International Journal of Software and Informatics, ISSN 1673-7288
http://www.ijsi.org, ijsi@iscas.ac.cn, +86-10-62661048
IJSI, 2022, 12(2): 213–230, doi: 10.21655/ijsi.1673-7288.00270

International Journal of Software and Informatics

Research Article

Deep Generative Crowdsourcing Learning with Worker Correlation Utilization

Shaoyuan Li (李绍园), Menglong Wei (韦梦龙), Shengjun Huang (黄圣君)

(College of Computer Science and Technology/College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)
Corresponding author: Shaoyuan Li, lisy@nuaa.edu.cn

Abstract Traditional supervised learning requires the groundtruth labels for the training data, which can be difficult to collect in many cases. In contrast, crowdsourcing learning collects noisy annotations from multiple non-expert workers and infers the latent true labels through some aggregation approach. In this paper, we notice that existing deep crowdsourcing work does not sufficiently model worker correlations, which is, however, shown to be helpful for learning by previous non-deep learning approaches. We propose a deep generative crowdsourcing learning approach to incorporate the strengths of Deep Neural Networks (DNNs) and exploit worker correlations. The model comprises a DNN classifier as a prior and an annotation generation process. A mixture model of workers' capabilities within each class is introduced into the annotation generation process for worker correlation modeling. For adaptive trade-off between model complexity and data fitting, we implement fully Bayesian inference. Based on the natural-gradient stochastic variational inference techniques developed for the Structured Variational AutoEncoder (SVAE), we combine variational message passing for conjugate parameters and stochastic gradient descent for DNN parameters into a unified framework for efficient end-to-end optimization. Experimental results on 22 real crowdsourcing datasets demonstrate the effectiveness of the proposed approach.

Keywords crowdsourcing learning; deep generative model; worker correlation; Bayesian; natural-gradient stochastic variational inference

Citation Li SY, Wei ML, Huang SJ. Deep generative crowdsourcing learning with worker correlation utilization, *International Journal of Software and Informatics*, 2022, 12(2): 213–230. http://www.ijsi.org/1673-7288/270.htm

As one of the most studied and widely used learning paradigms in machine learning, supervised learning requires training samples and their labels. However, in many real-world tasks, collecting true labels is not easy. On the one hand, true labels need be annotated by domain experts, who are usually expensive and limited resources; on the other hand, when the sample size is large, annotation becomes time-consuming and labor-intensive. Since the emergence of crowdsourcing platforms such as Amazon Mechanical Turk (AMT) and Crowdsflower, crowdsourcing provides an easier approach to collecting labels. By assigning the annotating

This is the English version of the Chinese article "利用标注者相关性的深度生成式众包学习. 软件学报, 2022, 33(4): 1274–1286. doi: 10.13328/j.cnki.jos.006479"
Funding items: National Natural Science Foundation of China (61906089); Jiangsu Province Basic Research Program (BK20190408); China Postdoc Science Foundation (2019TQ0152)
Received 2021-05-31; Revised 2021-07-16; Accepted 2021-08-27; IJSI published online 2022-06-25

tasks to non-expert workers[1] that are easily accessible from the Internet, crowdsourcing can quickly collect a large amount of supervised information. It is therefore widely used in fields such as natural language processing[2], medical diagnosis[3], image recognition[4], and named entity recognition[5].

Since the annotations are provided by non-expert workers, crowdsourcing usually assigns the task to multiple workers to reduce data errors. It then couples multiple annotations to estimate the true labels of a sample. Depending on whether Deep Neural Network (DNN) classifier models are used, existing crowdsourcing methods can be classified into non-deep crowdsourcing learning and deep crowdsourcing learning. The majority voting method, a common method that uses annotated information, takes the most frequently annotated class as the true label for each sample. Although this method is simple, fast, and easy to implement, it ignores the differences in the abilities of different workers. By considering true labels as unknown latent variables, the probabilistic graphical model-based approach uses different parameters to characterize the annotation capabilities of different workers. As an early representative work in this regard, Dawid and Skene[6] proposed the Dawid-Skene (DS) model that portrayed worker capabilities with classification accuracy. This model iteratively estimated true labels and worker accuracy by Expectation Maximization (EM) optimization to solve the problem of estimating more reliable conclusions from the diagnoses of multiple non-expert workers (medical students). Since then, many works have improved and extended the DS model from the perspectives of worker capability portrayal, sample difficulty portrayal, optimization implementation, and annotation correlation modeling and produced favorable results[7–12].

To exploit sample features, Raykar *et al*.[3] introduced a logistic regression classifier into the DS model as the true label prior and solved the classifier and worker parameters iteratively by EM. This idea was then extended to other types of classifier priors, such as the Gaussian process classifier[13]. As deep learning achieves significant progress in various fields such as visual speech[14], deep crowdsourcing learning that incorporates DNNs (to assimilate their representational learning advantages) has become a research trend in crowdsourcing[15–17]. Since EM-style optimization requires optimizing the classifier in each iteration, its computation complexity is undoubtedly high when the neural network model is complex[15], which renders efficient optimization a concern for deep crowdsourcing learning. Considering the multi-layer structure of neural networks, an approach to solve this problem is to add a layer of coefficients behind the output layer of the neural network classifier to portray worker capability. Thus, the final output layer of the classifier corresponds to the crowdsourcing annotation prediction[16, 17]. The optimization can be performed in an end-to-end manner, i.e., stochastic gradient descent of all parameters in the network is conducted on the loss of crowdsourcing annotation predictions and the classifier and worker capabilities are estimated simultaneously. Although this implementation avoids the computation complexity of iterative optimization, it loses the interpretable structured representation property of the probabilistic graphical model and fails to guarantee the maximization of the annotation likelihood or its lower bound.

In this study, crowdsourcing learning is conducted with a deep generative model, which retains both the representational learning advantages of DNNs and the structured representation of the probabilistic graphical model. Most of the works in this regard are based on the variational autoencoder model and its improvements[18–20]. Works on the same topic as this study are still relatively scarce[21–25] and ignore the use of correlations among workers. Nevertheless, available works on non-deep crowdsourcing[10–12] show that worker correlation modeling and utilization can help to improve the crowdsourcing learning effect. This paper proposes a deep generative crowdsourcing learning approach with worker correlation utilization, in which the subclass mixture model in Ref. [12] is extended to portray worker correlations and a

DNN is utilized as a classifier. This model uses confusion matrices to describe worker capabilities and the generation process of crowdsourcing annotations for each subclass, and its subclass mixture model shares the latent variables for true labels with the neural network classifier and the annotation generation process. To fit the parameters adaptively and thus avoid manual parameter selection, we implement a fully Bayesian model to describe the model parameters with probability distributions. To implement Bayesian inference, this paper, resorting to the optimization techniques for the Structured Variational AutoEncoder (SVAE), combines variational message passing for conjugate parameters with natural-gradient variational inference and efficiently updates all parameters in an end-to-end fashion[20]. The overall optimization process guarantees that the lower bound of the crowdsourcing annotation likelihood is maximized. Experimental results on 22 real crowdsourcing classification datasets with those of several methods compared show that the combination of deep representational learning with worker correlation can effectively improve the crowdsourcing learning effect. The contributions of this paper are summarized as follows:

(1) the deep generative crowdsourcing learning approach with worker correlation utilization is proposed for the first time, and it combines representational learning advantages with interpretability;

(2) efficient end-to-end natural-gradient stochastic variational inference is achieved, with a time complexity comparable to that of training DNN classifiers with true labels;

(3) the experimental results on a large number of real datasets validate the effectiveness of the proposed approach.

The rest of this paper is organized as follows: Section 1 outlines the related works; Section 2 provides a formal description of the crowdsourcing learning problem and the related background; Sections 3 and 4 present the proposed approach and the optimization implementation respectively; Section 5 illustrates the experimental results, and the last section concludes the whole paper.

