

# Bayesian Crowdsourcing with Constraints <sup>\*</sup>

Panagiotis A. Traganitis ✉ and Georgios B. Giannakis

Electrical & Computer Engineering Department, and Digital Technology Center,  
University of Minnesota, Minneapolis MN 55414, USA  
{traga003,georgios}@umn.edu.

**Abstract.** Crowdsourcing has emerged as a powerful paradigm for efficiently labeling large datasets and performing various learning tasks, by leveraging crowds of human annotators. When additional information is available about the data, constrained or semi-supervised crowdsourcing approaches that enhance the aggregation of labels from human annotators are well motivated. This work deals with constrained crowdsourced classification with instance-level constraints, that capture relationships between pairs of data. A Bayesian algorithm based on variational inference is developed, and its quantifiably improved performance, compared to unsupervised crowdsourcing, is analytically and empirically validated on several crowdsourcing datasets.

**Keywords:** Crowdsourcing · Variational inference · Bayesian · Semi-supervised · Ensemble learning

## 1 Introduction

Crowdsourcing, as the name suggests, harnesses crowds of human annotators, using services such as Amazon’s Mechanical Turk [16], to perform various learning tasks such as labeling, image tagging and natural language annotations, among others [11]. Even though crowdsourcing can be efficient and relatively inexpensive, inference of true labels from the noisy responses provided by multiple annotators of unknown expertise can be challenging, especially in the typical *unsupervised* scenario, where no ground-truth data is available.

It is thus practical to look for side information that can be beneficial to the crowdsourcing task, either from experts or from physical constraints associated with the task. For example, queries with known answers may be injected in crowdsourcing tasks in Amazon’s Mechanical Turk, in order to assist with the evaluation of annotator reliability. Methods that leverage such information fall under the constrained or semi-supervised learning paradigm [5]. Seeking improved performance in the crowdsourcing task, we focus on constrained crowdsourcing, and investigate instance-level or pairwise constraints, such as must- and cannot-link constraints, which show up as side information in unsupervised tasks, such as clustering [3]. Compared to label constraints, instance-level ones provide ‘weaker information,’ as they describe relationships between pairs of data, instead

---

<sup>\*</sup> Work in this paper was supported by NSF grant 1901134.

of anchoring the label of a datum. Semi-supervised learning with few label constraints is typically employed when acquiring ground-truth labels is time consuming or expensive, e.g. annotation of biomedical images from a medical professional. Instance-level constraints on the other hand are used when domain knowledge is easier to encode between pairs of data, e.g. road lane identification from GPS data.

To accommodate such available information, the present work capitalizes on recent advances in Bayesian inference. Relative to deterministic methods, the Bayesian approach allows for seamless integration of prior information for the crowdsourcing task, such as annotator performance from previous tasks, and at the same time enables uncertainty quantification of the fused label estimates and parameters of interest. Our major contributions can be summarized as follows: (i) We develop a Bayesian algorithm for crowdsourcing with pairwise constraints; (ii) We derive novel error bounds for the unsupervised variational Bayes crowdsourcing algorithm, in Thm. 1. These error bounds are extended for the proposed constrained algorithm in Thm. 2; and (iii) Guided by the aforementioned theoretical analysis, we provide a constraint selection scheme. In addition to the error bounds, the performance of the proposed algorithm is evaluated with extensive numerical tests. Those corroborate that there are classification performance gains to be harnessed, even when using weaker side information, such as the aforementioned must- and cannot-link constraints.

**Notation:** Unless otherwise noted, lowercase bold fonts,  $\mathbf{x}$ , denote vectors, uppercase ones,  $\mathbf{X}$ , represent matrices, and calligraphic uppercase,  $\mathcal{X}$ , stand for sets. The  $(i, j)$ th entry of matrix  $\mathbf{X}$  is denoted by  $[\mathbf{X}]_{ij}$ .  $\Pr$  or  $p$  denotes probability, or the probability mass function;  $\sim$  denotes “distributed as,”  $|\mathcal{X}|$  is the cardinality of set  $\mathcal{X}$ ,  $\mathbb{E}[\cdot]$  denotes expectation, and  $\mathbb{1}(\mathcal{A})$  is the indicator function for the event  $\mathcal{A}$ , that takes value 1, when  $\mathcal{A}$  occurs, and is 0 otherwise.

## 2 Problem formulation and preliminaries

Consider a dataset consisting of  $N$  data  $\{x_n\}_{n=1}^N$ , with each datum belonging to one of  $K$  classes with corresponding labels  $\{y_n\}_{n=1}^N$ ; that is,  $y_n = k$  if  $x_n$  belongs to class  $k$ . Consider now  $M$  annotators that observe  $\{x_n\}_{n=1}^N$ , and provide estimates of labels. Let  $\check{y}_n^{(m)} \in \{1, \dots, K\}$  be the label estimate of  $x_n$  assigned by the  $m$ -th annotator. If an annotator has not provided a response for datum  $n$  we set  $\check{y}_n^{(m)} = 0$ . Let  $\check{\mathbf{Y}}$  be the  $M \times N$  matrix of annotator responses with entries  $[\check{\mathbf{Y}}]_{mn} = \check{y}_n^{(m)}$ , and  $\mathbf{y} := [y_1, \dots, y_N]^\top$  the  $N \times 1$  vector of ground truth labels. The task of *crowdsourced classification* is: Given only the annotator responses in  $\check{\mathbf{Y}}$ , the goal is find the ground-truth label estimates  $\{\hat{y}_n\}_{n=1}^N$ .

