Aggregating Crowdsourced Ordinal Labels via Bayesian Clustering

Xiawei Guo(✉) and James T. Kwok

Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{xguoae,jamesk}@cse.ust.hk

Abstract. Crowdsourcing allows the collection of labels from a crowd of workers at low cost. In this paper, we focus on ordinal labels, whose underlying order is important. Crowdsourced labels can be noisy as there may be amateur workers, spammers and/or even malicious workers. Moreover, some workers/items may have very few labels, making the estimation of their behavior difficult. To alleviate these problems, we propose a novel Bayesian model that clusters workers and items together using the nonparametric Dirichlet process priors. This allows workers/items in the same cluster to borrow strength from each other. Instead of directly computing the posterior of this complex model, which is infeasible, we propose a new variational inference procedure. Experimental results on a number of real-world data sets show that the proposed algorithm is more accurate than the state-of-the-art, and is more robust to sparser labels.

1 Introduction

In many real-world classification applications, acquisition of labels is difficult and expensive. Recently, crowdsourcing provides an attractive alternative. With platforms like the Amazon Mechanical Turk, cheap labels can be efficiently obtained from non-expert workers. However, the collected labels are often noisy because of the presence of inexperienced workers, spammers and/or even malicious workers.

To clean these labels, a simple approach is majority voting [14]. By assuming that most workers are reliable, labels on a particular item are aggregated by selecting the most common label. However, this ignores relationships among labels provided by the same worker. To alleviate this problem, one can assume that labels of each worker are generated according to an underlying confusion matrix, which represents the probability that the worker assigns a particular label conditioned on the true label [7,19]. Others have also modeled the difficulties in labeling various items [2,13,23–25] and workers' dedications to the labeling task [2].

On the other hand, besides the commonly encountered binary and multiclass labels, labels can also be ordinal. For example, in web search, the relevance of a query-URL pair can be labeled as "irrelevant", "relevant" and "highly-relevant". Unlike nominal labels, it is important to exploit the underlying order of ordinal

P. Frasconi et al. (Eds.): ECML PKDD 2016, Part I, LNAI 9851, pp. 426–442, 2016.
DOI: 10.1007/978-3-319-46128-1_27

labels. In particular, adjacent labels are often more difficult to differentiate than those that are further apart.

To solve the aforementioned problem, Lakshminarayanan and Teh [16] assumed that the ordinal labels are generated by the discretization of some continuous-valued latent labels. The latent label for each worker-item pair is drawn from a normal distribution, with its mean equal to the true label and its variance related to the worker's reliability and the item's difficulty. While this model is useful for "good" workers, it is not appropriate for malicious workers whose labels can be very different from the true label. Moreover, it can be too simplistic to use only one reliability (resp. difficulty) parameter to model each worker (resp. item).

A more recent model is the minimax entropy framework [26], which is extended from the minimax conditional entropy approach for multiclass label aggregation [25]. To encode ordinal information, they compare the worker and item labels with a reference label that can take all possible label values. The confusion for each worker-item pair as obtained from the model is then constrained to be close to its empirical counterpart. Finally, the true labels and probabilities are obtained by solving an optimization problem derived from the minimax entropy principle. In comparison with [16], ordering of the ordinal labels is now explicitly considered.

In crowdsourcing applications, some workers may only provide very few labels. Similarly, some items may receive very few labels. Parameter estimation for these workers and items can thus be unreliable. To alleviate this problem, one can consider the latent connections among workers and items. Intuitively, workers with similar characteristics (e.g., gender, age, and nationality) tend to have similar behaviors, and similarly for items. By clustering them together, one can borrow strength from one worker/item to another. Kajino *et al.* [12] formulated label aggregation as a multitask learning problem [8]. Each worker is modeled as a classifier, and the classifiers of similar workers are encouraged to be similar. However, the ground-truth classifier, which generates the true labels, is required to lie in one of the worker clusters. Moreover, this algorithm requires access to item features and cannot be used with ordinal labels. Venanzi *et al.* [21] proposed to cluster multiclass labels by using the Dirichlet distribution. However, the number of clusters needs to be pre-specified, which may not be practical. Moreover, item grouping is not considered. Lakkaraju *et al.* [15] modeled both item and worker groups. However, they again require the use of both worker and item features. Moreover, clustering and label inference are performed as separate tasks.

In this paper, motivated by the conditional probability derived in the dual of [26], we propose a novel algorithm to aggregate ordinal labels. Different from [26], a full Bayesian model is constructed, in which clustering of both workers and items are encouraged via the Dirichlet process (DP) [9] priors. DP is a nonparametric model which is advantageous in that the number of clusters does not need to be pre-specified. The resultant Bayesian model allows detection of clustering structure, learning of worker/item characteristics and label aggregation be performed simultaneously. Empirically, it also significantly outperforms

the state-of-the-art. However, as we use DP priors with non-conjugate base distributions, exact inference is infeasible. To address this problem, we extend the techniques in [11], and derive a mean field variational inference algorithm for parameter estimation.

2 Ordinal Label Aggregation by Minimax Conditional Entropy

Let there be N workers, M items, and m ordinal label classes. We use i, j, m to index the workers, items, and labels, respectively. The true label of item j is denoted Y_j, with probability distribution Q. The label assigned by worker i to item j is X_{ij}, and Ξ is the set of (i, j) tuples with X_{ij}'s observed. We assume that there is at least one observed X_{ij} for each worker i, and at least one observed X_{ij} for each item j.

Zhou *et al.* [26] formulated label aggregation as a constrained minimax optimization problem, in which $H(X|Y) - H(Y) - \frac{1}{\alpha}\Omega(\xi) - \frac{1}{\beta}\Psi(\zeta)$ is maximized w.r.t. $P(X_{ij} = k \mid Y_j = c)$ and minimized w.r.t. $Q(Y_j = c)$. Here, $H(X|Y)$ is the conditional entropy of X given Y, $H(Y)$ is the entropy of Y, $\Omega(\xi) = \sum_{i,s}(\xi_{is}^{\triangle,\nabla})^2, \Psi(\zeta) = \sum_{j,s}(\zeta_{js}^{\triangle,\nabla})^2$ are ℓ_2-regularizers on the slack variables $\xi_{is}^{\triangle,\nabla}, \zeta_{js}^{\triangle,\nabla}$ (in (1) and (2)), and α, β are regularization parameters. Let $\phi_{ij}(c, k) = Q(Y_j = c)P(X_{ij} = k|Y_j = c)$ be the expected confusion from label c to label k by worker i on item j, and $\hat{\phi}_{ij}(c, k) = Q(Y_j = c)\mathbb{I}(X_{ij} = k)$ be its empirical counterpart. Besides the standard normalization constraints on probability distributions P and Q, Zhou *et al.* [26] requires $\phi_{ij}(c, k)$ be close to the empirical $\hat{\phi}_{ij}(c, k)$:

$$\sum_{c \triangle s}\sum_{k \nabla s}\sum_{j}[\phi_{ij}(c,k) - \hat{\phi}_{ij}(c,k)] = \xi_{is}^{\triangle,\nabla}, \forall i, \forall 2 \le s \le m, \tag{1}$$

$$\sum_{c \triangle s}\sum_{k \nabla s}\sum_{i}[\phi_{ij}(c,k) - \hat{\phi}_{ij}(c,k)] = \zeta_{js}^{\triangle,\nabla}, \forall j, \forall 2 \le s \le m. \tag{2}$$

Here, s is a reference label for comparing the true label c with worker label k, and $\triangle, \nabla$ is a binary relation operator (either $\ge$ or $<$). Together, they allow consideration of the four cases: (i) $c < s, k < s$; (ii) $c < s, k \ge s$; (iii) $c \ge s, k < s$; and (iv) $c \ge s, k \ge s$.