1 Related Work

Due to the possible errors in crowdsourcing annotation, estimating true labels is a research focus in crowdsourcing learning. As a straightforward approach, the majority voting method uses the most frequently annotated classes as the true labels. Since it assumes that such annotations are equally correct, it does not work well when a lot of errors occur. By treating true labels as unknown latent variables and modeling the annotation generation process, the probabilistic graphical model provides an alternative way to study the problem. As an early representative work in this regard, the DS model[6]used accuracy to portray individual worker capability and iteratively estimated worker accuracy and true labels by the EM method with the goal of maximizing annotation likelihood. Many subsequent works improved and extended the DS model; for example, Ref. [7] proposed variational inference including belief propagation and a mean-field model from the optimization perspective; Ref. [8] introduced a sample difficulty parameter so that the annotation quality can be related to both workers and samples; Ref. [9] used a confusion matrix parameter to characterize the annotation quality of the workers for each sample and estimated true labels of the samples and parameters according to the min/max entropy principle. Refs. [10, 26, 27] extended the DS model from a Bayesian perspective by introducing a Dirichlet prior for the accuracy parameter and implementing Bayesian inference through Gibbs sampling, variational inference, and EM, respectively, to avoid manual parameter selection. Recently, the modeling and exploitation of correlations among workers have attracted researchers considering that correlations often exist among annotations in crowdsourcing problems. Ref. [10] described the dependency between any two annotations with an undirected

Markov network; Ref. [11] theoretically analyzed the min/max probability of error of the confusion matrix-based crowdsourcing model in light of the assumption of worker clustering; Ref. [12] reflected the tensor structure of crowdsourcing annotations by building a subclass mixture model for real classes to describe the correlations among the workers.

To facilitate crowdsourcing learning with sample features, Ref. [3] proposed using a logistic regression classifier from features to true labels as a prior assumption for true labels in DS models. This idea was subsequently extended to other types of classifier models such as the Gaussian process classifier[13]. As deep learning booms, deep crowdsourcing learning using DNNs as classifier models has become a research trend in the crowdsourcing field[15–17]. In Ref. [15], a convolutional neural network was used as a classifier, and iterative optimization was performed through EM to solve the optimal neural network classifier under the current worker parameters in each iteration. To avoid the computation overhead of the EM algorithm, Refs. [16, 17] added a layer of coefficients behind the classifier output as the worker capability parameter in light of the structure of neural networks. Thus, the classifier parameters and the worker capability parameters could be considered as parameters at different layers of the network and further updated in an end-to-end fashion by stochastic gradient descent. Although Refs. [16, 17] avoided the high computation complexity of EM optimization, their DNNs not only lacked the interpretable structure of the probabilistic graphical model but also failed to guarantee that the annotation likelihood or its lower bound was maximized.

In this paper, we propose the deep generative crowdsourcing learning approach by drawing on the development of deep generative models and their optimization techniques, mainly the variational autoencoder model[18] and SVAE[20]. The variational autoencoder[18] is a representative deep generative model that uses neural networks to learn the latent space representation of samples and reconstructs the original space of the samples from the latent space. By describing the latent space with probability distributions, the variational autoencoder can randomly sample and reconstruct the data in the latent space to generate new samples. It thus has wide applicability in data generation and is considered an important research method in the field of non-supervised learning. Resorting to the reparameterization technique, the variational autoencoder fits the probability distribution parameters with neural networks and implements efficient end-to-end optimization of parameters through variational inference optimization, thereby providing an optimization framework for deep generative models. SVAE[20] designs and utilizes the conjugated structure of the probabilistic graphical model to represent distributions as exponential family distributions and fits the parameters of exponential family distributions with neural networks. This enables the autoencoder to utilize the fast Bayesian inference method based on the conjugate distribution structure in traditional generative models such as the topic model[28]. We note that Refs. [21–25] also proposed deep generative crowdsourcing learning approaches on the basis of variational autoencoders and their extended models. For example, Ref. [21] minimized the reconstruction errors in crowdsourcing annotation with a variational autoencoder[18]; and Refs. [22–24] proposed semi-supervised crowdsourcing classification and clustering learning methods using unlabeled data based on semi-supervised variational autoencoders[19]. Ref. [25] proposed a fully Bayesian deep generative crowdsourcing classification method through SVAE optimization[20]. Nevertheless, they were all based on the assumed conditional independence of the workers and did not consider worker correlation modeling. In this paper, we extend the subclass mixture model from Ref. [12] to a deep generative model that learns classifiers to implement efficient Bayesian parameter inference by using the optimization techniques developed for SVAE[20].

The above related works mainly focus on the annotation aggregation for single-labeled classification tasks that are also the focus of this paper. The extensive needs in other fields which

have given rise to numerous crowdsourcing-related research will not be elaborated here, such as multi-labeled crowdsourcing learning[29], interactive feature selection based on crowdsourcing learning[30], and research on trusted crowdsourcing mechanisms[31].

2 Problem Formalization and Related Background

2.1 Problem formalization

In this paper, the set of N samples is expressed as $X = \{x_1, \cdots, x_N\}$, where $x_i \in R^d$ refers to the ith sample and d refers to the spatial dimension of the sample features. When the number of workers is W, the crowdsourcing annotation result on the sample set X can be expressed as $L \in \{0, 1, \cdots, K\}^{N \times W}$, where K refers to the number of classes and L_{ij} refers to the annotation of the jth worker on the sample x_i. $L_{ij} = k$, for $k \neq 0$, indicates that the jth worker has annotated x_i as belonging to the kth class, whereas, $L_{ij} = 0$ represents that the sample x_i has not been annotated by the jth worker, i.e., the annotation is absent. In crowdsourcing data, each worker usually annotates part of the sample data, so annotations are often absent. The goal of crowdsourcing learning is to estimate the true class $Y = \{y_1, \cdots, y_N\}$ of the sample X from the annotations L.

Next, we outline the classical crowdsourcing learning model based on the assumption of worker independence and then present our model.

2.2 Worker independence model

Fig. 1 shows the classical crowdsourcing generation process based on the assumption of worker independence. For descriptive convenience and consistency with the Bayesian framework considered in this paper, the independent Bayesian Classifier Combination (iBCC) model in Ref. [10] is used here as an example.

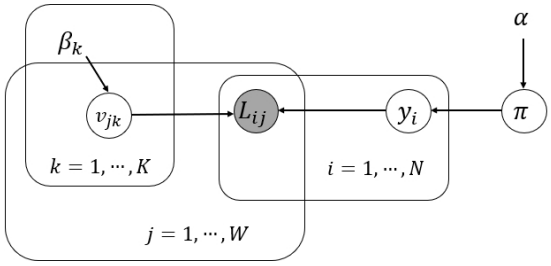


Figure 1 Probabilistic graphical representation of the iBCC model

Giving no consideration to sample feature utilization, this model consists of two parts: annotation generation process $p(L_{ij}|y_i, V_j)$ and true label prior $p(y_i|\pi)$. $V_j = \{\nu_{jk}\}_{k=1}^{K}$ indicates the parameter of the annotation process corresponding to the j^{th} worker, $\nu_{jk} = [\nu_{jk1}, \cdots, \nu_{jkK}] \in [0, 1]^{K \times 1}$ represents the probability that the sample is annotated as belonging to each class when the true label is $y_i = k$, and π is the prior distribution parameter of the true label. β_k and α correspond to the distribution parameters of V_{jk} and π respectively. This model assumes that the workers are conditionally independent, i.e., when the sample x_i and its true class y_i are given, the crowdsourcing annotations are independent of each other. Taking $y_i = k$ as an example, we have

$$p(L_{i1}, \cdots, L_{iW}|y_i = k, \{V_j\}_{j=1}^{W}) = \prod_{j=1}^{W} \mathcal{I}(L_{ij} \neq 0) p(L_{ij}|y_i = k, \nu_{jk}) \tag{1}$$

where $\mathcal{I}(\cdot)$ is the indicator function that is set to 1 when the condition in parentheses is satisfied and to 0 when it is not. Assuming that the samples are independent, the joint distribution of the crowdsourcing annotation L, true label Y, and parameters $V = \{\nu_{jk}\}$ and π can be expressed as

$$\begin{aligned} p(L, Y, V, \pi) &= p(L|Y, V)p(Y|\pi)p(\pi)p(V) \\ &= p(\pi) \cdot \prod_{i=1}^{N} p(y_i|\pi) \cdot \prod_{j=1}^{W} \mathcal{I}(L_{ij} \neq 0) p(L_{ij}|y_i, V_j) \cdot \prod_{k=1}^{K} p(\nu_{jk}) \end{aligned} \tag{2}$$

The model in Fig. 1 can be regarded as a Bayesian extension of the DS model[6]. In contrast to the DS model that solves the point estimation for the parameters V and π, this Bayesian model estimates the posterior distributions of the parameters V and π, which enables it to describe more uncertainties. For the model in Fig. 1, existing works proposed parameter estimation methods based on inference techniques such as Gibbs sampling[10], Bayesian variational mean-field inference[26], and EM[27].

