a

SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

Crowdsourcing Aggregation with Deep Bayesian Learning

Shao-Yuan Li[1*], Sheng-Jun Huang[1,2] & Songcan Chen[1]

[1]*Nanjing University of Aeronautics and Astronautics, Nanjing* 211106, *China;*
[2]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093 *China*

Abstract In this paper we consider crowdsourcing classification problem where labeling information from crowds are aggregated to infer latent true labels. We propose a fully Bayesian deep generative crowdsourcing model (BayesDGC), which combines the strength of Deep Neural Networks (DNN) on automatic representation learning and the interpretable probabilistic structure encoding of probabilistic graphical models. The model composes a DNN classifier as prior for the true labels, and a probabilistic model for annotation generation process. The DNN classifier and the annotation generation process share the latent true label variables. To address the inference challenge, we develop a natural-gradient stochastic variational inference, which combines variational message passing for conjugate parameters and sgd for DNN, and learns distributions of latent true labels and workers' confusion matrix through end-to-end training. We illustrate the effectiveness of the proposed model with empirical results on 22 real-world datasets.

Keywords crowdsourcing, classification, fully bayesian deep generative models, natural gradient, stochastic variational inference

Citation Shao-Yuan Li, Sheng-Jun Huang, Songcan Chen. Crowdsourcing Aggregation with Deep Bayesian Learning. Sci China Inf Sci, for review

1 Introduction

Typical supervised learning requires labels for training, whereas for many real-world tasks, acquiring the gold standard labels is not possible or too expensive. On the other hand, in recent years, crowdsourcing [5, 28] has established itself as a reliable solution to collecting annotations for data. With the advent of crowdsourcing services like Amazon Mechanical Turk[1)] and Crowdflower[2)], crowdsourcing has been used for collecting vast annotated datasets in a short time in numerous fields such as natural language understanding [24], medical diagnosis [18], vision image tagging [29] and entity resolution [12].

While crowdsourcing is scalable enough, the annotations provided by the annotators are inherently subjective, and there can be substantial amount of disagreement among different annotators. The noise associated with annotations can lead to limited performance of conventional learning algorithms. Consequently, one core task in crowdsourcing has been estimating the hidden groundtruth labels from collected annotations. A number of methods have been proposed for this purpose, in this paper, based on whether they only use the annotation information or also use data feature information, we categorize them into two groups. Among the methods built only on annotation information, the simplest and most common technique is majority voting, which treats annotators equally reliable and grants workers equal votes. To take into account the annotators' skill and data difficulty variation, probabilistic models have emerged. As the initiating work, Dawid and Skene [2] parameterized the annotators' reliability using their error rates and models the annotations as noisy observations of the latent groundtruth. Extending DS with finer levels of annotation generation process and optimization techniques, various following works were proposed [14, 30, 32].

* Corresponding author (email: lisy@nuaa.edu.cn)

1) https://www.mturk.com
2) http://crowdflower.com

Considering that the features of data contain complementary information, another line of work proceed by integrating the features into learning model [1,3,18,19,21,25]. The work of Raykar *et al.* [18] is one of the most prominent which extended DS [2] to jointly learn the worker parameters and a logistic regression classifier. Treating the classifier as prior of the groundtruth label, the optimization naturally follows the expectation-maximization (EM) procedure of DS [2]. This idea was later extended to other types of classifier models such as Gaussian process classifiers [19] and, recently, to deep neural networks, called as deep crowd learning (DCL). The works of [3, 21, 25] are three representatives in DCL. [21] exploited convolutional neural network as the classifier and used the EM optimization procedure. To avoid the computational overhead of iterative EM in [21], [3, 25] proposed to treat the latent true labels as one hidden layer of the deep neural network, and take the crowds' annotations as the output layer. The whole DNN was directly trained on the noisy labels end-to-end using back propagation.

While [3, 25] combated the computational issue of EM-style algorithms, their heuristic optimization implementation is not guaranteed to maximizing certain lower bound of the original learning objective, in contrary to the EM-based algorithms. Besides, they lose the probabilistic structure interpretation of the DNN classifier output and the workers' parameterization. In this paper, to keep the strength of DNN on automatic representation learning and the flexibility of probabilistic graphical models on encoding interpretable probabilistic structure, we propose a fully Bayesian deep generative crowdsourcing model (BayesDGC).

Specifically, we exploit the principle idea of regarding the DNN classifier as prior for the true labels, and parameterize each worker's reliability using a confusion matrix. The latent true label variables are shared among the DNN classifier and the annotation generation process. In contrast to [3, 25] which heuristically learn uninterpretable deterministic parameters, and require human tuning, our model is fully Bayesian. BayesDGC performs distribution inference for the latent true labels and the workers' confusion matrix, automatically trade-off between the model complexity and data fitting. To address the inference challenge, we develop a natural-gradient stochastic variational inference algorithm, which combines variational message passing for conjugate structures and the sgd of DNN, and conducts all parameter training in an end-to-end manner using back propagation. Besides, the optimization process is guaranteed to maximizing a variational lower bound of observed annotations' likelihood.

While our modeling of independent worker confusion matrix in this paper is basic, we point out that the proposed deep Bayesian inference is general enough to apply to more sophisticated parameterizations, as long as the parameter conjugate structures can be exploited. We take deep Bayesian crowdsourcing on more sophisticated annotation generation process such as correlated workers [6, 14] for future work.