Per datum  $x_n$ , the true label  $y_n$  is assumed drawn from a categorical distribution with parameters  $\boldsymbol{\pi} := [\pi_1, \dots, \pi_K]^\top$ , where  $\pi_k := \Pr(y_n = k)$ . Further, consider that each learner has a fixed probability of deciding that a datum belongs to class  $k'$ , when presented with a datum of class  $k$ ; thus, annotator behavior is presumed invariant across the dataset. The performance of an annotator  $m$  is then characterized by the so-called *confusion* matrix  $\boldsymbol{\Gamma}^{(m)}$ , whose

$(k, k')$ -th entry is  $[\mathbf{\Gamma}^{(m)}]_{k,k'} := \gamma_{k,k'}^{(m)} = \Pr(\tilde{y}_n^{(m)} = k' | y_n = k)$ . The  $K \times K$  confusion matrix showcases the statistical behavior of an annotator, as each row provides the annotator’s probability of deciding the correct class, when presented with a datum from each class. Collect all annotator confusion matrices in  $\mathbf{\Gamma} := [\mathbf{\Gamma}^{(1)}, \dots, \mathbf{\Gamma}^{(m)}]$ . Responses of different annotators per datum  $n$  are presumed conditionally independent, given the ground-truth label  $y_n$ ; that is,  $p(\tilde{y}_n^{(1)} = k_1, \dots, \tilde{y}_n^{(M)} = k_M | y_n = k) = \prod_{m=1}^M p(\tilde{y}_n^{(m)} = k_m | y_n = k)$ . The latter is a standard assumption that is commonly employed in crowdsourcing works [7, 13, 36]. Finally, most annotators are assumed to be better than random.

## 2.1 Prior works

Arguably the simplest approach to fusing crowdsourced labels is majority voting, where the estimated label of a datum is the one most annotators agree upon. This presumes that all annotators are equally “reliable,” which may be unrealistic. Aiming at high-performance label fusion, several approaches estimate annotator parameters, meaning the confusion matrices as well as class priors. A popular approach is joint maximum likelihood (ML) estimation of the unknown labels  $\mathbf{y}$ , and the aforementioned confusion matrices using the expectation-maximization (EM) algorithm [7]. As EM guarantees convergence to a locally optimal solution, alternative estimation methods have been recently advocated. Spectral methods invoke second- and third-order moments of annotator responses to infer the unknown annotator parameters [13, 36, 31, 12].

A Bayesian treatment of crowdsourced learning, termed *Bayesian Classifier Combination* was introduced in [15]. This approach used Gibbs sampling to estimate the parameters of interest, while [28] introduced a variational Bayes EM (VBEM) method for the same task. Other Bayesian approaches infer communities of annotators, enhancing the quality of aggregated labels [34, 19]. For sequential or networked data, [23, 20, 32, 33] advocated variational inference and EM-based alternatives. All aforementioned approaches utilize only  $\tilde{\mathbf{Y}}$ . When features  $\{x_n\}_{n=1}^N$  are also available, parametric models [22], approaches based on Gaussian Processes [26, 25], or deep learning [1, 24, 27, 2] can be employed to classify the data, and simultaneously learn a classifier.

Current constrained or semi-supervised approaches to crowdsourcing extend the EM algorithm of Dawid and Skene [30], by including a few ground-truth labels. The model in [35] includes a graph between datapoints to enable label prediction for data that have not been annotated, while [17] puts forth a max-margin majority vote method when label constraints are available. When features  $\{x_n\}_{n=1}^N$  are available, [14] relies on a parametric model to learn a binary classifier from crowd responses and expert labels.

The present work develops a Bayesian *constrained* approach to crowdsourced classification, by adapting the popular variational inference framework [4]. In addition, it provides novel performance analysis for both Bayesian unsupervised and semi-supervised approaches. The proposed method does not require features

$\{x_n\}_{n=1}^N$ , but relies only on annotator labels – a first attempt at incorporating instance-level constraints for the crowdsourced classification task.

### 3 Variational inference for crowdsourcing

Before presenting our constrained approach, this section will recap a Bayesian treatment of the crowdsourcing problem. First, variational Bayes is presented in Sec. 3.1, followed by an inference algorithm for crowdsourcing [28] in Sec. 3.2. A novel performance analysis for the VBEM algorithm is also presented in Sec. 3.2.

#### 3.1 Variational Bayes

Consider a set of observed data collected in  $\mathbf{X}$ , and a set of latent variables and parameters in  $\mathbf{Z}$  that depend on  $\mathbf{X}$ . Variational Bayes seeks  $\mathbf{Z}$  that maximize the marginal of  $\mathbf{X}$ , by treating both latent variables and parameters as random [4]. EM in contrast treats  $\mathbf{Z}$  as deterministic and provides point estimates. The log-marginal of  $\mathbf{X}$  can be written as

$$\ln p(\mathbf{X}) = \int q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z} - \int q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z} = L(q) + \text{KL}(q||p) \quad (1)$$

where  $L(q) := \int q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z}$ , and  $\text{KL}(q||p) = - \int q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z}$  denotes the Kullback-Leibler divergence between pdfs  $q$  and  $p$  [6]. This expression is maximized for  $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X})$ , however when  $p(\mathbf{Z}|\mathbf{X})$  is an intractable pdf, one may seek distributions  $q$  from a prescribed tractable family  $\mathcal{Q}$ , such that  $\text{KL}(q||p)$  is minimized. One such family is the family of factorized distributions, which also go by the name of mean field distributions. Under the mean field paradigm, the variational distribution  $q(\mathbf{Z})$  is decomposed into a product of single variable factors,  $q(\mathbf{Z}) = \prod_i q(\mathbf{Z}_i)$ , with  $q(\mathbf{Z}_i)$  denoting the variational distribution corresponding to the variable  $\mathbf{Z}_i$ .