At optimality, it can be shown that

$$P(X_{ij} = k|Y_j = c) = \exp[\sigma_i(c,k) + \tau_j(c,k)]/Z_{ijc}, \tag{3}$$

where Z_{ijc} is a normalization factor,

$$\sigma_i(c,k) = \sum_{1 \le s \le m}\sum_{\triangle,\nabla}\sigma_{is}^{\triangle,\nabla}\mathbb{I}(c \triangle s, k \nabla s), \tag{4}$$

$$\tau_j(c,k) = \sum_{1 \le s \le m}\sum_{\triangle,\nabla}\tau_{js}^{\triangle,\nabla}\mathbb{I}(c \triangle s, k \nabla s), \tag{5}$$

$\sigma_{is}^{\triangle,\nabla}$, $\tau_{js}^{\triangle,\nabla}$ are Lagrange multipliers for the constraints (1) and (2), respectively, and $\mathbb{I}(\cdot)$ is the indicator function. Note that $\sigma_i(c,k)$ controls how likely worker i assigns label k when the true label is c, and $\tau_j(c,k)$ controls how likely item j is assigned label k when the true label is c. Equations (4) and (5) can be written more compactly as $\sigma_i(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\sigma}_i$ and $\tau_j(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\tau}_j$, where $\boldsymbol{\sigma}_i = [\sigma_{is}^{\triangle,\nabla}]$, $\boldsymbol{\tau}_j = [\tau_{js}^{\triangle,\nabla}]$, and $\mathbf{t}_{ck} = [\mathbb{I}(c \triangle s, k \nabla s)]$. Moreover, let $\mathbf{X} = [X_{ij}]_{(i,j)\in\Xi}$, and $\mathbf{Y} = [Y_j]$. Equation (3) can be rewritten as

$$
\begin{aligned}
P(\mathbf{X}|\mathbf{Y}) &= \prod_{(i,j)\in\Xi} \prod_{c,k} P(X_{ij} = k | Y_j = c)^{\mathbb{I}(X_{ij}=k, Y_j=c)} \\
&= \prod_{(i,j)\in\Xi} \prod_{c,k} \left[\frac{1}{Z_{ijc}} \exp[\mathbf{t}_{ck}^T (\boldsymbol{\sigma}_i + \boldsymbol{\tau}_j)] \right]^{\mathbb{I}(X_{ij}=k, Y_j=c)} . \qquad (6)
\end{aligned}
$$

3 Bayesian Clustering of Workers and Items

Note that each worker i (resp. item j) has its own set of variables $\{\sigma_i(c,k)\}$ (resp. $\{\tau_j(c,k)\}$). When the data are sparse, i.e., the set Ξ of observed labels is small, an accurate estimation of these variables can be difficult. In this section, we alleviate this data sparsity problem by clustering workers and items. While the minimax optimization framework in [26] can utilize ordering information in the ordinal labels, it is non-Bayesian and clustering cannot be easily encouraged. In this paper, we propose a full Bayesian model, and encourage clustering of workers and items using the Dirichlet process (DP) [9]. The DP prior is advantageous in that the number of clusters does not need to be specified in advance. However, with the non-conjugate priors and DPs involved, inference of the proposed model becomes more difficult. By extending the work in [11], we derive a variational Bayesian inference algorithm to infer the parameters and aggregate labels.

3.1 Model

Recall that $\sigma_i(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\sigma}_i$ controls how likely worker i assigns label k when the true label is c. To encourage worker clustering, we define a prior G_a on $\{\boldsymbol{\sigma}_i\}_{i=1}^N$. G_a is drawn from the Dirichlet process $\mathrm{DP}(\beta_a, G_{a0})$, where β_a is the concentration parameter, and G_{a0} is the base distribution (here, we use the normal distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$). Similarly, as $\tau_j(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\tau}_j$ controls how likely item j assigns label k when the true label is c, we define a prior $G_b \sim \mathrm{DP}(\beta_b, G_{b0})$ on $\{\boldsymbol{\tau}_j\}_{j=1}^M$ to encourage item clustering (with $G_{b0} = \mathcal{N}(\boldsymbol{\nu}_0, \boldsymbol{\Omega}_0)$).

To make variational inference possible, we use the stick-breaking representation [20] and rewrite G_a as $\sum_{l=1}^{\infty} \phi_l \delta_{\boldsymbol{\sigma}_l^*}$, where $\phi_l = V_l \prod_{d=1}^{l-1}(1 - V_d), V_l \sim \mathrm{Beta}(1, \beta_a)$, and $\boldsymbol{\sigma}_l^* \sim G_{a0}$. When we draw $\boldsymbol{\sigma}$ from G_a, $\delta_{\boldsymbol{\sigma}_l^*} = 1$ if $\boldsymbol{\sigma} = \boldsymbol{\sigma}_l^*$, and 0 otherwise. Similarly, $G_b = \sum_{i=1}^{\infty} \varphi_h \delta_{\boldsymbol{\tau}_h^*}$, where $\varphi_h = H_h \prod_{d=1}^{h-1}(1 - H_d)$, $H_h \sim \mathrm{Beta}(1, \beta_b)$ and $\boldsymbol{\tau}_h^* \sim G_{b0}$. Similar to $\sigma_i(c,k), \tau_j(c,k)$ in [26], $\sigma_l^*(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\sigma}_l^*$

controls how likely workers in cluster l assign label k when the true label is c, and $\tau_h^*(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\sigma}_h^*$ controls how likely items in cluster h are assigned label k when the true label is c Let z_i (resp. u_j) indicate the cluster that worker i (resp. item j) belongs to. We then have $\sigma_i(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\sigma}_{z_i}^*$ and $\tau_j(c,k) = \mathbf{t}_{ck}^T \boldsymbol{\tau}_{u_j}^*$ for all i, j, c, k. Putting these into (6), we obtain the conditional probability as

$$P(\mathbf{X}|\mathbf{Y}, \mathbf{z}, \mathbf{u}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*) = \prod_{(i,j)\in\Xi} \prod_{c,k} \left[\frac{1}{Z_{ijc}} \exp\left[\mathbf{t}_{ck}^T (\boldsymbol{\sigma}_{z_i}^* + \boldsymbol{\tau}_{u_j}^*) \right] \right]^{\mathbb{I}(X_{ij}=k, Y_j=c)}, \tag{7}$$

where $Z_{ijc} = \sum_k \exp[\mathbf{t}_{ck}^T (\boldsymbol{\sigma}_{z_i}^* + \boldsymbol{\tau}_{u_j}^*)]$, $\boldsymbol{\sigma}^* = [\boldsymbol{\sigma}_l^*], \boldsymbol{\tau}^* = [\boldsymbol{\tau}_h^*], \mathbf{z} = [z_i]$, and $\mathbf{u} = [u_j]$. In other words, rating X_{ij} is generated from a softmax function [3] conditioned on $Y_j, z_i, u_j, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*$. Finally, the true label Y_j of item j is drawn from the multinomial distribution $\text{Mult}(\pi_1, \pi_2, \ldots, \pi_m)$, where $\pi_1, \ldots, \pi_m$ are drawn from a Dirichlet prior with hyperparameter α. The whole label generation process is shown in Algorithm 1. A graphical representation of the Bayesian model, which will be called Cluster-based Ordinal Label Aggregation (COLA) in the sequel, is shown in Fig. 1.