2 Related work

In the past few years, many methods have been proposed for crowd aggregation to address the annotation noise and trustworthiness issues. Among them, majority voting (MV) is the most straightforward and widely used, which conducts simple voting over all workers. Considering that MV ignores the quality differences of workers' annotations, [22] and [26] respectively proposed strategies by considering the certainty information of the majority classes and minority classes, and also quality differences of workers over different instances. Probabilistic approaches modeling the workers' expertise and instances' difficulties were another exploration line. The DS model [2] is one key early contribution. To deal with the clinical diagnostics problem, DS proposed to use error rates to parameterize the worker labels conditioned on the item's true label, and proposed an EM algorithm to estimate the error rates and latent true labels. The generative annotation modeling idea has been the basis of many other variants, which model the annotation generation process in finer levels using different optimization techniques. For example, [29,30] also took item difficulty into account and proposed an EM algorithm to infer the most probable label. [32] used a confusion matrix on each item and estimates the latent true labels via a minimax entropy principle, promoting the true label distribution close to empirical worker annotation distributions. [15] conducted the optimization in crowdsourcing from variational inference perspective, and proposed variational inference methods including belief propagation and mean field. Bayesian extensions of DS were also explored, such as [8, 23, 27] which generalized DS to fully Bayesian by introducing Dirichlet priors, and conducted inference respectively through Gibbs sampling, variational Bayesian inference, and EM. Recently, rather than treating the workers independently, modeling correlation between workers has attracted attention. In [17], a non-parametric Dirichlet process is used to explicitly model workers in clusters within which

confusion matrices are likely similar. [6] derived a minimax error rate for general confusion-matrix-based models and proposed a worker clustering model. [14] proposed a mixture model for classes and made connection with annotation tensor decomposition.

Rather than purely relying on the annotations to infer the truth, works on utilizing the data feature information to help improve estimating true labels have been studied. Extending DS by using a logistic regression classifier as the true label prior, [18] was one of the pioneers in this direction. Other types of classifier models such as Gaussian process classifier and supervised latent Dirichlet allocation [19, 20] were also proposed. They mainly work by integrating a supervised learning model as prior of the true labels, and add it into the probabilistic annotation generation model. [31] proposed to construct local linear neighborhood graph in the feature space and conduct annotation distribution propagation in the label space.

With the success of deep neural networks which allow flexible data representations to be learned [11], deep crowd learning (DCL) attempting to combine DNN into crowdsourcing were performed [1, 3, 16, 21, 25]. [21] used a cnn classifier as label prior and used the EM optimization procedure. [3, 25] avoided the computational overhead of EM by heuristically conducting direct loss minimization on the noisy annotations, thus the DNN sgd optimization can be used. Technically, our work is inspired by [1, 16], which exploited deep generative models and conducted the inference using end-to-end backpropagation. Our work differs in problem scenario and the implementation. [1] considered semi-supervised crowd classification based on the inference technique of non-Bayesian semi-supervised variational autoencoder (SVAE) [9]. [16] considered clustering problem, thus the model construction and inference implementations are totally different.

3 The proposed model

We denote the set of N examples observations by $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in R^d$ denotes the d-dimensional feature values of the i-th example. The collected annotations provided by W workers are denoted as $\mathbf{L} \in \{0, 1, \cdots, K\}^{N \times W}$, with $\mathbf{L}_{ij}$ representing the label assignment of example i given by worker j. When $\mathbf{L}_{ij} = k (k \neq 0)$, it means that the i-th example is categorized as k-th class by the j-th worker. When $\mathbf{L}_{ij} = 0$, it means that the annotations of worker j for example i is not observed. Our target is to estimate the latent true labels $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ for $\mathbf{X}$ making the best use of the feature and crowds' annotations.

3.1 Fully Bayesian deep generative crowdsourcing (BayesDGC)

The graphical model of the proposed approach is shown in Figure 1. The model contains two main parts: the annotation generation process $p(\mathbf{L}_{ij}|\mathbf{y}_i; \mathbf{V}_j)$ and the prior model for latent true labels $p(\mathbf{y}_i|\mathbf{x}_i, \pi)$.

For the annotation generation part, we adopt the classic independent confusion matrix parameterization for each worker. Specifically, the confusion matrix of the $j-$th worker can be characterized by parameters $\mathbf{V}_j = \{\boldsymbol{\nu}_{j1}, \cdots, \boldsymbol{\nu}_{jK}\}$, with vector $\boldsymbol{\nu}_{jk} = \{\nu_{jk1}, \cdots, \nu_{jkK}\}$. Given the true label $\mathbf{y}_i$ of example $\mathbf{x}_i$, the generation likelihood of annotation $\mathbf{L}_{ij}$ is

$$p(\mathbf{L}_{ij} = l|\mathbf{y}_i = k, \mathbf{V}_j) = \nu_{jkl}. \tag{1}$$

Assuming the examples being independent and the annotations for each example are generated independently by different workers, the total likelihood of the annotations can be written as

$$P(\mathbf{L}|\mathbf{Y}, \mathbf{V}) = \prod_{i=1}^{N} \prod_{j=1}^{W} \mathcal{I}[\mathbf{L}_{ij} \neq 0] p(\mathbf{L}_{ij}|\mathbf{y}_i, \mathbf{V}_j). \tag{2}$$

For the latent true labels' prior model, we exploit two priors, one data invariant prior $p(\mathbf{y}_i; \boldsymbol{\pi})$ and one feature dependent neural network classifier prior $p(\mathbf{y}_i|\mathbf{x}_i; \boldsymbol{\gamma})$ parameterized by $\boldsymbol{\gamma}$, respectively defined as

$$p(\mathbf{y}_i, \boldsymbol{\pi}) = \prod_{k=1}^{K} \boldsymbol{\pi}_k, \qquad p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma}) = Categorical(\tau(\mathbf{x}_i; \boldsymbol{\gamma})) \tag{3}$$

Assuming the examples being independent, the prior of $\mathbf{Y}$ can be written as

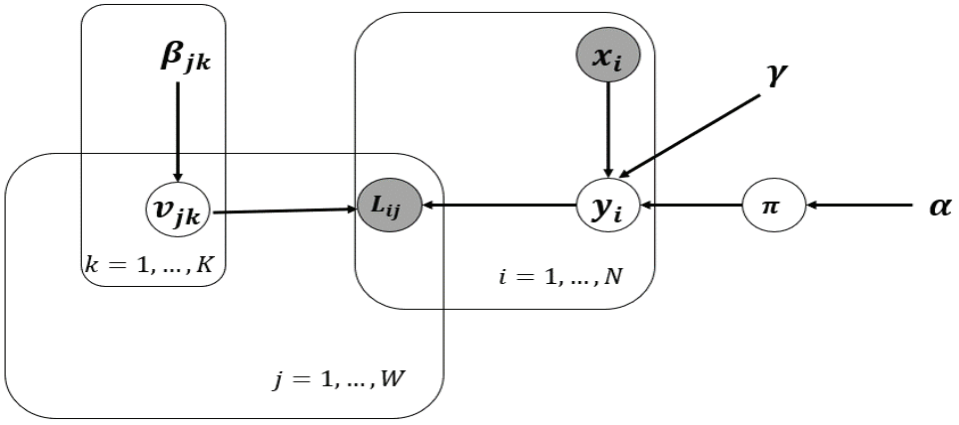

Figure 1 The plate notation for our proposed BayesDGC

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\gamma}) = p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma})p(\mathbf{Y}|\boldsymbol{\pi}) = \prod_{i=1}^{N} p(\mathbf{y}_i, \boldsymbol{\pi})p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma}). \tag{4}$$

