompetent because they are unskilled or they may be sloppy and work as fast as possible [45]. Sometimes they may ignore the rules provided by task requesters [28]. Even if they are skilled and make an effort, unintentional errors can still occur because many microtasks are complex and even tedious. While requesters may specify additional qualifications that workers must meet to be able to work on the tasks, such as requiring the historical approval rate on MTurk above a certain threshold or using a few "control questions" as an entrance exam [51, 41, 32], there is still no guarantee for the reliability of the selected workers or the quality of their responses[3] [44]. A more common strategy employed by task requesters in practice is to use repeated labeling [37], i.e., assigning each task to multiple workers and asking each worker to complete multiple tasks. Such a strategy raises two key challenges in a microtask crowdsourcing application:

1. how to estimate the ground truth labels from the redundant and noisy labels provided by a crowd of workers with heterogeneous (albeit unknown) reliability;

2. how to better evaluate workers and quantify their annotation performance, instead of relying on their historical approval rate.

Much of the existing literature on learning from crowd labeling has focused on the single-label setting (see Section 2.2.1 for related work), in which each item to be annotated can be assigned *one and only one* label. It includes both binary-class labeling tasks, e.g., determining whether an email message is spam or not, and multi-class labeling tasks, e.g., identifying the digit value $\{0, 1, \ldots, 9\}$ from an image. However, in various application domains it is quite common that data sets are inherently *multi-label* in nature. In electronic health records (EHRs), each patient record can be assigned multiple codes representing different symptoms and diagnoses. See Figure 2.1 for two more examples of multi-label datasets. Given the prevalence of the multi-label setting, which subsumes the single-label problem as a special case, it is critical to go beyond the single-label assumption.

[3] This is further confirmed by one of our MTurk experiments (see Section 2.5.3 and Figure 2.10), in which we find that workers with an over 90% HIT approval rate in our empirical study can have an accuracy of only 60-70% when they are asked to annotate a multi-label image dataset.



(a) NUS-WIDE-OBJECT (b) WNUT16

Figure 2.1: **A Screenshot of MTurk Experiments on Two Multi-label Datasets.** The ground truth labels are highlighted, though not shown to workers during the experiments. See Section 2.5 for a detailed description of the datasets and experimental setup.

There are several approaches to transforming a multi-label crowdsourcing problem into a single-label problem. The simplest and most direct method casts a multi-label problem as a *multi-class* problem by treating each possible label combination as a unique class, resulting in 2^C distinct classes where C is the number of label candidates. The main disadvantage is that the number of unique classes is exponential in the number of label candidates, which raises both estimation and computational challenges, especially in applications where the potential number of label candidates reaches several hundred [80]. An alternative approach is to reduce a multi-label problem to multiple independent *binary-class* problems, one for each label candidate, such that existing binary-class crowdsourcing methods can be employed to combine workers' responses to each label candidate separately. The limitation of this approach is that it completely ignores the dependency among multiple labels, which can arise naturally in real-world applications. For example, a document in the `military` category is more likely to be associated with `politics` than `education` or `health`. In the early stages of a multi-label crowdsourcing project, the number of annotations is too small to accurately estimate each of the true labels separately, so it is important to exploit label correlations (though unknown in the data collection phase) to improve the quality of aggregated results and to reduce the label acquisition cost.

Contributions and Technical Insights. In this work, we propose a new Bayesian

hierarchical framework that incorporates label dependency modeling for solving the multi-label crowdsourcing problem, i.e., inferring ground truth labels of each item and estimating reliability of each worker from a set of crowdsourced annotations. We first introduce the Bayesian Model with Independent Labels (BIL), in which multiple labels of each item are independently generated and hence no label dependency is being considered (Section 2.3.2). BIL extends the MV algorithm by modeling heterogeneous worker quality with a variant of the "two-coin" model [69]. Next, as the main contribution of this work, we construct a second model that uses a mixture of Bernoulli distributions [50, 7] to capture label dependency in an effective manner, and then we integrate this mixture model into the annotation process of crowd workers (Section 2.3.3). We also develop a nonparametric extension based on Dirichlet process mixtures [26, 3] to automatically infer the number of mixture components (Section 2.3.4). The resulting Bayesian Model with Mixture of Bernoulli (BMB) and its Bayesian nonparametric extension (BNMB), along with an efficient collapsed variational inference algorithm (Section 2.4), allow us to achieve superior performance in multi-label crowdsourcing applications compared with state-of-the-art alternative approaches (Section 2.5). We conclude the paper by discussing the implications of our work and by highlighting possible future research directions (Section 2.6).

2.2 Related Work

Our work contributes to the growing literature on statistical learning approaches to crowd labeling, with a focus on the general multi-label setting to simultaneously infer the unknown ground truth labels and worker reliability from crowdsourced data. In this section, we provide a brief review of the literature that is pertinent to the two key factors underlying our model development: heterogeneous worker quality and complex label dependency.

2.2.1 Worker Quality in Crowdsourcing

There is a substantial body of literature on modeling and estimating worker quality for the single-label crowdsourcing problem. When the annotation task is binary,

the most commonly used probabilistic models for specifying how individual workers annotate items are the one-coin model [30, 15, 44] and the two-coin model [69]. The one-coin model assumes that each worker has a single parameter to encode his labeling quality, i.e., the probability of providing the correct annotation to an assigned item. By contrast, the two-coin model uses two parameters per worker, the sensitivity and specificity, to represent the accuracy of correctly identifying the true positives and true negatives, respectively. For multi-class crowd labeling tasks, most existing approaches are based on the seminal Dawid-Skene model [17], in which each annotator is associated with a worker-specific confusion matrix[4] and the (a, b)-th entry of the matrix corresponds to the probability of labeling an item with true class a as in class b. Given the observed labels, the true labels and worker confusion matrices can be jointly estimated by the expectation-maximization (EM) algorithm [20]. Based on the estimated worker confusion matrices, Ipeirotis et. al [38] presented an approach that enables the incorporation of cost-sensitive classification errors in quantifying the expected cost of each worker. Kim [46] provided a Bayesian extension of the Dawid-Skene model, referred to as independent Bayesian Combination of Classifiers (iBBC), by imposing a conjugate prior on each row of the worker confusion matrices. A variety of extensions of the Dawid-Skene and iBCC models have been proposed, mainly by including other relevant factors of the microtask crowdsourcing problem as additional variables, such as task difficulty [89], clusters or groupings of workers [82, 83, 63], and feature information of items [88, 69].

Relatively less research has been conducted in the multi-label context, where each item can be associated with multiple categories from a set of label candidates. [4] proposed an extension of the iBCC model for the setting where both ground truth labels and worker annotations have to specify the proportion of each category, e.g., {anger: 50%, surprise: 25%, disgust: 25%} for a document and {flower: 40%, leaf: 60%} for an image. This extra step of asking workers to exert additional effort to estimate the proportion of each category in each item is neither suitable nor necessary

[4] When the total number of classes is two, the Dawid-Skene model becomes equivalent to the two-coin model.

for every multi-label crowdsourcing scenario, because most real-world applications only require to identify the presence or absence of each category. Duan [23] introduced a method that treats each subset of label candidates as a unique class and then uses the Dawid-Skene model (or its variants) to parameterize the annotation process of each worker. This multi-class based approach can easily suffer from the sparsity of annotations, because a large number of parameters in the confusion matrices need to be estimated when the number of label candidates is large. Padmanabhan [65] proposed a variant of the one-coin model, in which each annotator is characterized by a single worker-specific parameter that indicates the probability of reporting the ground truth for each label candidate. Bragg [10] considered a variant of the two-coin model, where the accuracy of identifying a label present in an item (i.e., sensitivity) is separated from the accuracy of recognizing a label that is not present in an item (i.e., specificity). However, the model assumes that all workers have an equal level of sensitivity and specificity, an assumption that rarely holds in practice. In this work, we develop a variant of the two-coin model that allows for heterogeneity in worker quality (Section 2.3.2). As observed in [10] and our own empirical study on MTurk (Figure 2.10), worker specificity is much higher than worker sensitivity. This is due to the nature of the multi-label annotation tasks, in which workers are more likely to miss a label that is actually present in an item than to select a label that is not present. Therefore, using two parameters to represent worker quality is more flexible, accurate, and effective in the multi-label crowdsourcing setting[5].

2.2.2 Label Dependency in Multi-label Learning

Label dependency has been mainly discussed in the *supervised* multi-label classification literature; see [79] and [92] for an overview. The goal is to build a classifier that maps an instance to a subset of labels by taking advantage of potential correlations among labels in the training set. It has been shown that effective exploitation of la-

[5]For each model proposed in this paper, we have also developed its one-coin version. Empirical results show that the two-coin version of each model significantly outperforms the corresponding one-coin version.

bel dependency can significantly improve the prediction performance of a multi-label classifier.

Exploiting label dependency in the *unsupervised* multi-label crowdsourcing problem is more challenging, because label correlations cannot be directly computed from the noisy annotations provided by a crowd of workers. Existing approaches include assuming no dependency in the ground truth labels [65], modeling pairwise co-occurrence between label candidates [10], capturing only negative correlation [36], and considering all possible label combinations [23]. All of these approaches have their own limitations. In this work, we propose to use a mixture of Bernoulli distributions (Section 2.3.3), a flexible and powerful model that is capable of capturing complex label dependency with a fairly small number of mixture components [50, 7, 52]. Moreover, it is able to model both positive and negative correlations, and the sign of correlation is completely data-driven rather than being encoded in the model assumption.

[91] also made an attempt to use a mixture distribution to capture label dependency in the multi-label crowdsourcing context. Their approach is based on a two-level probabilistic model (referred to as MCMLD) in which there is no generative probabilistic process for worker reliability parameters, as well as other parameters. Moreover, it assumes a separate worker reliability parameter for each label candidate, and the classical expectation-maximization (EM) algorithm is applied to perform point estimation for all parameters in the MCMLD model. This leads to the following statistical, computational, and practical issues. (1) Point estimation of worker reliability parameters can easily suffer from sparsity of annotations as well as low label cardinality (average number of labels present in an item) in the assigned tasks. (2) The computational complexity of the E-step is exponential in the number of label candidates, making their method only applicable to a limited number of crowdsourcing applications. (3) As worker reliability is not probabilistically modeled, there is no natural way to incorporate each worker's historical performance and to update worker reliability profiles in a sequential manner. In this work, we propose a variational Bayesian approach that can overcome all of these issues, which is the topic of the next two sections. In particular, we treat the worker reliability parame-

ters as *latent variables* rather than a large number of parameters to be estimated, we define appropriate generative probabilistic models for those latent variables, and we can maintain and update posterior distributions over them with a computationally efficient variational inference algorithm.

2.3 Hierarchical Bayesian Modeling Framework

In this section, we begin with a formulation of the microtask crowdsourcing problem in the general multi-label setting, and then we present our hierarchical Bayesian modeling assumptions underlying the annotation process of crowd workers.

2.3.1 Problem Setup

We consider a typical microtask crowdsourcing application in which task requesters wish to make use of an online crowdsourcing service (e.g., MTurk) to recruit workers to annotate a set of unlabeled items. In the multi-label setting, each item $i \in \{1, 2, \ldots, N\}$ can be associated with a subset of all label candidates $\{1, 2, \ldots, C\}$. We use a binary vector $\mathbf{z}_i \in \{0, 1\}^C$ to represent its ground truth labels, where $z_{i,j}$ indicates if item i has label j.

Suppose that there are L crowd workers to provide annotations for N items. We focus on the static scenario where annotations are collected *all at once* before the aggregation and inference step. The set of items labeled by worker l is denoted by $N(l) \subseteq \{1, 2, \ldots, N\}$. Because crowd workers can have a wide range of reliability and expertise, the collected labels are a set of noisy annotations $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^L\}$, with $\mathbf{y}^l \in \{0, 1\}^{|N(l)| \times C}$ provided by worker l. Each entry $y^l_{i,j} \in \{0, 1\}$ indicates whether worker l assigns label j to item i. An equivalent representation is $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, where $\mathbf{y}_i \in \{0, 1\}^{|L(i)| \times C}$ and $L(i) \subseteq \{1, 2, \ldots, L\}$ is the set of workers who have annotated item i. We will use both representations throughout the paper.

Given noisy annotations $\mathbf{Y}$ collected from a crowdsourcing application, the goal is to *simultaneously* infer (1) the ground truth labels $\mathbf{z} \doteq \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\} \in \{0, 1\}^{N \times C}$

of N items, and (2) the reliability of L workers, i.e., sensitivity $\boldsymbol{\Psi} \in [0,1]^L$ and specificity $\bar{\boldsymbol{\Psi}} \in [0,1]^L$ (to be formally defined in Section 2.3.2).

2.3.2 Bayesian Model with Independent Labels (BIL)

First, we propose a hierarchical model (see Figure 2.2a) in which multiple labels of an item are independently generated. Therefore, the model assumes that there is no label dependency. Specifically, we use τ_j of $\boldsymbol{\tau} \in [0,1]^C$ to denote the frequency of label $j \in \{1, 2, \ldots, C\}$. Given the label frequency parameter $\boldsymbol{\tau}$, the multiple true labels $\{z_{i,1}, \ldots, z_{i,C}\}$ of item i are conditionally independent:

$$z_{i,j} \mid \tau_j \;\sim\; \text{Bernoulli}(\tau_j). \tag{2.1}$$

Given the true labels $\mathbf{z}_i$, the annotation process of each worker $l \in L(i)$ is assumed to follow a variant of the two-coin model [69]: for each label present in the ground truth, the worker l provides a correct annotation with *sensitivity* $\Psi^l := \mathbb{P}(y^l_{i,j} = 1 \mid z_{i,j} = 1)$; for each label that is not present in the ground truth, the worker l provides a correct annotation with *specificity* $\bar{\Psi}^l := \mathbb{P}(y^l_{i,j} = 0 \mid z_{i,j} = 0)$. Hence, the conditional probability of each collected annotation $y^l_{i,j}$ follows[6]

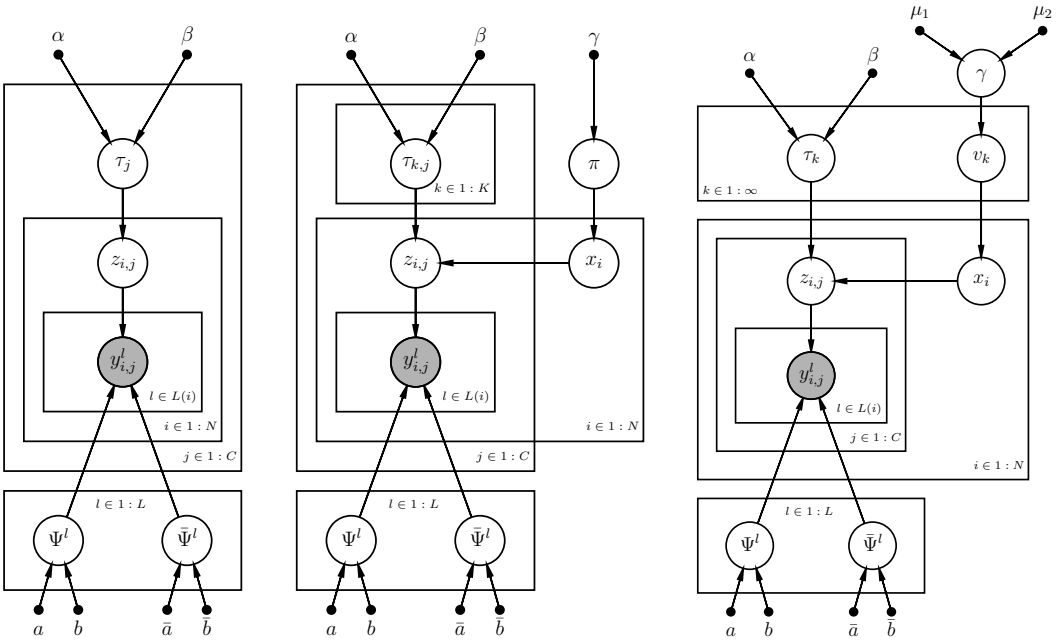
$$y^l_{i,j} \mid \Psi^l, \bar{\Psi}^l, \mathbf{z}_i \;\sim\; \text{Bernoulli}\big((\Psi^l)^{z_{i,j}}(1-\bar{\Psi}^l)^{1-z_{i,j}}\big). \tag{2.2}$$

Finally, we place conjugate priors over the parameters $\Psi^l, \bar{\Psi}^l$, and τ_j to complete our Bayesian Model with Independent Labels (BIL).

$$\begin{aligned} \Psi^l \mid a, b \;&\sim\; \text{Beta}(a, b), \\ \bar{\Psi}^l \mid a, b \;&\sim\; \text{Beta}(\bar{a}, \bar{b}), \\ \tau_j \mid \alpha, \beta \;&\sim\; \text{Beta}(\alpha, \beta). \end{aligned} \tag{2.3}$$

The hyperparameters $\{a, b, \bar{a}, \bar{b}\}$ indicate how strong our prior belief is about worker reliability, and $\{\alpha, \beta\}$ specify our prior belief about the proportion of present labels in each item. The choice of those hyperparameters is often context-dependent. See Section 2.5.1 for more details.

[6] The assumption made is that the worker sensitivity and specificity are independent of the label candidate under consideration.



(a) BIL (b) BMB (c) BNMB

Figure 2.2: **Probabilistic Graphical Model Representation of the BIL, BMB, and BNMB Models.** The observed binary variables $y_{i,j}^l$, indicated by shaded nodes, represent whether worker l assigns label j to item i. Solid nodes represent hyperparameters. All the other variables, including the ground truth labels $\mathbf{z}_i$ and worker reliability $\{\Psi^l, \bar{\Psi}^l\}$, are hidden. The annotation process is repeated for each item $i \in \{1, \ldots, N\}$ and for each worker $l \in L(i)$.

2.3.3 Bayesian Model with Mixture of Bernoulli (BMB)

In the BIL model, label dependency is completely ignored because it is easy to verify that $\mathbb{P}(z_{i,1}, z_{i,2}, \ldots, z_{i,C} \mid \alpha, \beta) = \prod_{j=1}^{C} \mathbb{P}(z_{i,j} \mid \alpha, \beta)$ by integrating out the parameters τ_j. However, dependency among different labels arises naturally in practice. It is crucial to exploit such dependency (though unknown in the data collection phase) so as to improve the overall performance of the model, especially in the early stages of a multi-label crowdsourcing application. At this stage, there is a limited amount of annotations available to accurately infer each of the true labels independently, so it is important to develop models that can share statistical strength among different label candidates.

To capture dependency among multiple labels, we propose to use a mixture of Bernoulli distributions [50, 7, 52]. Formally, we assume that the joint probability distribution over the ground truth labels $\mathbf{z}_i \in \{0,1\}^C$ is a weighted combination of K mixture components, in which multiple binary labels $\{z_{i,1}, z_{i,2}, \ldots, z_{i,C}\}$ are assumed to be independent Bernoulli within each component,

$$p(\mathbf{z}_i \mid \boldsymbol{\pi}, \boldsymbol{\tau}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{C} \tau_{k,j}^{z_{i,j}} (1 - \tau_{k,j})^{1 - z_{i,j}}. \tag{2.4}$$

The parameters $\pi_k \in [0,1]$ are the mixing coefficients satisfying $\sum_{k=1}^{K} \pi_k = 1$, and $\boldsymbol{\tau}_k \in [0,1]^C$ is the label frequency parameter for the kth Bernoulli mixture component. When there is only one mixture component (i.e., $K = 1$), the joint distribution of multiple binary labels in (2.4) reduces to the one used in the BIL model (2.1) so that all labels are independent. When $K \geq 2$, the C distinct binary labels become dependent in the joint distribution, even though they are independent within each mixture component. To see this, notice that the covariance matrix of $\mathbf{z}_i$ is *non-diagonal* [7, §9.3.3]:

$$\operatorname{cov}(\mathbf{z}_i) = \sum_{k=1}^{K} \pi_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\tau}_k \boldsymbol{\tau}_k^T \right) - \mathbb{E}(\mathbf{z}_i) \mathbb{E}(\mathbf{z}_i)^T, \tag{2.5}$$

where $\boldsymbol{\Sigma}_k = \operatorname{diag}\{\tau_{k,j}(1 - \tau_{k,j})\}$ and $\mathbb{E}(\mathbf{z}_i) = \sum_{k=1}^{K} \pi_k \boldsymbol{\tau}_k$. This flexible mixture distribution is capable of modeling both positive and negative correlations among multivariate binary variables.