Algorithm 1. The proposed generation process.

for $j = 1, 2, \ldots, M$ **do** ▷ Generate true labels
  draw $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_m] \sim \text{Dir}(\alpha/m, \alpha/m, \ldots, \alpha/m)$; // Dirichlet distribution
  draw $Y_j \sim \text{Mult}(\boldsymbol{\pi})$;
end for
for $l = 1, 2, \ldots$ **do** ▷ Generate worker clusters
  draw $V_l \sim \text{Beta}(1, \beta_a)$;
  $\phi_l = V_l \prod_{d=1}^{l-1} (1 - V_d)$;
  draw $\boldsymbol{\sigma}_l^* \sim G_{a0} = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$;
end for
for $i = 1, 2, \ldots, N$ **do** ▷ Generate workers from worker clusters
  draw $z_i \sim \text{Mult}(\boldsymbol{\phi})$;
end for
for $h = 1, 2, \ldots$ **do** ▷ Generate item clusters
  draw $H_h \sim \text{Beta}(1, \beta_b)$;
  $\varphi_h = H_h \prod_{d=1}^{h-1} (1 - H_d)$;
  draw $\boldsymbol{\tau}_h^* \sim G_{b0} = \mathcal{N}(\boldsymbol{\nu}_0, \boldsymbol{\Omega}_0)$;
end for
for $j = 1, 2, \ldots, M$ **do** ▷ Generate items from item clusters
  draw $u_j \sim \text{Mult}(\boldsymbol{\varphi})$;
end for
for $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, M$ **do** ▷ Generate worker labels
  draw $X_{ij} \sim P(X_i | Y_j, z_i, u_j, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$;
end for

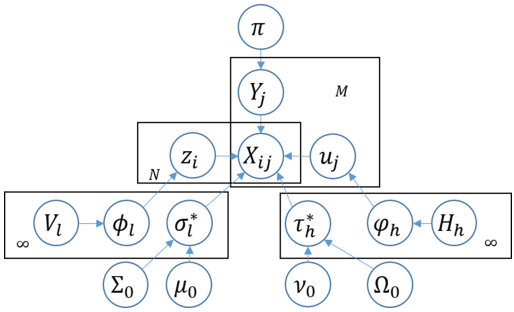


Fig. 1. Graphical representation of the proposed model.

3.2 Inference Procedure

The joint distribution can be written as

$$\begin{aligned} &P(\mathbf{X}, \mathbf{Y}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \mathbf{z}, \mathbf{u}, \mathbf{V}, \mathbf{H} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\nu}_0, \boldsymbol{\Omega}_0, \alpha, \beta_a, \beta_b) \\ &= P(\mathbf{X}|\mathbf{Y}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \mathbf{z}, \mathbf{u}) P(\boldsymbol{\sigma}^*|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) P(\boldsymbol{\tau}^*|\boldsymbol{\nu}_0, \boldsymbol{\Omega}_0) \\ &\quad P(\mathbf{Y}|\boldsymbol{\pi}) P(\boldsymbol{\pi}|\alpha) P(\mathbf{z}|\mathbf{V}) P(\mathbf{V}|\beta_1) P(\mathbf{u}|\mathbf{H}) P(\mathbf{H}|\beta_2), \end{aligned}$$

where $\mathbf{V} = [V_l]$, and $\mathbf{H} = [H_h]$. Monte Carlo Markov Chain (MCMC) sampling [1] can be used to approximate the posterior distribution. However, it can be slow and its convergence is difficult to diagnose [4]. Another approach is variational inference [11], which approximates the posterior distribution by maximizes a lower bound of the marginal likelihood. However, due to the infinite number of variables in the DPs and our use of non-conjugate priors, standard variational inference cannot be used. To solve this problem, we propose an integration of the techniques in [4,5] with variational inference. Specifically, we infer the variational parameters of the DPs based on an extension of [4], and handle the non-conjugate priors by a technique similar to [5].

Let $\boldsymbol{\theta} = \{\boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \mathbf{Y}, \boldsymbol{\pi}, \mathbf{z}, \mathbf{u}, \mathbf{V}, \mathbf{H}\}$. In variational inference, the posterior $P(\boldsymbol{\theta}|\mathbf{X})$ is approximated by a distribution $q(\boldsymbol{\theta})$. The log likelihood of the marginal distribution of $\mathbf{X}$ is $\log P(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q \| P_{\boldsymbol{\theta}|\mathbf{X}})$, where $\mathcal{L}(q) = \int q(\boldsymbol{\theta}) \log \frac{P(\mathbf{X}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}$, and $\mathrm{KL}(q \| P_{\boldsymbol{\theta}|\mathbf{X}}) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|\mathbf{X})} \mathrm{d}\boldsymbol{\theta}$ is the KL divergence between q and $P_{\boldsymbol{\theta}|\mathbf{X}}$. As $\mathrm{KL}(q \| P_{\boldsymbol{\theta}|\mathbf{X}}) \geq 0$, we simply maximize the lower bound $\mathcal{L}(q)$ of $\log P(\mathbf{X})$. Using the variational mean field approach, $q(\boldsymbol{\theta})$ is assumed to be factorized as $\prod_{n=1}^{S} q_n(\boldsymbol{\theta}_n)$, where S is the number of factors, $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_S\}$ is a partition of $\boldsymbol{\theta}$, and q_n is the variational distribution of $\boldsymbol{\theta}_n$ [22]. We perform alternating maximization of $\mathcal{L}\left(\prod_{n=1}^{S} q_n(\boldsymbol{\theta}_n)\right)$ w.r.t. q_n's. It can be shown that the optimal q_n is given by

$$q_n^*(\boldsymbol{\theta}_n) = \exp\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg n})} \log P(\mathbf{X}, \boldsymbol{\theta})\right] + \text{constant}, \tag{8}$$

where $\boldsymbol{\theta}_{\neg n}$ is the subset of variables in $\boldsymbol{\theta}$ excluding $\boldsymbol{\theta}_n$, and $\mathbb{E}_{q(\boldsymbol{\theta}_{\neg n})}$ is the corresponding expectation operator.

As there are infinite variables in the stick-breaking representations of $\mathrm{DP}(\beta_a, G_{a0})$ and $\mathrm{DP}(\beta_b, G_{b0})$, we set a maximum on the numbers of clusters as in [4]. Note that the exact distributions of the stick-breaking process are not truncated. The factorized variational distribution is

$$\begin{aligned}
&q(\boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \mathbf{Y}, \boldsymbol{\pi}, \mathbf{z}, \mathbf{u}, \mathbf{V}, \mathbf{H}) \\
&= q_{\boldsymbol{\sigma}^*}(\boldsymbol{\sigma}^*) q_{\boldsymbol{\tau}^*}(\boldsymbol{\tau}^*) q_{\mathbf{Y}}(\mathbf{Y}) q_{\boldsymbol{\pi}}(\boldsymbol{\pi}) q_{\mathbf{z}}(\mathbf{z}) q_{\mathbf{u}}(\mathbf{u}) q_{\mathbf{V}}(\mathbf{V}) q_{\mathbf{H}}(\mathbf{H}) \\
&= \prod_{l=1}^{K_1} q_{\boldsymbol{\sigma}_l^*}(\boldsymbol{\sigma}_l^*) \prod_{h=1}^{K_2} q_{\boldsymbol{\tau}_h^*}(\boldsymbol{\tau}_h^*) \prod_{j=1}^{M} q_{Y_j}(Y_j) q_{\boldsymbol{\pi}}(\boldsymbol{\pi}) \\
&\quad \prod_{i=1}^{N} q_{z_i}(z_i) \prod_{j=1}^{M} q_{u_j}(u_j) \prod_{l=1}^{K_1} q_{V_l}(V_l) \prod_{h=1}^{K_2} q_{H_h}(H_h),
\end{aligned}$$

where K_1, K_2 are the truncated numbers of clusters for workers and items, respectively. Using (8), it can be shown that the variational distributions of $\{\mathbf{Y}, \boldsymbol{\pi}, \mathbf{z}, \mathbf{u}, \mathbf{H}, \mathbf{V}\}$ can be easily obtained as:

$$\begin{aligned}
q_{Y_j}^*(Y_j) = \mathrm{Mult}(\mathbf{r}_j^Y), \;\; q_{\boldsymbol{\pi}}^*(\boldsymbol{\pi}) = \mathrm{Dir}(\alpha_1, \alpha_2, \ldots \alpha_m), \\
q_{z_i}^*(z_i) = \mathrm{Mult}(\mathbf{r}_i^z), \;\; q_{u_j}^*(u_j) = \mathrm{Mult}(\mathbf{r}_j^u), \\
q_{V_l}^*(V_l) = \mathrm{Beta}(\gamma_{l,1}, \gamma_{l,2}), \;\; q_{H_h}^*(H_h) = \mathrm{Beta}(\eta_{h,1}, \eta_{h,2}),
\end{aligned}$$