Apart from the above annotation generation process and true label prior, we also assume conjugate Dirichlet priors over the global parameters $\Theta = \{\mathbf{V}, \boldsymbol{\pi}\}$, given by

$$p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}), \quad p(\mathbf{V}) = \prod_{j=1}^{W}\prod_{k=1}^{K} p(\boldsymbol{\nu}_{jk}) = Dir(\boldsymbol{\nu}_{jk}|\boldsymbol{\beta}_{jk}) \tag{5}$$

Thus the overall joint distribution of the observed annotations $\mathbf{L}$, the latent true labels $\mathbf{Y}$ and global parameters $\boldsymbol{\Theta} = \{\mathbf{V}, \boldsymbol{\pi}\}$ can be represented as:

$$p(\mathbf{L}, \mathbf{Y}, \boldsymbol{\Theta}|\mathbf{X}, \boldsymbol{\gamma}) = p(\boldsymbol{\pi})p(\mathbf{Y}|\boldsymbol{\pi})p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma})p(\mathbf{L}|\mathbf{Y}, \mathbf{V})p(\mathbf{V}). \tag{6}$$

Our objective is to estimate the posterior distribution of global parameters $p(\boldsymbol{\Theta}|\mathbf{L}, \mathbf{X})$ and true labels $p(\mathbf{Y}|\mathbf{L}, \mathbf{X})$, through maximizing the likelihood of observed annotations $p(\mathbf{L})$.

In case of without the DNN classifier prior, our model degenerate to the Bayesian extension to the DS model, which was independently implemented using optimization procedure such as Gibbs sampling [8], mean-field variational Bayes [23] and EM [27]. Whereas in our deep crowd model, when combined with nonlinear DNN classifier $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma})$, the Gibbs sampling and EM would be too slow, since they involve expensive sampling loop per data or iterative procedure per epoch. The variational mean-field message passing is quite efficient, whereas it depends on conjugate exponential family likelihood to preserve tractable structures, which does not hold for general data models such as neural networks. In the next subsection, we build on recent advances in structured variational autoencoders (SVAE), and develop a stochastic variational inference algorithm for our model, which is able to conduct efficient end-to-end training of all parameters and guaranteed to maximize some variational lower bound of the log likelihood of annotations $\log p(\mathbf{L})$.

3.2 Natural-gradient stochastic variational inference algorithm

To perform efficient inference in deep probabilistic graphical models, the variational autorencoder (VAE) [10] used reparameterization technique, and proposed to use a recognition network to fit a mapping from the data to concerned distribution's parameters. Thus the posterior distribution can be inferred through end-to-end optimizing over the whole neural networks. In structured vae (SVAE) [7], the authors extended VAE with idea from natural gradient stochastic variation inference (SVI) [4] for conditional conjugate models. The idea is straightforward, instead of using the recognition network to output the posterior distribution's parameters, they used the recognition network to output the conjugate graphical model potentials, which are then used for mean-field variational message passing and natural gradient computation. The advantage of SVAE is that, it can leverage conjugate structure to efficiently compute natural gradients of variational parameters, which enables effective second-order optimization.

In this paper, we follow the optimization procedure of SVAE, and implement the natural gradient stochastic variational inference algorithm for our BayesDGC. In the following, we give the detailed implementation. As VAE [10], the variational evidence lower bound (ELBO) is derived as

$$\log p(\mathbf{L}) \geqslant \mathcal{L}(\mathbf{Y}, \boldsymbol{\Theta}, \boldsymbol{\gamma}) \triangleq \mathbb{E}_{q(\boldsymbol{\Theta}, \mathbf{Y})} [\log \frac{p(\mathbf{L}, \mathbf{Y}, \boldsymbol{\Theta} | \mathbf{X}, \boldsymbol{\gamma})}{q(\mathbf{Y}) q(\boldsymbol{\Theta})}]. \tag{7}$$

Here we exploit a mean-field variational family $q(\boldsymbol{\Theta}, \mathbf{Y}) = q(\boldsymbol{\Theta})\, q(\mathbf{Y})$. To make use of the conjugate structure of our model, we rewrite the distribution of $p(\boldsymbol{\pi}), p(\mathbf{V}), p(\mathbf{Y}|\boldsymbol{\pi})$ defined in Eq. 5, 4 in their exponential family form:

$$p(\boldsymbol{\pi}) = \exp\{< \boldsymbol{\eta}_{\boldsymbol{\pi}},\ \boldsymbol{t}(\boldsymbol{\pi}) > - \log\ Z(\boldsymbol{\eta}_{\boldsymbol{\pi}})\} \tag{8}$$

$$p(\boldsymbol{\nu}_{jk}) = \exp\{< \boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}},\ \boldsymbol{t}(\boldsymbol{\nu}_{jk}) > - \log\ Z(\boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}})\} \tag{9}$$

$$p(\mathbf{y}|\boldsymbol{\pi}) = \exp\{< \boldsymbol{\eta}_{\mathbf{y}}\ (\boldsymbol{\pi}),\ \boldsymbol{t}(\mathbf{y}) > - \log\ Z(\boldsymbol{\eta}_{\mathbf{y}}\ (\boldsymbol{\pi}))\} = \exp\{< \boldsymbol{t}(\boldsymbol{\pi}), (\boldsymbol{t}(\mathbf{y}), \mathbf{1}) >\} \tag{10}$$