Most existing crowdsourcing learning algorithms are based on the conditional independence assumption of the workers. To characterize worker correlations, Ref. [10] proposed using the undirected Markov network to describe the dependence between any two annotations; however, it cannot handle the case with absent sample annotations. Ref. [12] proposed building a subclass mixture model for true classes and assumed that the workers were conditionally independent on a given subclass to describe the worker correlations. In this paper, we extend this subclass mixture model to deep generative model learning classifiers using sample features. The specific implementation process is presented in the following section.

3 Deep Generative Crowdsourcing Learning Approach with Worker Correlation Utilization

Fig. 2 shows the structure of the proposed method in the form of a probabilistic graphical model. Similarly to the model shown in Fig. 1, this model also consists of two main parts: annotation generation process $p(L_{ij}|y_i, g_i, V_j)$ and true label prior $p(y_i|x_i, \gamma, \pi)$. The difference is that the true label prior in this model depends on sample characteristics. A DNN classifier with the parameter γ is used to characterize the dependency process; moreover, an additional latent variable $g_i \in \{1, \cdots, M\}$ is introduced into the annotation generation process to characterize worker correlation. The number M refers to the value range of the latent variable g_i. The parameter $\nu_{jkm} \in [0, 1]^{K \times 1}$ corresponds to the probability that the sample is annotated as belonging to each class by the jth worker when the latent variables are $\{y_i = k, g_i = m\}$. We set $V_j = \{\nu_{jkm}\}_{k,m}$, where $k = 1, \cdots, K$ and $m = 1, \cdots, M$. The specific meaning of each part of the model is given below.

3.1 Annotation generation process

Unlike Eq. (1), which assumes that the workers are independent of each other when true labels are given, Ref. [12] portrays each class as a mixture model with M subclasses, i.e., the dataset is divided into a total of $K \times M$ subclasses, and the workers are conditionally independent in each subclass. By taking class k as an example, its subclass distribution $p(g|y = k)$ is represented by a discrete probability distribution on the class M with the parameter $\tau_k \in [0, 1]^{M \times 1}$:

$$p(g|y = k) \triangleq p(g|y = k, \tau_k) = Categorical(g|\tau_k) \tag{3}$$

For each sample x_i, two latent variables $y_i \in \{1, \cdots, K\}$ and $g_i \in \{1, \cdots, M\}$ are used to represent the true class y_i of the sample and one of its subclasses g_i respectively. With

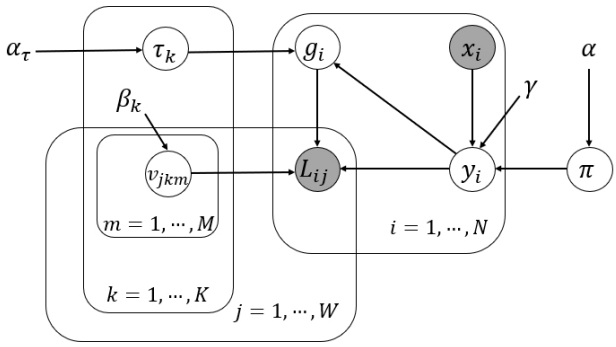


Figure 2 Probabilistic graphical representation of the proposed model

$y_i = k, g_i = m$ as an example, if the parameter $\nu_{jkm} \in [0,1]^{K\times 1}$ refers to the probability that the sample is annotated as belonging to each class by the jth worker, the probability of the annotation L_{ij} can be obtained by Eq. (4).

$$p(L_{ij} = l|y_i = k, g_i = m, V_j) \triangleq p(L_{ij} = l|y_i = k, g_i = m, \nu_{jkm}) = \nu_{jkml} \tag{4}$$

Then, the joint distribution of multiple annotations for sample x_i can be expressed as

$$p(L_{i1}, \cdots, L_{iW}|y_i = k, g_i = m, \{V_j\}_{j=1}^W) = \prod_{j=1}^W \mathcal{I}(L_{ij} \neq 0) p(L_{ij}|y_i = k, g_i = m, \nu_{jkm}) \tag{5}$$

Eq. (3) and Eq. (5) are combined to obtain Eq. (6):

$$\begin{aligned} &p(L_{i1}, \cdots, L_{iW}|y_i = k, \{V_j\}_{j=1}^W) \\ &= \sum_{m=1}^M p(g_i = m|y_i = k) p(L_{i1}, \cdots, L_{iW}|y_i = k, g_i = m, \nu_{jkm}) \end{aligned} \tag{6}$$

By comparing Eq. (6) with Eq. (1) of the worker independence model in the previous section, we can see that the crowdsourcing annotations are no longer independent of each other when the true labels of the sample are given. Assuming that the samples are independent of each other, the generation process of annotation L can be expressed as

$$P(L|Y, G, V) = \prod_{i=1}^N \prod_{j=1}^W \mathcal{I}(L_{ij} \neq 0) p(L_{ij}|y_i, g_i, V_j) \tag{7}$$

3.2 True label prior

For the prior model $p(y_i|x_i, \gamma, \pi)$ of true labels, $p(y_i|\pi)$ and $p(y_i|x_i, \gamma)$ are used to represent the feature-independent prior and the feature-related prior, respectively, and the DNN classifier $f(\cdot)$ with the parameter γ is used to implement $p(y_i|x_i, \gamma)$. The two priors are defined as below:

$$p(y_i|\pi) = Categorical(y_i|\pi), \quad p(y_i|x_i, \gamma) = Categorical(f(x_i; \gamma)) \tag{8}$$

Assuming that the samples are independent of each other, the prior with regard to Y can be expressed as

$$p(Y|X, \gamma, \pi) = p(Y|X, \gamma) p(Y|\pi) = \prod_{i=1}^N p(y_i|\pi) p(y_i|x_i, \gamma) \tag{9}$$

3.3 Parameter conjugate prior

In addition to the above annotation generation process and true label prior, this paper assumes that the parameters ν_{jkm}, π, and τ_k obey the conjugate Dirichlet prior distribution as defined below:

$$p(\nu_{jkm}) = Dir(\nu_{jkm}|\beta_k), p(\pi) = Dir(\pi|\alpha), p(\tau_k) = Dir(\tau_k|\alpha_\tau) \tag{10}$$

3.4 Joint distribution

According to the above definition, the joint distribution of the crowdsourcing annotation L, latent variables Y and G, and parameter $\Theta = \{V = \{\nu_{jkm}\}, \pi, \tau = \{\tau_k\}\}$ can be expressed as

$$\begin{aligned} p(L, Y, G, \Theta|X, \gamma) &= p(L, Y, G, V, \pi, \tau|X, \gamma) \\ &= p(L|Y, G, V)p(Y|\pi)p(Y|X, \gamma)p(G|Y, \tau)p(V)p(\pi)p(\tau) \end{aligned} \tag{11}$$

3.5 Learning objective

For the model expressed by Eq. (11), the objective of this paper is to estimate the posterior distributions $p(Y|L, X)$, $p(G|L, X)$, and $p(\Theta|L, X)$ of true label Y of the sample, subclass G, and parameter $\Theta = \{V, \pi, \tau\}$ by maximizing the crowdsourcing annotation likelihood $p(L)$.