The graphical model representation of our Bayesian Model with Mixture of Bernoulli (BMB) is shown in Figure 2.2b. For each item $i \in \{1, 2, \ldots, N\}$, we introduce a latent variable $x_i \in \{1, 2, \ldots, K\}$ to indicate which mixture component it is drawn from, i.e., $\mathbb{P}(x_i = k \mid \boldsymbol{\pi}) = \pi_k$ for $k = 1, 2, \ldots, K$. Combining the mixture of Bernoulli distributions over the ground truth labels with the two-coin model of the annotation process in (2.2), we obtain the following generative model:

$$\begin{aligned} x_i \mid \boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}), \\ z_{i,j} \mid x_i, \boldsymbol{\tau} &\sim \text{Bernoulli}(\tau_{x_i,j}), \\ y^l_{i,j} \mid \Psi^l, \bar{\Psi}^l, \mathbf{z}_i &\sim \text{Bernoulli}\big((\Psi^l)^{z_{i,j}}(1-\bar{\Psi}^l)^{1-z_{i,j}}\big). \end{aligned} \tag{2.6}$$

Finally, we impose conjugate priors over the parameters Ψ^l,$\bar{\Psi}^l$, $\tau_{k,j}$, and $\boldsymbol{\pi}$ as follows:

$$\begin{aligned} \Psi^l \mid a, b &\sim \text{Beta}(a, b), \\ \bar{\Psi}^l \mid a, b &\sim \text{Beta}(\bar{a}, \bar{b}), \\ \tau_{k,j} \mid \alpha, \beta &\sim \text{Beta}(\alpha, \beta), \\ \boldsymbol{\pi} \mid \gamma &\sim \text{Dirichlet}(\gamma). \end{aligned} \tag{2.7}$$

2.3.4 Bayesian Nonparametric Extension of BMB (BNMB)

Using the BMB model requires the pre-specification of the number of mixture components. Traditional model selection methods such as cross-validation, AIC, and BIC are not applicable because the crowdsourcing problem is unsupervised and we do not have access to a validation set for which the ground truth labels $\mathbf{z}$ is known. To automatically infer the number of mixture components K, we propose a Bayesian nonparametric extension based on *Dirichlet process mixture models* [26, 3]. In the Bayesian nonparametric approach, the number of mixture components K is part of the posterior distribution, and it is allowed to grow with data size rather than being fixed in advance. Formally, we assume that the joint probability distribution over the ground truth labels $\mathbf{z}_i$ follows the Dirichlet process mixture with a countably infinite number of Bernoulli components, in contrast to the finite mixture model used in (2.4). We construct the infinite mixture model with the stick-breaking representation [72] as

follows. First, we generate an independent sequence of random variables $\mathbf{v} = (v_k)_{k=1}^{\infty}$ as $v_k \mid \gamma \sim \text{Beta}(1, \gamma)$, and then define $\pi_k(\mathbf{v}) = v_k \prod_{m=1}^{k-1}(1 - v_m)$ for $k = 1, 2, \ldots, \infty$. It is known that this infinite sequence $\boldsymbol{\pi}(\mathbf{v}) = (\pi_k(\mathbf{v}))_{k=1}^{\infty}$ satisfies $\sum_{k=1}^{\infty} \pi_k(\mathbf{v}) = 1$ with probability one, so we can interpret $\pi_k(\mathbf{v})$ as the mixing coefficient of the kth component. The distribution of $\boldsymbol{\pi}(\mathbf{v})$ is called the Griffiths-Engen-McCloskey (GEM) distribution, written as $\boldsymbol{\pi}(\mathbf{v}) \sim \text{GEM}(\gamma)$. Next, we generate an independent sequence of random vectors $\boldsymbol{\tau} = (\boldsymbol{\tau}_k)_{k=1}^{\infty}$, where each $\boldsymbol{\tau}_k \in [0, 1]^C$ specifies the parameter of the kth Bernoulli mixture component with $\tau_{k,j} \mid \alpha, \beta \sim \text{Beta}(\alpha, \beta)$ for $j = 1, 2, \ldots, C$. Finally, for each item $i \in \{1, 2, \ldots, N\}$, the ground truth labels $\mathbf{z}_i \in \{0, 1\}^C$ are drawn from an infinite mixture model:

$$\begin{aligned} x_i \mid \mathbf{v} &\sim \text{Discrete}(\boldsymbol{\pi}(\mathbf{v})), \\ z_{i,j} \mid x_i, \boldsymbol{\tau} &\sim \text{Bernoulli}(\tau_{x_i,j}), \end{aligned} \tag{2.8}$$

where $x_i \in \{1, 2, \ldots, \infty\}$ is the latent variable indicating which mixture component item i is drawn from, i.e., $\mathbb{P}(x_i = k \mid \mathbf{v}) = \pi_k(\mathbf{v})$ for $k = 1, 2, \ldots, \infty$. Other probabilistic components, including the annotation process of crowd workers, are the same as in the BMB model. The graphical model representation of our proposed Bayesian Nonparametric Model with Mixture of Bernoulli (BNMB) is shown in Figure 2.2c, with a conjugate Gamma(u_1, u_2) prior placed on the concentration parameter γ of the Dirichlet process [25, 8].

2.4 Variational Inference Algorithm

Recall that our goal is to infer the ground truth labels $\mathbf{z} \in \{0, 1\}^{N \times C}$ of N items as well as sensitivity $\boldsymbol{\Psi} \in [0, 1]^L$ and specificity $\bar{\boldsymbol{\Psi}} \in [0, 1]^L$ of L workers, based on a collection of noisy crowdsourced annotations $\mathbf{Y}$. This is accomplished by computing the posterior distribution $p(\mathbf{z}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}} \mid \mathbf{Y})$ in our hierarchical models, which is unfortunately intractable for exact inference. Although there exist several "black box" approximate inference algorithms, such as the No-U-Turn-Sampling [NUTS; 34] and automatic differentiation variational inference [ADVI; 48] as well as their implementa-

tions in modern probabilistic programming packages (e.g., Stan[7] and PyMC3[8]), these gradient-based methods require the latent variables to be continuous and discrete variables to be explicitly marginalized out. However, marginalization is not tractable for all probabilistic models. Take our BMB model as a concrete example, marginalizing our the binary latent vector $\mathbf{z}_i \in \{0,1\}^C$ in the likelihood is intractable because the computational complexity scales exponentially with the number of label candidates. Therefore, these generic and derivation-free approaches cannot be applied to our models, and we must derive our own inference algorithm. In this section, we first introduce the mean-field (MF) variational inference [43, 85, 9], and we then develop a collapsed variational Bayesian (CVB) inference algorithm for the BMB model[9].

2.4.1 Mean-Field Variational Inference

The variational inference approach is a faster alternative to Monte Carlo sampling methods with competitive accuracy. Its key idea is to approximate the true but intractable posterior p by a simpler distribution q within a more tractable family of distributions, such that the Kullback-Leibler (KL) divergence $\mathrm{KL}(q \parallel p)$ is minimized. In the standard mean-field approximation, the true posterior of *all* latent variables $p(\mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}} \mid \mathbf{Y})$ is approximated by a fully factorized variational distribution

$$q_{\mathrm{MF}}(\mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}) = q(\mathbf{z} \mid \boldsymbol{\lambda}) q(\mathbf{x} \mid \mathbf{r}) q(\boldsymbol{\pi} \mid \mathbf{m}) q(\boldsymbol{\tau} \mid \mathbf{e}, \mathbf{f}) q(\boldsymbol{\Psi} \mid \mathbf{g}, \mathbf{h}) q(\bar{\boldsymbol{\Psi}} \mid \bar{\mathbf{g}}, \bar{\mathbf{h}}), \tag{2.9}$$

where the distributions q on the right-hand side are further factorized with respect to their components. As a consequence, each latent variable is governed by its own variational distribution q that is parameterized by a free "variational parameter". It is well-known that minimizing the KL divergence is equivalent to maximizing the *evidence lower bound* (ELBO), a lower bound $\mathcal{L}(\theta_v)$ on the log likelihood $\log p(\mathbf{Y})$ obtained by Jensen's inequality, with respect to the variational parameters $\theta_v =$

[7] https://mc-stan.org/

[8] https://docs.pymc.io/

[9] The inference algorithm for the BIL model (Section 2.3.2) can be derived by setting $K = 1$, and the inference procedure for the nonparametric extension (i.e., the BNMB model in Section 2.3.4) is derived in Appendix A.1.

$\{\boldsymbol{\lambda}, \mathbf{r}, \mathbf{m}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}, \bar{\mathbf{g}}, \bar{\mathbf{h}}\}$,

$$\log p(\mathbf{Y}) \geq \mathcal{L}(\theta_v) \doteq \mathbb{E}_{q_{\mathrm{MF}}}[\log p(\mathbf{Y}, \mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}})] - \mathbb{E}_{q_{\mathrm{MF}}}[\log q_{\mathrm{MF}}(\mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}})]. \tag{2.10}$$

The mean-field variational inference algorithm proceeds by applying coordinate ascent updates to maximize $\mathcal{L}(\theta_v)$. Specifically, we iteratively optimize over each coordinate of the variational parameters, while holding the other coordinates fixed. This procedure is guaranteed to converge to a local optimum of the ELBO.

2.4.2 Collapsed Variational Bayesian Inference

Although computationally efficient, the independence assumption made in the mean-field approximation (2.9) can potentially lead to inaccurate posterior estimates. One can obtain a better approximation by considering a strictly larger family of variational posterior

$$q_{\mathrm{CVB}}(\mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}) = q(\boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}} \mid \mathbf{z}, \mathbf{x}) q(\mathbf{z} \mid \boldsymbol{\lambda}) q(\mathbf{x} \mid \mathbf{r}), \tag{2.11}$$

where the components of latent variables $\mathbf{z}$ and $\mathbf{x}$ are still assumed to be mutually independent but there is no assumption on the posterior of $\{\boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}\}$ given $\mathbf{z}$ and $\mathbf{x}$. Because this family of distributions makes a strictly weaker assumption on the form of variational posterior than mean-field (2.9), it leads to a more accurate approximation of the true posterior and a tighter ELBO. It has been shown that assuming a variational distribution in the form of (2.11) is equivalent to marginalizing out the latent variables $\{\boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}\}$ in the joint distribution $p(\mathbf{Y}, \mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}})$ before applying mean-field approximation to the posterior of $\mathbf{z}$ and $\mathbf{x}$ [76, 39]. The resulting inference algorithm is known as collapsed variational Bayesian inference.

For the BMB model, the full joint distribution is given by

$$p(\mathbf{Y}, \mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}) = \left[\frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K} \prod_{k=1}^{K} \pi_k^{\gamma+n_k-1}\right] \cdot \left[\frac{1}{B(\alpha,\beta)^{KC}} \prod_{k=1}^{K}\prod_{j=1}^{C} \tau_{k,j}^{\alpha+n_{k,j,1}-1}(1-\tau_{k,j})^{\beta+n_{k,j,0}-1}\right] \cdot \left[\frac{1}{B(a,b)^L B(\bar{a},\bar{b})^L} \prod_{l=1}^{L} (\Psi^l)^{a+u^l_{1,1}-1}(1-\Psi^l)^{b+u^l_{1,0}-1}(\bar{\Psi}^l)^{\bar{a}+u^l_{0,0}-1}(1-\bar{\Psi}^l)^{\bar{b}+u^l_{0,1}-1}\right], \tag{2.12}$$

where Γ is the gamma function, B is the beta function, $n_k = \#\{i : x_i = k\}$, $n_{k,j,s} = \#\{i : x_i = k, z_{i,j} = s\}$ for $s \in \{0,1\}$, and $u^l_{s,t} = \#\{(i,j) : z_{i,j} = s, y^l_{i,j} = t\}$ for $s,t \in \{0,1\}$.

Marginalizing out the latent variables $\{\boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}\}$ in the joint distribution (2.12), we obtain the marginal distribution over $\{\mathbf{Y}, \mathbf{z}, \mathbf{x}\}$

$$p(\mathbf{Y}, \mathbf{z}, \mathbf{x}) = \left[\frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K} \frac{\prod_{k=1}^{K}\Gamma(\gamma+n_k)}{\Gamma(K\gamma+N)}\right] \cdot \left[\frac{1}{B(\alpha,\beta)^{KC}} \prod_{k=1}^{K}\prod_{j=1}^{C} B(\alpha+n_{k,j,1}, \beta+n_{k,j,0})\right] \cdot \left[\frac{1}{B(a,b)^L B(\bar{a},\bar{b})^L} \prod_{l=1}^{L} B(a+u^l_{1,1}, b+u^l_{1,0}) B(\bar{a}+u^l_{0,0}, \bar{b}+u^l_{0,1})\right]. \tag{2.13}$$

We now proceed to apply mean-field variational inference to approximate the posterior distribution $p(\mathbf{z}, \mathbf{x} \mid \mathbf{Y})$ with a fully factorized variational distribution

$$q_{\mathrm{MF}}(\mathbf{z}, \mathbf{x}) = \prod_{i=1}^{N}\prod_{j=1}^{C} q(z_{i,j} \mid \lambda_{i,j}) \prod_{i=1}^{N} q(x_i \mid \mathbf{r}_i), \tag{2.14}$$

where each component $z_{i,j}$ of the ground truth labels is parameterized as $q(z_{i,j}) = \mathrm{Bernoulli}(\lambda_{i,j})$ and each component x_i of the mixture component indices is parameterized as $q(x_i) = \mathrm{Discrete}(\mathbf{r}_i)$. It is worth noting that the mean-field assumption made here is in the collapsed space of latent variables $\{\mathbf{z}, \mathbf{x}\}$ instead of in the joint space of all latent variables $\{\mathbf{z}, \mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}\}$ as in (2.9).

Applying coordinate ascent updates to maximize the ELBO $\mathcal{L}(\theta_v)$

$$\log p(\mathbf{Y}) \;\geq\; \mathcal{L}(\theta_v) \doteq \mathbb{E}_{q_{\mathrm{MF}}}[\log p(\mathbf{Y}, \mathbf{z}, \mathbf{x})] - \mathbb{E}_{q_{\mathrm{MF}}}[\log q_{\mathrm{MF}}(\mathbf{z}, \mathbf{x})] \tag{2.15}$$

with respect to the variational parameters $\theta_v = \{\boldsymbol{\lambda}, \mathbf{r}\}$, we obtain the collapsed variational Bayesian (CVB) inference algorithm for the BMB model. The entire procedure

is summarized in Algorithm 1, where a dot indicates that the corresponding index is being summed over, e.g., $u^l_{1\cdot} = u^l_{1,1} + u^l_{1,0}$, and we use the superscript $\neg i$ and $\neg(i,j)$ to indicate the corresponding variables being excluded from the counts, e.g., $n^{\neg i}_{k,j,1} = \sum_{i' \neq i} \mathbb{I}[x_{i'} = k, z_{i',j} = 1]$. We aim to apply coordinate ascent updates to maximize the ELBO $\mathcal{L}(\theta_v)$ in (2.15)

$$\log p(\mathbf{Y}) \geq \mathcal{L}(\theta_v) := \mathbb{E}_{q_{\text{MF}}}[\log p(\mathbf{Y}, \mathbf{z}, \mathbf{x})] - \mathbb{E}_{q_{\text{MF}}}[\log q_{\text{MF}}(\mathbf{z}, \mathbf{x})] \tag{2.16}$$

with respect to the variational parameters $\theta_v = \{\boldsymbol{\lambda}, \mathbf{r}\}$. In the variational posterior $q_{\text{MF}}(\mathbf{z}, \mathbf{x})$, each component $z_{i,j}$ of the ground truth labels is parameterized as $q(z_{i,j}) = \text{Bernoulli}(\lambda_{i,j})$ and each component x_i of the mixture component indices is parameterized as $q(x_i) = \text{Discrete}(\mathbf{r}_i)$. According to Equation (17) in [9], the update equation for $\lambda_{i,j}$ in the coordinate ascent algorithm satisfies

$$\begin{aligned} \lambda_{i,j} &\propto \exp\left\{\mathbb{E}_q\Big[\log p\left(z_{i,j} = 1 \mid \mathbf{z}^{\neg(i,j)}, \mathbf{x}, \mathbf{Y}\right)\Big]\right\}, \\ 1 - \lambda_{i,j} &\propto \exp\left\{\mathbb{E}_q\Big[\log p\left(z_{i,j} = 0 \mid \mathbf{z}^{\neg(i,j)}, \mathbf{x}, \mathbf{Y}\right)\Big]\right\}, \end{aligned} \tag{2.17}$$

where the expectation is with respect to the variational distribution q over all the other latent variables $\{\mathbf{z}^{\neg(i,j)}, \mathbf{x}\}$. Similarly, the update equation for $\mathbf{r}_i$ in the coordinate ascent algorithm satisfies

$$r_{i,k} \propto \exp\left\{\mathbb{E}_q\Big[\log p\left(x_i = k \mid \mathbf{z}, \mathbf{x}^{\neg i}, \mathbf{Y}\right)\Big]\right\}, \tag{2.18}$$

where the expectation is with respect to the variational distribution q over all the other latent variables $\{\mathbf{z}, \mathbf{x}^{\neg i}\}$. Therefore, it remains to calculate the exponentiated expected log of the conditional probabilities in the update equations (2.17) and (2.18).