It can be shown that the optimal updates for each factor are given by [4]

$$\ln q^*(\mathbf{Z}_i) = \mathbb{E}_{-\mathbf{Z}_i} [\ln p(\mathbf{X}, \mathbf{Z})] + c \quad (2)$$

where  $c$  is an appropriate constant, and the  $-\mathbf{Z}_i$  subscript denotes that the expectation is taken w.r.t. the terms in  $q$  that do not involve  $\mathbf{Z}_i$ , that is  $\prod_{j \neq i} q(\mathbf{Z}_j)$ . These optimal factors can be estimated iteratively, and such a procedure is guaranteed to converge at least to a local maximum of (1).

#### 3.2 Variational EM for crowdsourcing

Next, we will outline how variational inference can be used to derive an iterative algorithm for crowdsourced classification [28]. The Bayesian treatment of the crowdsourcing problem dictates the use of prior distributions on the parameters of interest, namely  $\boldsymbol{\pi}$ , and  $\boldsymbol{\Gamma}$ . The probabilities  $\boldsymbol{\pi}$  are assigned a Dirichlet distribution

prior with parameters  $\boldsymbol{\alpha}_0 := [\alpha_{0,1}, \dots, \alpha_{0,K}]^\top$ , that is  $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}_0)$ , whereas for an annotator  $m$ , the columns  $\{\boldsymbol{\gamma}_k^{(m)}\}_{k=1}^K$  of its confusion matrix are considered independent, and  $\boldsymbol{\gamma}_k^{(m)}$  is assigned a Dirichlet distribution prior with parameters  $\boldsymbol{\beta}_{0,k}^{(m)} := [\beta_{0,k,1}^{(m)}, \dots, \beta_{0,k,K}^{(m)}]^\top$ , respectively. These priors on the parameters of interest are especially useful when only few data have been annotated, and can also capture annotator behavior from previous crowdsourcing tasks. The joint distribution of  $\mathbf{y}, \check{\mathbf{Y}}, \boldsymbol{\pi}$  and  $\boldsymbol{\Gamma}$  is

$$p(\mathbf{y}, \boldsymbol{\pi}, \check{\mathbf{Y}}, \boldsymbol{\Gamma}; \boldsymbol{\alpha}_0, \mathbf{B}_0) = \prod_{n=1}^N \pi_{y_n} \prod_{m=1}^M \prod_{k'=1}^K (\gamma_{y_n, k'}^{(m)})^{\delta_{n, k'}^{(m)}} p(\boldsymbol{\pi}; \boldsymbol{\alpha}_0) p(\boldsymbol{\Gamma}; \mathbf{B}_0) \quad (3)$$

with  $\mathbf{B}_0$  collecting all prior parameters  $\boldsymbol{\beta}_{0,k}^{(m)}$  for  $k = 1, \dots, K$  and  $m = 1, \dots, M$ , and  $\delta_{n, k}^{(m)} := \mathbf{1}(\check{y}_n^{(m)} = k)$ . The parametrization on  $\boldsymbol{\alpha}_0, \mathbf{B}_0$  will be henceforth implicit for brevity.

With the goal of estimating the unknown variables and parameters of interest, we will use the approach outlined in the previous subsection, to approximate  $p(\mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\Gamma} | \check{\mathbf{Y}})$  using a variational distribution  $q(\mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\Gamma})$ . Under the mean-field class, this variational distribution factors across the unknown variables as  $q(\mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\Gamma}) = q(\mathbf{y})q(\boldsymbol{\pi})q(\boldsymbol{\Gamma})$ . Since the data are assumed i.i.d. [cf. Sec. 2],  $q(\mathbf{y})$  is further decomposed into  $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$ . In addition, since annotators are assumed independent and the columns of their confusion matrices are also independent, we have  $q(\boldsymbol{\Gamma}) = \prod_{m=1}^M \prod_{k=1}^K q(\boldsymbol{\gamma}_k^{(m)})$ . The variational Bayes EM algorithm is an iterative algorithm, with each iteration consisting of two steps: the variational E-step, where the latent variables  $\mathbf{y}$  are estimated; and the variational M-step, where the distribution of the parameters of interest  $\boldsymbol{\pi}, \boldsymbol{\Gamma}$  are estimated.

At iteration  $t + 1$ , the variational distribution for the unknown label  $y_n$  is given by

$$\ln q_{t+1}(y_n) = \mathbb{E}_{-y_n} [\ln p(\mathbf{y}, \boldsymbol{\pi}, \check{\mathbf{Y}}, \boldsymbol{\Gamma})] + c. \quad (4)$$

The subscript of  $q$  denotes the iteration index. The expectation in (4) is taken w.r.t. the terms in  $q_t$  that do not involve  $y_n$ , that is  $\prod_{n' \neq n} q_t(y_{n'}) q_t(\boldsymbol{\pi}) q_t(\boldsymbol{\Gamma})$ . Upon expanding  $p(\mathbf{y}, \boldsymbol{\pi}, \check{\mathbf{Y}}, \boldsymbol{\Gamma})$ , (4) becomes

$$\ln q_{t+1}(y_n) = \mathbb{E}_{\boldsymbol{\pi}} [\ln \pi_{y_n}] + \mathbb{E}_{\boldsymbol{\Gamma}} \left[ \sum_{m=1}^M \sum_{k'=1}^K \delta_{n, k'}^{(m)} \ln \gamma_{y_n, k'}^{(m)} \right] + c. \quad (5)$$

Accordingly, the update for the class priors in  $\boldsymbol{\pi}$  is

$$\ln q_{t+1}(\boldsymbol{\pi}) = \mathbb{E}_{-\boldsymbol{\pi}} [\ln p(\mathbf{y}, \boldsymbol{\pi}, \check{\mathbf{Y}}, \boldsymbol{\Gamma})] + c \quad (6)$$

where the expectation is taken w.r.t.  $q_{t+1}(\mathbf{y}) q_t(\boldsymbol{\Gamma})$ . Based on (6), it can be shown [28] that

$$q_{t+1}(\boldsymbol{\pi}) \propto \text{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}_{t+1}) \quad (7)$$

where  $\boldsymbol{\alpha}_{t+1} := [\alpha_{t+1,1}, \dots, \alpha_{t+1,K}]^\top$  with  $\alpha_{t+1,k} = N_{t+1,k} + \alpha_{0,k}$ ,  $N_{t+1,k} := \sum_{n=1}^N q_{t+1}(y_n = k)$ . As a direct consequence of this, the term involving  $\boldsymbol{\pi}$  in (5)

is given by  $\mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_k] = \psi(\alpha_{t,k}) - \psi(\bar{\alpha}_t)$ , with  $\psi$  denoting the digamma function, and  $\bar{\alpha}_t := \sum_{k=1}^K \alpha_{t,k}$ .