When a DNN classifier is not used as a true label prior, the model in this paper is equivalent to the non-deep model with worker correlation utilization in Ref. [12], and it can directly use the optimization methods for the model in Fig. 1, which assumes the workers are independent. For example, the Bayesian variational mean-field message passing method[26] was employed for fast inference in Ref. [12]. However, iterative optimization with Gibbs sampling[10] and EM[27]are inefficient when a neural network classifier $p(Y|X, \gamma)$ is introduced. On the other hand, the variational mean-field message passing method[26] requires the annotation likelihood $p(L)$ to conform to the conjugate exponential family distribution, which is not satisfied by the nonlinear neural network classifier prior. In the next section, we propose an optimization algorithm with stochastic variational inference for the proposed model by drawing on the optimization progress of SVAE[20], such that to implement efficient end-to-end optimization of the parameters in the model, and guarantee that the variational lower bound of the logarithmic annotation likelihood is maximized.

4 Natural-gradient Stochastic Variational Inference

As a representative deep generative model, the variational autoencoder[18], by utilizing the reparameterization technique, fits the parameters of a probability distribution with neural networks and performs end-to-end gradient descent. Ref. [20] extended it to the SVAE in the Bayesian framework by designing and exploiting the conjugated structure of the probabilistic graphical model, so that it can use the fast Bayesian inference method based on the conjugated distribution structure as traditional generative models such as the topic model[28]. Specifically, Ref. [20] expressed distributions as exponential family distributions in light of the natural-gradient Stochastic Variational Inference (SVI) framework in Ref. [28]. Parameters for the exponential family distributions of true labels are fitted with a neural network, and mean-field message passing and natural gradient calculation are performed, which achieve efficient second-order optimization. Drawing on Ref. [20], this paper proposes an optimization method for the deep generative crowdsourcing model with labeling correlation utilization. The implementation details are as follows.

Assume that the posterior distributions of true label Y, subclass G, and parameter $\Theta = \{V, \pi, \tau\}$ obey the variational mean-field distribution, i.e., $q(Y, G, \Theta) = q(Y)q(G)q(\Theta)$.

Similarly to the case of the variational autoencoder, the variational Evidence Lower BOund (ELBO) of the logarithmic annotation likelihood $\log p(L)$ for Eq. (11) can be expressed as

$$\log p(L) \geq \mathcal{L}(Y, G, \Theta, \gamma) \triangleq E_{q(Y,G,\Theta)}\left[\log \frac{p(L, Y, G, \Theta|X, \gamma)}{q(Y)q(G)q(\Theta)}\right] \tag{12}$$

To utilize the conjugated structure of the distribution, we use η to represent natural parameters in exponential family distribution, $t(\,\cdot\,)$ to denote the sufficient statistics, and $\log Z(\,\cdot\,)$ to indicate the logarithmic partition functions. In this paper, the equations $p(\nu_{jkm}, p(\pi), p(tau_k), p(Y|\pi)$, and $p(g|y = k, \tau_k)$ defined in Eqs. (10), (8), and (3) are rewritten as exponential family distributions:

$$p(\nu_{jkm}) = \exp\{\langle \eta_{\nu_{jkm}}, t(\nu_{jkm})\rangle - \log Z(\eta_{\nu_{jkm}})\} \tag{13}$$

$$p(\pi) = \exp\{\langle \eta_{\pi}, t(\pi)\rangle - \log Z(\eta_{\pi})\} \tag{14}$$

$$p(\tau_k) = \exp\{\langle \eta_{\tau_k}, t(\tau_k)\rangle - \log Z(\eta_{\tau_k})\} \tag{15}$$

$$p(y|\pi) = \exp\{\langle \eta_y(\pi), t(y)\rangle - \log Z(\eta_y(\pi))\} = \exp\{\langle t(\pi), (t(y), 1)\rangle\} \tag{16}$$

$$p(g|y = k, \tau_k) = \exp\{\langle \eta_g(\tau_k), t(g)\rangle - \log Z(\eta_g(\tau_k))\} = \exp\{\langle t(\tau_k), (t(g), 1)\rangle\} \tag{17}$$

For Eqs. (13)–(17), the specific values of $\eta, t(\cdot), \log Z(\cdot)$ are

$$\eta_{\nu_{jkm}} = \begin{bmatrix} \beta_{k1} - 1 \\ \vdots \\ \beta_{kK} - 1 \end{bmatrix}, \eta_{\pi} = \begin{bmatrix} \alpha_1 - 1 \\ \vdots \\ \alpha_K - 1 \end{bmatrix}, \eta_{\tau_k} = \begin{bmatrix} \alpha_{\tau 1} - 1 \\ \vdots \\ \alpha_{\tau M} - 1 \end{bmatrix},$$

$$\eta_y(\pi) = \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_K \end{bmatrix}, \eta_g(\tau_k) = \begin{bmatrix} \log \tau_{k1} \\ \vdots \\ \log \tau_{kM} \end{bmatrix} \tag{18}$$

$$t(\nu_{jkm}) = \begin{bmatrix} \log \nu_{jkm1} \\ \vdots \\ \log \nu_{jkmK} \end{bmatrix}, t(\pi) = \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_K \end{bmatrix}, t(\tau_k) = \begin{bmatrix} \log \tau_{k1} \\ \vdots \\ \log \tau_{kM} \end{bmatrix},$$

$$t(y) = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix}, t(g) = \begin{bmatrix} g_1 \\ \vdots \\ g_M \end{bmatrix} \tag{19}$$

$$\log Z(\eta_{\nu_{jkm}}) = \sum_{l=1}^{K} \log \Gamma(\beta_{kl}) - \log \Gamma\left(\sum_{k=1}^{K} \beta_{kl}\right),$$

$$\log Z(\eta_{\pi}) = \sum_{k=1}^{K} \log \Gamma(\alpha_k) - \log \Gamma\left(\sum_{k=1}^{K} \alpha_k\right) \tag{20}$$

$$\log Z(\eta_{\tau_k}) = \sum_{m=1}^{M} \log \Gamma(\alpha_{\tau m}) - \log \Gamma\left(\sum_{m=1}^{M} \alpha_{\tau m}\right),$$

$$\log Z(\eta_y(\pi)) = 0, \log Z(\eta_g(\tau_k)) = 0 \tag{21}$$

where $\Gamma(\cdot)$ represents the Gamma functions. Similarly, the variational posterior distributions $q(Y)$, $q(G)$, and $q(\Theta)$ can also be written as exponential family distributions:

$$q(\theta) = \exp\{\langle \eta_{\theta}, t(\theta)\rangle - \log Z(\theta)\},\ \theta \in Y \cup G \cup \Theta \tag{22}$$

Through the above expressions of exponential family distributions, the variational ELBO given in Eq. (12) can be rewritten as an objective with respect to the natural parameters:

$$\mathcal{L}(\eta_Y, \eta_G, \eta_\Theta, \gamma) \triangleq E_{q(Y,G,\Theta)}\left[\log \frac{p(L, Y, G, \Theta|X, \gamma)}{q(Y)q(G)q(\Theta)}\right] \tag{23}$$

To exploit the conjugated nature of the model, similarly to the case of SVAE[20], the potential function of the prior $p(y_i|x_i, \gamma)$ under a conjugate model is constructed with the output of DNNs $\gamma(\cdot)$:

$$\psi(y_i|x_i, \gamma) \triangleq \langle \gamma(x_i), t(y_i)\rangle \tag{24}$$

The variational lower bound is obtained using the potential function $\psi(y_i|x_i, \gamma)$ to substitute $p(y_i|x_i, \gamma)$:

$$\hat{\mathcal{L}}(\eta_Y, \eta_G, \eta_\Theta, \gamma) \triangleq E_{q(Y,G,\Theta)}\left[\log \frac{p(L, Y, G, \Theta)\exp\{\psi(Y|X, \gamma)\}}{q(Y)q(G)q(\Theta)}\right] \tag{25}$$

The distributions in the variational ELBO $\hat{\mathcal{L}}$ are now exponential family distributions with conjugated structures, and their parameters are the natural parameters η_Y, η_G, and η_Θ in the exponential family distributions and the neural network parameter γ. The natural-gradient SVI in Ref. [28] can then be used to solve the parameters as follows.