From the marginal distribution of $\{\mathbf{Y}, \mathbf{z}, \mathbf{x}\}$ in (2.13), we have the following conditional probabilities:

$$
\begin{aligned}
p\left(z_{i,j}=1 \mid \mathbf{z}^{\neg(i,j)}, \mathbf{x}, \mathbf{Y}\right) &\propto \prod_{k=1}^{K}\left(\alpha+n_{k,j,1}^{\neg i}\right)^{\mathbb{I}[x_i=k]} \prod_{l \in L(i)} \frac{\left(a+u_{1,1}^{l\neg(i,j)}\right)^{y_{i,j}^{l}}\left(b+u_{1,0}^{l\neg(i,j)}\right)^{1-y_{i,j}^{l}}}{a+b+u_{1 \cdot}^{l\neg(i,j)}}, \\
p\left(z_{i,j}=0 \mid \mathbf{z}^{\neg(i,j)}, \mathbf{x}, \mathbf{Y}\right) &\propto \prod_{k=1}^{K}\left(\beta+n_{k,j,0}^{\neg i}\right)^{\mathbb{I}[x_i=k]} \prod_{l \in L(i)} \frac{\left(\bar{a}+u_{0,0}^{l\neg(i,j)}\right)^{1-y_{i,j}^{l}}\left(\bar{b}+u_{0,1}^{l\neg(i,j)}\right)^{y_{i,j}^{l}}}{\bar{a}+\bar{b}+u_{0 \cdot}^{l\neg(i,j)}}, \\
p\left(x_i=k \mid \mathbf{z}, \mathbf{x}^{\neg i}, \mathbf{Y}\right) &\propto\left(\gamma+n_{k}^{\neg i}\right) \prod_{j=1}^{C} \frac{\left(\alpha+n_{k,j,1}^{\neg i}\right)^{z_{i,j}}\left(\beta+n_{k,j,0}^{\neg i}\right)^{1-z_{i,j}}}{\alpha+\beta+n_{k}^{\neg i}}.
\end{aligned}
\tag{2.19}
$$

Plugging into (2.17) and (2.18), and making use of the assumption that all components of $\{\mathbf{z}, \mathbf{x}\}$ are mutually independent under the variational distribution q_{MF}, we arrive at the following update equations:

$$
\begin{aligned}
\lambda_{i,j} \propto \exp\Bigg\{ & \sum_{k=1}^{K} r_{i,k} \mathbb{E}_q\left[\log\left(\alpha+n_{k,j,1}^{\neg i}\right)\right] \\
& + \sum_{l \in L(i)}\left(y_{i,j}^{l} \mathbb{E}_q\left[\log\left(a+u_{1,1}^{l\neg(i,j)}\right)\right]+\left(1-y_{i,j}^{l}\right) \mathbb{E}_q\left[\log\left(b+u_{1,0}^{l\neg(i,j)}\right)\right]\right. \\
& \left.-\mathbb{E}_q\left[\log\left(a+b+u_{1 \cdot}^{l\neg(i,j)}\right)\right]\right)\Bigg\},
\end{aligned}
\tag{2.20}
$$

$$
\begin{aligned}
1-\lambda_{i,j} \propto \exp\Bigg\{ & \sum_{k=1}^{K} r_{i,k} \mathbb{E}_q\left[\log\left(\beta+n_{k,j,0}^{\neg i}\right)\right] \\
& + \sum_{l \in L(i)}\left(\left(1-y_{i,j}^{l}\right) \mathbb{E}_q\left[\log\left(\bar{a}+u_{0,0}^{l\neg(i,j)}\right)\right]+y_{i,j}^{l} \mathbb{E}_q\left[\log\left(\bar{b}+u_{0,1}^{l\neg(i,j)}\right)\right]\right. \\
& \left.-\mathbb{E}_q\left[\log\left(\bar{a}+\bar{b}+u_{0 \cdot}^{l\neg(i,j)}\right)\right]\right)\Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
r_{i,k} \propto \exp\Bigg\{ & \mathbb{E}_q\left[\log\left(\gamma+n_{k}^{\neg i}\right)\right] \\
& + \sum_{j=1}^{C}\left(\lambda_{i,j} \mathbb{E}_q\left[\log\left(\alpha+n_{k,j,1}^{\neg i}\right)\right]+\left(1-\lambda_{i,j}\right) \mathbb{E}_q\left[\log\left(\beta+n_{k,j,0}^{\neg i}\right)\right]\right) \\
& - C \mathbb{E}_q\left[\log\left(\alpha+\beta+n_{k}^{\neg i}\right)\right]\Bigg\}.
\end{aligned}
\tag{2.21}
$$

Finally, following [76], we apply the Gaussian (second-order) approximation to cal-

culate each expectation term in (2.20) and (2.21). We use $\mathbb{E}_q\left[\log\left(\alpha + n_{k,j,1}^{\neg i}\right)\right]$ and $\mathbb{E}_q\left[\log\left(a + u_{1,1}^{l\neg(i,j)}\right)\right]$ as running examples; other terms can be computed in a similar way. Under the Gaussian approximation, we have

$$\mathbb{E}_q\left[\log\left(\alpha + n_{k,j,1}^{\neg i}\right)\right] \quad \approx \quad \log\left(\alpha + \mathbb{E}_q\left[n_{k,j,1}^{\neg i}\right]\right) - \frac{\mathrm{Var}_q\left[n_{k,j,1}^{\neg i}\right]}{2\left(\alpha + \mathbb{E}_q\left[n_{k,j,1}^{\neg i}\right]\right)^2}, \tag{2.22}$$

where the count $n_{k,j,1}^{\neg i} = \sum_{i' \neq i} \mathbb{I}[x_{i'} = k, z_{i',j} = 1]$ is simply a sum of independent Bernoulli variables. Hence, its mean and variance under the variational distribution q are given by

$$\mathbb{E}_q\left[n_{k,j,1}^{\neg i}\right] = \sum_{i' \neq i} r_{i',k}\lambda_{i',j}, \qquad \mathrm{Var}_q\left[n_{k,j,1}^{\neg i}\right] = \sum_{i' \neq i} r_{i',k}\lambda_{i',j}(1 - r_{i',k}\lambda_{i',j}). \tag{2.23}$$

Similarly, the count $u_{1,1}^{l\neg(i,j)} = \sum_{(i',j') \neq (i,j)} \mathbb{I}[z_{i'j'} = 1] y_{i',j'}^l$ is also a sum of independent Bernoulli variables, and its mean and variance under the variational distribution q are

$$\mathbb{E}_q\left[u_{1,1}^{l\neg(i,j)}\right] = \sum_{(i',j') \neq (i,j)} \lambda_{i',j'} y_{i',j'}^l, \qquad \mathrm{Var}_q\left[u_{1,1}^{l\neg(i,j)}\right] = \sum_{(i',j') \neq (i,j)} \lambda_{i',j'}(1 - \lambda_{i',j'}) y_{i',j}^l. \tag{2.24}$$

Plugging the Gaussian approximation (2.22) of each expectation term into the update

equations (2.20) and (2.21), we obtain the CVB inference algorithm updates.

$$\begin{aligned}
\lambda_{i,j} \propto & \prod_{k=1}^{K}(\alpha + \mathbb{E}_q[n_{k,j,1}^{\neg i}])^{r_{i,k}} \prod_{k=1}^{K} \exp\left\{ - r_{i,k} \frac{\mathrm{Var}_q[n_{k,j,1}^{\neg i}]}{2(\alpha + \mathbb{E}_q[n_{k,j,1}^{\neg i}])^2} \right\} \\
& \prod_{l \in L(i)} (a + \mathbb{E}_q[u_{1,1}^{l\neg(i,j)}])^{y_{i,j}^l} (b + \mathbb{E}_q[u_{1,0}^{l\neg(i,j)}])^{1-y_{i,j}^l} (a + b + \mathbb{E}_q[u_{1.}^{l\neg(i,j)}])^{-1} \\
& \prod_{l \in L(i)} \exp\left\{ - y_{i,j}^l \frac{\mathrm{Var}_q[u_{1,1}^{l\neg(i,j)}]}{2(a + \mathbb{E}_q[u_{1,1}^{l\neg(i,j)}])^2} - (1 - y_{i,j}^l) \frac{\mathrm{Var}_q[u_{1,0}^{l\neg(i,j)}]}{2(b + \mathbb{E}_q[u_{1,0}^{l\neg(i,j)}])^2} \right\} \\
& \prod_{L \in L(i)} \exp\left\{ \frac{\mathrm{Var}_q[u_{1.}^{l\neg(i,j)}]}{2(a + b + \mathbb{E}_q[u_{1.}^{l\neg(i,j)}])^2} \right\},
\end{aligned} \tag{2.25}$$

$$\begin{aligned}
1 - \lambda_{i,j} \propto & \prod_{k=1}^{K}(\beta + \mathbb{E}_q[n_{k,j,0}^{\neg i}])^{r_{i,k}} \prod_{k=1}^{K} \exp\left\{ - r_{i,k} \frac{\mathrm{Var}_q[n_{k,j,0}^{\neg i}]}{2(\beta + \mathbb{E}_q[n_{k,j,0}^{\neg i}])^2} \right\} \\
& \prod_{l \in L(i)} (\bar{a} + \mathbb{E}_q[u_{0,0}^{l\neg(i,j)}])^{1-y_{i,j}^l} (\bar{b} + \mathbb{E}_q[u_{0,1}^{l\neg(i,j)}])^{y_{i,j}^l} (\bar{a} + \bar{b} + \mathbb{E}_q[u_{0.}^{l\neg(i,j)}])^{-1} \\
& \prod_{l \in L(i)} \exp\left\{ - (1 - y_{i,j}^l) \frac{\mathrm{Var}_q[u_{0,0}^{l\neg(i,j)}]}{2(\bar{a} + \mathbb{E}_q[u_{0,0}^{l\neg(i,j)}])^2} - y_{i,j}^l \frac{\mathrm{Var}_q[u_{0,1}^{l\neg(i,j)}]}{2(\bar{b} + \mathbb{E}_q[u_{0,1}^{l\neg(i,j)}])^2} \right\} \\
& \prod_{l \in L(i)} \exp\left\{ \frac{\mathrm{Var}_q[u_{0.}^{l\neg(i,j)}]}{2(\bar{a} + \bar{b} + \mathbb{E}_q[u_{0.}^{l\neg(i,j)}])^2} \right\}.
\end{aligned} \tag{2.25}$$

$$\begin{aligned}
r_{i,k} \propto & \frac{\gamma + \mathbb{E}_q[n_k^{\neg i}]}{(\alpha + \beta + \mathbb{E}_q[n_k^{\neg i}])^{-C}} \exp\left\{ - \frac{\mathrm{Var}_q[n_k^{\neg i}]}{2(\gamma + \mathbb{E}_q[n_k^{\neg i}])^2} + C \frac{\mathrm{Var}_q[n_k^{\neg i}]}{2(\alpha + \beta + \mathbb{E}_q[n_k^{\neg i}])^2} \right\} \\
& \prod_{j=1}^{C} (\alpha + \mathbb{E}_q[n_{k,j,1}^{\neg i}])^{\lambda_{i,j}} (\beta + \mathbb{E}[n_{k,j,0}^{\neg i}])^{1-\lambda_{i,j}} \\
& \prod_{j=1}^{C} \exp\left\{ - \lambda_{i,j} \frac{\mathrm{Var}_q[n_{k,j,1}^{\neg i}]}{2(\alpha + \mathbb{E}_q[n_{k,j,1}^{\neg i}])^2} - (1 - \lambda_{i,j}) \frac{\mathrm{Var}_q[n_{k,j,0}^{\neg i}]}{2(\beta + \mathbb{E}_q[n_{k,j,0}^{\neg i}])^2} \right\}.
\end{aligned} \tag{2.26}$$

In practice, we run the inference algorithm with three random initializations of the variational parameters and retain the results with the highest value of ELBO. For each random initialization, we repeat the update rules (2.25) and (2.26) until the relative improvement in the ELBO falls below a small threshold (e.g., 0.0001) or the algorithm reaches the maximum number of iterations (e.g., 200 iterations).

Input : A set of crowdsourced annotations $\mathbf{Y}$ and hyperparameters $\theta_h = \{a, b, \bar{a}, \bar{b}, \alpha, \beta, \gamma\}$.

Output: Estimated ground truth labels $\mathbf{z}_i$ of each item i, sensitivity Ψ^l and specificity $\bar{\Psi}^l$ of each worker l, and distribution over all label combinations.

Initialize variational parameters $\theta_v = \{\boldsymbol{\lambda}, \mathbf{r}\}$ randomly.

repeat

  Update $\boldsymbol{\lambda}$ for the ground truth labels: for $i = 1, \ldots, N$ and $j = 1, \ldots, C$ according to (2.25).

  Update $\mathbf{r}$ for the mixture component indices: for $i = 1, \ldots, N$ and $k = 1, \ldots, K$ according to (2.26)

until *the ELBO converges*

Algorithm 1: Collapsed variational inference for BMB.

The computational complexity of the CVB inference algorithm is $\mathcal{O}(NC(K+T))$ per iteration, which scales linearly with the number of items N, the number of labels C, the number of mixture components K, and the average number of annotations per item T. Compared with mean-field variational inference, the CVB inference is not only more accurate but also computationally more efficient, because much fewer variational parameters need to be updated in the collapsed space and the resulting algorithm does not involve computationally expensive digamma functions.

Once the algorithm converges, we have solved the aforementioned multi-label crowdsourcing problem: (1) the ground truth $z_{i,j}$ can be estimated by examining the variational parameter $\lambda_{i,j}$, which indicates the posterior probability of having label j in item i; (2) the sensitivity Ψ^l and specificity $\bar{\Psi}^l$ of worker l can be estimated by comparing collected annotations $\mathbf{y}^l$ with the estimated ground truth labels[10]. As a by-product, the procedure also provides an estimation of the complete joint distribu-

[10] We have also considered estimating the posterior means of Ψ^l and $\bar{\Psi}^l$. Empirical results show that directly comparing each worker's annotations with the estimated true labels provides a better performance improvement.

tion $P = \{p(\mathbf{z}) \mid \mathbf{z} \in \{0,1\}^C\}$ over 2^C distinct label combinations. This can be easily computed by substituting estimated posterior means of $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ into the mixture of Bernoulli distributions (2.4):

$$\widehat{P} = \left\{ \hat{p}(\mathbf{z}) \mid \hat{p}(\mathbf{z}) = \sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{C} \hat{\tau}_{k,j}^{z_j} \left(1 - \hat{\tau}_{k,j}\right)^{1-z_j}, \mathbf{z} \in \{0,1\}^C \right\}, \tag{2.27}$$

where $\hat{\pi}_k = (\gamma + \sum_i r_{i,k})/(K\gamma + \sum_i \sum_{k'} r_{i,k'})$ and $\hat{\tau}_{k,j} = (\alpha + \sum_i r_{i,k}\lambda_{i,j})/(\alpha + \beta + \sum_i r_{i,k})$.

2.5 Empirical Studies

To evaluate the performance of our proposed approach, we conduct two comprehensive empirical studies using five real-world multi-label datasets. The first is a set of well-controlled experiments in which each worker's quality and annotations are simulated based on the ground truth labels of these datasets, and the second is a real-world experiment in which annotations are collected from human workers on MTurk. Both studies show that the BMB and BNMB models significantly outperform the BIL model and several baseline methods in terms of ground truth recovery and worker reliability estimation.

Dataset	Data type	# items N	# labels C	Label cardinality
`NUS-WIDE-OBJECT`	image	1,000	31	2.51
`CelebA`	image	1,000	26	6.82
`WNUT16`	text	500	9	2.14
`SemEval`	text	100	6	3.7
`Emotions`	music	593	6	1.87

Table 2.1: **Summary statistics of the multi-label datasets used in the experiments.** Label cardinality is defined as the average number of labels present in an item.

2.5.1 Datasets and Experimental Settings

Datasets. The evaluation is conducted based on five public, real-world, multi-label datasets, as summarized in Table 2.1.

- The `NUS-WIDE-OBJECT` dataset [14] is a collection of 30,000 images with real-world objects, such as flowers, rocks, and sun. There are 31 object categories in total. We randomly sample 1,000 images that are associated with more than one object category. See Figure 2.1a for two representative examples.

- The `CelebA` dataset [56] is a collection of more than 200,000 images of celebrity faces. Each image can be associated with multiple facial attributes, such as smiling and wearing eyeglasses. We randomly select 1,000 images with 26 different facial attributes.

- The `WNUT16` dataset [70] contains 2,400 tweets, to which entities of 10 different categories (including "`product`", "`facility`", and "`sports team`") have been assigned to particular words in each tweet. We drop out the "`other`" category and randomly sample 500 tweets containing more than one entity. See Figure 2.1b for two illustrative examples.

- The `SemEval` dataset [75, 74] consists of judgments of 6 emotional tags (such as "`anger`" and "`joy`") for 100 news headlines. The ground truth was constructed based on a set of expert judgments. The dataset also contains 1,000 crowdsourced judgments submitted by 38 MTurk workers, in which 10 judgments were collected for each headline. In the original dataset, each judgment takes the form of numerical values between 0 and 100 for each of the emotional tags. Following [91], we preprocess the dataset by considering a nonzero judgment as an affirmative answer to the presence of an emotion in a headline.

- The `Emotions` dataset [78] contains 593 pieces of music. Each music piece is associated with one or more tags out of 6 emotional categories, such as "`happy-pleased`" and "`sadly-lonely`". The ground truth labels were provided by a panel of music experts.

Baselines. We select the following baseline methods for comparison with our proposed models.

- `MV`: If a label candidate appears in more than half of annotations for an item, the Majority Voting (MV) algorithm predicts it to be a label present in this item. It is easy to implement and has been applied in several multi-label crowdsourcing contexts [64, 21]. However, the algorithm is known to be error-prone because it treats each worker's quality as equal and completely ignores any label dependency.

- `MLNB`: The Multi-Label Naïve Bayes model [10] improves upon the MV algorithm by taking both worker quality and label dependency into account. However, the model only captures pairwise label co-occurrence and assumes homogeneous worker quality, i.e., all workers have an equal level of sensitivity and specificity.

- `BLN`: This approach stands for Bayesian Model with Logit-Normal[11]. It is similar to BIL except that the prior over the label frequency parameter $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_C\} \in [0,1]^C$ is a multivariate logit-normal distribution instead of an independent Beta distribution as in (2.3), thus introducing correlation between τ_j and $\tau_{j'}$ (and also dependency between labels $z_{i,j}$ and $z_{i,j'}$) for $j \neq j'$. It is possible to derive a variational inference algorithm based on Laplace approximation for this non-conjugate model [86]. We defer details to Appendix A.2.

- `iBCC+Powerset`: The iBCC model [46] is a Bayesian extension of the Dawid-Skene model [17] to perform inference from crowdsourced *multi-class* annotations. A direct approach to applying iBCC in our multi-label setting would be to treat each label combination as a distinct class, with a total number of 2^C classes. We can only run this method on the `SemEval` and `Emotions` datasets when the number of label candidates C is small.

- `iBCC+Pairwise`: This approach applies iBCC to certain subsets of label combinations, where those subsets are formed by grouping label candidates into

[11] This baseline was suggested by an anonymous reviewer.

pairs [23]. It does not scale well either because of the exponential number of pairing patterns that must be considered.

- `MCMLD`: This model [91] also makes use of a mixture distribution to capture label dependency. However, it is a two-level probabilistic model in the sense that there is no generative process for worker reliability parameters. Moreover, it assumes a separate worker reliability parameter for each label candidate, and the classical EM algorithm is applied to perform point estimation of all parameters. Because the computational complexity of the E-step is $\mathcal{O}(2^C)$, we can only run this algorithm on the `SemEval` and `Emotions` datasets.

Hyperparameters. We adopt an empirical Bayes approach to choosing the hyperparameters for worker sensitivity Ψ^l and specificity $\bar{\Psi}^l$ in (2.3) and (2.7). In particular, we maximize the ELBO with respect to the hyperparameters $a, b, \bar{a}, \bar{b}$ via Minka's fixed-point updates [60]. The hyperparameters of $\boldsymbol{\tau}$ are chosen to be $\alpha = 0.1, \beta = 1$. Finally, we choose the Dirichlet hyperparameter $\gamma = 0.1$ in (2.7) for the BMB model, and place a Gamma$(0.5, 0.5)$ prior on the concentration parameter of the Dirichlet process in the BNMB model (Figure 2.2c).