Here $\boldsymbol{\eta}$ denotes the natural parameters, $\boldsymbol{t}(\cdot)$ denotes the sufficient statistics, and $\log Z(\cdot)$ denotes the log partition function. For Eq. 8- Eq. 10, their expressions are respectively:

$$\boldsymbol{\eta}_{\boldsymbol{\pi}} = \begin{bmatrix} \boldsymbol{\alpha}_1 - 1 \\ \vdots \\ \boldsymbol{\alpha}_K - 1 \end{bmatrix}, \boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}} = \begin{bmatrix} \boldsymbol{\beta}_{jk1} - 1 \\ \vdots \\ \boldsymbol{\beta}_{jkK} - 1 \end{bmatrix}, \boldsymbol{\eta}_{\mathbf{y}}\ (\boldsymbol{\pi}) = \begin{bmatrix} \log \boldsymbol{\pi}_1 \\ \vdots \\ \log \boldsymbol{\pi}_K \end{bmatrix}$$

$$\boldsymbol{t}(\boldsymbol{\pi}) = \begin{bmatrix} \log \boldsymbol{\pi}_1 \\ \vdots \\ \log \boldsymbol{\pi}_K \end{bmatrix}, \boldsymbol{t}(\boldsymbol{\nu}_{jk}) = \begin{bmatrix} \log \boldsymbol{\nu}_{jk1} \\ \vdots \\ \log \boldsymbol{\nu}_{jkK} \end{bmatrix}, \boldsymbol{t}(\mathbf{y}) = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}$$

$$\log\ Z(\boldsymbol{\eta}_{\boldsymbol{\pi}}) = \sum_{k=1}^{K} \log \Gamma(\boldsymbol{\alpha}_k) - \log \Gamma(\sum_{k=1}^{K} \boldsymbol{\alpha}_k),\ \log\ Z(\boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}}) = \sum_{l=1}^{K} \log \Gamma(\boldsymbol{\beta}_{jkl}) - \log \Gamma(\sum_{k=1}^{K} \boldsymbol{\beta}_{jkl}),\ \log\ Z(\boldsymbol{\eta}_{\mathbf{y}}\ (\boldsymbol{\pi})) = 0$$

Here Γ is the Gamma function. Similarly, rewriting the posterior distribution in its exponential family form $q(\theta) = \exp\{< \boldsymbol{\eta}_\theta, \boldsymbol{t}(\theta) > - \log Z(\theta)\}$, $\theta \in \boldsymbol{\Theta} \cup \mathbf{Y}$. Substituting the above exponential family expressions for distributions, the ELBO $\mathcal{L}(\mathbf{Y}, \boldsymbol{\Theta}; \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ in Eq. 7 now becomes $\mathcal{L}(\boldsymbol{\eta}_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}; \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ with $\boldsymbol{\eta}$ as parameters:

$$\mathcal{L}(\boldsymbol{\eta}_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}) \triangleq \mathbb{E}_{q(\boldsymbol{\Theta}, \mathbf{Y})} [\log \frac{p(\mathbf{L}, \mathbf{Y}, \boldsymbol{\Theta} | \mathbf{X}, \boldsymbol{\gamma})}{q(\mathbf{Y}) q(\boldsymbol{\Theta})}]. \tag{11}$$

To leverage the conjugate structure of our model, as SVAE [7], we use the DNN classifier to form conjugate graphical model potentials:

$$\psi(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\gamma}) \triangleq < \boldsymbol{\gamma}(\mathbf{x}_i), \boldsymbol{t}(\mathbf{y}_i) > . \tag{12}$$

Replacing $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma})$ with the conjugate term defined by $\psi(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma})$, we have the following surrogate objective $\hat{\mathcal{L}}$:

$$\hat{\mathcal{L}}(\boldsymbol{\eta}_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}) \triangleq \mathbb{E}_{q(\boldsymbol{\Theta}, \mathbf{Y})} [\log \frac{p(\mathbf{L}, \mathbf{Y}, \boldsymbol{\Theta}) \exp\{\psi(\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma})\}}{q(\mathbf{Y}) q(\boldsymbol{\Theta})}]. \tag{13}$$

Algorithm 1 Bayesian deep generative crowdsourcing (BayesDGC)

Input: example features $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, crowd annotations $\mathbf{L} \in \{0, 1, \cdots, K\}^{N \times W}$, initial value for the global variational parameters $\boldsymbol{\eta}_{\boldsymbol{\Theta}}$ and neural network parameters $\boldsymbol{\gamma}$

1: **Repeat**:
2: Given $\boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}$, update the true label's natural parameter $\boldsymbol{\eta}^*_{\mathbf{y}_i}$ for each example using Eq.14
3: Given estimated $\boldsymbol{\eta}^*_{\mathbf{y}_i}$, compute the gradient of $\boldsymbol{\eta}_{\boldsymbol{\Theta}}$ using Eq. 17- 18 and $\boldsymbol{\gamma}$ via DNN back propagation, then conduct sgd updating for them
4: **Until** The lower bound $\mathcal{J}(\boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma})$ converges or the maximum number of epochs is reached.

Then similar as SVI [4], we can conduct the natural gradient stochastic variational inference. With the global variational parameters $\boldsymbol{\eta}_{\boldsymbol{\Theta}}$ fixed, the optimal solution for $q^*(\mathbf{Y})$ factorizes over examples, $q^*(\mathbf{Y}) = \prod_{i=1}^N q^*(\mathbf{y}_i)$. Each $q^*(\mathbf{y}_i)$ is derived in closed form:

$$\begin{aligned} \log q^*(\mathbf{y}_i) &= \mathbb{E}_{q(\boldsymbol{\pi})} \log p(\mathbf{y}_i|\boldsymbol{\pi}) + < \boldsymbol{\gamma}(\mathbf{x}_i), \boldsymbol{t}(\mathbf{y}_i) > + \mathbb{E}_{q(\mathbf{V})} \log p(\mathbf{L}|\mathbf{Y}, \mathbf{V}) + const, \\ \boldsymbol{\eta}^*_{\mathbf{y}_i} &= \mathbb{E}_{q(\boldsymbol{\pi})} \boldsymbol{t}(\boldsymbol{\pi}) + \boldsymbol{\gamma}(\mathbf{x}_i) + \sum_{i=1}^N \sum_{j=1}^W \mathcal{I}(\mathbf{L}_{ij} \neq 0) \mathbb{E}_{q(\boldsymbol{\nu}_{j\mathbf{L}_{ij}})} \boldsymbol{\nu}_{j\mathbf{L}_{ij}} \end{aligned} \tag{14}$$