(1) Solution of η_Y with other variables given

When the other variables are given, the optimal solution $q^*(Y)$ for the latent variable Y is independent of the samples, i.e., $q^*(Y) = \prod_{i=1}^N q^*(y_i)$, and the distribution $q^*(y_i)$ of each sample has a closed-form solution as follows:

$$\begin{aligned}\log q^*(y_i) = {} & E_{q(\pi)} \log p(y_i|\pi) + \langle \gamma(x_i), t(y_i)\rangle \\ & + E_{q(g_i)q(V)} \log p(\{L_{ij}\}_{j\in W_i}|y_i, g_i, V) + const\end{aligned} \tag{26}$$

$$\eta^*_{y_i} = E_{q(\pi)}t(\pi) + \gamma(x_i) + \sum_{j=1}^{W} \mathcal{I}(L_{ij} \neq 0)[E_{q(\nu_{jL_{ij}})}t(\nu_{jL_{ij}})] * [E_{q(g_i)}t(g_i)] \tag{27}$$

where the L_{ij} in $\nu_{jL_{ij}}$ is a subscript. With $L_{ij} = k$ as an example, ν_{jk} refers to the matrix in which ν_{jkm} is the column, specifically $\nu_{jk} = [\nu_{jk1}, \cdots, \nu_{jkM}] \in [0, 1]^{K\times M}$. $E_{q(\nu_{jL_{ij}})}$ and $t(\nu_{jL_{ij}})$ refer to operations on each column in the matrix $\nu_{jL_{ij}}$, and * refers to matrix multiplication.

(2) Solution of $\boldsymbol{\eta}_G$ with other variables given

When the other variables are given, the optimal solution $q^*(G)$ for the latent variable G is independent of the samples, i.e., $q^*(G) = \prod_{i=1}^N q^*(g_i)$, and the distribution $q^*(g_i)$ of each sample has the following closed-form solution:

$$\log q^*(g_i) = E_{q(y_i)q(\tau_{y_i})} \log p(g_i|y_i, \tau_{y_i}) + E_{q(y_i)q(V)} \log p(\{L_{ij}\}_{j\in W_i}|y_i, g_i, V) + const \tag{28}$$

$$\begin{aligned}\eta^*_{g_i} = {} & [E_{q(\tau)}t(\tau)] * [E_{q(y_i)}t(y_i)] \\ & + \sum_{j=1}^{W} \mathcal{I}(L_{ij} \neq 0)[E_{q(\nu_{jL_{ij}})}t(\nu_{jL_{ij}})]^{\mathrm{T}} * [E_{q(y_i)}t(y_i)]\end{aligned} \tag{29}$$

Here, τ refers to a matrix with τ_k as the column in the form $\tau = [\tau_1, \cdots, \tau_K] \in [0,1]^{M\times K}$; $E_{q(\tau)}$ and $t(\tau)$ refer to the operations on each column in the matrix τ. T stands for the matrix transpose, and * is the matrix multiplication.

(3) Solution of $\boldsymbol{\eta_\Theta}$ and γ with $\boldsymbol{\eta_Y}$ and $\boldsymbol{\eta_G}$ given

η_Y^* and η_G^* are substituted into Equation (23) to obtain the following optimization objective with regard to η_Θ and γ:

$$\mathcal{J}(\eta_\Theta, \gamma) \triangleq \mathcal{L}(\eta_Y^*, \eta_G^*, \eta_\Theta, \gamma) \tag{30}$$

In terms of Eq. (30), Ref. [20] proves that $\mathcal{J}(\eta_\Theta, \gamma)$ is the optimal lower bound regarding Eq. (23), i.e.,

$$\max_{\eta_Y, \eta_G} \mathcal{L}(\eta_Y, \eta_G, \eta_\Theta, \gamma) \geq \mathcal{J}(\eta_\Theta, \gamma) \tag{31}$$

According to Ref. [20], the gradient of $\mathcal{J}(\cdot)$ with respect to η_Θ can be derived as follows:

$$\tilde{\nabla}_{\eta_\Theta}\mathcal{J} = \left[\eta_\Theta^0 + E_{q^*(Y)q^*(G)}(t(Y,G,X,L),1) - \eta_\Theta\right] + \nabla_{\eta_Y,\eta_G}\left(\mathcal{L}\left(\eta_Y^*, \eta_G^*, \eta_\Theta, \gamma\right), 0\right) \tag{32}$$

Here, η_Θ^0 refers to the natural parameter of the prior distribution for parameter Θ when the model is used. For the model in this paper, the following equations can be derived for the natural gradients of $\eta_{\nu_{jkm}}$, η_π, and η_{τ_k}:

$$\tilde{\nabla}_{\eta_{v_{jk}}}\mathcal{J} = \eta_{v_{jk}}^0 + \sum_{i=1}^{N}\mathcal{I}(L_{ij}\neq 0)\left[E_{q^*(y_i)}t(y_i)\otimes\overline{L_{ij}}\right]^*\left[E_{q^*(g_i)}t(g_i)\right]^{\mathrm{T}} - \eta_{v_{jk}} \tag{33}$$

$$\tilde{\nabla}_{\eta_\pi}\mathcal{J} = \eta_\pi^0 + \sum_{i=1}^{N}E_{q^*(y_i)}t(y_i) - \eta_\pi \tag{34}$$

$$\tilde{\nabla}_{\eta_\tau}\mathcal{J} = \eta_\tau^0 + \sum_{i=1}^{N}\left[E_{q^*(g_i)}t(g_i)\right] * \left[E_{q^*(y_i)}t(y_i)\right]^{\mathrm{T}} - \eta_\tau \tag{35}$$

Here, $\eta_{\nu_{jk}}$ refers to the matrix with $\eta_{\nu_{jkm}}$ as the column in the form $\eta_{\nu_{jk}} = [\eta_{\nu_{jk1}}, \cdots, \eta_{\nu_{jkM}}] \in [0,1]^{K\times M}$, η_τ is the matrix with η_{τ_k} as the column in the form $\eta_\tau = [\eta_{\tau_1}, \cdots, \eta_{\tau_K}] \in [0,1]^{M\times K}$. $\tilde{\nabla}_{\nu_{jk}}\mathcal{J}$ and $\tilde{\nabla}_{\eta_\tau}\mathcal{J}$ represent the derivative operation on each column of the matrixes $\eta_{\nu_{jk}}$ and η_τ respectively. $\overline{L_{ij}}$ refers to the one-hot encoding representation of L_{ij}. As for the neural network parameter γ, its gradient $\nabla_\gamma\mathcal{J}$ can be calculated by using existing DNN back propagation.

After the model training is completed, the probability of the true labels of the samples and the worker capability parameters can be obtained with the expectations $E_{q^*(y_i)}t(y_i)$ and $E_{q^*(\nu_{jkm})}t(\nu_{jkm})$ corresponding to sufficient statistics of posterior distributions:

$$E_{q^*(y_i)}t(y_i) = \begin{bmatrix}\pi_{y_{i1}}\\ \vdots \\ \pi_{y_{iK}}\end{bmatrix}, E_{q^*(\nu_{jkm})}t(\nu_{jkm}) = \begin{bmatrix}\varphi(\beta_{k1}^*)\\ \vdots \\ \varphi(\beta_{kK}^*)\end{bmatrix} - \varphi(\sum_{l=1}^{K}\beta_{kl}^*) \tag{36}$$

The overall implementation process is given in Algorithm 1. It can be seen that, compared with the ordinary neural network training process, the calculation of closed-form solutions $\eta_{y_i}^*$ and $\eta_{g_i}^*$ is added to each iteration besides the gradient update of the parameters η_Θ and γ. The overall computation complexity of the algorithm depends on the number of gradient updates and proves to be comparable to that of the ordinary neural network training process.