where $\{\mathbf{r}_j^Y\}_{j=1}^M, \{\mathbf{r}_i^z\}_{i=1}^N, \{\mathbf{r}_j^u\}_{j=1}^M, \{\alpha_c\}_{c=1}^m, \{\gamma_{l,1}, \gamma_{l,2}\}_{l=1}^{K_1}$, and $\{\eta_{h,1}, \eta_{h,2}\}_{h=1}^{K_2}$ are variational parameters. All these have closed-form updates as

$$r_{jc}^Y \leftarrow \frac{1}{Z_j^Y} \exp\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg \mathbf{Y}})} \log \pi_c + \sum_{i:(i,j)\in\Xi}^{N} \sum_{l=1}^{K_1} \sum_{h=1}^{K_2} r_{il}^z r_{jh}^u U_{ijlhc}\right],$$

$$r_{il}^z \leftarrow \frac{1}{Z_i^z} \exp\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg \mathbf{z}})} \log \phi_l + \sum_{j:(i,j)\in\Xi} \sum_{c=1}^{m} \sum_{h=1}^{K_2} r_{jh}^u r_{jc}^Y U_{ijlhc}\right],$$

$$r_{jh}^u \leftarrow \frac{1}{Z_j^u} \exp\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg \mathbf{u}})} \log \varphi_h + \sum_{i:(i,j)\in\Xi} \sum_{c=1}^{m} \sum_{l=1}^{K_1} r_{il}^z r_{jc}^Y U_{ijlhc}\right],$$

$$\alpha_c \leftarrow \frac{\alpha}{m} + \sum_{j=1}^{M} r_{jc}^Y, \quad \gamma_{l,1} \leftarrow 1 + \sum_{i=1}^{N} r_{il}^z, \quad \gamma_{l,2} \leftarrow \beta_a + \sum_{i=1}^{N} \sum_{d=l+1}^{K_1} r_{id}^z,$$

$$\eta_{h,1} \leftarrow 1 + \sum_{j=1}^{M} r_{jh}^u, \quad \eta_{h,2} \leftarrow \beta_b + \sum_{j=1}^{M} \sum_{d=h+1}^{K_2} r_{jd}^u,$$

where Z_j^Y, Z_i^z, Z_j^u are normalization constants,

$$U_{ijlhc} = \sum_{k=1}^{m} I(X_{ij} = k) \mathbf{t}_{ck}^T [\mathbb{E}_{q(\boldsymbol{\sigma}_l^*)}(\boldsymbol{\sigma}_l^*) + \mathbb{E}_{q(\boldsymbol{\tau}_h^*)}(\boldsymbol{\tau}_h^*)] - \mathbb{E}_{q(\boldsymbol{\sigma}_l^*, \boldsymbol{\tau}_h^*)} Z_{lhc}^* \quad (9)$$

and $Z^*_{lhc} = \sum_{k=1}^{m} \exp[\mathbf{t}_{ck}^T(\boldsymbol{\sigma}_l^* + \boldsymbol{\tau}_h^*)]$. Computing U_{ijlhc} requires knowing the variational distributions of $\boldsymbol{\sigma}_l^*$ and $\boldsymbol{\tau}_h^*$, and will be derived in the following.

Recall that $P(\boldsymbol{\sigma}_l^*), P(\boldsymbol{\tau}_h^*)$ are normal distributions. These are not the conjugate prior of $P(\mathbf{X}|\mathbf{Y}, \mathbf{z}, \mathbf{u}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ in (7). Thus, on maximizing $\mathcal{L}$, $q(\boldsymbol{\sigma}^*)$ and $q(\boldsymbol{\tau}^*)$ do not have closed-form solutions. Note that the $\frac{1}{Z_{ijc}} \exp\left[\mathbf{t}_{ck}^T(\boldsymbol{\sigma}_{z_i}^* + \boldsymbol{\tau}_{u_j}^*)\right]$ term in (7) is a softmax function similar to that in [5], which uses variational inference to learn discrete choice models. However, while the parameters of different sets of choices in [5] are conditionally independent, here in (7) they are coupled together. Thus, the inference procedure in [5] cannot be directly applied and has to be extended.

First, (7) can be rewritten as

$$P(\mathbf{X}|\mathbf{Y}, \mathbf{z}, \mathbf{u}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*) = \prod_{(i,j)\in\Xi} \prod_{c,k,l,h} \left[\frac{\exp\left[\mathbf{t}_{ck}^T(\boldsymbol{\sigma}_l^* + \boldsymbol{\tau}_h^*)\right]}{Z^*_{lhc}}\right]^{\mathbb{I}(z_i=l, u_j=h, X_{ij}=k, Y_j=c)}.$$

Since $P(\boldsymbol{\sigma}_l^*), P(\boldsymbol{\tau}_h^*)$ are normal distributions, we constrain the variational distributions of $\boldsymbol{\sigma}_l^*$ and $\boldsymbol{\tau}_h^*$ to be also normal, i.e., $q_{\boldsymbol{\sigma}_l^*}(\boldsymbol{\sigma}_l^*) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$, and $q_{\boldsymbol{\tau}_h^*}(\boldsymbol{\tau}_h^*) = \mathcal{N}(\boldsymbol{\nu}_h, \boldsymbol{\Omega}_h)$. Let $\boldsymbol{\mu} = [\boldsymbol{\mu}_l]$, $\boldsymbol{\nu} = [\boldsymbol{\nu}_h]$, $\boldsymbol{\Sigma} = [\boldsymbol{\Sigma}_l]$, and $\boldsymbol{\Omega} = [\boldsymbol{\Omega}_h]$. On maximizing $\mathcal{L}(q)$, it can be shown that the variational parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\nu}, \boldsymbol{\Omega}\}$ can be obtained as

$$\begin{aligned}\min_{\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\nu},\boldsymbol{\Omega}} \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log q_{\boldsymbol{\sigma}^*}(\boldsymbol{\sigma}^*) q_{\boldsymbol{\tau}^*}(\boldsymbol{\tau}^*)\right] - \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log P(\boldsymbol{\sigma}^*)P(\boldsymbol{\tau}^*)\right] \\ -\mathbb{E}_{q(\boldsymbol{\theta})}\left[\log P(\mathbf{X}|\mathbf{Y}, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \mathbf{z}, \mathbf{u})\right]. \end{aligned} \tag{10}$$

The first term is the entropy of the normal distribution, and the second term is the cross-entropy of two normal distributions. Both are easy to compute. The last term can be rewritten as $\sum_{(i,j)\in\Xi} \sum_{c,k,l,h} r_{il}^z r_{jh}^u r_{jc}^Y \mathbb{I}(X_{ij} = k) \left(\mathbf{t}_{ck}^T(\boldsymbol{\mu}_l + \boldsymbol{\nu}_h) - \mathbb{E}_{q(\boldsymbol{\sigma}^*,\boldsymbol{\tau}^*)} \log Z^*_{lhc}\right)$. The term $\mathbb{E}_{q(\boldsymbol{\sigma}^*,\boldsymbol{\tau}^*)} \log Z^*_{lhc}$, which also appears in (9), can be approximated as in [5]:

$$\log \sum_{k=1}^{m} \exp(\mathbf{t}_{ck}^T \boldsymbol{\mu}_l + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Sigma}_l \mathbf{t}_{ck}) \exp(\mathbf{t}_{ck}^T \boldsymbol{\nu}_h + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Omega}_h \mathbf{t}_{ck}).$$