2.5.2 Simulation Experiments

We first evaluate the performance of different methods in a set of simulation experiments. In these well-controlled experiments, we make use of the true labels and their dependency from the five real-world multi-label datasets in Table 2.1, and simulate multiple workers of varying quality to mimic the real-world crowdsourcing scenario. In particular, we simulate three types of workers—reliable, normal, and random—by following the previous literature [15, 65], and we then generate multiple annotations per item with each worker labeling multiple items according to the annotation process defined in (2.2). For reliable workers, the sensitivity Ψ^l is drawn from a uniform distribution Uni$(0.75, 0.85)$ and the specificity $\bar{\Psi}^l$ is sampled from Uni$(0.90, 1.00)$. The sensitivity and specificity of normal workers are drawn from Uni$(0.55, 0.75)$ and Uni$(0.80, 0.90)$, respectively. Finally, random workers have their sensitivity sampled

from Uni$(0.45, 0.55)$ and specificity sampled from Uni$(0.70, 0.80)$[12]. We name the ratio of the three worker types "heterogeneity ratio", denoted by R. All simulation results below are obtained by averaging over 50 independent runs of the experiment, with a different randomization seed in each run. For each dataset, experiment, and seed, we also conduct a statistical test to compare the performance of BMB and its closest competing method (excluding BNMB). We then report the median p-value for the 50 seeds for each experiment and dataset.

The number of workers L is set to be 980, 980, 490, 98, and 700 for the `NUS-WIDE-OBJECT`, `CelebA`, `WNUT16`, `SemEval`, and `Emotions` datasets, respectively. Unless otherwise noted, for the BMB and MCMLD models, the number of mixture components is $K = 6$ for the `NUS-WIDE-OBJECT` and `CelebA` datasets; for the `WNUT16`, `SemEval`, and `Emotions` datasets with a few label candidates, we set $K = 4$ by following Zhang and Wu [91].

Estimation of Ground Truth Labels. We evaluate all methods in terms of ground truth recovery under various values of heterogeneity ratio R (Figure 2.3) and different numbers of annotations per item T (Figure 2.4). The performance of a method is evaluated by the F1 score, i.e., the harmonic mean of precision and recall. For presentation purposes, we omit the plot of low-performing baseline methods (e.g., MV, iBCC, and MCMLD) to focus on differences between the top-performing models. Figures showing the full results of all methods are presented in Appendix A.3.

First, we fix $T = 10$ and vary R from 6:6:2 (less than 15% random workers) to 3:4:7 (50% random workers). As shown in Figure 2.3, although the performance of all methods drops as the proportion of low-reliability workers increases, BMB/BNMB significantly outperform the competing methods. The performance gap between BMB/BNMB and BIL clearly demonstrates the benefit of taking label dependency into account. We note that although BLN can also capture label dependency, it often

[12]The ranges of sensitivity and specificity are chosen based on our empirical study on MTurk (see Figure 2.10). In accordance with findings in previous research [10], each worker's specificity is assumed to be higher than his or her sensitivity. This is because when working on a multi-label annotation task, even random workers can have a large specificity value $\bar{\Psi}^l := \mathbb{P}(y^l_{i,j} = 0 \mid z_{i,j} = 0)$ as they tend to ignore most of the label candidates.

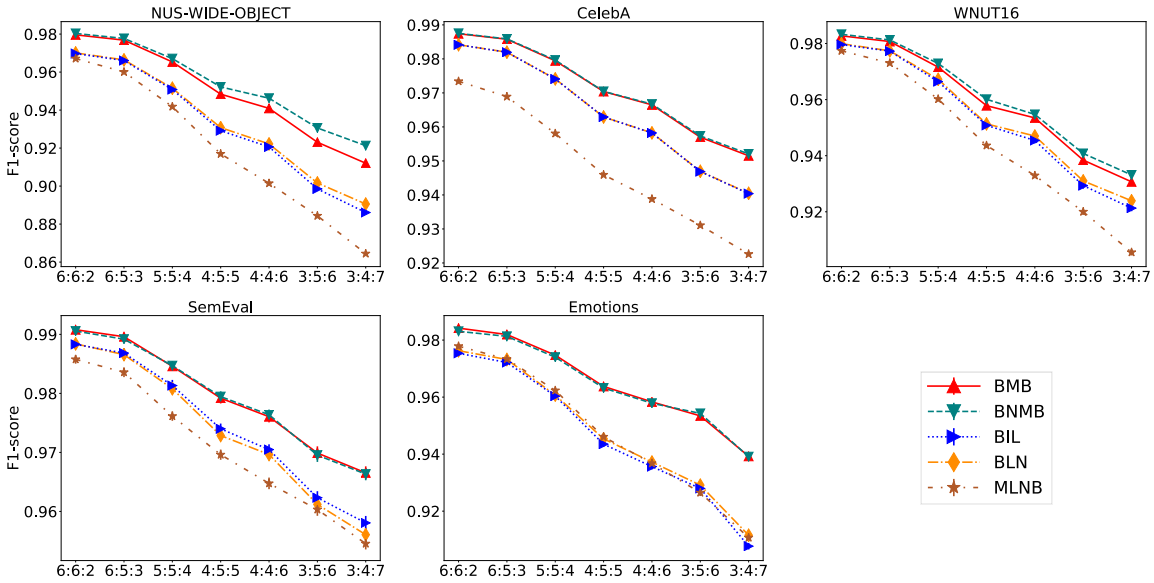


Figure 2.3: **Accuracy of the Inferred Labels in Simulation Experiments (Measured by the F1 Score) as a Function of the Heterogeneity Ratio of Worker Types** R**.** Each item receives $T = 10$ annotations.
The median p-values for the statistical test comparing BMB and its closest competing method (excluding BNMB) are < 0.001, < 0.001, 0.012, 0.049, and < 0.001, respectively. See Figure A.1 in Appendix A.3 for the complete figure showing the results of all methods.

performs comparably to BIL. This is perhaps because BLN is a non-conjugate model and the derived variational inference based on Laplace approximation can incur additional inference and estimation errors (see Appendix A.2), which offsets its larger model capacity when compared with BIL. The improvement of BMB/BNMB over MLNB is due to the superiority of using a powerful mixture model over modeling pairwise label co-occurrence and the consideration of heterogeneous worker quality. When the proportion of unreliable workers is high, the performance gap between BMB/BNMB and BIL/MLNB becomes even larger. This suggests that *joint* modeling of both heterogeneous worker reliability and label dependency is more effective in this setting, as low worker reliability can potentially be offset with sufficient knowledge of label dependency. Since BIL and MLNB only manage to model one of these two factors, their performance suffers more in this low-reliability regime. When the number of label candidates C is small, we are able to run the MCMLD model and the

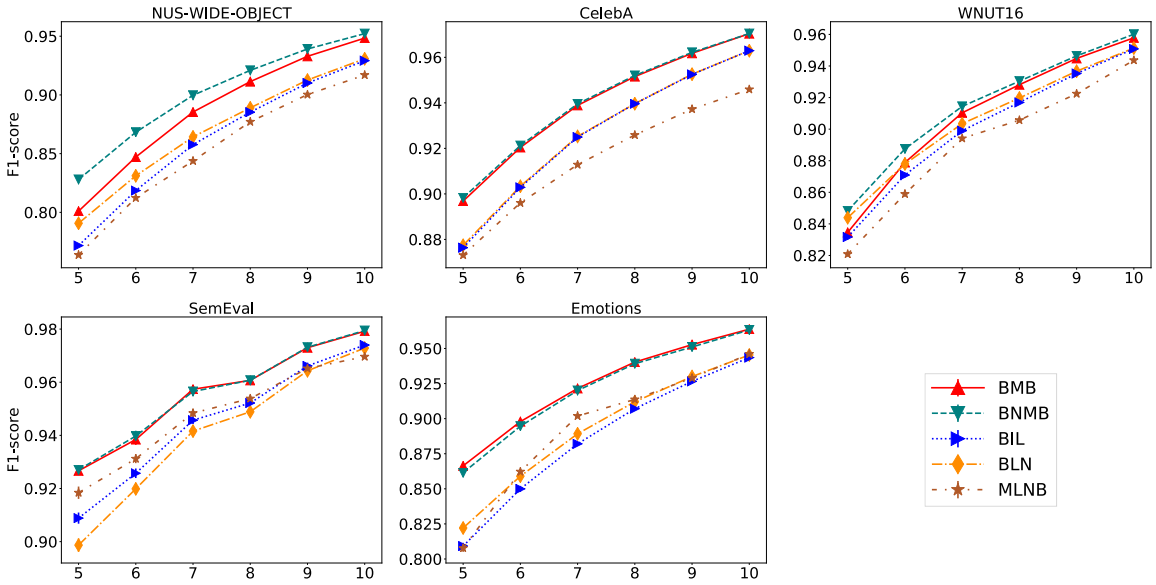


Figure 2.4: **Accuracy of the Inferred Labels in Simulation Experiments (Measured by the F1 Score) as a Function of the Number of Annotations per Item T.** The heterogeneity ratio of worker types is R =4:5:5.
The median p-values for the statistical test comparing BMB and its closest competing method (excluding BNMB) are < 0.001, < 0.001, 0.011, 0.002, and < 0.001, respectively. See Figure A.2 in Appendix A.3 for the complete figure showing the results of all methods.

two iBCC based methods on the `SemEval` and `Emotions` datasets (see Figure A.1 in Appendix A.3). Their inferior performance demonstrates the severe issue of the use of point estimation for all parameters in the MCMLD model, and suggests that it is not recommended to apply multi-class crowdsourcing approaches to the multi-label setting.

Next, we fix the heterogeneity ratio R =4:5:5 and vary the number of annotations per item from $T = 5$ to $T = 10$. We draw similar conclusions from the results shown in Figure 2.4 as BMB/BNMB consistently outperforms all other models. As expected, the performance of most methods improves with an increasing number of annotations per item T, as more label information is available for inferring the ground truth.

Estimation of Worker Quality. We also evaluate the accuracy of the estimated worker sensitivity and specificity. Following Raykar [69], we measure the estimation

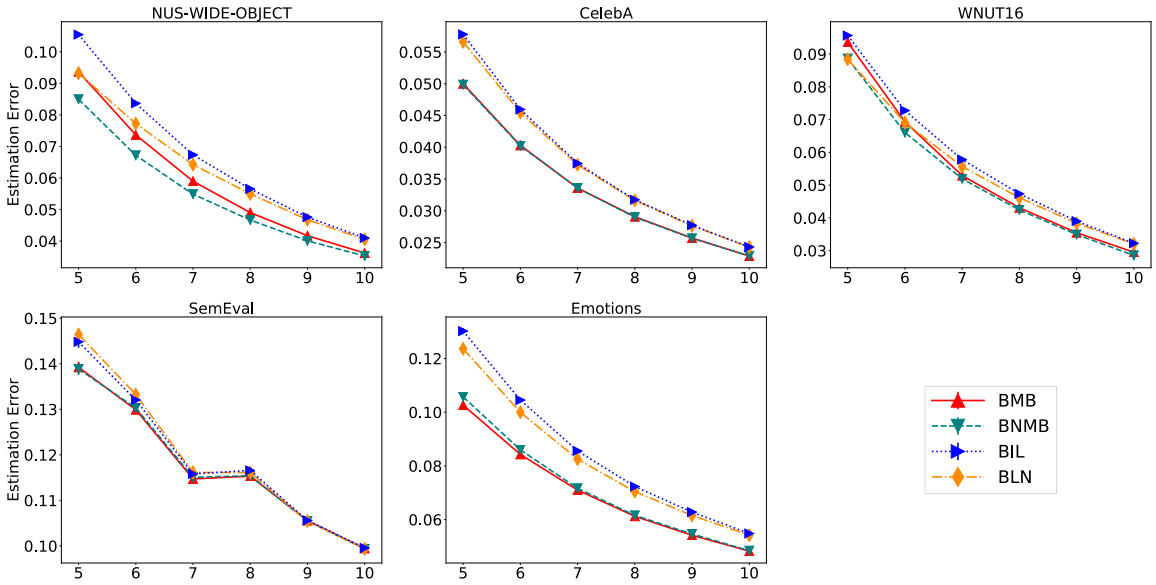
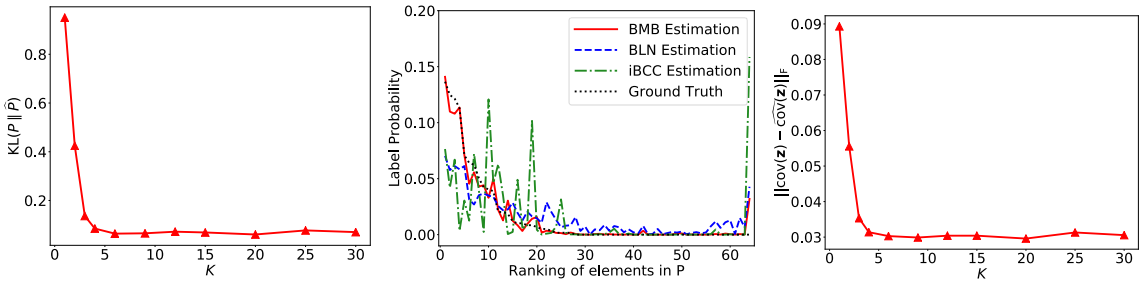


Figure 2.5: **Estimation Error of the Worker Reliability Parameters in Simulation Experiments (Measured by Mean L_2 Distance) as a Function of the Number of Annotations per Item T.** The heterogeneity ratio of worker types is R =4:5:5. The median p-values for the statistical test comparing BMB and its closest competing method (excluding BNMB) are < 0.001, < 0.001, 0.011, 0.150, and < 0.001, respectively.
See Figure A.4 in Appendix A.3 for the complete figure showing the results of all methods.

error by the mean L_2 distance between the true worker sensitivity/specificity pairs $\{(\Psi^l, \bar{\Psi}^l) : l = 1, \ldots, L\}$ and their estimated values. MV is not included in the comparison since worker reliability is not being modeled. The two iBCC based methods are not included either because each worker's quality is parameterized and represented by a worker-specific confusion matrix.

Figure 2.5 shows the performance of different methods in the setting of R =4:5:5; the results for other values of R are qualitatively similar. For presentation purposes, we defer the plot of the two low-performing baseline methods MLNB and MCMLD to Appendix A.3. The estimation error of MLNB is much higher than the corresponding error of all other methods that allow for heterogeneous worker reliability (at least 66.36% higher than the corresponding BMB error), as MLNB assumes that all workers have the same sensitivity and specificity. BMB/BNMB consistently shows superior

performance because more effective modeling of label dependency helps to improve the accuracy of the inferred labels (as shown in Figure 2.3 and Figure 2.4), which in turn can provide more accurate label information to better estimate each worker's quality. We also note that all results except for the much smaller `SemEval` dataset ($N = 100$, $L = 98$), the advantage of BMB over its closest competitor method (excluding BNMB) is statistically significant at the 0.05 level.



(a) $\mathrm{KL}(P||\widehat{P})$ (b) P versus $\widehat{P}$ (c) $\|\mathrm{cov}(\mathbf{z}) - \widehat{\mathrm{cov}}(\mathbf{z})\|_{\mathrm{F}}$

Figure 2.6: **Results of BMB for Estimating the Label Distribution and Dependency on the `Emotions` Dataset.** Each item receives $T = 10$ annotations, and the heterogeneity ratio of worker types is R =4:5:5. (a) $\mathrm{KL}(P \parallel \widehat{P})$ for a range of values of K. (b) The element-wise difference between P and $\widehat{P}$ estimated by BMB (with $K = 4$), BLN, and iBCC + Powerset. For clarity of presentation, the elements of $\widehat{P}$ have been rearranged according to their decreasing orders in P. (c) The Frobenius norm between the empirical and estimated covariance matrix of the ground truth labels for a range of values of K.

Estimation of Label Distribution and Dependency. In addition to the ground truth labels and worker reliability, our proposed BMB model can also provide an estimation of label distribution and dependency from noisy crowdsourced annotations. Given that there is an exponential number of label combinations, we will only use the `Emotions` dataset with $C = 6$ to demonstrate how well the BMB estimator $\widehat{P}$ in (2.27) can approximate the empirical distribution $P = \{p(\mathbf{z}) \mid \mathbf{z} \in \{0,1\}^C\}$ over all 2^C possible label combinations. Figure 2.6a reports the KL divergence between P and $\widehat{P}$ for a range of values of K. It is clear that P can be well estimated by $\widehat{P}$ with a moderate value of K (such as 4). Figure 2.6b shows the element-wise difference between P and $\widehat{P}$ estimated with $K = 4$ mixture components. For comparison, we

also include the label distribution estimated by BLN and iBCC+Powerset[13]. We see that BMB is much better at estimating the empirical distribution than BLN and iBCC+Powerset. Finally, Figure 2.6c reports the Frobenius norm between the empirical covariance matrix of the ground truth labels and the estimated covariance matrix based on Equation (2.5) for a range of values of K.

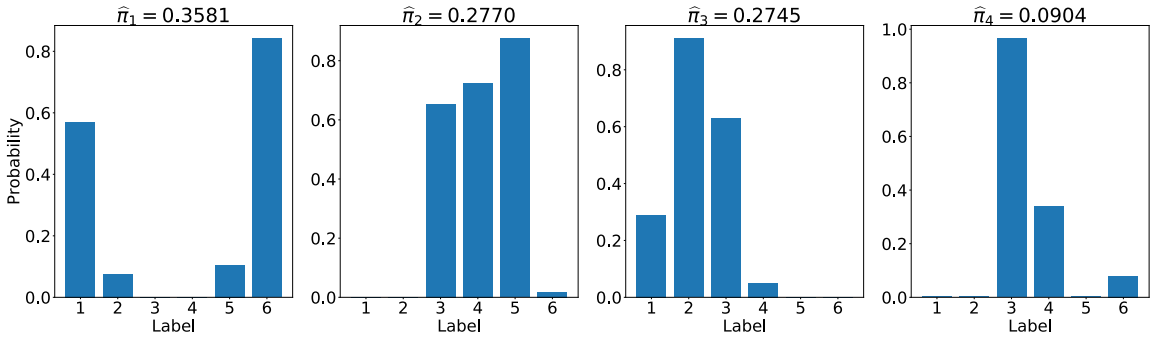


Figure 2.7: **The Mixture Components Estimated by the BMB Model with $K = 4$ from the `Emotions` Dataset.** The 6 label candidates in x-axis, from left to right, are "`amazed-surprised`", "`happy-pleased`", "`relaxing-calm`", "`quiet-still`", "`sad-lonely`", and "`angry-aggressive`", respectively. $\hat{\pi}_k$ is the estimated mixing coefficient of the kth component, and each bar represents the estimated probability of the corresponding label candidate within that mixture component, i.e., $\tau_{k,j} = \mathbb{P}(z_{i,j} = 1 \mid x_i = k)$.

Illustration of Estimated Mixture Components. Figure 2.7 shows the mixture components estimated by the BMB model with $K = 4$ on the `Emotions` dataset, under the same setup as in Figure 2.6. There are 6 label candidates: "`amazed-surprised`", "`happy-pleased`", "`relaxing-calm`", "`quiet-still`", "`sad-lonely`", and "`angry-aggressive`". Each of the top 3 most influential components characterizes a certain type of label co-occurrence: (1) the first component captures co-occurrence of two strong emotional labels "`amazed-surprised`" and "`angry-aggressive`"; (2) the second component indicates that quiet music tends to be relaxing, calm, and lonely; (3) the third component implies that happy and pleased music is likely to be relaxing and calm.

[13]MLNB is not included because we are unable to obtain a joint distribution over all 2^C label combinations from the estimated pairwise label co-occurrence.

Impact of Number of Mixture Components K. Figure 2.8 reports the accuracy of the inferred labels by BMB with a range of values for K and by its Bayesian nonparametric extension. As shown in the figure, the nonparametric approach—which integrates over the number of mixture components—performs comparably to or slightly better than the best-fitting BMB model estimated with a moderate value of K (such as 4 or 6). We note that the small performance gap between the best-fitting BMB and BNMB on the `NUS-WIDE-OBJECT` and `WNUT16` datasets can also be observed at the corresponding experimental setting in Figure 2.3 (at R =4:5:5) and in Figure 2.4 (at $T = 10$).