Algorithm 1. Deep generative crowdsourcing learning algorithm with worker correlation utilization

Input: Training samples $X = \{x_1, \cdots, x_N\}$, annotations $L \in \{0, 1, \cdots, K\}^{N \times W}$, and parameter Θ (corresponding to the natural parameter η_Θ^0)
Output: Predicted true label $Y = \{y_1, \cdots, y_N\}$
1. **Initialization:** Initialize the parameters $\eta_\Theta, \gamma, \eta_Y, \eta_G$
2. **repeat**
3. With fixed $\eta_\Theta, \gamma, \eta_G$, calculate the natural parameter $\eta_{y_i}^*$ of the posterior distribution of the true label for each sample x_i by Eq. (27)
4. With fixed $\eta_\Theta, \gamma, \eta_Y$, calculate the natural parameter $\eta_{g_i}^*$ of the posterior distribution of the subclasses for each sample x_i by Eq. (29)
5. With fixed $\eta_{y_i}^*, \eta_{g_i}^*$, calculate the natural gradient $\tilde{\nabla}_{\eta_\Theta} \mathcal{J}$ of η_Θ by Eqs. (33)–(35), calculate the gradient $\nabla_\gamma \mathcal{J}$ of neural network parameter γ by back propagation, and perform stochastic gradient ascent update for η_Θ, γ
6. **until** The variational lower bound $\mathcal{J}(\eta_\Theta, \gamma)$ converges or the maximum number of iterations is reached
7. **Prediction:** Obtain predicted true labels by Eq. (36)

5 Experiments

5.1 Experimental setup

5.1.1 *Experimental data*

In this paper, two multi-labeled crowdsourcing image datasets, i.e., `dataset1` and `dataset2`, collected by Ref. [32] are used. They contain 700 and 1,495 images corresponding to 6 and 16 classes respectively. The original data contain annotations from 18 and 15 workers respectively. Annotation accuracy is calculated for the sample subset corresponding to each worker, and the MacroF1 results are mainly distributed in [0.700, 0.800], indicating that most of the workers are reliable and thus using these two datasets to verify crowdsourcing learning is feasible. The experimental results in Ref. [32] show that most methods tend to be consistent when the number of workers reaches 10. Therefore, to improve experimental efficiency, we choose the 9 workers with the largest number of annotated samples and the original 1248-dimensional Fisher vector features. Single label crowdsourcing learning is performed independently on each class to obtain 22 datasets of binary classification tasks.

5.1.2 *Comparison methods*

In this paper, three groups of representative crowdsourcing learning methods are compared: (1) the Majority Voting (MV) method using annotated information, the DS model[6], and the MaxEn model based on the min/max entropy principle[9]; (2) the non-deep generative method Yutc[3] with a logistic regression classifier as its true label prior; (3) deep generative model BayesDGC[25] with no regard to worker correlations.

The proposed method is denoted as BayesDGC-w, and it is equivalent to BayesDGC when worker correlations are not considered, i.e., the number M of subclasses in the subclass mixture model of each class is 1. When the DNN classifier is not used as the true label prior, BayesDGC-w is equivalent to the EBCC method in Ref. [12] that utilizes annotated information. Therefore, as for the deep generative models BayesDGC and BayesDGC-w, this paper also compares their non-deep Bayesian variants, i.e., BayesGC and BayesGC-w that do not use sample features, to examine the respective effects of DNN classifiers and worker correlations on crowdsourcing learning.

The proposed method BayesDGC-w uses a perceptron with a single latent layer (the number of nodes is 100) as a DNN classifier. The number M of subclasses in the subclass mixture model is set to 3, and Dirichlet priors $Dir(\nu_{jkm}|\beta_k^0)$, $Dir(\pi|\alpha^0)$, and $Dir(\tau_k|\tau^0)$ are adopted for the

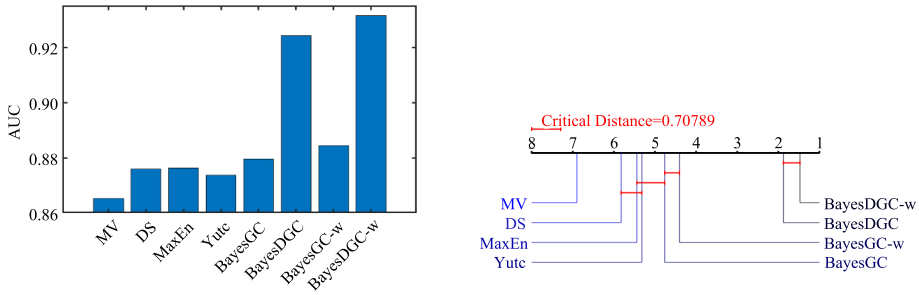
parameters ν_{jkm}, π, and τ_k respectively. As for the worker capability parameter ν_{jkm}, its prior parameter $\beta_k^0 \in R^K$ is set to $\beta_{kk}^0 = 5$, $\beta_{kk'}^0 = 2$, when $k \neq k'$, to reflect the superiority of the worker capabilities over random guesses. For $\alpha^0 \in R^K$ and $\tau^0 \in R^M$, the value of each element is 1.1. The optimization process is carried out using Adam optimizer[33], with learning rate of 0.001 and the number epochs is 400. The BayesDGC method is implemented under the same settings for the other parameters when $M = 1$. The BayesGC and BayesGC-w are implemented in the absence of a neural network classifier. Except for the DS model which uses the confusion matrix to characterize the worker capability, all the other methods under comparison use the parameter settings suggested in the original paper.

To test the effect of the size of crowdsourcing annotations on each learning method, the mean and standard deviation of 10 repetitions of the experiment are recorded by randomly retaining 10%–100% of the annotations at 10% intervals. Since the original data are based on multi-labeled tasks, the classes are severely unbalanced. For example, each image has 1.24 positive labels on average in the 6 classes corresponding to `dataset1`, while such a number is 1.80 for the images in the 16 classes corresponding to `dataset2`. The area under the receiver operating characteristic (ROC) curve (AUC) is used as the evaluation measure.