Problem (10) can then be solved via gradient-based methods, such as L-BFGS [17]. Denote the objective by f. It can be shown that

$$\frac{\partial f}{\partial \boldsymbol{\mu}_l} = -\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu}_l - \boldsymbol{\mu}_0) + \sum_{(i,j)\in\Xi} \sum_{c=1}^{m} \sum_{h=1}^{K_2} r_{il}^z r_{jh}^u r_{jc}^Y \sum_{k=1}^{m} [\mathbb{I}(X_{ij} = k) - w_{klh}]\mathbf{t}_{ck},$$

where

$$w_{klh} = \frac{\exp(\mathbf{t}_{ck}^T \boldsymbol{\nu}_h + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Omega}_h \mathbf{t}_{ck}) \exp(\mathbf{t}_{ck}^T \boldsymbol{\mu}_l + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Sigma}_l \mathbf{t}_{ck})}{\sum_{k=1}^{m} \exp(\mathbf{t}_{ck}^T \boldsymbol{\nu}_h + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Omega}_h \mathbf{t}_{ck}) \exp(\mathbf{t}_{ck}^T \boldsymbol{\mu}_l + \frac{1}{2}\mathbf{t}_{ck}^T \boldsymbol{\Sigma}_l \mathbf{t}_{ck})}.$$

Moreover, as $\mathbf{\Sigma}_l \succeq 0$, we assume that $\mathbf{\Sigma}_l = \mathbf{L}_l^T \mathbf{L}_l$, where $\mathbf{L}_l$ is lower-triangular. It can then be shown that

$$\frac{\partial f}{\partial \mathbf{L}_l} = \mathbf{L}_l^{-T} - \left[\mathbf{\Sigma}_0^{-1} - \sum_{(i,j)\in\Xi} \sum_{c=1}^{m} \sum_{h=1}^{K_2} r_{il}^z r_{jh}^u r_{jc}^Y \left(\sum_{k=1}^{m} w_{klh} \mathbf{t}_{ck}^T \mathbf{t}_{ck} \right) \right] \mathbf{L}_l.$$

Recall that $\mathbf{L}_l$ is lower-triangular, so in updating $\mathbf{L}_l$, we only need the diagonal elements $(\mathbf{L}_l^{-T})_{ii} = 1/(\mathbf{L}_l)_{ii}$. of the upper-triangular $\mathbf{L}_l^{-T}$ [5]. The gradients of f w.r.t. $\boldsymbol{\nu}$ and $\mathbf{\Omega}$ can be obtained in a similar manner.

4 Experiments

4.1 Synthetic Data Set

In this section, we perform experiments on synthetic data. Workers are generated from three clusters ($w1, w2, w3$), items from three clusters ($i1, i2, i3$), and ordinal labels in $\{1, 2, 3, 4, 5\}$. The cluster parameters $\boldsymbol{\sigma}^*, \boldsymbol{\tau}^*$ are sampled independently from the normal distribution with means in Table 1 and standard deviation 0.1. Confusion matrices[1] of the clusters are shown in Fig. 2. As can be seen, workers in cluster $w1$ are the least confused in label assignment. This is followed by cluster $w2$, and workers in cluster $w3$ are most confused (spammers). Similarly, items in cluster $i1$ are the least confused, while those in $i3$ are the most confused.

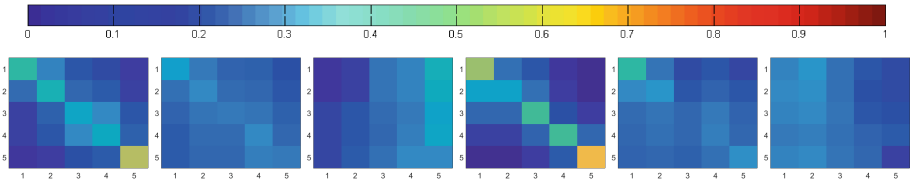
Table 1. Parameter means of the worker clusters ($w1, w2, w3$) and item clusters ($i1, i2, i3$).

	$(\sigma_{ls}^*)^{<,<}$	$(\sigma_{ls}^*)^{<,\geq}$	$(\sigma_{ls}^*)^{\geq,<}$	$(\sigma_{ls}^*)^{\geq,\geq}$
$w1$	1	0	0	1
$w2$	1	0.8	0.8	1
$w3$	0.3	1	0.5	1

	$(\tau_{hs}^*)^{<,<}$	$(\tau_{hs}^*)^{<,\geq}$	$(\tau_{hs}^*)^{\geq,<}$	$(\tau_{hs}^*)^{\geq,\geq}$
$i1$	1	0	0	1
$i2$	1	0.8	0.8	1
$i3$	1	0.5	1	0.5

We generate two data sets. Both have 300 workers from the 3 clusters ($w1$, $w2$, $w3$), with sizes 200, 50, and 50, respectively. The first data set (D1) has 1200 items coming from the 3 clusters ($i1$, $i2$, $i3$), with sizes 800, 200, and 200, respectively. Each item is labeled by 6 randomly selected workers. The second data set (D2) has 300 items coming from the 3 clusters with sizes 200, 50, and 50, respectively. Each item is labeled by 30 workers.

[1] To obtain the confusion matrices of items, we remove the effects of workers by assuming that workers assign labels randomly. Using (7), it can be shown that the (k, c)th entry of the confusion matrix of item cluster h is $\exp\left(\mathbf{t}_{ck}^\top \boldsymbol{\tau}_h^*\right) / \sum_k \exp\left(\mathbf{t}_{ck}^\top \boldsymbol{\sigma}_l^*\right)$. Similarly, for worker cluster l, the (k, c)th entry of its confusion matrix is $\exp\left(\mathbf{t}_{ck}^\top \boldsymbol{\sigma}_l^*\right) / \sum_k \exp\left(\mathbf{t}_{ck}^\top (\boldsymbol{\sigma}_l^*)\right)$.



(a) $w1$. (b) $w2$. (c) $w3$. (d) $i1$. (e) $i2$. (f) $i3$.

Fig. 2. True confusion matrices of the worker and item clusters.

We set the truncated numbers of clusters K_1, K_2 in COLA to 8, and $\boldsymbol{\mu}_0 = \mathbf{0}, \boldsymbol{\nu}_0 = \mathbf{0}$, $\boldsymbol{\Sigma}_0 = \frac{1}{\lambda_a}\mathbf{I}, \boldsymbol{\Omega}_0 = \frac{1}{\lambda_b}\mathbf{I}$, where $\lambda_b = \lambda_a M/N$, as in [26]. Parameters β_a, β_b, α and λ_a are tuned by maximizing the log-likelihood as in [26]. Latent variables are initialized in an non-informative manner: $\boldsymbol{\mu}_l = \mathbf{0}, \boldsymbol{\nu}_h = \mathbf{0}$, $\mathbf{r}_i^z = [1/K_1, \ldots, 1/K_1]^T$, $\mathbf{r}_j^u = [1/K_2, \ldots, 1/K_2]^T$, $\eta_{h,1} = 1, \eta_{h,2} = \beta_a$ $\gamma_{l,1} = 1, \gamma_{l,2} = \beta_b$, and $\mathbf{r}_j^Y$ is from the empirical probabilities of the observed labels. We compare the proposed algorithm with the following state-of-the-art:

1. Ordinal minimax entropy (OME) [26], with the hyperparameters tuned by the cross-validation method suggested in [26].
2. Ordinal mixture (ORDMIX) [16]: The predicted labels are obtained by discretizing (normally distributed) continuous-valued latent labels.
3. Dawid-Skene model (DS) [7]: A well-known approach for label aggregation, which estimates a confusion matrix for each worker.
4. Majority voting (MV) [14], which has been commonly used as a simple baseline.