For the  $k$ -th column of  $\boldsymbol{\Gamma}^{(m)}$  the update takes the form

$$\ln q_{t+1}(\boldsymbol{\gamma}_k^{(m)}) = \mathbb{E}_{-\boldsymbol{\gamma}_k^{(m)}} [\ln p(\mathbf{y}, \boldsymbol{\pi}, \tilde{\mathbf{Y}}, \boldsymbol{\Gamma})] + c \quad (8)$$

Using steps similar to the update of  $q(\boldsymbol{\pi})$ , one can show that

$$q_{t+1}(\boldsymbol{\gamma}_k^{(m)}) \propto \text{Dir}(\boldsymbol{\gamma}_k^{(m)}; \boldsymbol{\beta}_{t+1,k}^{(m)}) \quad (9)$$

with  $\boldsymbol{\beta}_{t+1,k}^{(m)} := [\beta_{t+1,k,1}^{(m)}, \dots, \beta_{t+1,k,1}^{(m)}]$ ,  $\beta_{t+1,k,k'}^{(m)} := N_{t+1,k,k'}^{(m)} + \beta_{0,k,k'}^{(m)}$  and  $N_{t+1,k,k'}^{(m)} := \sum_{n=1}^N q_{t+1}(y_n = k) \delta_{n,k'}^{(m)}$ . Consequently at iteration  $t+1$ , and upon defining  $\bar{\beta}_{t+1,k}^{(m)} := \sum_{\ell} \beta_{t+1,k,\ell}^{(m)}$ , we have  $\mathbb{E}_{\boldsymbol{\Gamma}}[\ln \gamma_{k,k'}^{(m)}] = \psi(\beta_{t+1,k,k'}^{(m)}) - \psi(\bar{\beta}_{t+1,k}^{(m)})$ . Given initial values for  $\mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_{y_n}]$  and  $\mathbb{E}_{\boldsymbol{\Gamma}}[\sum_{m=1}^M \ln \gamma_{y_n, \hat{y}_n^{(m)}}^{(m)}]$  per variational E-step, first the variational distribution for each datum  $q(y_n)$  is computed using (4). Using the recently computed  $q(\mathbf{y})$  at the variational M-step, the variational distribution for class priors  $q(\boldsymbol{\pi})$  is updated via (6) and the variational distributions for each column of the confusion matrices  $q(\boldsymbol{\gamma}_k^{(m)})$  are updated via (8). The E- and M-steps are repeated until convergence. Finally, the fused labels  $\hat{\mathbf{y}}$  are recovered at the last iteration  $T$  as

$$\hat{y}_n = \underset{k}{\operatorname{argmax}} q_T(y_n = k). \quad (10)$$

The overall computational complexity of this algorithm is  $\mathcal{O}(NKMT)$ . The updates of this VBEM algorithm are very similar to the corresponding updates of the EM algorithm for crowdsourcing [7].

**Performance analysis** Let  $\boldsymbol{\pi}^*$  comprise the optimal class priors,  $\boldsymbol{\Gamma}^*$  the optimal confusion matrices, and  $\mu_m := \Pr(\hat{y}_n^{(m)} \neq 0)$  the probability an annotator will provide a response. The following theorem establishes that when VBEM is properly initialized, the estimation errors for  $\boldsymbol{\pi}, \boldsymbol{\Gamma}$  and the labels  $\mathbf{y}$  are bounded. Such proper initializations can be achieved, for instance, using the spectral approaches of [36, 31, 12].

**Theorem 1.** *Suppose that  $\gamma_{k,k'}^{*(m)} \geq \rho_{\gamma}$  for all  $k, k', m$ ,  $\pi_k^* \geq \rho_{\pi}$  for all  $k$ , and that VBEM is initialized so that  $|\mathbb{E}[\pi_k] - \pi_k^*| \leq \varepsilon_{\pi,0}$  for all  $k$  and  $|\mathbb{E}[\gamma_{k,k'}^{(m)}] - \gamma_{k,k'}^{*(m)}| \leq \varepsilon_{\gamma,0}$  for all  $k, k', m$ . It then holds w.p. at least  $1 - \nu$  that for iterations  $t = 1, \dots, T$ , update (4) yields*

$$\begin{aligned} \max_k |q_t(y_n = k) - \mathbb{1}(y_n = k)| &\leq \varepsilon_{q,t} := K \exp(-U) \\ U &:= D + f_{\pi}(\varepsilon_{\pi,t-1}) + M f_{\gamma}(\varepsilon_{\gamma,t-1}), \end{aligned}$$

where  $D$  is a quantity related to properties of the dataset and the annotators, while  $f_\pi, f_\gamma$  are decreasing functions. Consequently for  $k, k' = 1, \dots, K$ ,  $m = 1, \dots, M$  updates (6) and (8) yield respectively

$$|\mathbb{E}[\pi_k] - \pi_k^*| \leq \epsilon_{\pi,k,t} := \frac{N(\varepsilon_{q,t} + g_\pi(\nu)) + \alpha_{0,k} + \rho_\pi \bar{\alpha}_0}{N + \bar{\alpha}_0} \quad (11)$$