5.2 Experimental results

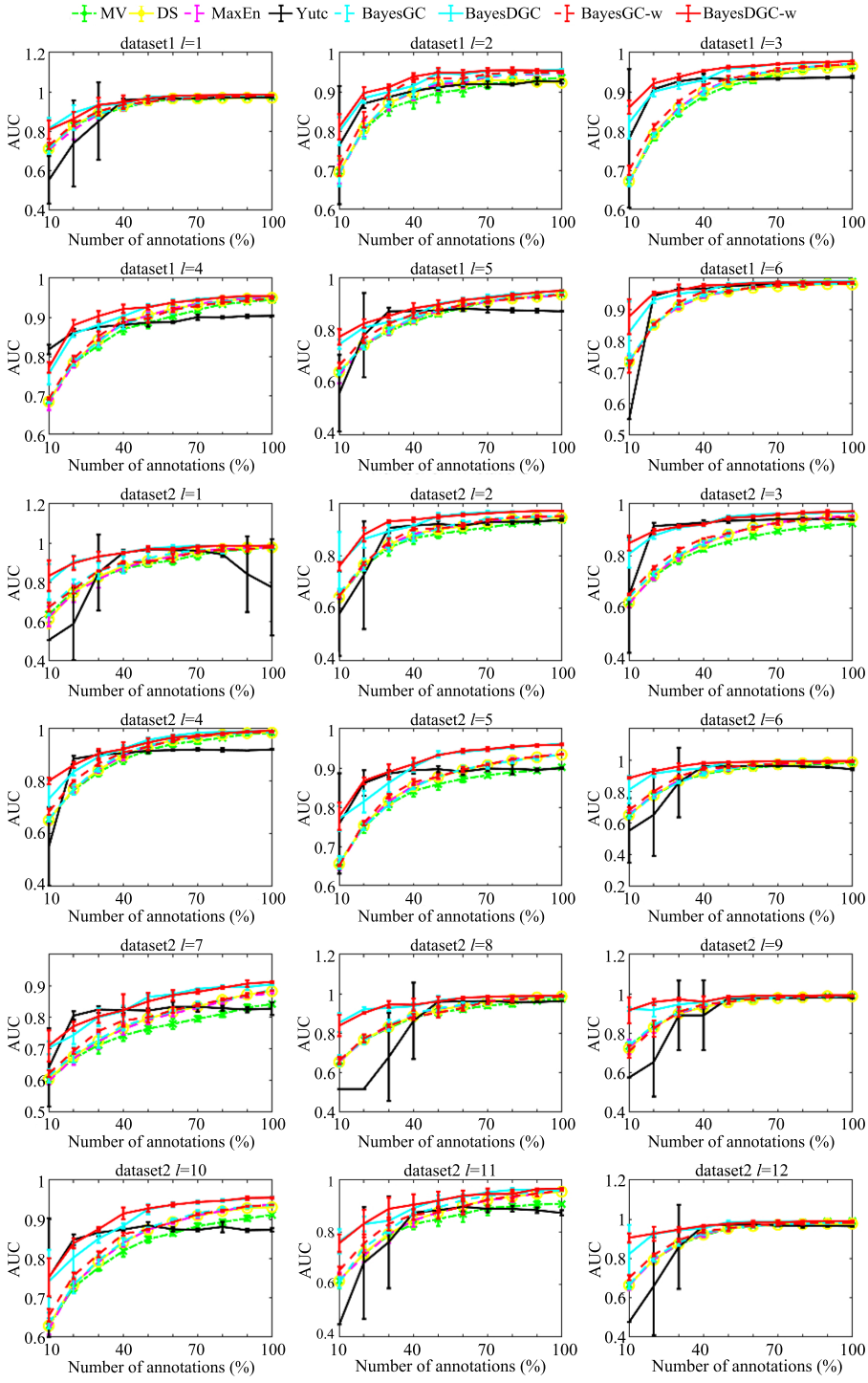
Fig. 3(a) shows the average AUC results of BayesDGC-w and the comparison methods under 10 labeling ratios (220 experiments) on 22 datasets; we can see that the MV method without considering annotation quality modeling is the least effective. The deep crowdsourcing methods BayesDGC-w and BayesDGC are far better than the non-deep crowdsourcing methods, and the average AUC of BayesDGC-w (BayesGC-w) that considers the utilization of labeling correlations is significantly better than that of BayesDGC (BayesGC) that does not consider the utilization of labeling correlations. Fig. 3(b) shows the results of the Nemenyi tests. The Nemenyi test is a common test to compare the overall performance of multiple methods on multiple datasets[34]. The numbers on the upper horizontal line in Fig. 3(b) indicate the average ranking of each method over 220 experiments. When the difference in the average ranking of two methods is greater than a Critical Difference (CD), it indicates that the two methods hold a significant statistic difference, otherwise they do not. CD depends on the number of methods compared, the number of experiments, and the significance p. By setting $p = 0.05$, we obtain CD $= 0.70789$ for our experiments. The red lines in the figure connect the algorithms whose difference in ranking is less than CD. It can be seen that the deep generative method BayesDGC-w (BayesDGC) significantly outperforms the non-deep methods and that BayesDGC-w with labeling relation utilization has effects comparable with those of BayesDGC.

The lth class corresponding to the two datasets in the binary classification task is expressed



(a) Mean AUC (b) Nemenyi test

Figure 3 Overall AUC effects of BayesDGC-w and 7 other comparison methods on 22 real datasets


MV
DS
MaxEn
Yutc
BayesGC
BayesDGC
BayesGC-w
BayesDGC-w
dataset1 l=1
dataset1 l=2
dataset1 l=3
dataset1 l=4
dataset1 l=5
dataset1 l=6
dataset2 l=1
dataset2 l=2
dataset2 l=3
dataset2 l=4
dataset2 l=5
dataset2 l=6
dataset2 l=7
dataset2 l=8
dataset2 l=9
dataset2 l=10
dataset2 l=11
dataset2 l=12
AUC
Number of annotations (%)

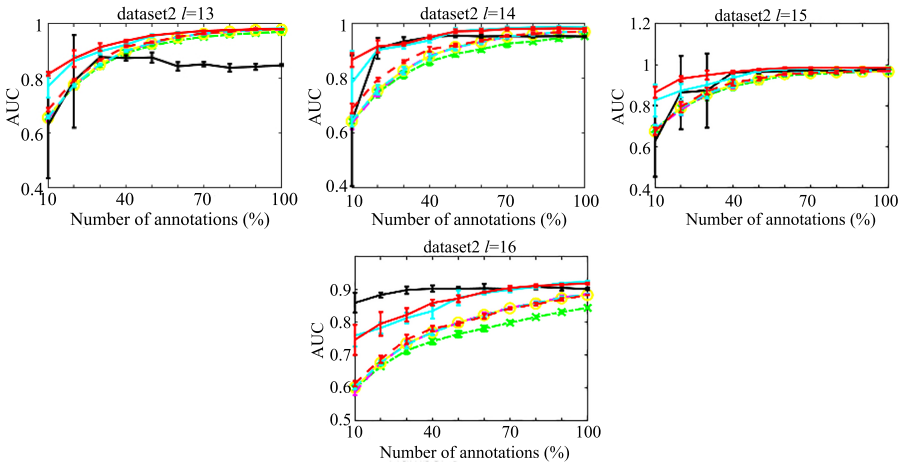


Figure 4 AUC results of comparison methods on 22 real datasets

by l. The specific experimental results on 22 datasets are given in Fig. 4. It can be seen that in most cases the proposed BayesDGC-w method consistently outperforms the other methods. The AUC effect of each method tends to increase steadily with the labeling size. Compared with BayesDGC-w, BayesDGC, and Yutc using sample features, especially in the case of fewer annotations ($\leq 40\%$), the methods (MV, DS, MaxEn, BayesGC, and BayesGC-w) only using annotations have equivalent effects, which are nevertheless significantly poorer than those of BayesDGC-w and BayesDGC. This indicates that sample features contain important complementary information. The method Yutc using a logistic regression classifier achieves comparable or even better results on some datasets such as `dataset1` ($l = 6$) and `dataset2` ($l = 3$, 14, 16). However, its results are not stable on some datasets such as `dataset2` ($l = 5$, 10, 13), probably because the linear model complexity is lower than that required by data fitting and the non-Bayesian implementation of this method significantly affects the parameter setting on the results. In contrast, the DNNs of BayesDGC-w and BayesDGC provide sufficient learning capabilities, and the parameters that best fit the data are automatically fitted by Bayesian inference, thus saving the need of manual parameter selection. The comparison between BayesDGC-w (BayesGC-w) and BayesDGC (BayesGC) demonstrates that the utilization of worker correlations helps to improve the crowdsourcing learning effect, which is consistent with the results in Ref. [12]. The next section presents the effects of this method when the number M of subclasses is set to different values.

5.3 Parameter discussion

This section discusses the influence of the number M of subclasses in the subclass mixture model on the BayesDGC-w model. Fig. 5 shows the AUC results of BayesDGC-w on four datasets when $M = 1, 3, 5, 10, 20, 30$, and 100% annotations are used. The case of $M = 1$ is equivalent to that where annotation correlations are not considered. Two sets of representative results are shown here: (1) `dataset2` ($l = 1$, 12) correspond to the case where considering the worker correlations ($M > 1$) helps to improve learning; (2) `dataset2` ($l = 3$, 11) correspond to the case where too many subclasses ($M = 20, 30$) are not conducive to learning, due to the over large hypothesis space of the worker capability parameter $\{\nu_{jkm}\}$, which renders the optimization prone to local optima. Therefore, we set the number of subclasses to $M = 3$ for the sake of optimization stability. We will further investigate this problem from the perspective of regularizing the worker capability parameters or nonparametric Bayesian learning.