To allow statistical significance testing, we learn the model using 90 % of the items and run each experiment for 10 repetitions. As in [16,26], the following measures are used: (i) mean squared error: MSE $= \frac{1}{|S|}\sum_{j\in S}((\mathbb{E}_Q[Y_j] - Y_j^*)^2)$, where S is the set of items with ground-truth labels Y_j^*'s; (ii) ℓ_0 error $= \frac{1}{|S|}\sum_{j\in S}\mathbb{E}_Q[\mathbb{I}(Y_j \neq Y_j^*)]$; (iii) ℓ_1 error $= \frac{1}{|S|}\sum_{j\in S}\mathbb{E}_Q[|Y_j - Y_j^*|]$; and (iv) ℓ_2 error $= \sqrt{\frac{1}{|S|}\sum_{j\in S}\mathbb{E}_Q[(Y_j - Y_j^*)^2]}$.

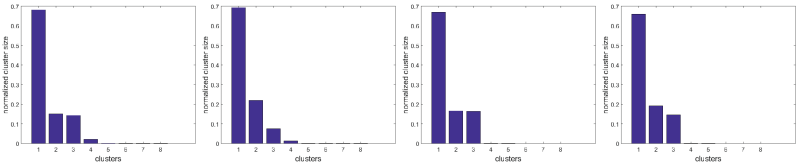
Results are shown in Table 2. As can be seen, on data set D1, the proposed COLA significantly outperforms the other methods. On D2, with more labels per item, inference becomes easier and all methods have improved performance. COLA is still better in terms of MSE and ℓ_2 error, but can be slightly outperformed by the simpler methods of DS and MV in terms of ℓ_0 and ℓ_1 errors.

Figure 3 shows the normalized sizes of the worker and item clusters obtained by COLA. Recall that $\mathbf{z}$ indicates the cluster memberships of workers. The normalized size of worker cluster l is defined as $\mathbb{E}(S_l^z)/\sum_{l=1}^{K_1}\mathbb{E}(S_l^z)$, where S_l^z is the size of worker cluster l, and $\mathbb{E}(S_l^z) = \mathbb{E}\left[\sum_{i=1}^N I(z_i = l)\right] = \sum_{i=1}^N P(z_i = l) = \sum_{i=1}^N r_{il}^z$. Similarly, the normalized size of item cluster h is $\mathbb{E}(S_h^u)/\sum_{h=1}^{K_2}\mathbb{E}(S_h^u)$, where S_h^u is the size of item cluster h, and $\mathbb{E}(S_h^u) = \sum_{j=1}^M r_{jh}^u$. As can be seen,

Table 2. Errors obtained on the synthetic data sets. The best results and those that are not statistically worse (using paired t-test at 95 % significance level) are in bold.

		COLA	OME	ORDMIX	DS	MV
D1	MSE	$\mathbf{0.249 \pm 0.006}$	0.314 ± 0.011	0.341 ± 0.025	0.446 ± 0.009	0.401 ± 0.006
	ℓ_0	$\mathbf{0.180 \pm 0.003}$	0.228 ± 0.001	0.229 ± 0.006	0.225 ± 0.003	0.225 ± 0.003
	ℓ_1	$\mathbf{0.209 \pm 0.002}$	0.284 ± 0.004	0.273 ± 0.012	0.289 ± 0.004	0.304 ± 0.005
	ℓ_2	$\mathbf{0.522 \pm 0.005}$	0.642 ± 0.009	0.616 ± 0.021	0.668 ± 0.007	0.717 ± 0.008
D2	MSE	$\mathbf{0.073 \pm 0.007}$	0.101 ± 0.005	0.112 ± 0.009	0.089 ± 0.013	0.268 ± 0.012
	ℓ_0	0.080 ± 0.010	0.81 ± 0.005	0.83 ± 0.013	0.074 ± 0.000	$\mathbf{0.073 \pm 0.004}$
	ℓ_1	0.081 ± 0.010	0.82 ± 0.004	0.84 ± 0.013	$\mathbf{0.079 \pm 0.004}$	0.083 ± 0.006
	ℓ_2	$\mathbf{0.282 \pm 0.017}$	0.310 ± 0.013	0.315 ± 0.018	0.298 ± 0.022	0.319 ± 0.020

the sizes of the three dominant worker clusters are close to the ground truth on both data sets. However, the item cluster (normalized) sizes on D1 are less accurate than those on D2. This is due to that each item in D1 only has 6 labels, while each item in D2 has 30 (in comparison, each worker on average has 24 labels for D1 and 30 labels for D2).



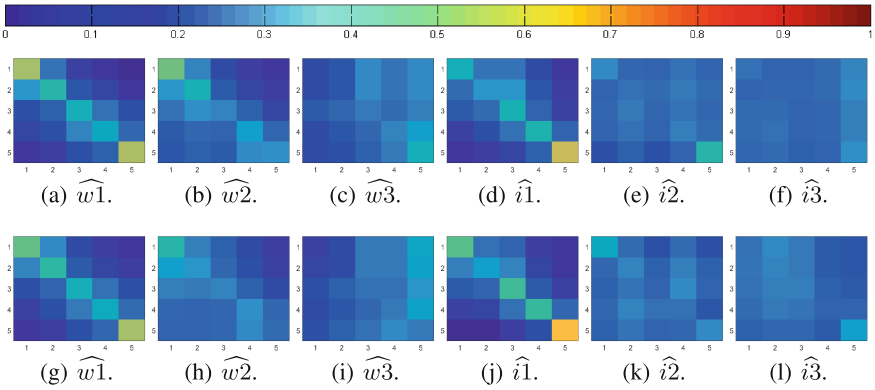
(a) worker (D1). (b) item (D1). (c) worker (D2). (d) item (D2).

Fig. 3. Normalized sizes of the worker and item clusters on D1 (left) and D2 (right). The true normalized sizes of the worker and item clusters are 0.667,0.167,0.167.

Next, we show the confusion matrices of the obtained clusters. Since we only have the distributions of $\boldsymbol{\sigma}^*$ and $\boldsymbol{\tau}^*$, we will use their expectations. Note that $\mathbb{E}[\boldsymbol{\sigma}_l^*] = \boldsymbol{\mu}_l$. From (7), the (c, k)th entry of the confusion matrix of worker cluster l can be represented as $\exp[\mathbf{t}_{ck}^\top \boldsymbol{\mu}_l] / \sum_k \exp[\mathbf{t}_{ck}^\top \boldsymbol{\mu}_l]$, and similarly, that of the item cluster h is $\exp[\mathbf{t}_{ck}^\top \boldsymbol{\nu}_h] / \sum_k \exp[\mathbf{t}_{ck}^\top \boldsymbol{\nu}_h]$. The obtained confusion matrices for worker and item clusters are shown in Fig. 4. Here, we focus on the three largest worker/item clusters, which can be seen to dominate in Fig. 3. Comparing with the ground-truth in Fig. 2, the 3 worker and item clusters can be well detected. Note again that the item clusters obtained on D2 are more accurate than those on D1, as each item in D2 has more labels for inference.

4.2 Real Data Sets

In this section, experiments are performed on three commonly-used data sets (Table 3).



(a) $\widehat{w1}$. (b) $\widehat{w2}$. (c) $\widehat{w3}$. (d) $\widehat{i1}$. (e) $\widehat{i2}$. (f) $\widehat{i3}$.

(g) $\widehat{w1}$. (h) $\widehat{w2}$. (i) $\widehat{w3}$. (j) $\widehat{i1}$. (k) $\widehat{i2}$. (l) $\widehat{i3}$.

Fig. 4. Confusion matrices of the obtained worker/item clusters on D1 (top) and D2 (bottom).

Table 3. Summary of the data sets used.

	#items		#workers	#classes	#observed labels	#labels/worker			#labels/item		
	total	w/ground truth				min	mean	max	min	mean	max
AC2	11,040	333	825	4	89,799	1	108.8	7551	1	8.1	27
TREC	19,721	3,250	762	3	90,244	1	118.4	7467	1	4.6	34
WEB	2,665	2,665	177	5	15,567	1	87.9	1225	1	5.8	12

1. AC2 [10]: This contains AMT judgments for website ratings, with the 4 levels: "G", "PG", "R", and "X";
2. TREC [6]: This is a web search data set, with the 3 levels: "NR" (non-relevant), "R" (relevant) and "HR" (highly relevant);
3. WEB [26]: This is another web search relevance data set, with the 5 levels: "P" (perfect), "E" (excellent), "G" (good), "F" (fair) and "B" (bad).