$$|\mathbb{E}[\gamma_{k,k'}^{(m)}] - \gamma_{k,k'}^{*(m)}| \leq \epsilon_{\gamma,k,k',t}^{(m)} := \frac{2N(g_\gamma(\nu) + \varepsilon_{q,t}) + \beta_{0,k,k'}^{(m)} + \bar{\beta}_{0,k}^{(m)}}{N\mu_m\pi_k^* - N\frac{g_\gamma(\nu)}{\gamma_{k,k'}^{*(m)}} - N\varepsilon_{q,t} + \bar{\beta}_k^{(m)}} \quad (12)$$

with  $g_\pi(\nu)$  and  $g_\gamma(\nu)$  being decreasing functions of  $\nu$  and  $\varepsilon_{\pi,t} := \max_k \epsilon_{\pi,k,t}$ ,  $\varepsilon_{\gamma,t} := \max_{k,k',m} \epsilon_{\gamma,k,k',t}^{(m)}$ .

Detailed theorem proofs are deferred to Appendix A of the supplementary material. Theorem 1 shows that lowering the upper bound on label errors  $\varepsilon_q$ , will reduce the estimation error upper bounds of  $\{\gamma_{k,k'}^{(m)}\}$  and  $\pi$ . This in turn can further reduce  $\varepsilon_{q,t}$ , as it is proportional to  $\varepsilon_\pi$ , and  $\varepsilon_\gamma$ . With a VBEM algorithm for crowdsourcing and its performance analysis at hand, the ensuing section will introduce our constrained variational Bayes algorithm that can incorporate additional information to enhance label aggregation.

## 4 Constrained crowdsourcing

This section deals with a constrained (or semi-supervised) variational Bayes approach to crowdsourcing. Here, additional information to the crowdsourcing task is available in the form of pairwise constraints, that indicate relationships between pairs of data. Throughout this section,  $N_C$  will denote the number of available constraints that are collected in the set  $\mathcal{C}$ .

First, we note that when label constraints are available, the aforementioned VBEM algorithm can readily handle them, in a manner similar to [30]. Let  $\mathcal{C}$  denote the set of indices for data with label constraints  $\{y_n\}_{n \in \mathcal{C}}$  available. These constraints can then be incorporated by fixing the values of  $\{q(y_n)\}_{n \in \mathcal{C}}$  to 1 for all iterations. The variational distributions for data  $n \notin \mathcal{C}$  are updated according to (5), while confusion matrices and prior probabilities according to (8) and (6), respectively.

Next, we consider the case of instance-level constraints, which are the main focus of this work. Such information may be easier to obtain than the label constraints of the previous subsection, as pairwise constraints encapsulate relationships between pairs of data and not “hard” label information. The pairwise constraints considered here take the form of must-link and cannot-link relationships, and are collected respectively in the sets  $\mathcal{C}_{\text{ML}}$  and  $\mathcal{C}_{\text{CL}}$ , and  $\mathcal{C} = \mathcal{C}_{\text{ML}} \cup \mathcal{C}_{\text{CL}}$ ,  $N_{\text{ML}} = |\mathcal{C}_{\text{ML}}|$ ,  $N_{\text{CL}} = |\mathcal{C}_{\text{CL}}|$ . All constraints consist of tuples  $(i, j) \in \{1, \dots, N\} \times \{1, \dots, N\}$ . A must-link constraint  $(i, j)$ ,  $i \neq j$  indicates that two data points,  $\mathbf{x}_i$  and  $\mathbf{x}_j$  must belong to the same class, i.e.  $y_i = y_j$ , whereas a cannot-link constraint  $(i', j')$  indicates that two points  $\mathbf{x}_{i'}$  and  $\mathbf{x}_{j'}$  are

not in the same class;  $y_{i'} \neq y_{j'}$ . Note that instance level constraints naturally describe a graph  $\mathcal{G}$ , whose  $N$  nodes correspond to the data  $\{x_n\}_{n=1}^N$ , and whose (weighted) edges are the available constraints. This realization is the cornerstone of the proposed algorithm, suggesting that instance-level constraints can be incorporated in the crowdsourcing task by means of a probabilistic graphical model.

Specifically, we will encode constraints in the marginal pmf of the unknown labels  $p(\mathbf{y})$  using a Markov Random Field (MRF), which implies that for all  $n = 1, \dots, N$ , the local Markov property holds, that is  $p(y_n | \mathbf{y}_{-n}) = p(y_n | \mathbf{y}_{\mathcal{C}_n})$ , where  $\mathbf{y}_{-n}$  denotes a vector containing all labels except  $y_n$ , while  $\mathbf{y}_{\mathcal{C}_n}$  a vector containing labels for  $y_{n'}$ , where  $n' \in \mathcal{C}_n$ , and  $\mathcal{C}_n$  is a set containing indices such that  $n' : (n, n') \in \mathcal{C}$ . By the Hammersley-Clifford theorem [10], the marginal pmf of the unknown labels is given by

$$p(\mathbf{y}) = \frac{1}{Z} \exp \left( \sum_{n=1}^N \ln \pi_{y_n} + \eta \sum_{n' \in \mathcal{C}_n} V(y_n, y_{n'}) \right) \quad (13)$$

where  $Z$  is the (typically intractable) normalization constant, and  $\eta > 0$  is a tunable parameter that specifies how much weight we assign to the constraints. Here, we define the  $V$  function as  $V(y_n, y_{n'}) = w_{n,n'} \mathbb{1}(y_n = y_{n'})$ , where

$$w_{n,n'} = \mathbb{1}((n, n') \in \mathcal{C}_{\text{ML}}) - \mathbb{1}((n, n') \in \mathcal{C}_{\text{CL}}) \quad (14)$$