Please note that in $\boldsymbol{\nu}_{j\mathbf{L}_{ij}}$ the $\mathbf{L}_{ij}$ acts as the second dimension index of $\boldsymbol{\nu}$, i.e., when $\mathbf{L}_{ij} = k$, $\boldsymbol{\nu}_{j\mathbf{L}_{ij}}$ becomes $\boldsymbol{\nu}_{jk}$. Plugging $\boldsymbol{\eta}^*_{\mathbf{Y}}$ back into $\mathcal{L}$, we can define the final optimization objective as:

$$\mathcal{J}(\boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}) \triangleq \mathcal{L}(\boldsymbol{\eta}^*_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}). \tag{15}$$

It was proved in SVAE [7] that $\mathcal{J}(\boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma})$ lower bounds the partially optimized mean field objective, i.e., $\max_{\boldsymbol{\eta}_{\mathbf{Y}}} \mathcal{L}(\boldsymbol{\eta}_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}) \geqslant \mathcal{J}(\boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma})$, thus $\mathcal{J}$ can serve as the variational lower bound of $\mathcal{L}$. According to [7], the natural gradient of $\mathcal{J}$ with respect to $\boldsymbol{\eta}_{\boldsymbol{\Theta}}$ is derived as:

$$\widetilde{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\Theta}}} \mathcal{J} = [\boldsymbol{\eta}^0_{\boldsymbol{\Theta}} + \mathbb{E}_{q^*(\mathbf{Y})}(\boldsymbol{t}(\mathbf{Y}, \mathbf{X}, \mathbf{L}), \mathbf{1}) - \boldsymbol{\eta}_{\boldsymbol{\Theta}}] + (\nabla_{\boldsymbol{\eta}(\mathbf{Y})} \mathcal{L}(\boldsymbol{\eta}^*_{\mathbf{Y}}, \boldsymbol{\eta}_{\boldsymbol{\Theta}}, \boldsymbol{\gamma}), \mathbf{0}) \tag{16}$$

Here $\boldsymbol{\eta}^0_{\boldsymbol{\Theta}}$ is the prior natural parameter value of $\boldsymbol{\Theta}$ set by users. For our problem, the natural gradients for $\widetilde{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\pi}}} \mathcal{J}$ and $\widetilde{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}}} \mathcal{J}$ are respectively derived as:

$$\widetilde{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\pi}}} \mathcal{J} = \boldsymbol{\eta}^0_{\boldsymbol{\pi}} + \sum_{i=1}^N \mathbb{E}_{q^*(\mathbf{y}_i)} \boldsymbol{t}(\mathbf{y}_i) - \boldsymbol{\eta}_{\boldsymbol{\pi}} \tag{17}$$

$$\widetilde{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}}} \mathcal{J} = \boldsymbol{\eta}^0_{\boldsymbol{\nu}_{jk}} + \sum_{i=1}^N \mathcal{I}(\mathbf{L}_{ij} \neq 0) \mathbb{E}_{q^*(\mathbf{y}_i)} \boldsymbol{t}(\mathbf{y}_i) \otimes \bar{\mathbf{L}}_{ij} - \boldsymbol{\eta}_{\boldsymbol{\nu}_{jk}} \tag{18}$$

Here $\bar{\mathbf{L}}_{ij}$ is the one-hot vector representation of $\mathbf{L}_{ij}$. And for parameters $\boldsymbol{\gamma}$, their gradient $\nabla_{\boldsymbol{\gamma}} \mathcal{J}$ can be directly computed within the DNN back propagation framework. The complete optimization process of our BayesDGC is shown in Algorithm 1.

After the training finished, we can estimate each example's true label assignment and each worker's confusion matrix using their respective expected sufficient statistics.

$$\mathbb{E}_{q(\mathbf{y})} \boldsymbol{t}(\mathbf{y}) = \begin{bmatrix} \boldsymbol{\pi}_1 \\ \vdots \\ \boldsymbol{\pi}_K \end{bmatrix}, \quad \mathbb{E}_{q(\boldsymbol{\nu}_{jk})} \boldsymbol{t}(\boldsymbol{\nu}_{jk}) = \begin{bmatrix} \varphi(\boldsymbol{\beta}_{jk1}) \\ \vdots \\ \varphi(\boldsymbol{\beta}_{jkK}) \end{bmatrix} - \varphi(\sum_{l=1}^K \boldsymbol{\beta}_{jkl}) \tag{19}$$

Here φ is the digamma function. For our classification problem, the DNN model with parameter $\boldsymbol{\gamma}$ can act as learned classifier. For new data with feature as input, their label probability can be predicted by applying softmax to the output of the DNN model.

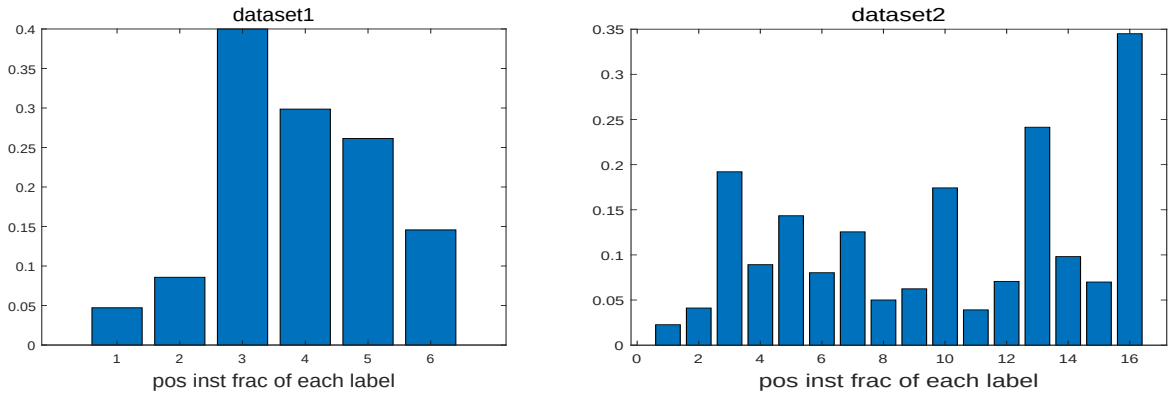

Figure 2 The positive instance fraction of each label for *dataset1* and *dataset2*.