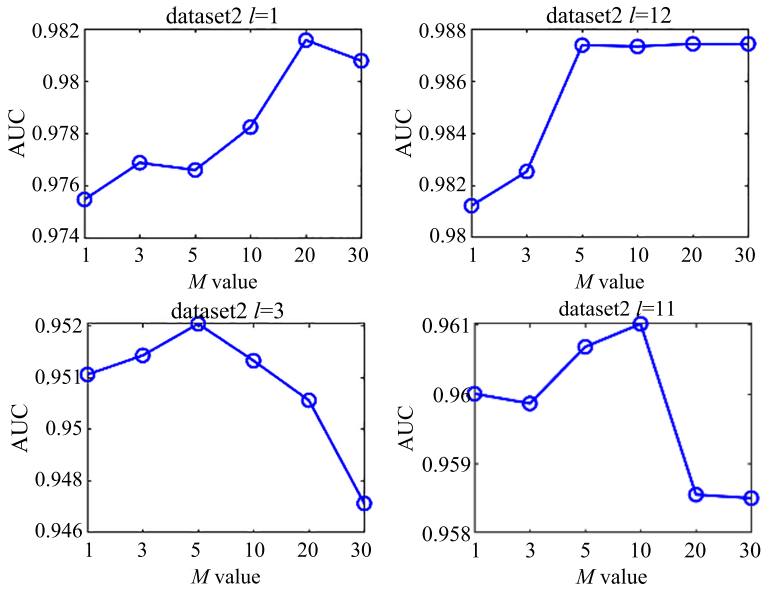


Figure 5 Influence of parameter M on learning performance of BayesDGC-w

6 Conclusion

In this paper, we proposed a deep generative crowdsourcing learning method with worker correlation utilization to capture annotation correlations by introducing a mixture model of workers' capabilities within each class into the annotation generation process. To implement Bayesian inference, this paper, resorting to the optimization technique of SVAE, used the conjugated structure of probability distributions to combine variational message passing with stochastic gradient descent for neural network parameters and thereby implement efficient end-to-end optimization. In this way, the proposed method avoids the iterative computational overheads of the EM algorithm and the Gibbs sampling method. It is found that the number of mixed components in the mixture model has a great influence on the model performance. In future work, we will explore this problem from the perspective of parameter regularization or nonparametric Bayesian learning, and try to extend the idea of worker correlation modeling to label correlation modeling for multi-label crowd sourcing learning.

References

[1] Weld D, Lin C, Bragg J. Artificial intelligence and collective intelligence. In: Malone T, Bernstein M, eds. The Collective Intelligence Handbook. 2015.

[2] Snow R, O'Connor B, Jurafsky D, *et al*. Cheap and fast—But is it good? Evaluating non-expert annotations for natural language tasks. Proc. of the Conf. on Empirical Methods in Natural Language Processing. 2008. 254–263.

[3] Raykar V, Yu S, Zhao L, *et al*. Learning from crowds. Journal of Machine Learning Research, 2010, 11: 1297–1322.

[4] Welinder P, Branson S, Belongie S, *et al*. The multidimensional wisdom of crowds. In: Lafferty J, Williams CI, Shawe-Taylor J, Zemel R, Culotta A, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.23. 2010. 2024–2432.

[5] Li Q, Li Y, Gao J, *et al*. A confidence-aware approach for truth discovery on long-tail data. Proc. of the VLDB Endowment, 2014, 8(4): 425–436.

[6] Dawid AP, Skene AM. Maximum likeihood estimation of observer error-rates using the em algorithm. Journal of the Royal Statistical Society, 1979, 28(1): 20–28.

[7] Liu Q, Peng J, Ihler A. Variational inference for crowdsourcing. In: Bartlett P, Pereira F, Burges C, Bottou L, Weinberger K, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.25. 2012. 692–700.

[8] Whitehill J, Ruvolo P, Wu T, *et al*. 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. Proc. of the Advances in Neural Information Processing Systems, Vol.22. 2009. 2035–2043.

[9] Zhou D, Basu S, Mao Y, *et al*. Learning from the wisdom of crowds by minimax entropy. In: Bartlett P, Pereira F, Burges C, Bottou L, Weinberger K, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.25. 2012. 2195–2203.

[10] Kim HC, Ghahramani Z. Bayesian classifier combination. In: Proc. of the 15th Int'l Conf. on Artificial Intelligence and Statistics. 2012. 619–627.

[11] Imamura H, Sato I, Sugiyama M. Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. Proc. of the 35th Int'l Conf. on Machine Learning. 2018. 2152–2161.

[12] Li Y, Rubinstein BIP, Cohn T. Exploiting worker correlation for label aggregation in crowdsourcing. Proc. of the 36th Int'l Conf. on Machine Learning. 2019. 3886–3895.

[13] Rodrigues F, Pereira F, Ribeiro B. Gaussian process classification and active learning with multiple annotators. Proc. of the 31th Int'l Conf. on Machine Learning. 2014. 433–441.

[14] LeCun Y, Bengio Y, Hinton GE. Deep learning. Nature, 2015, 521(7553): 436–444.

[15] Albarqouni S, Baur C, Achilles F, *et al*. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans. on Medical Imaging, 2016, 35(5): 1313–1321.

[16] Rodrigues F, Pereira FC. Deep learning from crowds. Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018. 1611–1618.

[17] Tanno R, Saeedi A, Sankaranarayanan S, *et al*. Learning from noisy labels by regularized estimation of annotator confusion. Proc. of the 2019 IEEE Conf. on Computer Vision and Pattern Recognition. 2019. 11244–11253.

[18] Kingma DP, Welling M. Auto-Encoding variational Bayes. Proc. of the 2nd Int'l Conf. on Learning Representations. 2014.

[19] Kingma DP, Mohamed S, Rezende DJ, *et al*. Semi-Supervised learning with deep generative models. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.27. 2014. 3581–3589.

[20] Johnson MJ, Duvenaud D, Wiltschko AB, *et al*. Composing graphical models with neural networks for structured representations and fast inference. In: Lee DD, Sugiyama M, von Luxburg U, Guyon I, Garnett R, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.29. 2016. 2946–2954.

[21] Yin L, Han J, Zhang W, *et al*. Aggregating crowd wisdoms with label-aware autoencoders. Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. 2017. 1325–1331.

[22] Atarashi K, Oyama S, Kurihara M. Semi-supervised learning from crowds using deep generative models. Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018. 1555–1562.

[23] Shi W, Sheng VS, Li X, *et al*. Semi-supervised multi-label learning from crowds via deep sequential generative model. Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2020. 1141–1149.

[24] Luo Y, Tian T, Shi J, *et al*. Semi-crowdsourced clustering with deep generative models. In: Bengio S, Wallach HM, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. Proc. of the Advances in Neural Information Processing Systems, Vol.31. 2018. 3216–3226.

[25] Li SY, Huang SJ, Chen S. Crowdsourcing aggregation with deep Bayesian learning. Science China Information Sciences, 2021, 64(3).

[26] Simpson E, Roberts S, Psorakis I, *et al*. Dynamic Bayesian combination of multiple imperfect classiers. Proc. of the Decision Making and Imperfection. 2013. 1–35.

[27] Venanzi M, Guiver J, Kazai G, *et al*. Community-based Bayesian aggregation models for crowdsourcing. Proc. of the 23rd Int'l Conf. on World Wide Web. 2014. 155–164.

[28] Homan MD, Blei DM, Wang C, *et al*. Stochastic variational inference. Journal of Machine Learning Research, 2013, 14(1): 1303–1347.

[29] Li SY, Jiang Y. Multi-Label crowdsourcing learning. Ruan Jian Xue Bao/Journal of Software, 2020, 31(5): 1497–1510 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5673.htm [doi: 10.13328/j.cnki.jos.005673]

[30] Chen CJ, Jiang L, Lei N, *et al*. An interactive feature selection method based on learning-from-crowds. Sci Sin Inform, 2020, 50: 794–812 (in Chinese with English abstract). [doi: 10.1360/SSI-2019-0208]

[31] Feng Y, Wang Y, Fang CR, *et al*. An approach for developing a highly trustworthy crowd-sourced workforce. Sci Sin Inform, 2019, 49: 1412–1427 (in Chinese with English abstract). [doi: 10.1360/N112018-00303]

[32] Li SY, Jiang Y, Chawla NV, *et al*. Multi-Label learning from crowds. IEEE Trans. on Knowledge and Data Engineering, 2019, 31(7): 1369–1382.

[33] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proc. of the 3rd Int'l Conf. on Learning Representations. CA, 2015.

[34] Demsar J. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research, 2006, 7: 1–30.

Shaoyuan Li, Ph.D., lecturer, CCF member. Her research interests include machine learning and data mining.

Shengjun Huang, Ph.D., doctoral supervisor, CCF member. His research interests include machine learning and pattern recognition.

Menglong Wei, graduate student, CCF student member. His research interests include machine learning and data mining.