The weights  $w_{n,n'}$  can also take other real values, indicating the confidence one has per constraint  $(n, n')$ , but here we will focus on  $-1, 0, 1$  values. This  $V(y_n, y_{n'})$  term promotes the same label for data with must-link constraints, whereas it penalizes similar labels for data with cannot-link constraints. Note that more sophisticated choices for  $V$  can also be used. Nevertheless, this particular choice leads to a simple algorithm with quantifiable performance, as will be seen in the ensuing sections. Since the number of constraints is typically small, for the majority of data the term  $\sum V(y_n, y_{n'})$  will be 0. Thus, the  $\ln \pi$  term in the exponent of (13) acts similarly to the prior probabilities of Sec. 3.2. Again, adopting a Dirichlet prior  $\text{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}_0)$ , and  $\gamma_k^{(m)} \sim \text{Dir}(\gamma_k^{(m)}; \boldsymbol{\beta}_{0,k}^{(m)})$  for all  $m, k$ , and using mean-field VB as before, the variational update for the  $n$ -th label becomes

$$\begin{aligned} \ln q_{t+1}(y_n = k) &= \mathbb{E}_{\boldsymbol{\pi}} [\ln \pi_k] + c \\ &+ \mathbb{E}_{\boldsymbol{\Gamma}} \left[ \sum_{m=1}^M \sum_{k'=1}^K \delta_{n,k'}^{(m)} \ln \gamma_{k,k'}^{(m)} \right] + \eta \sum_{n' \in \mathcal{C}_n} w_{n,n'} q_t(y_{n'} = k). \end{aligned} \quad (15)$$

where we have used  $\mathbb{E}_{y_{n'}} [\mathbb{1}(y_{n'} = k)] = q_t(y_{n'} = k)$ . The variational update for label  $y_n$  is similar to the one in (5), with the addition of the  $\eta \sum_{n' \in \mathcal{C}_n} w_{n,n'} q_t(y_{n'} = k)$  term that captures the labels of the data that are related (through the instance constraints) to the  $n$ -th datum. Updating label distributions via (15), updates for  $\boldsymbol{\pi}$  and  $\boldsymbol{\Gamma}$  remain identical to those in VBEM [cf. Sec. 3.2]. The instance-level

---

**Algorithm 1** Crowdsourcing with instance-level constraints
 

---

**Input:** Annotator responses  $\tilde{\mathbf{Y}}$ , initial  $\boldsymbol{\pi}$ ,  $\boldsymbol{\Gamma}$ ,  $\{q_0(y_n = k)\}_{n,k=1}^{N,K}$ , constraints  $\mathcal{C}$ , parameters  $\boldsymbol{\alpha}_0$ ,  $\mathbf{B}_0$ ,  $\eta$   
**Output:** Estimates  $\mathbb{E}[\boldsymbol{\pi}]$ ,  $\mathbb{E}[\boldsymbol{\Gamma}]$ ,  $\hat{\mathbf{y}}$ .  
**while** not converged **do**  
     Update  $q_{t+1}(y_n)$  via (15)  
     Update  $q_{t+1}(\boldsymbol{\pi})$  using  $q_{t+1}(y_n)$ , (6)  
     Update  $q_{t+1}(\boldsymbol{\gamma}_k^{(m)}) \forall m, k$ , using  $q_{t+1}(y_n)$ , (8).  
      $t \leftarrow t + 1$   
**end while**  
 Estimate fused data labels using (10).

---

semi-supervised crowdsourcing algorithm is outlined in Alg. 1. As with its plain vanilla counterpart of Sec. 3.2, Alg. 1 maintains the asymptotic complexity of  $\mathcal{O}(NMKT)$ .

#### 4.1 Performance analysis

Let  $\tilde{\mathcal{C}}$  with cardinality  $|\tilde{\mathcal{C}}| = \tilde{N}_C$  be a set comprising indices of data that take part in at least one constraint, and let  $\tilde{\mathcal{C}}^c$  with  $|\tilde{\mathcal{C}}^c| = \bar{N}_C$  denote its complement. The next theorem quantifies the performance of Alg. 1.

**Theorem 2.** *Consider the same setup as in Thm. 1, instance level constraints collected in  $\mathcal{C}$ , and the initialization of Alg. 1 satisfying  $\max_k |q_0(y_n = k) - \mathbb{1}(y_n = k)| \leq \varepsilon_{q,0}$  for all  $k, n$ . Then the following hold w.p. at least  $1 - \nu$ : For iterations  $t = 1, \dots, T$ , update (15) yields for  $n \in \tilde{\mathcal{C}}$*

$$\begin{aligned}
 \max_k |q_t(y_n = k) - \mathbb{1}(y_n = k)| &\leq \tilde{\varepsilon}_{q,t} & (16) \\
 \tilde{\varepsilon}_{q,t} &:= \max_n K \exp(-U - \eta W_n) = \max_n \varepsilon_{q,t} \exp(-\eta W_n) \\
 W_n &:= N_{\text{ML},n}(1 - 2\varepsilon_{q,t-1}) - 2N_{\text{CL},n}\varepsilon_{q,t-1} + N_{\text{CL},n,\min}
 \end{aligned}$$

with  $N_{\text{ML},n}, N_{\text{CL},n}$  denoting the number of must- and cannot-link constraints for datum  $n$  respectively, and  $N_{\text{CL},n,\min} := \min_k N_{\text{CL},n,k}$ , where  $N_{\text{CL},n,k}$  is the number of cannot-link constraints of datum  $n$ , that belong to class  $k$ . For data without constraints, that is  $n \in \tilde{\mathcal{C}}^c$

$$\max_k |q_t(y_n = k) - \mathbb{1}(y_n = k)| \leq \varepsilon_{q,t} = K \exp(-U),$$

with  $U$  as defined in Thm. 1. For  $k, k' = 1, \dots, K$ ,  $m = 1, \dots, M$  updates (6) and (8) yield respectively