4 Experiments

In this section, we compare the proposed approach with several baseline methods on real-world crowdsourcing datasets.

Data Sets: We use the two image crowdsourcing datasets that we have previously collected for multi-label crowdsourcing study [13]. Their names are *dataset1* and *dataset2*, which respectively concerns 6, 16 candidate labels and 700, 1495 images, with groundtruth labels annotated by human volunteers. The data analysis in [13] shows that the crowds' macro F1 scores vary mainly around $[0.70, 0.80]$, which indicates the reliability of majority workers and establishes the base of learning feasibility.

In this paper, we conduct experiments on each label independently, thus get 22 binary datasets. Originally in [13], annotations of 18 and 15 workers are kept for experiments. Observation from the results of [13] and our experiments show that, the performance of most methods converges when the number of workers exceeds 10, in this paper, for experiment efficiency, we keep annotations from 9 workers who annotated most data and conduct aggregation. The original 1248-dim fisher vector features are used.

Comparison Methods: We compared with four representative state of the art crowdsourcing methods MV, DS [2], MaxEn [32] and Yutc [18]. Besides, for our proposed BayesDGC mdoel, we also implement the none deep Bayesian variant BayesGC, for which the DNN classifier prior is not used.

For the proposed BayesDGC and BayesGC, the Dirichlet prior $Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}^0)$ with $\boldsymbol{\alpha}^0 = 1.1$ is used for $\boldsymbol{\pi}$. For $\boldsymbol{\nu}_{jk}$, one same prior is used for all workers, and $Dir(\boldsymbol{\nu}_{jk}|\boldsymbol{\beta}^0)$ with $\boldsymbol{\beta}^0_{kk} = 5, \boldsymbol{\beta}^0_{kk'} = 2, k \neq k'$ is used to encode that workers are better than random guessing. For BayesDGC, a one hidden layer (with 100 nodes) MLP is exploited as the deep neural network classifier. Adam Optimizer with 0.001 learning rate is used. For the baselines, we use codes provided by their authors and the default parameter suggested there are used. Except for DS, the two coin model implemented by [15] is used.

To test the approaches' performance dependence on the amount of annotations, we vary the observed fraction of annotations p from 10% to 100% in a uniform random manner, and report the average and standard deviation results for 10 times repetitions. As the datasets are originally for multi-label tasks, the 22 binary data are highly imbalanced, whose positive instance fraction are shown in Figure 2. It can be seen that most labels are extremely imbalanced. Thus to conduct the results evaluation, we treat the top k ranked labels of each method as its positive prediction and the rest as negative. Here k is the number of true positive labels for each label. The accuracy and F1 score results are reported. In the future, we plan to design algorithms to take into account the imbalance factor for crowdsourcing learning.

4.1 Results

For the 22 datasets, we use l to denote the corresponding binary classification task for the l-th label of dataset1 and dataset2. The accuracy results are shown in Figure 3. It can be seen that the proposed BayesDGC outperforms other methods significantly in most time, whereas the none deep Bayesian variant BayesGC is inferior. For all the approaches, an overall monotone performance increasing is observed as the number of annotations increases. It's notable that the four annotation only exploitation methods MV, DS, MaxEn, and BayesGC often achieve close performance, and exhibiting an obvious gap with BayesDGC, especially when the number of annotations is limited. This shows that as an essential information source,