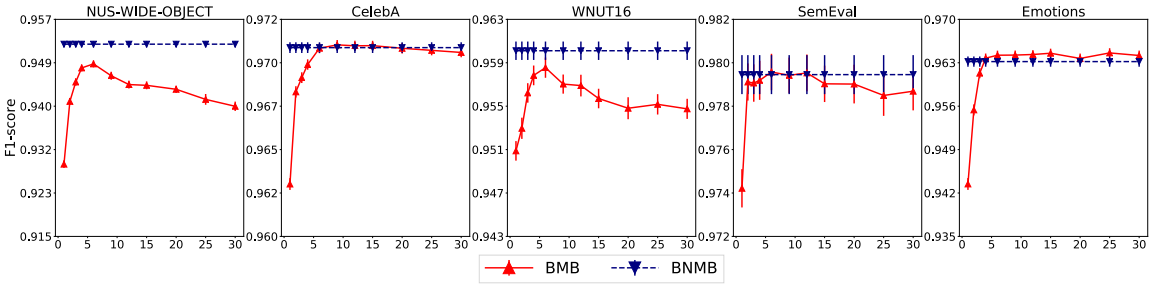


Figure 2.8: **Comparison of BMB (with Varying K) and its Bayesian Nonparametric Extension BNMB on Ground Truth Recovery in Simulation Experiments, Measured by the F1 Score.** Each item receives $T = 10$ annotations, and the heterogeneity ratio of worker types is R =4:5:5.

2.5.3 Real-World Experiments on MTurk

As pointed out in a recent crowdsourcing study [87], the drawback of simulation experiments is that the workers and their annotations are synthetically generated, which may or may not be realistic in a real-world crowdsourcing application. Therefore, to further validate our proposed methods, we conduct real-world experiments by recruiting human workers on the MTurk marketplace to provide annotations for the `NUS-WIDE-OBJECT`, `CelebA`, and `WNUT16` datasets[14]. For the `SemEval` dataset, we note that crowdsourced annotations collected from MTurk are already publicly available, as described in Section 2.5.1.

[14]The audio files in the `Emotions` dataset have already been preprocessed, so there are no raw audio/music data available to conduct a crowdsourcing experiment on MTurk.

Dataset	# items per HIT	Max # HITs per worker	Avg # items per worker	L
NUS-WIDE-OBJECT	10	6	22.32	448
CelebA	10	6	17.89	559
WNUT16	5	12	18.52	270
SemEval	20	5	26.32	38

Table 2.2: **Summary of the MTurk experiments.**

Following the common practice in the literature [35], each Human Intelligence Task (HIT) is designed to request a worker to annotate a certain number of different items. See Figure 2.1 for a screenshot of our MTurk experiments on the NUS-WIDE-OBJECT and WNUT16 datasets. We require each item to be labeled by 10 workers and each worker to complete no more than a certain maximum number of HITs. Furthermore, each worker has to complete at least one HIT to be included in the study, thus ensuring a minimum number of annotated items per worker. The summary statistics of the MTurk experiments is described in Table 2.2.

Similar to the simulation setting, we examine how the performance of each method varies with the number of annotations per item. From the 10 total annotations collected from MTurk for each item, we randomly sample $5 \leq T \leq 10$ of them, and then run our proposed models and any applicable baseline methods on this subset of annotations to infer the ground truth labels and worker reliability. For each value of T, we repeat this procedure 50 times. The results below are obtained by averaging over all trials.

Estimation of Ground Truth Labels. We first evaluate the accuracy of the inferred labels as a function of T, the number of annotations per item. The results are shown in Figure 2.9. Consistent with the findings in simulation experiments (Figure 2.4), either BMB or BNMB performs the best at recovering the ground truth. This implies that our proposed approach requires fewer annotations and lower labeling cost than other methods to achieve the same level of aggregation accuracy. We note that on the smaller SemEval dataset ($N = 100, L = 38$) with an average worker

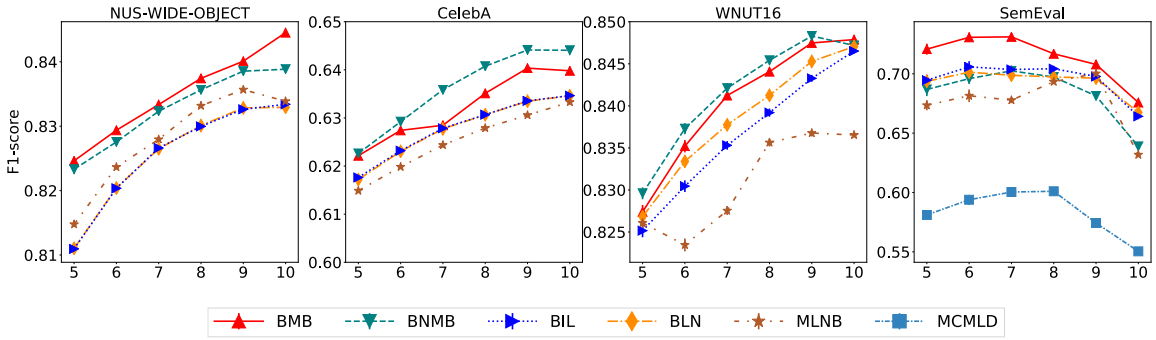


Figure 2.9: **Accuracy of the Inferred Labels in MTurk Experiments (Measured by the F1 Score) as a Function of the Number of Annotations per Item T.** The median p-values for the statistical test comparing BMB and its closest competing method (excluding BNMB) are < 0.001, < 0.001, 0.013, 0.002, respectively.
See Figure A.3 in Appendix A.3 for the complete figure showing the results of all methods.

sensitivity below 0.5 (see Figure 2.10), BNMB can become comparable or worse than a few other competing methods, suggesting that the nonparametric approach may need more samples or more reliable workers to perform better. The poor condition of worker reliability in this dataset also causes the performance of most methods to decrease at higher values of T. As the number of label candidates is small on this dataset ($C = 6$), we are able to run the two iBCC based methods (see Figure A.3 in Appendix A.3) and MCMLD. Their performance is substantially inferior to our proposed models, consistent with our simulation experiments.

Estimation of Worker Quality.

Since the ground truth labels are all known with certainty, we can measure the sensitivity and specificity of all workers by comparing their submitted annotations with the ground truth (see Figure 2.10). We treat the measured sensitivity and specificity as the "true quality" of these workers. It is important to emphasize that "in practice, workers' true quality values are always unknown" [87], because the ground truth labels themselves are unavailable and need to be inferred in a crowdsourcing application. Consistent with a previous study [10], we find that worker specificity is much higher than worker sensitivity. This is because in typical multi-label annotation

tasks, workers are more likely to miss a label that is actually present in an item than to select a label that is not present. The narrow range and very high values of worker specificity on the NUS-WIDE-OBJECT dataset are due to its low label density[15], so there is a higher fraction of labels not present in the ground truth (i.e., $z_{i,j} = 0$). Figure 2.11 shows the performance of different methods in estimating worker quality. Either BMB or BNMB achieves the lowest estimation error, and the advantage of BMB over its closest competitor method (excluding BNMB) is statistically significant at the 0.01 level. MLNB substantially underperforms relative to other methods as it assumes

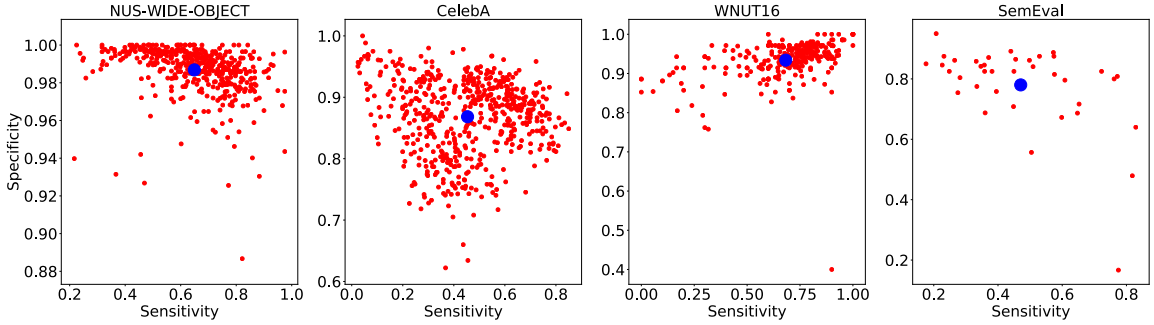


Figure 2.10: **True Sensitivity and Specificity of MTurk Workers for Each Dataset.** The mean sensitivity and specificity of all workers is marked with a larger blue circle.

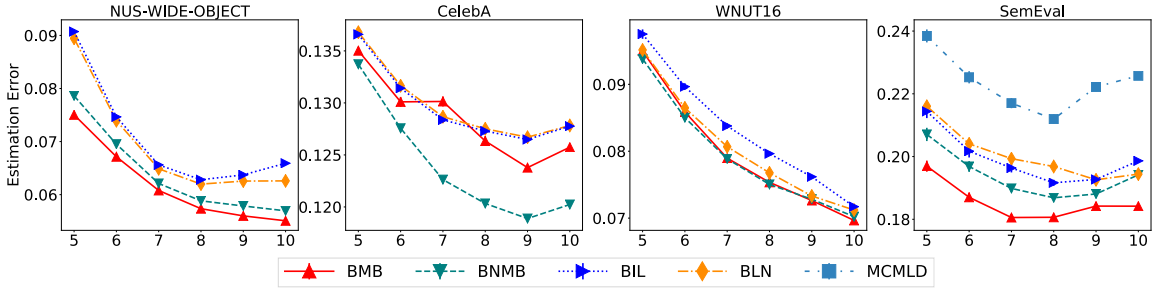


Figure 2.11: **Estimation Error of the Worker Reliability Parameters in MTurk Experiments (Measured by Mean L_2 Distance) as a Function of the Number of Annotations per Item T.** The median p-values for the statistical test comparing BMB and its closest competing method (excluding BNMB) are < 0.001, < 0.001, 0.009, and 0.001, respectively. See Figure A.5 in Appendix A.3 for the complete figure showing the results of all methods.

[15] Label density is defined as label cardinality divided by the number of label candidates C (see Table 2.1).

homogeneous worker quality (see Figure A.5 in Appendix A.3). On the `SemEval` dataset, our proposed models significantly outperform MCMLD, showing that the use of a separate worker reliability parameter for each label candidate can easily suffer from the sparsity of annotations in real-world crowdsourcing applications.

2.6 Conclusions

Given the popularity of leveraging microtask crowdsourcing platforms to collect large-scale labeled data, learning from a crowd of workers with heterogeneous labeling quality is becoming an increasingly important problem. In this work, we contribute to the literature by proposing an efficient statistical learning approach that can jointly infer the ground truth labels and worker quality from crowdsourced annotations. Our research fits into the broad theme of integrating human cognitive power and AI techniques to optimize the design of crowdsourcing systems. We focus on the general multi-label setting, where each item to be annotated can be associated with multiple labels *simultaneously*. Our endeavor is mainly motivated by the observation that correlations among different labels is very likely to appear in practice (e.g., document categories, image tags, and symptoms in medical records), and it is highly beneficial to exploit such label dependency (albeit unknown in the data collection phase) to improve the performance of a crowdsourcing system. The improvement is especially valuable in the early stages of a multi-label crowdsourcing project, when the number of annotations is too low to accurately estimate each of the true labels separately.

To solve the multi-label crowd labeling problem, we develop a novel Bayesian hierarchical framework for the underlying annotation process of crowd workers by incorporating both heterogeneous worker quality and label dependency. At the core of our method is a flexible and powerful mixture of Bernoulli distributions that is capable of capturing complex label dependency in an effective data-driven manner. We also present a nonparametric Bayesian extension of the model based on Dirichlet process mixtures to infer the number of mixture components in a principled way. An efficient collapsed variational Bayesian (CVB) inference algorithm is proposed to jointly estimate ground truth labels, worker reliability, and label dependency. Using

extensive simulation studies and real-world experiments on MTurk, we demonstrate that our method provides a significant improvement in the accuracy of ground truth recovery and worker quality estimation, over state-of-the-art alternative approaches.

2.6.1 Future Directions

We provide some future directions for further investigation. First, we note that our model takes place in the static setting where all crowdsourced annotations have been collected prior to the inference step. We may also wish to consider a scenario where the data collection phase and the inference phase are interleaved, so that the current estimation of worker quality can be leveraged to assign tasks in the next phase. This adaptive approach has been shown to be effective for the single-label crowdsourcing problem [33, 37, 44, 87]. Extending our algorithm to this dynamic setting may further improve the performance. Second, our statistical model assumes that the set of label candidates is given. We plan to combine our approach with an existing label elucidation method [10], which can brainstorm a set of label candidates efficiently using only a small number of items. Third, the probabilistic nature of our approach allows us to easily extend the model by considering other factors of the problem. Better performance can be expected by modeling task difficulty and incorporating feature information of items [89, 88].

We believe that our work makes an important contribution to the literature on effectively combining human and computer intelligence to solve important practical problems. Our proposed algorithm has demonstrated state-of-the-art results on annotation aggregation and worker quality estimation, and we have identified great potential utility of our approach both for crowdsourcing platforms and task requesters. We are hopeful that future approaches in other crowdsourcing problems may be able to build upon our framework and ideas.

CHAPTER 3

A PSEUDOLIKELIHOOD METHOD FOR INFERENCE OF PASSIVE SCALAR TURBULENCE

3.1 Introduction

Turbulence is one of the most difficult unsolved problems of the classical 19th century science, which is also of a paramount importance for many multi-physics challenges in ocean, atmosphere and climate modeling, astrophysics, and applications in aerospace, mechanical and bio- engineering [58, 77, 62, 27].

Difficulty in resolving turbulence originates from its highly non-equilibrium, multi-scale nature. Developed turbulence is typically characterized by a dimensionless parameter associated with the ratio of large, energy containing scale to the small scale where the energy dissipates. The ratio, also related to Reynolds number in fluid turbulence and to the Schmidt number in the passive scalar turbulence associated with the concentration of a pollutant, is large in the regime of developed turbulence. When the turbulence is excited at the largest scales, kinetic energy of the flow or positive definite functions of the pollutant concentration can only dissipate at the smallest (viscous or diffusive) scales. Transfer from largest scales to smallest scales is realized via a cascade, therefore involving the entire range of scales from the largest to the smallest, making turbulence highly non-equilibrium in contrast with an equilibrium mechanics where energy is injected and dissipated at comparable scales. This multi-scale phenomena also result in an extremely non-Gaussian, intermittent statistics at the smallest scales, and then significant sensitivity of relevant statistics at the largest scales to extreme fluctuations at the smallest scales linked to the so-called back-scattering phenomena.

All in all, it leads to our inability to predict statistics of turbulent fluctuations at the largest scales, which are often the prime focus of inquiry,through purely theoretical

analysis. To compensate for the theory handicap we relay largely on laboratory and numerical experiments. Well-controlled experiment and high-fidelity Direct Numerical Simulations (DNS) [1] ought to resolve all scales to provide a credible prediction. In many cases of practical interest, e.g. in astrophysics and climate science, setting up experiment correspondent to large Reynolds or Schmidt number is not possible. With regards to DNS one needs to simulate fluid-mechanics equations over spatial grid with a unit cell as small as the smallest scale of turbulence also extending this grid to the domain comparable to the energy-containing scale of the turbulence. Temporal resolution should also be chosen to be sufficiently small – resolving turnover time associated with the smallest spatial scale. In the result our computational abilities deteriorate dramatically with increase of the Reynolds or Schmidt number, making DNS unreliable to cover this regime of developed turbulence as well.

This curse of dimensionality and lack of reliable approaches to describing the large-scale structure of turbulence have resulted in emergence of heuristics, also called Reduced Order Models (ROMs) [58], such as Large Eddy Simulations (LES) [2] and Reynolds-Averaged-Navier-Stocks (RANS) [59], so that practitioners can quickly study a system's dominant effects using minimal computational resources. Putting aside lack of theoretical backing, main difficulty with traditional ROMs, consists in the fact that to represent large scale flows, and their statistics, quantitatively one needs to calibrate ROM for each new setting (e.g. new geometry of the large scale injection) anew. The calibration is costly and any new approaches and hints for improvements are thought after.

In this manuscript we consider one such approach coming from the field of computer science. The approach which is, to the best of our knowledge, was not considered for this problem before. Specifically, we show how recently developed method of *Graphical Model Learning* [13, 5, 68, 12, 11, 84, 57] can help to boost quantitatively the speed of reconstruction of the large scale statistics in turbulence.

The main idea of the method is in combining some Physics-of-Turbulence-Informed (PTI) assumptions about large scale statistics of turbulence with training data over dense spatial grid originating from either high-fidelity simulations or experiments.

The PTI assumptions will be succinctly stated in terms of a parameterized statistical model, called Graphical Model (GM), for the probability distribution function of the scalar field (chosen to be the main object of inquiry in this manuscript) over a spatially coarse grid. Parameters of the GM will be learned at the training stage which requires solving a data driven GM with the PTI assumptions embedded. We will adapt the GM learning methodology suggested in [84, 57] to the special case of multivariate continuously valued non-Gaussian GM. We will then validate the approach on the high-fidelity data samples from [16] for a multi-scale statistically stationary homogeneous and isotropic distribution of a scalar field. We choose to work with a highly symmetric turbulent flow and limited ourselves to the simpler (than general) case of scalar turbulence because testing new ideas first on the simplest turbulence case is the right approach, which allows us in the future to proceed to practical ROM challenges of inhomogeneous, anisotropic and non-stationary turbulence.

Some important technical highlights of the method are:

- We focus on the object stated as the probability of observing scalar at a point of the coarse grid, conditioned to values of the scalar at the neighboring grid points. Choice of the conditional probability is advantageous as it allows efficient computation of the correlation functions.

- As typical for the off-line applications, efficiency of the method is a concern for the post-training, inference stage of the method evaluation.

- We illustrate value of the PTI assumptions through direct comparison of the method's prediction for correlation functions resolving scalar fluctuations at the large (coarse) scale against naive empirical estimations of the correlation functions.

Material in the rest of the paper is organized as follows: we provide a description of the data in Section 3.2. In Section 3.3, we describe our two competing models for the passive scalar distribution: one based on a Gaussian distribution (Section 3.3.1) and our proposed model of a higher order moments model (Section 3.3.2), which is our main technical contribution. To validate the use of our models, we first provide

results on parameter estimation (Section 3.4.1) showing our higher order moments model differs in a substantial way from our Gaussian model. We then provide results showing that our higher order moments model is able to estimate important conditional moments of unseen data in Section 3.4.2. Finally, we provide some concluding remarks and future directions in Section 3.5.

3.2 Data

Our data involves a $128 \times 128 \times 128$ homogeneous isotropic turbulent flow with periodic boundary conditions. The data includes, for each (x, y, z) node, the velocity components and passive scalars generated by the method presented in [16]. We further coarse grain these flows by averaging $r \times r \times r$ boxes, thereby creating $d \times d \times d$ boxes where $d = 128/r$. The passive scalar is normalized to have a mean of 0 and are bounded between -1 and 1.

Concerning notation, we will refer to the scalar field as ϕ, and use ϕ_i for the value of the passive scalar at a specific node i in our simulation grid with three components (x, y, z).