$$|\mathbb{E}[\pi_k] - \pi_k^*| \leq \epsilon_{\pi,k,t} := \frac{\tilde{N}_C \tilde{\varepsilon}_{q,t} + \bar{N}_C \varepsilon_{q,t} + N g_\pi(\nu) + \alpha_{0,k} + \rho_\pi \bar{\alpha}_0}{N + \bar{\alpha}_0} \quad (17)$$

$$|\mathbb{E}[\gamma_{k,k'}^{(m)}] - \gamma_{k,k'}^{*(m)}| \leq \epsilon_{\gamma,k,k',t}^{(m)} := \frac{2N g_\gamma(\nu) + 2\tilde{N}_C \tilde{\varepsilon}_{q,t} + 2\bar{N}_C \varepsilon_{q,t} + \beta_{0,k,k'}^{(m)} + \bar{\beta}_{0,k}^{(m)}}{N \mu_m \pi_k^* - N \frac{g_\gamma(\nu)}{\gamma_{k,k'}^{*(m)}} - \tilde{N}_C \tilde{\varepsilon}_{q,t} - \bar{N}_C \varepsilon_{q,t} + \bar{\beta}_k^{(m)}} \quad (18)$$

and  $\varepsilon_{\pi,t} := \max_k \varepsilon_{\pi,k,t}, \varepsilon_{\gamma,t} := \max_{k,k',m} \varepsilon_{\gamma,k,k',t}^{(m)}$ .

Comparing Thm. 2 to Thm. 1, the error bounds for the labels involved in constraints introduced by Alg. 1 will be smaller than the corresponding bounds by VBEM, as long as  $\max_n W_n > 0$ . This in turn reduces the error bounds for the parameters of interest, thus justifying the improved performance of Alg. 1 when provided with good initialization. Such an initialization can be achieved by using spectral methods, or, by utilizing the output of the aforementioned unconstrained VBEM algorithm.

## 4.2 Choosing $\eta$ .

Proper choice of  $\eta$  is critical for the performance of the proposed algorithm, as it represents the weight assigned to the available constraints. Here, using the number of violated constraints  $N_V := \sum_{(n,n') \in \mathcal{C}_{ML}} \mathbf{1}(y_n \neq y_{n'}) + \sum_{(n,n') \in \mathcal{C}_{CL}} \mathbf{1}(y_n = y_{n'})$  as a proxy for performance, grid search can be used to select  $\eta$  from a (ideally small) set of possible values  $\mathcal{H}$ .

## 4.3 Selecting instance-level constraints

In some cases, acquiring pairwise constraints, can be costly and time consuming. This motivates judicious selection of data we would prefer to query constraints for. Here, we outline an approach for selecting constraints using only the annotator responses  $\mathbf{Y}$ . In addition to providing error bounds for VBEM under instance level constraints, Thm. 2 reveals how to select the  $N_C$  constraints in  $\mathcal{C}$ . Equations (17) and (18) show that in order to minimize the error bounds of the priors and confusion matrices, the data with the largest label errors  $|q(y_n = k) - \mathbf{1}(y_n = k)|$  should be included in  $\mathcal{C}$ . The smaller error bounds of (17) and (18) then result in smaller error bounds for the labels, thus improving the overall classification performance. Let  $\tilde{\mathcal{C}}_u$  denote the set of data for which we wish to reduce the label error. In order to minimize the error  $\tilde{\varepsilon}_q$  per element of  $\tilde{\mathcal{C}}_u$ , the term  $W_n$  in the exponent of (16) should be maximized. To this end, data in  $\tilde{\mathcal{C}}_u$  should be connected with instance-level constraints to data that exhibit small errors  $\varepsilon_q$ . Data connected to each  $n \in \tilde{\mathcal{C}}_u$ , are collected in the set  $\tilde{\mathcal{C}}_c(n)$ .

Since the true label errors are not available, one has to resort to surrogates for them. One class of such surrogates, typically used in active learning [9], are the so-called uncertainty measures. These quantities estimate how uncertain the classifier (in this case the crowd) is about each datum. Intuitively, data for which the crowd is highly uncertain are more likely to be misclassified. In our setup, such uncertainty measures can be obtained using the results provided by VBEM, and specifically the label posteriors  $\{q_T(y_n)\}_{n=1}^N$ . As an alternative, approximate posteriors can be found without using the VBEM algorithm, by taking a histogram of annotator responses per datum. Using these posteriors, one can measure the uncertainty of the crowd for each datum. Here, we opted for the so-called best-versus-second-best uncertainty measure, which per datum

**Table 1.** Dataset properties

Dataset	$N$	$M$	$K$	$\tilde{\delta}$
RTE	800	164	2	48.78
Sentence Polarity	5,000	203	2	136.68
Dog	807	109	5	74.03
Web	2,665	177	5	87.94

$n$  is given by

$$H(\hat{y}_n) = \max_k q_T(y_n = k) - \max_{k' \neq k} q_T(y_n = k'). \quad (19)$$

This quantity shows how close the two largest posteriors for each datum are; larger values imply that the crowd is highly certain for a datum, whereas smaller ones indicate uncertainty.

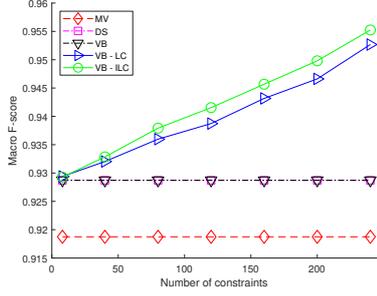
Using the results of VBEM and the uncertainty measure in (19),  $\lfloor N_C/K \rfloor$  are randomly selected without replacement, to be included in  $\tilde{\mathcal{C}}_u$ , with probabilities  $\lambda_n \propto 1 - H(\hat{y}_n)$ . For each  $n \in \tilde{\mathcal{C}}_u$ ,  $K$  data are randomly selected, with probabilities  $\lambda_n^\dagger \propto H(\hat{y}_n)$ , and are included in  $\tilde{\mathcal{C}}_c(n)$ . Using this procedure, data in  $\tilde{\mathcal{C}}_c(n)$ ,  $n \in \tilde{\mathcal{C}}_u$  are likely the ones that the crowd is certain of. Finally, the links  $\{(n, n'), n \in \tilde{\mathcal{C}}_u, n' \in \tilde{\mathcal{C}}_c(n)\}$  are queried and included in the constraint set  $\mathcal{C}$ . Note here that each uncertain data point is connected to  $K$  certain ones, to increase the likelihood that its label error will decrease.