The ordinal labels are converted to numbers (e.g., on the AC2 data set, "G", "PG", "R", and "X" are converted to 1, 2, 3, 4, respectively). As can be seen from Table 3, the number of labels provided by the workers can vary significantly.

For the proposed algorithm, we set the truncated numbers of clusters K_1, K_2 to 10 (on WEB, $K_1 = 15$). Larger values do not improve performance. The other parameters of all the algorithms are set as in Sect. 4.1. To allow statistical significance testing, again we learn the model using 90 % of the items[2], and repeat this process 10 times.

Results are shown in Table 4. As can be seen, COLA consistently outperforms all the other methods on AC2 and WEB. Moreover, ORDMIX is competitive with COLA on TREC, but much inferior on AC2 and WEB. As AC2 and

[2] For AC2 and TREC, since performance can only be evaluated on items with ground-truth labels and these two data sets have fewer such items, all these items (with ground-truth labels) are always selected into the 90 % subset.

Table 4. Errors obtained on the real-world data sets. The best results and those that are not statistically worse (using paired t-test at 95 % significance level) are in bold.

		COLA	OME	ORDMIX	DS	MV
AC2	MSE	**0.262 ± 0.003**	0.317 ± 0.001	0.364 ± 0.028	0.302 ± 0.007	0.292 ± 0.000
	ℓ_0	**0.228 ± 0.002**	**0.230 ± 0.004**	0.279 ± 0.015	0.271 ± 0.006	0.241 ± 0.000
	ℓ_1	**0.245 ± 0.002**	0.255 ± 0.005	0.348 ± 0.030	0.283 ± 0.006	0.297 ± 0.000
	ℓ_2	**0.513 ± 0.005**	0.546 ± 0.005	0.712 ± 0.053	0.564 ± 0.007	0.643 ± 0.000
TREC	MSE	0.641 ± 0.006	0.679 ± 0.001	**0.603 ± 0.019**	0.750 ± 0.004	0.649 ± 0.000
	ℓ_0	**0.492 ± 0.003**	**0.495 ± 0.001**	0.557 ± 0.006	0.513 ± 0.003	0.543 ± 0.000
	ℓ_1	**0.602 ± 0.004**	0.615 ± 0.002	**0.606 ± 0.011**	0.635 ± 0.003	0.661 ± 0.000
	ℓ_2	0.886 ± 0.005	0.924 ± 0.003	**0.838 ± 0.013**	0.938 ± 0.004	0.947 ± 0.000
WEB	MSE	**0.105 ± 0.003**	**0.106 ± 0.003**	0.360 ± 0.032	0.230 ± 0.005	0.517 ± 0.004
	ℓ_0	**0.096 ± 0.003**	0.103 ± 0.004	0.194 ± 0.003	0.169 ± 0.006	0.269 ± 0.002
	ℓ_1	**0.108 ± 0.004**	0.117 ± 0.004	0.242 ± 0.010	0.204 ± 0.006	0.425 ± 0.004
	ℓ_2	**0.369 ± 0.007**	0.381 ± 0.008	0.633 ± 0.024	0.534 ± 0.008	0.923 ± 0.006

WEB have more label classes than TREC (Table 3), ORDMIX, which has fewer parameters and is less flexible than COLA, is unable to sufficiently model the confusion matrices of workers and items. The performance of DS is also poor, as ordinal information of the labels is not utilized. Finally, as expected, the simple MV performs the worst overall.

Figure 5(a)–(j) show the confusion matrices of worker clusters obtained on AC2. For most of them ($\widehat{w1} - \widehat{w8}$), the diagonal values for "G" and "X" are high, indicating that most clusters can identify these two types of websites easily. For the largest worker cluster ($\widehat{w1}$), the highest value on each row lies on the diagonal, and so the labels assigned by this cluster are mostly consistent with the ground truth. As for cluster $\widehat{w5}$, the diagonal entries are much larger than the non-diagonal ones. Hence, the worker labels are often the same as the ground truth, suggesting that these workers are experts. This is also confirmed in Table 5, which shows the ℓ_2 error for each cluster. On the other hand, workers in cluster $\widehat{w9}$ almost always predict "G". They are likely to be spammers as observed in many crowdsourcing platforms [18]. In cluster $\widehat{w10}$, the off-diagonal values are larger than the diagonal ones, indicating that workers in this cluster may not understand this website rating task or may even be malicious.

Table 5. ℓ_2 errors for worker clusters obtained on the AC2 data set.

Cluster	$\widehat{w1}$	$\widehat{w2}$	$\widehat{w3}$	$\widehat{w4}$	$\widehat{w5}$	$\widehat{w6}$	$\widehat{w7}$	$\widehat{w8}$	$\widehat{w9}$	$\widehat{w10}$
ℓ_2 error	0.714	0.627	0.789	0.938	0.618	0.922	0.732	0.950	1.133	1.445

Figure 5(k)–(o) show the confusion matrices of the obtained item clusters. In general, as each item has fewer labels than each worker, the clustering structure here is less obvious (as discussed in Sect. 4.1). For the item cluster $\widehat{i1}$, the diagnal

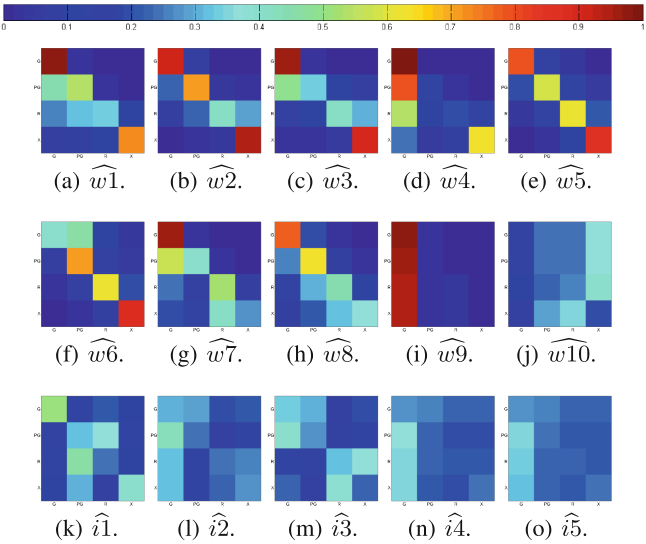


Fig. 5. Confusion matrices for worker clusters (top two rows) and item clusters (bottom row) obtained by COLA on the AC2 data set (clusters are ordered by decreasing size). In each cluster, columns are the cluster-assigned labels (left-to-right: "G", "PG", "R", "X"), and rows are the true labels (top-to-down: "G", "PG", "R", "X"). The five smallest item clusters occupy less than 2 % of the total size, and so are not shown.

elements have high values, indicating that items belonging to this cluster are relatively easy to distinguish. Item cluster $\widehat{i2}$ tends to assign label "G" more often. In $\widehat{i3}$, "G" and "PG" are sometimes confused, and so are "R" and "X".

Varying the Number of Items. In this experiment, we use item subsets of different sizes to learn the model. With a smaller number of items, the number of labels per worker is also reduced (Fig. 6), and estimating the workers' behavior become more difficult. Here, we focus on the two top performers in Table 4, namely, COLA and OME. Figure 7(a)–(i) show the errors averaged over 10 repetitions. As can be seen, as OME does not consider any structure among workers and items, its performance deteriorates significantly with fewer worker labels. On the other hand, COLA clusters workers and items. Thus, information within a cluster can be shared, and the performance is less affected.