3.3 Models

3.3.1 Gaussian baseline model

We focus on probabilistic models that represent the $d \times d \times d$ box as a random box, where each node $i = (x, y, z)$ is represented with a random variable ϕ_i. For a particular node i, we assume that ϕ_i is conditionally independent of all non-neighboring nodes, given its six immediately adjacent nodes. To make this precise, for a node i, denote $N(i)$ as its set of six immediately adjacent nodes. For example, if $i = (1, 1, 1)$, then $N(i) = \{(d, 1, 1), (1, d, 1), (1, 1, d), (2, 1, 1), (1, 2, 1), (1, 1, 2)\}$. (Note the periodic boundary conditions). Furthermore, given the isotropy and homogeneity of the turbulence, we assume a node's distribution does not depend on its location and we also assume that any interactions between (possibly more than two) nodes only depend on the distances between the nodes.

As a first baseline model, we model the entire box of passive scalars as a multivariate Gaussian distribution. Once we account for the conditional independence assumption and isotropy and homogeneity, we arrive at the following conditional distribution for ϕ_i which is parameterized by two parameters c_1 and c_2:

$$p(\phi_i|\phi_{\setminus i}) = p(\phi_i|\phi_{i'}, i' \in N(i)) \propto \exp\left\{-c_1\phi_i^2 - c_2 \sum_{i' \in N(i)} \phi_i\phi_{i'}\right\} \tag{3.1}$$

where $\setminus i$ refers to all nodes in the $d \times d \times d$ box except for i. We contextualize this model by noting that such a distribution is the maximum entropy distribution with respect to second moment constraints. The idea of using a maximum entropy distribution has been widely considered standard in statistical physics [67]. The main motivation is that if we wish to infer a distribution given certain constraints; all else equal, we would pick the distribution satisfying those constraints that exhibits the most disorder or randomness otherwise. Any other distribution would have smaller degeneracy and would implicitly exclude some viable characteristics of a constraint-satisfying distribution.

We also note that the full joint distribution is available and able to be parametrized as a d^3 dimensional zero-mean Gaussian distribution with an inverse covariance matrix where the diagonal is equal to c_1 and the off diagonal elements are equal to c_2 if there exists an edge between the two corresponding nodes and equal to 0 otherwise. From this joint distribution, we are able to write down the conditional distribution in Equation 3.1. Thus a constrained maximum likelihood estimation procedure is possible. However, due to computational concerns, we will instead maximize the pseudolikelihood[6] with respect to c_1 and c_2 using Newton's method implmeneted in the automatic differentiation package JuMP for Julia[24]:

$$\ell(c_1, c_2) := \sum_i \log p(\phi_i|\phi_{\setminus i}) = \sum_i \log p(\phi_i|\phi_{i'}, i' \in N(i)) \tag{3.2}$$

where $c_2' = c_2/c_1$.

With regards to the maximum pseudolikelihood estimator, it fits into a broad class of parameter estimators known as M-estimators. Under mild conditions, it is

known for M-estimators that the estimator is consistent and asymptotically normal [81] with variance [31] given by

$$\mathbb{E}[\nabla^2 \ell(c_1^*, c_2^*)]^{-T} \text{var}(\nabla \ell(c_1^*, c_2^*)) \mathbb{E}[\nabla^2 \ell(c_1^*, c_2^*)]^{-1} \tag{3.3}$$

where c_1^* and c_2^* are the true values of c_1 and c_2.

3.3.2 Higher order moments model

We are motivated by the principle of using maximum entropy distributions to model the joint distribution. We also wish to move beyond Gaussian models by incorporating higher order moments, while keeping the same assumptions on conditional independence, isotropy, and homogeneity. To define the model, we first introduce various subsets of pairs and triples of neighbors. Let $N^{(2)}(i) = N(i) \times N(i) \backslash \{(i', i') \forall i' \in N(i)\}$. Assuming $i = (x, y, z)$, we define the following:

$$N_1^{(2)}(i) = \{(i', i'') \in N^{(2)}(i) \text{ s.t. } \|i' + i'' - 2i\| = 0\}$$
$$N_2^{(2)}(i) = \{(i', i'') \in N^{(2)}(i) \text{ s.t. } \|i' + i'' - 2i\| = \sqrt{2}\}$$

where $\| \cdot \|$ denotes the Euclidean norm for $(x \mod d, y \mod d, z \mod d)$.

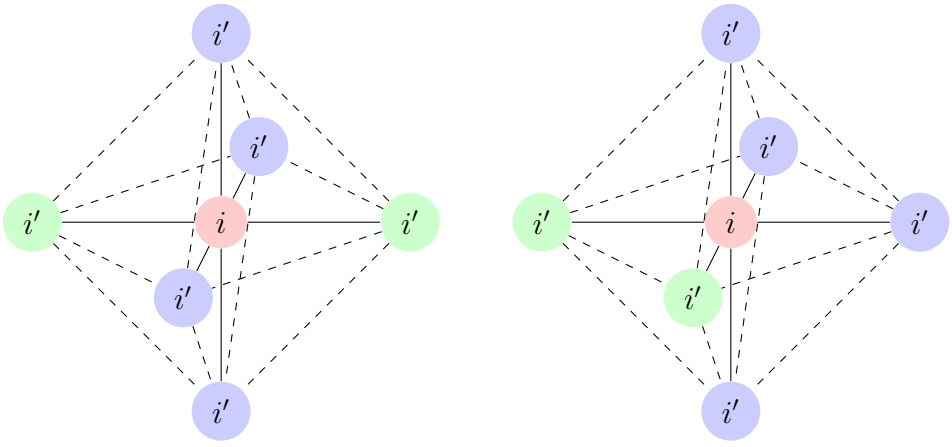


Figure 3.1: Left: Green nodes are example of a pair of nodes in $N_1^{(2)}(i)$ because they are colinear with the central node i. Right: Green nodes are an example of a pair of nodes in $N_2^{(2)}(i)$ because they are not colinear with the central node i.

In plain English, we separate pairs into pairs that are colinear with the central node ($N_1^{(2)}(i)$) and pairs that are not ($N_2^{(2)}(i)$). To more closely see this, note that for all $i' \in N(i)$,

$$i' - i \in \{(1,0,0),(0,1,0),(0,0,1),(-1,0,0),(0,-1,0),(0,0,-1)\} =: \mathcal{N}.$$

Then $N_1^{(2)}(i)$ are all pairs of neighbors i, i' such that $i' - i$ and $i'' - i$ are nonzero in the same coordinate. Because the relative positions of pairs of neighbors (even after considering translation and rotation invariance) are not the same, we anticipate the statistical interactions to be different as well even with the isotropy and homogeneity assumptions. Let $N^{(3)}(i) = N(i) \times N(i) \times N(i) \setminus \{(i', i', i')\forall i' \in N(i)\}$ We further define the following sets of triples of nodes:

$$N_1^{(3)} = \{(i', i'', i''') \in N^{(3)}(i) \text{ s.t. } \|i' + i'' + i''' - 3i\| = 1\}$$
$$N_2^{(3)} = \{(i', i'', i''') \in N^{(3)}(i) \text{ s.t. } \|i' + i'' + i''' - 3i\| = \sqrt{3}\}$$

This amounts to separating triples into triples that lie coplanar with the central node ($N_1^{(3)}(i)$) and those that are not ($N_2^{(3)}(i)$). We can then define our *higher*

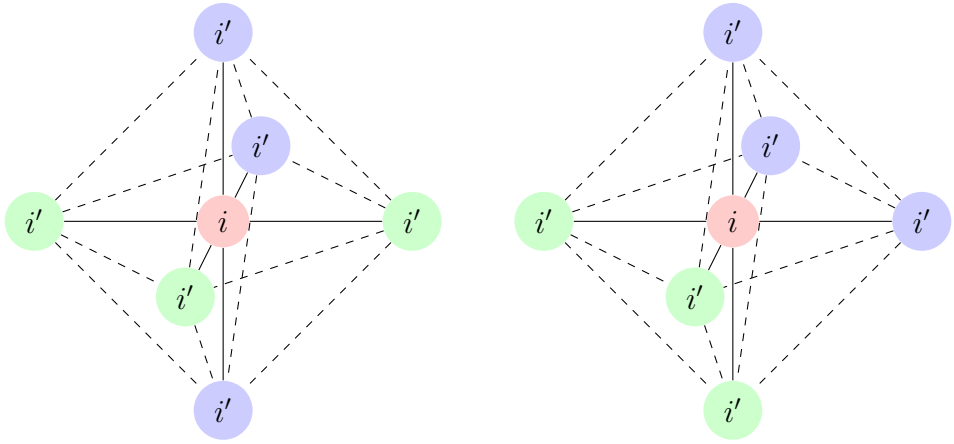


Figure 3.2: Left: Green nodes are example of a triplet of nodes in $N_1^{(3)}(i)$ because they are coplanar with the central node i. Right: Green nodes are an example of a triplet of nodes in $N_2^{(3)}(i)$ because they are not coplanar with the central node i.

order moments model parametrized by $c_1, \ldots, c_{12}$, which we shorten to HOMM:

$$p(\phi_i|\phi_{\setminus i}) \propto \exp\Big(- c_1\phi_i^2 - \sum_{i'\in N(i)} c_2\phi_i\phi_{i'} - c_3\phi_i^4 - c_4 \sum_{i'\in N(i)} \phi_i^3\phi_{i'} - c_5 \sum_{i'\in N(i)} \phi_i\phi_{i'}^3 - c_6 \sum_{i'\in N(i)} \phi_i^2\phi_{i'}^2 - c_7 \sum_{(i',i'')\in N_1^{(2)}(i)} \phi_i^2\phi_{i'}\phi_{i''} - c_8 \sum_{(i',i'')\in N_2^{(2)}(i)} \phi_i^2\phi_{i'}\phi_{i''} - c_9 \sum_{(i',i'')\in N_1^{(2)}(i)} \phi_i\phi_{i'}^2\phi_{i''} - c_{10} \sum_{(i',i'')\in N_2^{(2)}(i)} \phi_i\phi_{i'}^2\phi_{i''} - c_{11} \sum_{(i',i'',i''')\in N_1^{(3)}(i)} \phi_i\phi_{i'}\phi_{i''}\phi_{i'''} - c_{12} \sum_{(i',i'',i''')\in N_2^{(3)}(i)} \phi_i\phi_{i'}\phi_{i''}\phi_{i'''}\Big) \tag{3.4}$$

In this formulation, we can see that our higher order moments model is the maximum entropy distribution with fourth order moment constraints accounting for homogeneity and isotropy. We defer a discussion of the joint probability distribution to the Appendix.

Again, we will maximize the pseudolikelihood using Newton's method. We use Gauss-Hermite quadrature to calculate the normalization factor for the conditional probability density function.

3.4 Results

3.4.1 Parameter Estimation

We fit the higher order moments model to our turbulent passive scalar coarse grained snapshots with $r = 4$. We will first investigate whether our higher order moments model can fit our data differently than our Gaussian model. We run the pseudolikelihood method separately on our snapshots, so that we have parameter estimates for each snapshot. We report the mean of the estimated parameter values as well as the estimated standard deviations, which are obtained from the "subshape" procedure [73]. We note that for $c_3, \ldots, c_{12}$, values away from 0 show deviation from a Gaussian model.

We can see from Table 3.1 that our estimated parameter values suggest that the passive scalar distribution is quite different from a Gaussian distribution.

Parameter	Mean	Standard Error
c_1	494.455	4.834
c_2	-177.668	1.814
c_3	11517.761	287.911
c_4	-7164.710	196.949
c_5	3079.558	64.940
c_6	-4812.824	111.443
c_7	8639.086	187.622
c_8	-3155.951	171.770
c_9	-1384.305	41.963
c_{10}	3376.820	109.550
c_{11}	-3127.825	83.998
c_{12}	-2293.649	151.001

Table 3.1: Results for parameter estimation with $r = 4$.

3.4.2 Conditional Moments Estimation

In order to quantify the improvement of our model, we look at the estimation of conditional moments. Given a configuration of passive scalar values for 6 adjacent nodes $\boldsymbol{\phi}_{N(i)} = \{\phi_{i'}, i' \in N(i)\}$, we define

$$\mathcal{M}_m(\boldsymbol{\phi}_{N(i)}) := \mathbb{E}\Big[\sum_{i' \in N(i)} \big|\phi_i - \phi_{i'}\big|^m \Big| \boldsymbol{\phi}_{N(i)} \Big]. \tag{3.5}$$

We note that $\mathcal{M}_m(\boldsymbol{\phi}_{\mathcal{N}})$ includes all m-th order moments between a configuration for 6 adjacent nodes and a central node.

We set up an experiment with a training dataset and a separate test dataset. On the training dataset, we estimate the parameters of our models.

We also include an estimation procedure that is purely empirical, namely the K-nearest neighbor estimator. To be more precise, given a configuration of adjacent nodes $\boldsymbol{\phi}_{\mathcal{N}}$, we find the K nodes $i(1), \ldots, i(K)$ in the training dataset that have the closest adjacent node configuration to $\boldsymbol{\phi}_{N(i)}$ and calculate the empirical mean of (3.5), i.e.

$$\widehat{\mathcal{M}}_m^{\text{emp}}(\boldsymbol{\phi}_{\mathcal{N}}) := \frac{1}{K} \sum_{k=1}^{K} \sum_{i' \in N(i(k))} |\phi_{i(k)} - \phi_{i(k)'}|^m. \tag{3.6}$$

For the two model-based estimation procedures, we can calculate directly estimators for (3.5) $\widehat{\mathcal{M}}_m^{\text{Gauss}}(\boldsymbol{\phi}_{\mathcal{N}})$ and $\widehat{\mathcal{M}}_m^{\text{HOMM}}(\boldsymbol{\phi}_{\mathcal{N}})$ using numerical integration.

Estimation Error	$m=2$	$m=3$	$m=4$
$\sum \left(\mathcal{M}_m^{(\text{test})}\left(\phi_i^{(\text{test})}, \boldsymbol{\phi}_{N(i)}^{(\text{test})}\right) - \widehat{\mathcal{M}}_m^{\text{emp}}(\boldsymbol{\phi}_{N(i)}) \right)^2$	.328232	.023817	.002094
$\sum \left(\mathcal{M}_m^{(\text{test})}\left(\phi_i^{(\text{test})}, \boldsymbol{\phi}_{N(i)}^{(\text{test})}\right) - \widehat{\mathcal{M}}_m^{\text{Gauss}}(\boldsymbol{\phi}_{N(i)}) \right)^2$	.138448	.011209	.001141
$\sum \left(\mathcal{M}_m^{(\text{test})}\left(\phi_i^{(\text{test})}, \boldsymbol{\phi}_{N(i)}^{(\text{test})}\right) - \widehat{\mathcal{M}}_m^{\text{HOMM}}(\boldsymbol{\phi}_{N(i)}) \right)^2$	.119689	.008564	.000792

Table 3.2: Estimation error for each of the three estimation procedures. The sum is taken over sufficiently spaced nodes in the test snapshots. Here $r = 4$ and $K = 6$.

On the test data set, we take test point $\phi_i^{(\text{test})}$ and $\boldsymbol{\phi}_{N(i)}^{(\text{test})}$ compare each of the estimators to the actual test value from the test data set:

$$\mathcal{M}_m^{(\text{test})}\left(\phi_i^{(\text{test})}, \boldsymbol{\phi}_{N(i)}^{(\text{test})}\right) := \sum_{i' \in N(i)} \left| \phi_i^{(\text{test})} - \phi_{i'}^{(\text{test})} \right|^m \tag{3.7}$$

We use the square distance to compare the empirical test values and the predicted values, keeping in mind that the true conditional mean is the minimizer of this distance, i.e.

$$\mathcal{M}_m(\boldsymbol{\phi}_{N(i)}) = \underset{f(\boldsymbol{\phi}_{N(i)})}{\arg\min}\, \mathbb{E}\left[\left(\mathcal{M}_m^{(\text{test})}\left(\phi_i^{(\text{test})}, \boldsymbol{\phi}_{N(i)}^{(\text{test})}\right) - f(\boldsymbol{\phi}_{N(i)}) \right)^2 \right]. \tag{3.8}$$

In the experiment, we use the first 20% of our snapshots as the training set, and we evaluate on the remaining snapshots as the test set. To ensure a suitable degree of independence for our test set evaluation nodes, we select nodes that are sufficiently spaced out. From the results in Table 3.2, we can see that our higher moments model most accurately captures the underlying dynamics of the passive scalar by accurately predicting the conditional moments of a central node given its adjacent nodes. In particular, the accuracy on the higher order moments (third and fourth) indicate that the model is uniquely able to capture the dynamics that go beyond Gaussian. We iterate that the prediction numbers are on unseen data and are not a result of overfitting on the data.

3.5 Conclusion

In this manuscript, we consider a Graphical Model approach to characterizing the statistical distribution of the passive scalar of homogeneous isotropic turbulence. We

first present a simple Gaussian baseline model based on homogeneity and isotropy. We extend this Gaussian baseline model into a new model, our proposed higher order moments model. We are able to show that our higher order moments model outperforms the Gaussian baseline model as well as a purely empirical model-free method in estimating conditional moments of unseen data. Our experimental results provide solid backing and evidence for the efficacy of our distribution over the Gaussian distribution, as well as solid backing for the usage of graphical modelling over model-free approaches.

We see this work as an important first step in using graphical models to understand turbulent flows. Previous work [61] has focused on using neural networks and deep learning to model turbulent flows. While powerful and effective, we note that such models are not easily interpretable and may not be well suited for scientific goals. Using a graphical model approach, we are able to directly involve physics principles. Furthermore, our framework easily allows for additional modification in models. For instance, a practitioner may believe that a model more focused on the tails of the probability distribution is more suitable for their turbulence problem. It is easy to add such constraints into the framework of maximum entropy distributions, such distributions were previously considered in the quantitative finance literature [29]. Thus, we see potential to develop different models and repeat the validation procedure in Section 3.4.2 in order to select the best performing models. These models can then help us further understand how the important statistical interactions in turbulent flow operate. In this paper, we consider a very simplified idealistic model of turbulence but we are excited to see further applications to richer and more complex dynamics.

CHAPTER 4

CONCLUSION

In this work, we discuss two novel applications of graphical model methodology. Despite the differences in domain and also in the manner of graphical model (directed vs. undirected), both applications share the commonality in that graphical models were instrumental in both establishing a baseline method as well as instrumental in guiding toward natural extensions that ultimately proved to be more effective.

In our work on crowdsourcing, we first establish a baseline model BIL that recognizes the need to model and incorporate individual worker reliability but naively assumes a simplistic structure for the distribution of labels in a multi-label setting. Using graphical models, we are able to propose candidates for possible distributions for the underlying labels. Using the graphical model framework, we were able to design and test out the effectiveness of a few different distributions including a logit-normal, a mixture model, and a nonparametric mixture model. Furthermore, the graphical model formulation also facilitated efficient factorization of the joint probability distribution, which was key in deriving our approximate inference algorithms in Section 2.4.2. Future work may include more involved modeling of the worker population (see [53] for work exploiting *worker* correlations) and using effective MCMC algorithms like collapsed Gibbs sampling [54] to more accurately infer the posterior distribution.

In our work on turbulence, we aim to apply graphical model principles to analyzing the statistical distribution of turbulent flows. This leads us to the recognition of the necessity to *coarse grain* at certain granularities in order for our graphical model framework to effectively capture statistical interactions. The framework also lends itself easily to utilize the maximum pseudolikelihood parameter estimation method. Our parametrization of the distribution follows from the maximum entropy principle and recognition that non-Gaussian interactions play a key role in understanding

turbulent flow. As such we are able to show that our model based on higher order moments provides a more accurate representation of the dynamics in our turbulent flow. Thus we demonstrate that the graphical model framework and methodology can be applied to this setting of turbulent flows.