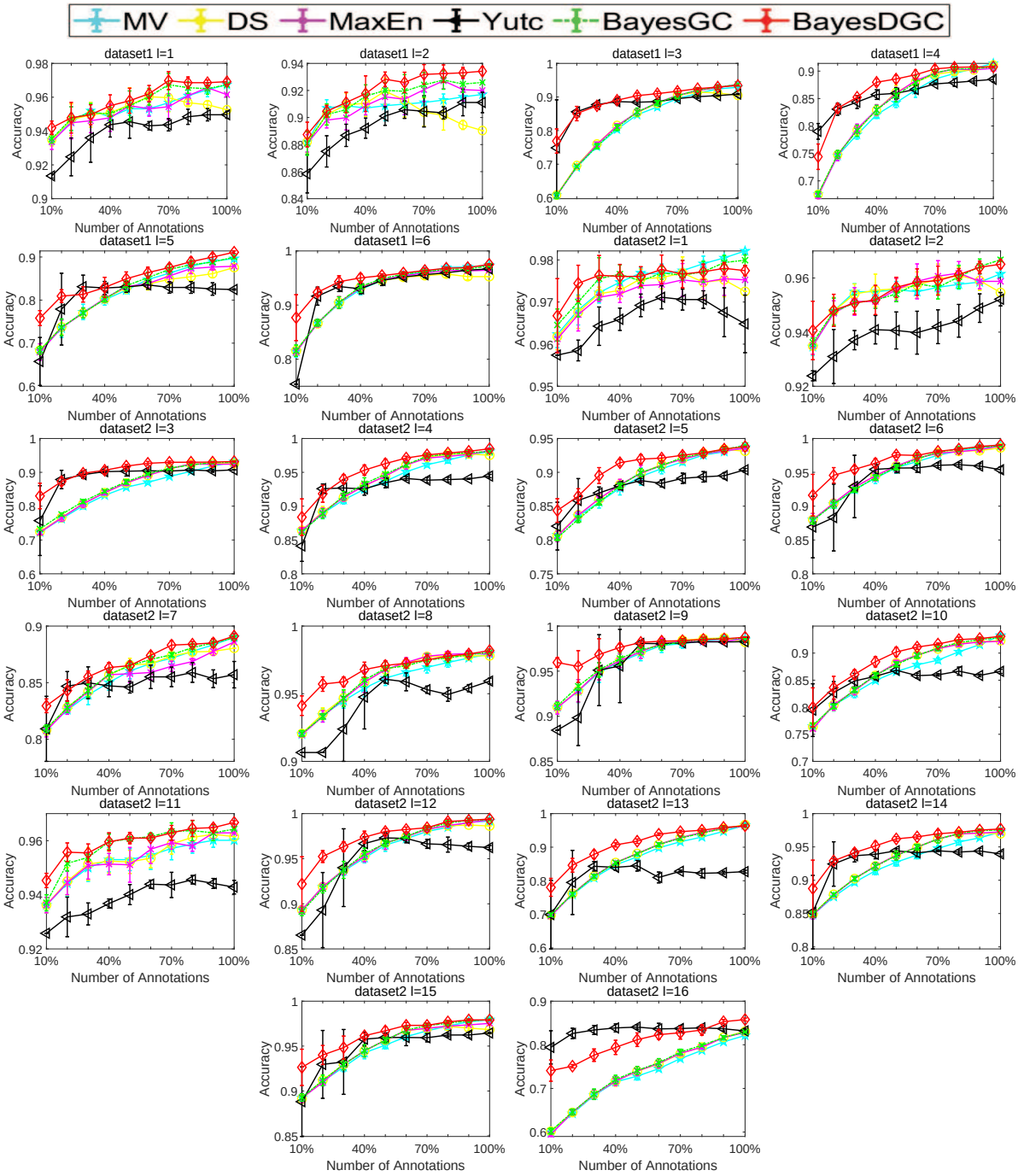

Figure 3 Accuracy results of all methods on 22 real-word datasets.

the data features should not be ignored. Then we look at the performance of Yutc, which exploited the feature information through a logistic regression classifier. On some data set such as dataset1 $l = 3, 6$, dataset2 $l = 3, 6, 16$, Yutc achieves comparable results or even better. But it's not stable, in some cases even perform the worst, e.g., dataset1 $l = 1, 2$, dataset2 $l = 1, 2, 8, 11$, which may be due to the linear model inefficiency or improper parameter setting, which means that for specific applications, careful parameter tuning should be conducted. Whereas for our fully Bayesian deep model, the DNN feature learning provides sufficient model capacity, and the parameter tuning is automatically conducted.

The results for the F1 score are shown in Figure 4. The comparison is similar with the accuracy result, but with much lower performance, indicating the class imbalance challenge for crowdsourcing learning.

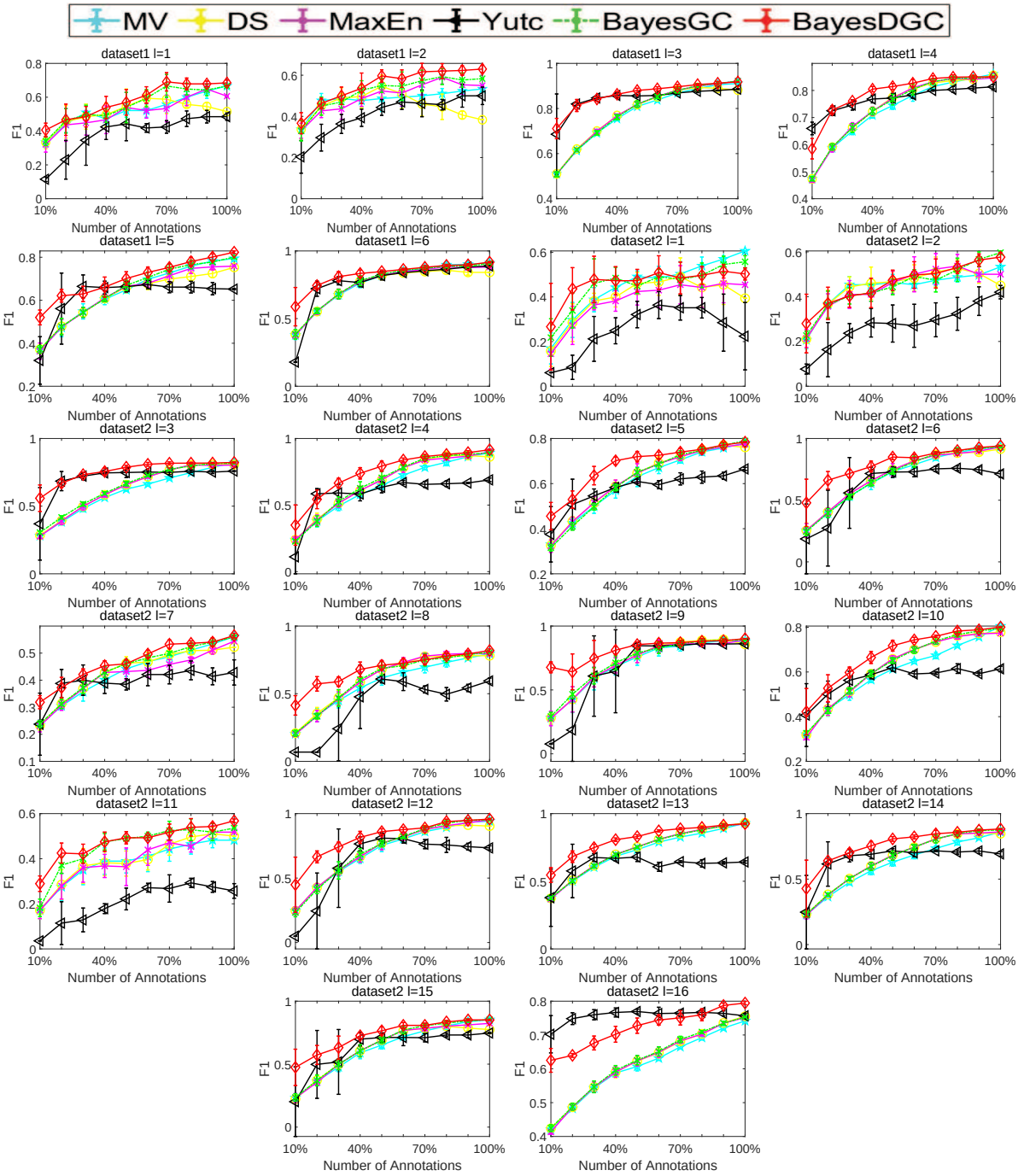

Figure 4 F1 score results of all methods on 22 real-word datasets.