Varying the Concentration Parameters. In this experiment, we study the effect of the DP's concentration parameters on the performance of COLA. In general, a smaller concentration parameter encourages fewer clusters, and vice verse. We first fix β_b to the value obtained in the previous experiment, and vary β_a from 0.1 to 3.5.

Figure 8(a) shows how the ℓ_2 error varies with β_a. Because of the lack of space, we only show results on the AC2 data set. COLA has stable performance over a

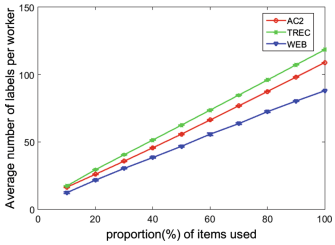
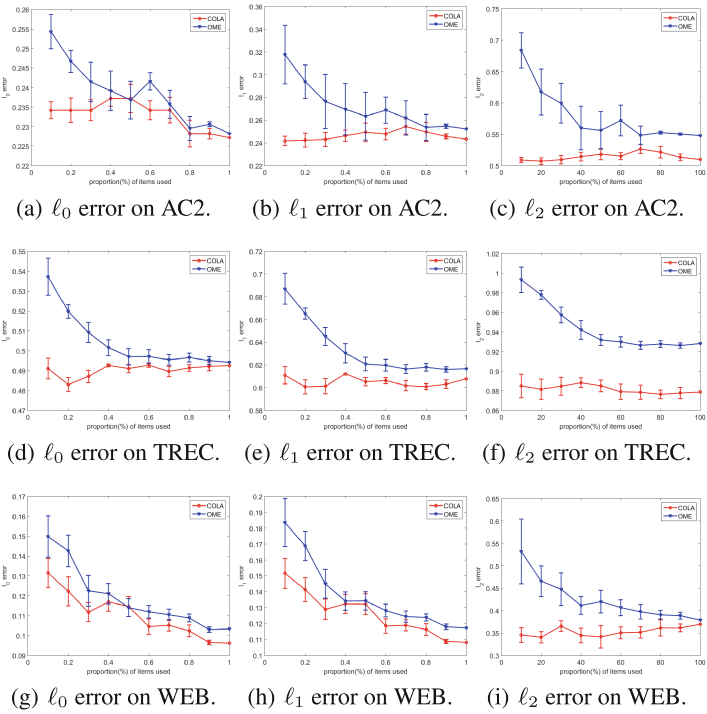


Fig. 6. Number of labels per worker with different numbers of items.



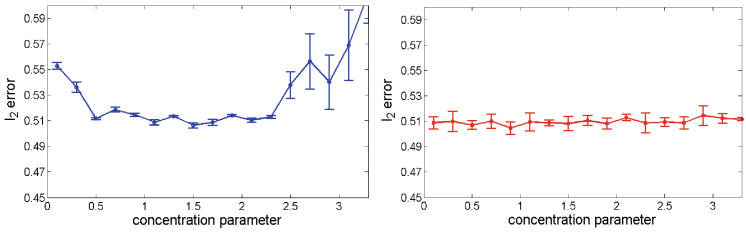
(a) ℓ_0 error on AC2. (b) ℓ_1 error on AC2. (c) ℓ_2 error on AC2.

(d) ℓ_0 error on TREC. (e) ℓ_1 error on TREC. (f) ℓ_2 error on TREC.

(g) ℓ_0 error on WEB. (h) ℓ_1 error on WEB. (i) ℓ_2 error on WEB.

Fig. 7. ℓ_0, ℓ_1 and ℓ_2 errors for COLA and OME with different proportions(%) of items used.

wide range of β_a. When β_a becomes too small, workers can only form very few clusters, and each cluster may not be coherent. When β_a is too large, clusters are split, and each cluster may not have enough data for accurate parameter estimation. Figure 8(b) shows the results on varying β_b. As can be seen, β_b has little influence on the performance. Again, this is consistent with the observation in the previous section that items' cluster structure is more difficult to identify.



(a) Varying β_a. (b) Varying β_b.

Fig. 8. ℓ_2 errors of COLA on the AC2 data set, with different values for β_a, β_b.

5 Conclusion

In this paper, we proposed a Bayesian clustering model to aggregate crowdsourced ordinal labels. Using the Dirichlet process, we encourage the formation of worker and item clusters in the label generating process, which leads to more accurate label estimation. While the probability model is complex and uses non-conjugate DPs, we derive an efficient variational inference procedure to infer the posterior distributions. Experimental results show that the proposed method yields significantly better accuracy than the state-of-the-art, and is more robust to sparser labels. Moreover, it detects meaningful clusters, which can help the user to study the group's behavior.

Acknowledgements. This research was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region (614513).

References

1. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Mach. Learn. **50**(1–2), 5–43 (2003)
2. Bi, W., Wang, L., Kwok, J.T., Tu, Z.: Learning to predict from crowdsourced data. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, pp. 72–82 (2014)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
4. Blei, D.M., Jordan, M.I.: Variational inference for Dirichlet process mixtures. Bayesian Anal. **1**(1), 121–143 (2006)
5. Braun, M., McAuliffe, J.: Variational inference for large-scale models of discrete choice. J. Am. Stat. Assoc. **105**(489), 324–335 (2010)
6. Buckley, C., Lease, M., Smucker, M.D.: Overview of the TREC 2010 relevance feedback track (Notebook). In: Proceedings of the Nineteenth Text Retrieval Conference Notebook (2010)
7. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. Appl. Stat. **28**(1), 20–28 (1979)
8. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)

9. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. Ann. Stat. **1**, 209–230 (1973)
10. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on Amazon Mechanical Turk. In: Proceedings of the SIGKDD Workshop on Human Computation, pp. 64–67 (2010)
11. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)
12. Kajino, H., Tsuboi, Y., Kashima, H.: Clustering crowds. In: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, pp. 1120–1127 (2013)
13. Kamar, E., Kapoor, A., Horvitz, E.: Identifying and accounting for task-dependent bias in crowdsourcing. In: Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, pp. 92–101 (2015)
14. Kazai, G., Kamps, J., Koolen, M., Milic-Frayling, N.: Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In: Proceedings of the 34th International Conference on Research and Development in Information Retrieval, pp. 205–224 (2011)
15. Lakkaraju, H., Leskovec, J., Kleinberg, J., Mullainathan, S.: A Bayesian framework for modeling human evaluations. In: Proceedings of SIAM International Conference on Data Mining, pp. 181–189 (2015)
16. Lakshminarayanan, B., Teh, Y.: Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. Technical report arXiv:1305.0015 (2013)
17. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**(151), 773–782 (1980)
18. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. J. Mach. Learn. Res. **13**, 491–518 (2012)
19. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. J. Mach. Learn. Res. **11**, 1297–1322 (2010)
20. Sethuraman, J.: A constructive definition of Dirichlet priors. Stat. Sin. **4**, 639–650 (1994)
21. Venanzi, M., Guiver, J., Kazai, G., Kohli, P., Shokouhi, M.: Community-based Bayesian aggregation models for crowdsourcing. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 155–164 (2014)
22. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Found. Trends Mach. Learn. **1**(1–2), 1–305 (2008)
23. Welinder, P., Branson, S., Belongie, S.J., Perona, P.: The multidimensional wisdom of crowds. In: Advances in Neural Information Processing Systems, pp. 2424–2432 (2010)
24. Yan, Y., Rosales, R., Fung, G., Schmidt, M.W., Valadez, G.H., Bogoni, L., Moy, L., Dy, J.G.: Modeling annotator expertise: learning when everybody knows a bit of something. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, pp. 932–939 (2010)
25. Zhou, D., Basu, S., Mao, Y., Platt, J.C.: Learning from the wisdom of crowds by minimax entropy. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2012)
26. Zhou, D., Liu, Q., Platt, J., Meek, C.: Aggregating ordinal labels from crowds by minimax conditional entropy. In: Proceedings of the 31st International Conference on Machine Learning, pp. 262–270 (2014)