APPENDIX A

Appendix for Chapter 2

A.1 Inference Algorithm for BNMB

In this section, we describe the details of deriving a collapsed variational inference algorithm for the Bayesian Nonparametric Extension of BMB. We utilize previous work on collapsed variational inference for Dirichlet Processes Mixture Models [49].

We use the truncated stick breaking process [40] which uses the infinite stick-breaking representation truncated after T mixture components. First, we generate an independent sequence of random variables $\mathbf{v} = (v_k)_{k=1}^{T-1}$ as $v_k \mid \gamma \sim \text{Beta}(1, \gamma)$ and $v_T = 1$, and then define $\pi_k(\mathbf{v}) = v_k \prod_{m=1}^{k-1}(1 - v_m)$ for $k = 1, 2, \ldots, T$. It is known that $\sum_{k=1}^{T} \pi_k(\mathbf{v}) = 1$ with probability one, so we can interpret $\pi_k(\mathbf{v})$ as the mixing coefficient of the kth component.

For this truncated BNMB model, the full joint distribution is given by

$$
\begin{aligned}
p(\mathbf{Y}, \mathbf{z}, \mathbf{x}, \mathbf{v}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}) = & \left[\prod_{k=1}^{T} v_k^{n_k} (1 - v_k)^{\gamma - 1 + n_{>k}} \right] \\
& \cdot \left[\frac{1}{B(\alpha, \beta)^{TC}} \prod_{k=1}^{T} \prod_{j=1}^{C} \tau_{k,j}^{\alpha + n_{k,j,1} - 1} (1 - \tau_{k,j})^{\beta + n_{k,j,0} - 1} \right] \\
& \cdot \left[\frac{1}{B(a, b)^{L} B(\bar{a}, \bar{b})^{L}} \prod_{l=1}^{L} (\Psi^l)^{a + u^l_{1,1} - 1} (1 - \Psi^l)^{b + u^l_{1,0} - 1} (\bar{\Psi}^l)^{\bar{a} + u^l_{0,0} - 1} (1 - \bar{\Psi}^l)^{\bar{b} + u^l_{0,1} - 1} \right],
\end{aligned}
\tag{A.1}
$$

where $n_{>k} = \#\{i : x_i > k\}$. By marginalizing out $\{\mathbf{v}, \boldsymbol{\tau}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}|\}$, we obtain the marginal distribution over $\{\mathbf{Y}, \mathbf{z}, \mathbf{x}\}$

$$p(\mathbf{Y},\mathbf{z},\mathbf{x}) = \left[\prod_{k=1}^{T}\frac{\Gamma(1+n_k)\Gamma(\gamma+n_{>k})}{\Gamma(1+\gamma+n_{\geq k})}\right]\cdot\left[\frac{1}{B(\alpha,\beta)^{TC}}\prod_{k=1}^{T}\prod_{j=1}^{C}B(\alpha+n_{k,j,1},\beta+n_{k,j,0})\right] \cdot \left[\frac{1}{B(a,b)^{L}B(\bar{a},\bar{b})^{L}}\prod_{l=1}^{L}B(a+u^l_{1,1},b+u^l_{1,0})B(\bar{a}+u^l_{0,0},\bar{b}+u^l_{0,1})\right], \tag{A.2}$$

where $n_{\geq k} = n_k + n_{>k}$. We note that this is equivalent to Equation (9) and (10) in [49] where their $\boldsymbol{\eta}$ is given by our $\{\boldsymbol{\tau},\boldsymbol{\Psi},\bar{\boldsymbol{\Psi}}\}$, but is further marginalized out.

From here, the derivation follows similarly as in the derivation for collapsed BMB. We define the same variational family as in Equation 2.14. The $\boldsymbol{\lambda}$ updates are virtually identical to the ones for BMB except replacing K with T. For the $\mathbf{r}$ updates, first note that the conditional distribution of x_i given $\{\mathbf{z},\mathbf{x}^{\neg i},\mathbf{Y}\}$ is given by

$$p\left(x_i = k \mid \mathbf{z},\mathbf{x}^{\neg i},\mathbf{Y}\right) \propto \frac{1+n_k^{\neg i}}{1+\gamma+n_{\geq k}^{\neg i}}\prod_{k'<k}\frac{\gamma+n_{>k'}^{\neg i}}{1+\gamma+n_{\geq k'}^{\neg i}}\prod_{j=1}^{C}\frac{\left(\alpha+n_{k,j,1}^{\neg i}\right)^{z_{i,j}}\left(\beta+n_{k,j,0}^{\neg i}\right)^{1-z_{i,j}}}{\alpha+\beta+n_k^{\neg i}}. \tag{A.3}$$

Then repeating the steps for collapsed variational inference for BMB and applying the Gaussian approximation, we have the update for $\mathbf{r}$:

$$\begin{aligned} r_{i,k} \propto\ & \frac{1+\mathbb{E}_q[n_k^{\neg i}]}{1+\gamma+\mathbb{E}_q[n_{\geq k}^{\neg i}]}\exp\left\{-\frac{\mathrm{Var}_q[n_k^{\neg i}]}{2(\gamma+\mathbb{E}_q[n_k^{\neg i}])^2}+\frac{\mathrm{Var}_q[n_{\geq k}^{\neg i}]}{2(1+\gamma+\mathbb{E}_q[n_{\geq k}^{\neg i}])^2}\right\} \\ & \prod_{k'<k}\frac{\gamma+\mathbb{E}_q[n_{>k'}^{\neg i}]}{1+\gamma+\mathbb{E}[n_{\geq k'}^{\neg i}]}\exp\left\{-\frac{\mathrm{Var}_q[n_{>k'}^{\neg i}]}{2(\gamma+\mathbb{E}_q[n_{>k'}^{\neg i}])^2}+\frac{\mathrm{Var}_q[n_{\geq k'}^{\neg i}]}{2(1+\gamma+\mathbb{E}_q[n_{\geq k'}^{\neg i}])^2}\right\} \\ & \prod_{j=1}^{C}(\alpha+\mathbb{E}_q[n_{k,j,1}^{\neg i}])^{\lambda_{i,j}}(\beta+\mathbb{E}[n_{k,j,0}^{\neg i}])^{1-\lambda_{i,j}} \\ & \prod_{j=1}^{C}\exp\left\{-\lambda_{i,j}\frac{\mathrm{Var}_q[n_{k,j,1}^{\neg i}]}{2(\alpha+\mathbb{E}_q[n_{k,j,1}^{\neg i}])^2}-(1-\lambda_{i,j})\frac{\mathrm{Var}_q[n_{k,j,0}^{\neg i}]}{2(\beta+\mathbb{E}_q[n_{k,j,0}^{\neg i}])^2}\right\} \\ & \exp\left\{C\frac{\mathrm{Var}_q[n_k^{\neg i}]}{2(\alpha+\beta+\mathbb{E}_q[n_k^{\neg i}])^2}\right\}. \end{aligned} \tag{A.4}$$

A.2 Bayesian Model with Logit-Normal (BLN)

As suggested by an anonymous reviewer, we consider an alternative approach to modeling dependency within the ground truth labels $\mathbf{z}_i \in \{0,1\}^C$. The resulting Bayesian Model with Logit-Normal (BLN) is similar to the BIL model (Section 2.3.2) except that the prior over the label frequency parameter $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_C\} \in [0,1]^C$ is a multivariate logit-normal distribution instead of an independent Beta distribution as in (2.3). Specifically, we assume that the logit transformation of $\{\tau_1, \ldots, \tau_C\}$ are distributed as a multivariate normal (MVN) with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$\left\{ \log\left(\frac{\tau_1}{1-\tau_1}\right), \log\left(\frac{\tau_2}{1-\tau_2}\right), \ldots, \log\left(\frac{\tau_C}{1-\tau_C}\right) \right\} \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{A.5}$$

The corresponding joint distribution of $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_C\}$ is called *multivariate logit-normal* with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ [42, §7.1.6]. As in BIL, the BLN model assumes that the multiple true labels $\{z_{i,1}, \ldots, z_{i,C}\}$ of item i are conditionally independent given the label frequency parameter $\boldsymbol{\tau}$, i.e., $z_{i,j} \mid \tau_j \sim \text{Bernoulli}(\tau_j)$, and that the annotation process of crowd workers follows the same two-coin model in (2.2). It can be shown (by integrating out $\boldsymbol{\tau}$) that the covariance matrix of the ground truth labels satisfies $\text{cov}(\mathbf{z}_i) = \text{cov}(\boldsymbol{\tau})$, which is non-diagonal provided that $\boldsymbol{\Sigma}$ is non-diagonal. Therefore, label dependency in the BLN model is introduced and characterized by the parameter $\boldsymbol{\Sigma}$.

To facilitate the presentation of the model, we introduce the natural parameters $\{\theta_j = \log(\tau_j/(1-\tau_j)) : j = 1, \ldots, C\}$ of Bernoulli and write the generative process of BLN in the following equivalent form:

$$\begin{aligned} \boldsymbol{\theta} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \\ z_{i,j} \mid \boldsymbol{\theta} &\sim \text{Bernoulli}(g(\theta_j)), \\ y^l_{i,j} \mid \Psi^l, \bar{\Psi}^l, \mathbf{z}_i &\sim \text{Bernoulli}\big((\Psi^l)^{z_{i,j}}(1-\bar{\Psi}^l)^{(1-z_{i,j})}\big), \end{aligned} \tag{A.6}$$

where $g : \mathbb{R} \to (0,1)$ is the logistic function defined as $g(x) = e^x/(1+e^x)$.

Unlike BIL and BMB, the BLN model is a non-conjugate because the normal prior over $\boldsymbol{\theta}$ is not conjugate to the Bernoulli$(g(\theta_j))$ likelihood. As a result, coordi-

nate ascent updates to maximize the ELBO are no longer available in closed form. We therefore develop an approximation algorithm based on *Laplace variational inference* [86]. The key idea is to apply Laplace approximation to update the variational distribution of the non-conjugate variable $\boldsymbol{\theta}$ and use standard coordinate ascent updates for the variational distribution of the conjugate variables $\{\boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}, \mathbf{z}\}$.

First, we introduce a family of variational posterior

$$q(\boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}, \mathbf{z}, \boldsymbol{\theta}) = q(\boldsymbol{\Psi} \mid \mathbf{g}, \mathbf{h}) q(\bar{\boldsymbol{\Psi}} \mid \bar{\mathbf{g}}, \bar{\mathbf{h}}) q(\mathbf{z} \mid \boldsymbol{\lambda}) q(\boldsymbol{\theta}). \tag{A.7}$$

With the exception of $\boldsymbol{\theta}$, the components of other latent variables are assumed to be further factorized. In particular, the variational posteriors over worker sensitivity Ψ^l and specificity $\bar{\Psi}^l$ are parameterized as $q(\Psi^l) = \text{Beta}(g^l, h^l)$ and $q(\bar{\Psi}^l) = \text{Beta}(\bar{g}^l, \bar{h}^l)$ respectively, and each component $z_{i,j}$ of the ground truth labels is parameterized as $q(z_{i,j}) = \text{Bernoulli}(\lambda_{i,j})$.

Next, according to Equation (4) in [86] or Equation (18) in [9], the optimal coordinate update for the variational distribution of the non-conjugate variable $\boldsymbol{\theta}$ satisfies

$$q(\boldsymbol{\theta}) \quad \propto \quad \exp\left\{\mathbb{E}_{q(\mathbf{z})}\left[\log p\left(\boldsymbol{\theta}, \mathbf{z}\right)\right]\right\}. \tag{A.8}$$

From the generative process specified in (A.6), we obtain the following joint distribution over $\boldsymbol{\theta}$ and $\mathbf{z}$:

$$p(\boldsymbol{\theta}, \mathbf{z}) \quad \propto \quad \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu})\right\} \prod_{i=1}^{N} \prod_{j=1}^{C} \left[g(\theta_j)^{z_{i,j}} (1 - g(\theta_j))^{1 - z_{i,j}}\right]. \tag{A.9}$$

Therefore, up to an additive constant, the exponent in (A.8) can be expressed as

$$f(\boldsymbol{\theta}) \doteq \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{\theta}, \mathbf{z})] = -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}) + \boldsymbol{\theta}^T \left(\sum_{i=1}^{N} \boldsymbol{\lambda}_i\right) - N \sum_{j=1}^{C} \log(1 + e^{\theta_j}), \tag{A.10}$$

where $\boldsymbol{\lambda}_i \doteq \{\lambda_{i,1}, \ldots, \lambda_{i,C}\} \in [0, 1]^C$. However, because the BLN model is non-conjugate, the coordinate update $q(\boldsymbol{\theta}) \propto \exp\{f(\boldsymbol{\theta})\}$ in (A.8) cannot be normalized in closed form. Laplace variational inference proceeds by taking a second-order Taylor

approximation of $f(\boldsymbol{\theta})$ around its maximum $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$,

$$f(\boldsymbol{\theta}) \quad \approx \quad f(\hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \nabla^2 f(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}), \tag{A.11}$$

where $\nabla^2 f(\hat{\boldsymbol{\theta}})$ is the Hessian matrix evaluated at $\hat{\boldsymbol{\theta}}$. In practice, the maximum $\hat{\boldsymbol{\theta}}$ can be found using a numerical optimization method such as gradient descent. Exponentiating (A.11) leads to the following approximate Gaussian posterior update for the non-conjugate variable $\boldsymbol{\theta}$:

$$q(\boldsymbol{\theta}) \quad \approx \quad \mathcal{N}(\hat{\boldsymbol{\theta}}, -\nabla^2 f(\hat{\boldsymbol{\theta}})^{-1}). \tag{A.12}$$

The optimal coordinate update for the variational distribution of the conjugate variables $\{\boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}, \mathbf{z}\}$ can be derived in a standard manner, and we omit the details here. The full Laplace variational inference procedure for the BLN model is described in Algorithm 2. The expectations of various quantities under the variational distribution q in (A.18) can be computed as follows: $\mathbb{E}_q[\log g(\theta_j)]$ can be calculated using a second-order Taylor expansion,

$$\mathbb{E}_q[\log g(\theta_j)] \quad \approx \quad \log g(\hat{\theta}_j) + \frac{g(\hat{\theta}_j)g''(\hat{\theta}_j) - g'(\hat{\theta}_j)^2}{2g(\hat{\theta}_j)^2} \mathrm{Var}_q[\theta_j], \tag{A.13}$$

where $\mathrm{Var}_q[\theta_j]$ is the jth diagonal element of $-\nabla^2 f(\hat{\boldsymbol{\theta}})^{-1}$; $\mathbb{E}_q[\log(1 - g(\theta_j))]$ can be computed similarly; the expectations of $\log \Psi^l$ and $\log(1 - \Psi^l)$ are

$$\mathbb{E}_q[\log \Psi^l] = \psi(g^l) - \psi(g^l + h^l) \quad \text{and} \quad \mathbb{E}_q[\log(1 - \Psi^l)] = \psi(h^l) - \psi(g^l + h^l), \tag{A.14}$$

where $\psi(\cdot)$ is the digamma function.

Finally, it is also necessary to update the hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in an empirical Bayes fashion. This amounts to setting $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as the empirical mean and covariance of $\{g^{-1}(\boldsymbol{\lambda}_i)\}_{i=1}^N$ at each iteration. We note that this step is equivalent to finding the approximate MLE of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using expected sufficient statistics under the variational distribution $q(\mathbf{z} \mid \boldsymbol{\lambda})$.

Input : A set of crowdsourced annotations $\mathbf{Y}$ and hyperparameters $\theta_h = \{a, b, \bar{a}, \bar{b}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

Output: Estimated ground truth labels $\mathbf{z}_i$ of each item i, sensitivity Ψ^l and specificity $\bar{\Psi}^l$ of each worker l, and distribution over all label combinations.

Initialize variational parameters $\theta_v = \{\mathbf{g}, \mathbf{h}, \bar{\mathbf{g}}, \bar{\mathbf{h}}, \boldsymbol{\lambda}\}$ randomly.

repeat

Update $\{\mathbf{g}, \mathbf{h}, \bar{\mathbf{g}}, \bar{\mathbf{h}}\}$ for the sensitivity and specificity: for $l = 1, \ldots, L$,

$$g^l = a + \sum_{i \in N(l)} \sum_{j=1}^{C} \lambda_{i,j} y_{i,j}^l, \quad h^l = b + \sum_{i \in N(l)} \sum_{j=1}^{C} \lambda_{i,j} (1 - y_{i,j}^l). \tag{A.15}$$

$$\bar{g}^l = \bar{a} + \sum_{i \in N(l)} \sum_{j=1}^{C} (1 - \lambda_{i,j})(1 - y_{i,j}^l), \quad \bar{h}^l = \bar{b} + \sum_{i \in N(l)} \sum_{j=1}^{C} (1 - \lambda_{i,j}) y_{i,j}^l. \tag{A.16}$$

Calculate $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$ from (A.10) and update $q(\boldsymbol{\theta})$ as

$$q(\boldsymbol{\theta}) \approx \mathcal{N}(\hat{\boldsymbol{\theta}}, (-\nabla^2 f(\hat{\boldsymbol{\theta}}))^{-1})) \tag{A.17}$$

Update $\boldsymbol{\lambda}$ for the ground truth labels: for $i = 1, \ldots, N$ and $j = 1, \ldots, C$,

$$\begin{aligned} \lambda_{i,j} &\propto \exp\left\{ \mathbb{E}_q[\log g(\theta_j)] + \sum_{l \in L(i)} \left(y_{i,j}^l \mathbb{E}_q[\log \Psi^l] + (1 - y_{i,j}^l) \mathbb{E}_q[\log(1 - \Psi^l)] \right) \right\}, \\ 1 - \lambda_{i,j} &\propto \exp\left\{ \mathbb{E}_q[\log(1 - g(\theta_j))] \right. \\ &\qquad \left. + \sum_{l \in L(i)} \left((1 - y_{i,j}^l) \mathbb{E}_q[\log \bar{\Psi}^l] + y_{i,j}^l \mathbb{E}_q[\log(1 - \bar{\Psi}^l)] \right) \right\}. \end{aligned} \tag{A.18}$$

until *the ELBO converges*

Algorithm 2: Laplace variational inference for BLN

A.3 Complete Figures

In this section, we provide the full results of all methods and discuss the low-performing baseline methods that are omitted in the main manuscript.

A.3.1 Estimation of Ground Truth Labels

Figures A.1 and A.2 show the accuracy of the inferred labels in simulation experiments (Section 2.5.2), and Figure A.3 shows the accuracy of the inferred labels in MTurk Experiments (Section 2.5.3). We note that the Majority Voting (MV) algorithm, as a simple baseline, significantly underperforms relative to other methods that model worker reliability. When the number of label candidates C is small, we are able to run the MCMLD model on the `SemEval` and `Emotions` datasets. The results show that it performs much worse than our proposed Bayesian hierarchical models, demonstrating the severe issue of performing point estimation for all parameters in the MCMLD model. Furthermore, the use of a separate worker reliability parameters for each label candidate suffers from the sparsity of annotations and ultimately results in poor performance. We are also able to run the two iBCC based methods on these two datasets. Their inferior performance suggests that it is not recommended to directly apply single-label (multi-class) crowdsourcing approaches to the multi-label setting since a large number of parameters in worker confusion matrices need to be inferred from a smaller number of annotations.