5 Conclusion

In this paper, we propose a fully Bayesian deep generative crowdsourcing model (BayesDGC) for classification. The model composes a probabilistic annotation generation process, and a deep neural network model for effective representation learning. To address the inference challenge, we implement an efficient end-to-end natural gradient stochastic variational inference algorithm, which avoids the computational overhead of EM and sampling approaches, and at the same time keeps the interpretable probabilistic structure. Experiments show the superiority of the proposed approach. In the future, we plan to extend the general enough inference algorithm to more sophisticated crowdsourcing aggregation problems, such as annotation correlation modeling and extension to multi-label tasks.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No.61906089), Jiangsu Province Basic Research Program (Grant No.BK20190408) and China Postdoc Science Foundation (The First Pre-station Special Grant).

References

1 Atarashi, K., Oyama, S., Kurihara, M.: Semi-supervised learning from crowds using deep generative models. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. pp. 1555–1562. New Orleans, Louisiana (2018)
2 Dawid, A.P., Skene, A.M.: Maximum likeihood estimation of observer error-rates using the em algorithm. Journal of the Royal Statistical Society **28**(1), 20–28 (1979)
3 Filipe Rodrigues, F.C.P.: Deep learning from crowds. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. pp. 1611–1618. New Orleans, Louisiana (2018)
4 Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.W.: Stochastic variational inference. Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
5 Horvitz, E.: Reflections on challenges and promises of mixed-initiative interaction. AI Magazine **28**(2), 13–22 (2007)
6 Imamura, H., Sato, I., Sugiyama, M.: Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. In: Proceedings of the 35th International Conference on Machine Learning. pp. 2152–2161. Stockholm, Sweden (2018)
7 Johnson, M.J., Duvenaud, D., Wiltschko, A.B., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 2946–2954 (2016)
8 Kim, H.C., Ghahramani, Z.: Bayesian classifier combination. Artificial Intelligence and Statistics p. 619627 (2012)
9 Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3581–3589 (2014)
10 Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the 2nd International Conference on Learning Representations. Banff, AB, Canada, (2014)
11 LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436444 (2015)
12 Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W., Han, J.: A confidence-aware approach for truth discovery on long-tail data. Proceedings of the VLDB Endowment **8**(4), 425436 (2014)
13 Li, S.Y., Jiang, Y., Chawla, N.V., Zhou, Z.H.: Multi-label learning from crowds. IEEE Transactions on Knowledge and Data Engineering **31**(7), 1369–1382 (2019)
14 Li, Y., Rubinstein, B.I.P., Cohn, T.: Exploiting worker correlation for label aggregation in crowdsourcing. In: Proceedings of the 36th International Conference on Machine Learning. pp. 3886–3895. Long Beach, California (2019)
15 Liu, Q., Peng, J., Ihler, A.: Variational inference for crowdsourcing. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 692–700 (2012)
16 Luo, Y., Tian, T., Shi, J., Zhu, J., Zhang, B.: Semi-crowdsourced clustering with deep generative models. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 3216–3226 (2018)
17 Moreno, P.G., Artes-Rodrguez, A., Teh, Y.W., Perez-Cruz, F.: Bayesian nonparametric crowdsourcing. Journal of Machine Learning Research **16**, 1607–1627 (2015)
18 Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of Machine Learning Research **11**, 1297–1322 (2010)
19 Rodrigues, F., Pereira, F., , Ribeiro, B.: Gaussian process classification and active learning with multiple annotators. In: Proceedings of the 31th International Conference on Machine Learning. p. 433441. Beijing, China (2014)
20 Rodrigues, F., Loureno, M., Ribeiro, B., Pereira:, F.C.: Learning supervised topic models for classification and regression from crowds. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(12), 2409–2422 (2017)
21 S., A., C., B., F., A., V., B., S., D., N., N.: Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. IEEE transactions on medical imaging **35**(5), 13131321 (2016)
22 Sheng, V.S., Zhang, J., Gu, B., Wu, X.: Majority voting and pairing with multiple noisy labeling. IEEE Transactions on Knowledge and Data Engineering **31**(7), 1355–1368 (2019)
23 Simpson, E., Roberts, S., Psorakis, I., Smith, A.: Dynamic bayesian combination of multiple imperfect classifiers. Decision making and imperfection p. 135 (2013)
24 Snow, R., O'Connor, B., Jurafsky, D., Ng, A.: 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. pp. 254–263. Honolulu, Hawaii (2008)
25 Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D.C., Silberman, N.: Learning from noisy labels by regularized estimation of annotator confusion. In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition. pp. 11244–11253. Long Beach, CA (2019)
26 Tao, F., Jiang, L., Li, C.: Label similarity-based weighted soft majority voting and pairing for crowdsourcing. Knowledge and Information Systems **62**(7), 2521–2538 (2020)
27 Venanzi, M., Guiver, J., Kazai, G.and Kohli, P., Shokouhi, M.: Community-based bayesian aggregation models for crowdsourcing. In: Proceedings of the 23rd international conference on World wide web. p. 155164. Seoul, Republic of Korea (2014)
28 Weld, D., Lin, C., Bragg, J.: Artificial intelligence and collective intelligence. In: Malone, T., Bernstein, M. (eds.) The Collective Intelligence Handbook (2015)
29 Welinder, P., Branson, S., Belongie, S., Perona, P.: The multidimensional wisdom of crowds. In: Lafferty, J., Williams, C.I., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 2024–2432 (2010)
30 Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 2035–2043 (2009)
31 Zhang, H., Jiang, L., Xu, W.: Multiple noisy label distribution propagation for crowdsourcing. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. pp. 1473–1479. Macao, China (2019)
32 Zhou, D., Basu, S., Mao, Y., Platt, J.: Learning from the wisdom of crowds by minimax entropy. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 2195–2203 (2012)