A.3.2 Estimation of Worker Reliability

Figures A.4 and A.5 show the estimation error of the worker reliability parameters in simulation experiments (Section 2.5.2) and MTurk Experiments (Section 2.5.3), respectively. MV is not included in the comparison because it does not model worker reliability. The two iBCC based methods model worker reliability through confusion matrices instead of sensitivity and specificity, so they are likewise not included. We note that MLNB performs much worse than all other methods that allow for heterogeneous worker reliability, as MLNB assumes that all workers have the same worker

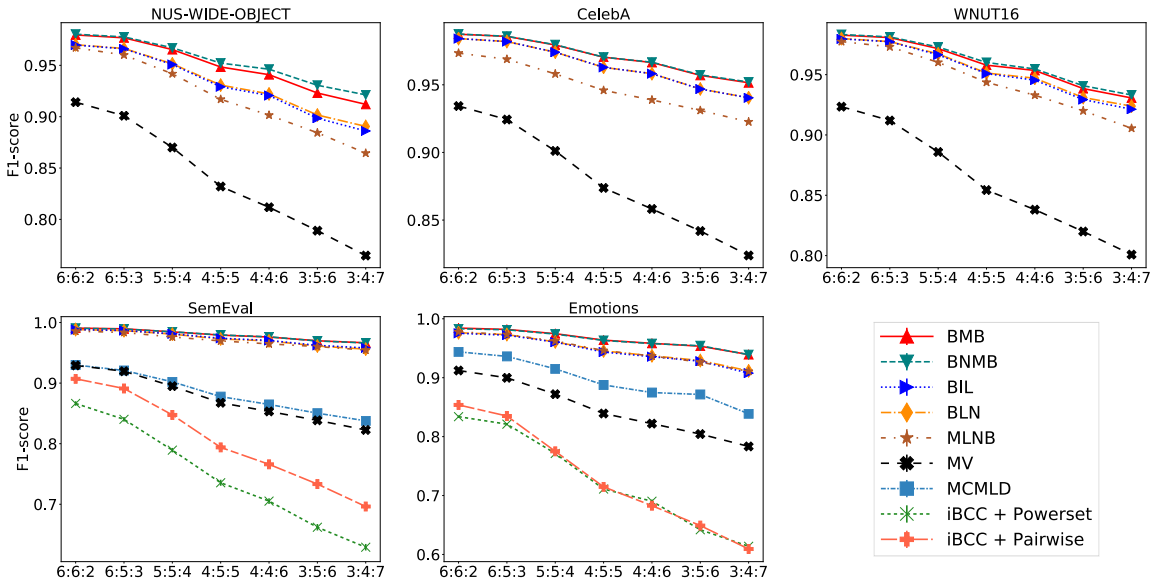


Figure A.1: **Accuracy of the Inferred Labels in Simulation Experiments (Measured by the F1 Score) as a Function of the Heterogeneity Ratio of Worker Types R.** Each item receives $T = 10$ annotations. The performance of MV, MCMLD, and iBCC is substantially lower than the competing methods.

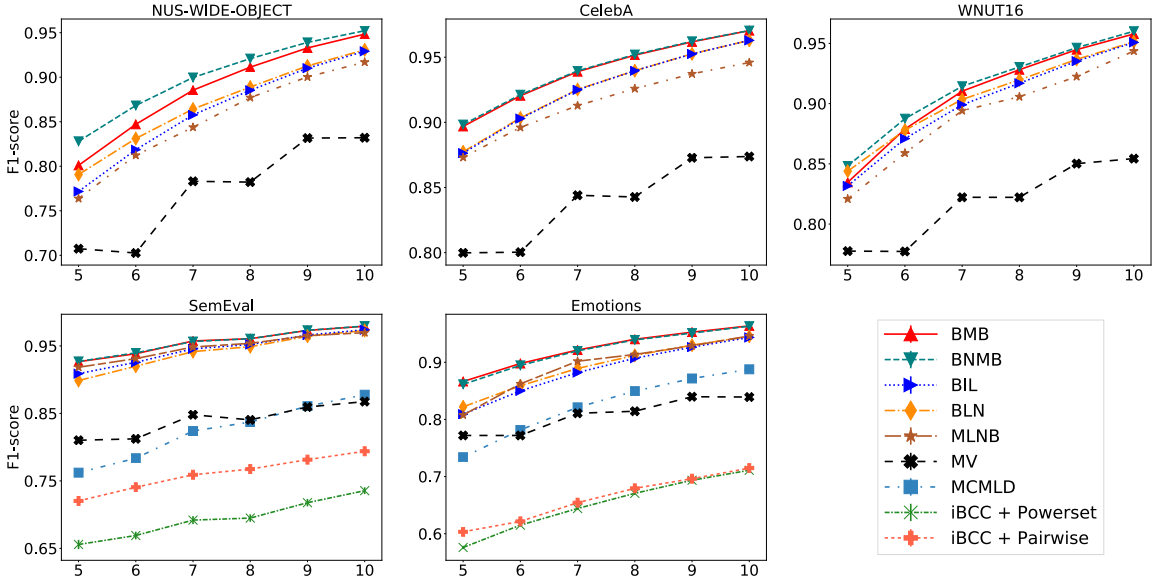


Figure A.2: **Accuracy of the Inferred Labels in Simulation Experiments (Measured by the F1 Score) as a Function of the Number of Annotations per Item T.** The heterogeneity ratio of worker types is R =4:5:5. The performance of MV, MCMLD, and iBCC is substantially lower than the competing methods.

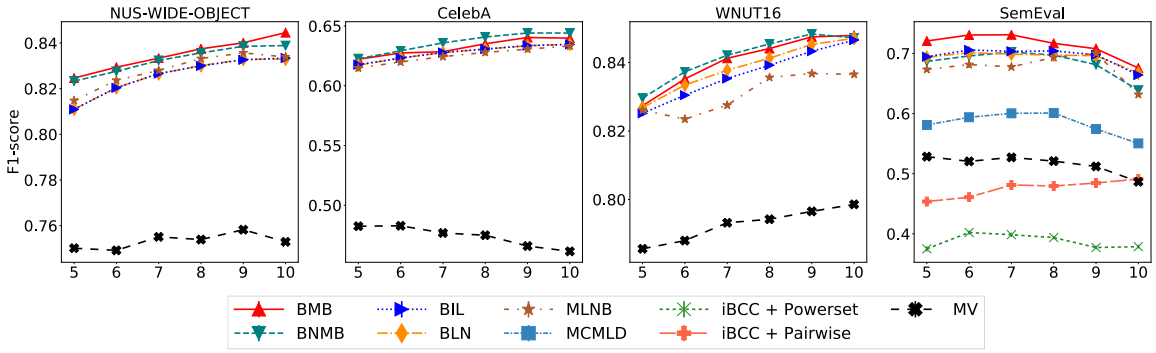


Figure A.3: **Accuracy of the Inferred Labels in MTurk Experiments (Measured by the F1 Score) as a Function of the Number of Annotations per Item T.** The performance of MV, MCMLD, and iBCC is substantially lower than the competing methods.

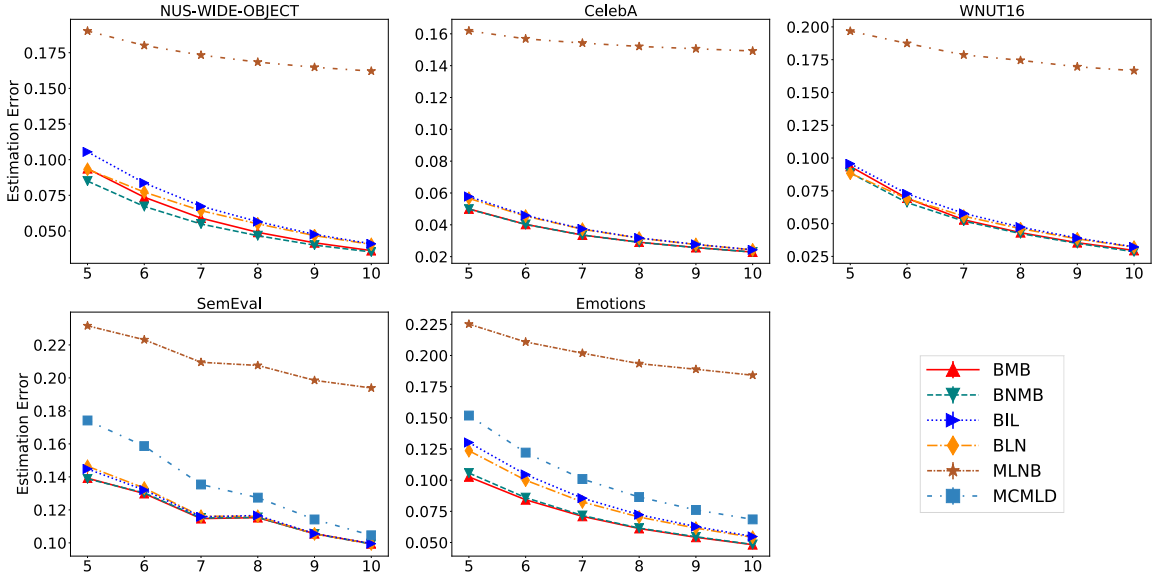


Figure A.4: **Estimation Error of the Worker Reliability Parameters in Simulation Experiments (Measured by Mean L_2 Distance) as a Function of the Number of Annotations per Item T.** The heterogeneity ratio of worker types is R =4:5:5. The performance of MLNB is substantially lower than the competing methods.

reliability.

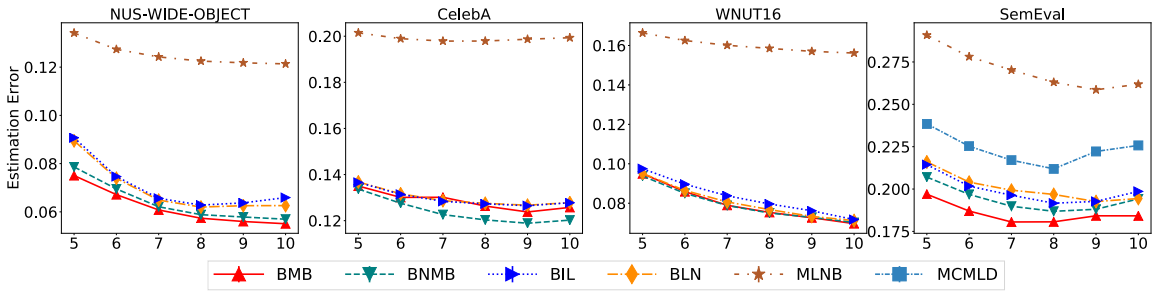


Figure A.5: **Estimation Error of the Worker Reliability Parameters in MTurk Experiments (Measured by Mean L_2 Distance) as a Function of the Number of Annotations per Item T.** The performance of MLNB is substantially lower than the competing methods.

APPENDIX B

Appendix for Chapter 3

B.1 Full Joint Distribution

We note that the joint probability distribution of (3.5) can be written in terms of a matrix $A \in \mathbb{R}^{d^3 \times d^3}$ and a symmetric four dimensional tensor $T \in \mathbb{R}^{d^3 \times d^3 \times d^3 \times d^3}$:

$$p(\boldsymbol{\phi}) \propto \exp\Big(- \sum_{i,i'} A_{i,i'} \phi_i \phi_{i'} - \sum_{i,i',i'',i'''} T_{i,i',i'',i'''} \phi_i \phi_{i'} \phi_{i''} \phi_{i'''} \Big), \tag{B.1}$$

and that the probability distribution is well defined if the four dimensional tensor is positive definite, leading to a log convex probability density function. This can be certified by finding the smallest eigenvector of the tensor using the tensor power method[18]. However, we bypass this step due to the overwhelmingly large number ($\sim d^{12}$) of elements of T and raise the open question of the possibility of taking advantage of structure and sparsity of T to certify whether T is positive definite. In practice, we note that due to the boundedness of ϕ_i, the distribution is well defined. We point out that maximum likelihood estimation would be intractable due to the lack of a closed form for the normalization factor for the joint probability density. For clarity, we iterate that our notation $p(x|y) \propto q(x,y)$ implies $p(x|y) = C(y)q(x,y)$ for some function $C(y)$, which for the purposes of pseudolikelihood is a sufficient notion of proportionality.

REFERENCES

[1] Giancarlo Alfonsi. Reynolds-averaged navier-stokes equations for turbulence modeling. *Applied Mechanics Review*, 62(4), 2009.

[2] Giancarlo Alfonsi. On direct numerical simulation of turbulent flows. *Applied Mechanics Review*, 64(2), 2011.

[3] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[4] A. Augustin, M. Venanzi, A. Rogers, and N. R. Jennings. Bayesian aggregation of categorical distributions with applications in crowdsourcing. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1411–1417, 2017.

[5] José Bento and Andrea Montanari. Which graphical models are difficult to learn? In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, page 1303–1311, Red Hook, NY, USA, 2009. Curran Associates Inc.

[6] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

[9] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[10] J. Bragg, Mausam, and D. S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing*, pages 25–33, 2013.

[11] Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 771–782, New York, NY, USA, 2015. Association for Computing Machinery.

[12] Guy Bresler, David Gamarnik, and Devavrat Shah. Hardness of parameter estimation in graphical models. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 1062–1070, Cambridge, MA, USA, 2014. MIT Press.

[13] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In Ashish Goel, Klaus Jansen, José D. P. Rolim, and Ronitt Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[14] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 48:1–48:9, 2009.

[15] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 285–294, 2013.

[16] Don Daniel, Daniel Livescu, and Jaiyoung Ryu. Reaction analogy based forcing for incompressible scalar turbulence. *Physical Review Fluids*, 3(9):094602, 2018.

[17] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.

[18] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[19] Deepjyoti Deka, Scott Backhaus, and Michael Chertkov. Estimating distribution grid topologies: A graphical learning based approach. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2016.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38, 1977.

[21] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014.

[22] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, 54(4):86–96, 2011.

[23] L. Duan, S. Oyama, H. Sato, and M. Kurihara. Separate or joint? Estimation of multiple labels from crowdsourced annotations. *Expert Systems with Applications*, 41(13):5723–5732, 2014.

[24] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[25] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

[26] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[27] Uriel Frisch. *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press, 1995.

[28] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1631–1640, 2015.

[29] Donald Geman, Hélyette Geman, and Nassim Nicholas Taleb. Tail risk constraints and maximum entropy. *Entropy*, 17(6):3724–3737, 2015.

[30] A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 167–176, 2011.

[31] Vidyadhar P Godambe. An optimum property of regular maximum likelihood estimation. *The Annals of Mathematical Statistics*, 31(4):1208–1211, 1960.

[32] J. K. Goodman, C. E. Cryder, and A. Cheema. Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples. *Journal of Behavioral Decision Making*, 26(3):213–224, 2013.

[33] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 45–51, 2012.

[34] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

[35] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 209–218, 2010.

[36] N. Q. V. Hung, H. H. Viet, N. T. Tam, M. Weidlich, H. Yin, and X. Zhou. Computing crowd consensus with partial agreement. *IEEE Transactions on Knowledge and Data Engineering*, 30(1):1–14, 2018.

[37] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.

[38] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67, 2010.

[39] Katsuhiko Ishiguro, Issei Sato, and Naonori Ueda. Averaged collapsed variational Bayes inference. *Journal of Machine Learning Research*, 18(1):1–29, 2017.

[40] Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[41] P. Isola, J. Xiao, A. Torralba, and A. Oliva. What makes an image memorable? In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 145–152, 2011.

[42] Harry Joe. *Multivariate Models and Multivariate Dependence Concepts*. CRC Press, 1997.

[43] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[44] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[45] G. Kazai, J. Kamps, and N. Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1941–1944, 2011.

[46] H.-C. Kim and Z. Ghahramani. Bayesian classifier combination. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 619–627, 2012.

[47] Daphne Koller and Nir Friedman. *Probabilistic Graphical models: Principles and Techniques*. MIT press, 2009.

[48] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017.

[49] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

[50] P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Houghton Mifflin, 1968.

[51] J. Le, A. Edmonds, V. Hester, and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Proceedings of the SIGIR Workshop on Crowdsourcing for Search Evaluation*, pages 17–20, 2010.

[52] C. Li, B. Wang, V. Pavlu, and J. Aslam. Conditional Bernoulli mixtures for multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2482–2491, 2016.

[53] Yuan Li, Benjamin Rubinstein, and Trevor Cohn. Exploiting worker correlation for label aggregation in crowdsourcing. In *International Conference on Machine Learning*, pages 3886–3895, 2019.

[54] Jun S Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[55] Qiang Liu and Alexander Ihler. Distributed parameter estimation via pseudo-likelihood. *arXiv preprint arXiv:1206.6420*, 2012.

[56] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[57] Andrey Y. Lokhov, Marc Vuffray, Sidhant Misra, and Michael Chertkov. Optimal structure and parameter learning of ising models. *Science Advances*, 4(3), 2018.

[58] J. L. Lumley. The structure of inhomogeneous turbulence. In A. M. Yaglom and V. I. Tatarski, editors, *Atmospheric Turbulence and Wave Propagation*, pages 166–178. Nauka, Moscow, 1967.

[59] C. Meneveau and J. Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1), 2000.

[60] Thomas Minka. Estimating a Dirichlet distribution. Technical report, 2000.

[61] Arvind Mohan, Don Daniel, Michael Chertkov, and Daniel Livescu. Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence. *arXiv preprint arXiv:1903.00033*, 2019.

[62] A. S. Monin and A. M. Yaglom. *Statistical Fluid Mechanics*, volume 2. MIT Press, Cambridge, MA, 1975.

[63] P. G. Moreno, A. Artes-Rodriguez, Y. W. Teh, and F. Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 16:1607–1627, 2015.

[64] S. Nowak and S. Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 557–566, 2010.

[65] D. Padmanabhan, S. Bhat, S. Shevade, and Y. Narahari. Topic model based multi-label classification. In *IEEE 28th International Conference on Tools with Artificial Intelligence*, pages 996–1003, 2016.

[66] Judea Pearl. *Causality*. Cambridge university press, 2009.

[67] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A Dill. Principles of maximum entropy and maximum caliber in statistical physics. *Reviews of Modern Physics*, 85(3):1115, 2013.

[68] Pradeep Ravikumar, Martin J. Wainwright, and John D. Lafferty. High-dimensional ising model selection using l1 -regularized logistic regression. *Ann. Statist.*, 38(3):1287–1319, 06 2010.

[69] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

[70] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, 2011.

[71] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

[72] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.

[73] Michael Sherman. Variance estimation for statistics computed from spatial lattice data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(3):509–523, 1996.

[74] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

[75] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 70–74, 2007.

[76] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1353–1360. 2007.

[77] H. Tennekes. Eulerian and lagrangian time microscales in isotropic turbulence. *Journal of Fluid Mechanics*, 67(3):561–567, 1975.

[78] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, (1):4, 2011.

[79] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[80] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. MULAN: A Java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

[81] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[82] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based Bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 155–164, 2014.

[83] M. Venanzi, W. T. L. Teacy, A. Rogers, and N. R. Jennings. Bayesian modelling of community-based multidimensional trust in participatory sensing under data sparsity. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 717–724, 2015.

[84] Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2595–2603. Curran Associates, Inc., 2016.

[85] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[86] Chong Wang and David M Blei. Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(Apr):1005–1031, 2013.

[87] J. Wang, P. G. Ipeirotis, and F. Provost. Cost-effective quality assurance in crowd labeling. *Information Systems Research*, 28(1):137–158, 2017.

[88] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

[89] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2035–2043, 2009.

[90] Jianxin Yin and Hongzhe Li. A sparse conditional gaussian graphical model for analysis of genetical genomics data. *The annals of applied statistics*, 5(4):2630, 2011.

[91] J. Zhang and X. Wu. Multi-label inference for crowdsourcing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2738–2747, 2018.

[92] M. Zhang and Z. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.