*Remark 3.* The principles outlined in this subsection can be leveraged to develop active semi-supervised crowdsourcing algorithms. We defer such approaches to future work.

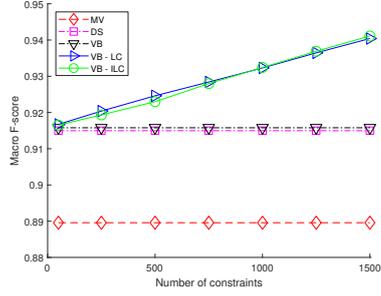
## 5 Numerical tests

Performance of the proposed semi-supervised approach is evaluated here on several popular crowdsourcing datasets. Variational Bayes with instance-level constraints (abbreviated as *VB - ILC*) [cf. Sec. 4, Alg. 1] is compared to majority voting (abbreviated as *MV*), the variational Bayes method of Sec. 3.2 that does not include side-information (abbreviated as *VB*), and the EM algorithm of [7] (abbreviated as *DS*) that also does not utilize any side information. Here, simple baselines are chosen to showcase the importance of including constraints in the crowdsourcing task. To further show the effect of pairwise constraints *VB - ILC* is compared to Variational Bayes using label constraints (abbreviated as *VB - LC*), as outlined at the beginning of Sec. 4.

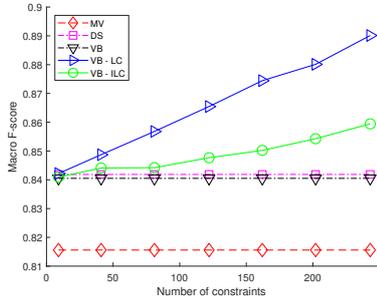
As figure of merit, the macro-averaged F-score [21] is adopted to indicate the per-class performance of an algorithm. The datasets considered here are the RTE [29], Sentence Polarity [25], Dog [8], and Web [37]; see Table 1 for their properties, where  $\tilde{\delta}$  denotes the average number of responses per annotator. For the datasets presented here, results indicating the micro-averaged F-score, alongside dataset descriptions and results with 6 additional datasets are included in Appendix



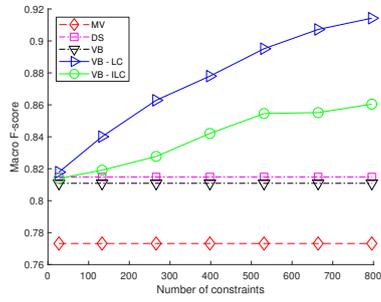
**Fig. 1.** Macro F-score for the RTE dataset, with the same number of constraints for *VB-LC* and *VB-ILC*



**Fig. 2.** Macro F-score for the Sentence Polarity dataset, with the same number of constraints for *VB-LC* and *VB-ILC*



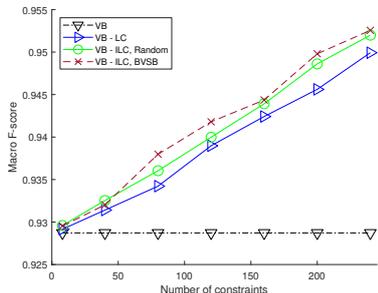
**Fig. 3.** Macro F-score for the Dog dataset, with the same number of constraints for *VB-LC* and *VB-ILC*



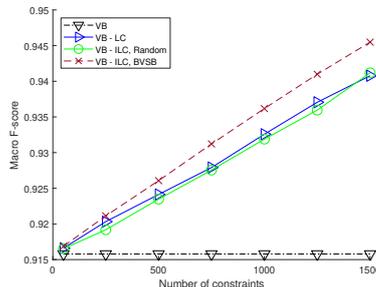
**Fig. 4.** Macro F-score for the Web dataset, with the same number of constraints for *VB-LC* and *VB-ILC*

B of the supplementary material. MATLAB [18] was used throughout, and all results represent the average over 20 Monte Carlo runs. In all experiments *VB* and *DS* are initialized using majority voting, while *VB-LC* and *VB-ILC* are initialized using the results of *VB*, since, as shown in Thm. 2 *VB-ILC* requires good initialization. For the RTE, Dog, and Sentence Polarity datasets, the prior parameters are set as  $\alpha_0 = \mathbf{1}$ , where  $\mathbf{1}$  denotes the all-ones vector, and  $\beta_{0,k}^{(m)}$  is a vector with  $K$  at its  $k$ -th entry and ones everywhere else, for all  $k, m$ . For the Web dataset, all priors are set to uniform as this provides the best performance for the VB based algorithms. The values of  $\eta$  for Alg. 1 are chosen as described in Sec. 4.2 from the set  $\mathcal{H} = \{0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 100, 500\}$ . When instance-level constraints in  $\mathcal{C}$  are provided to *VB-ILC*, constraints that can be logically derived from the ones in  $\mathcal{C}$  are also included. For example, if  $(i, j) \in \mathcal{C}_{ML}$  and  $(j, k) \in \mathcal{C}_{ML}$ , a new must-link constraint  $(i, k)$  will be added. Similarly, if  $(i, j) \in \mathcal{C}_{ML}$  and  $(j, k) \in \mathcal{C}_{CL}$ , then a cannot-link constraint  $(i, k)$  will be added.

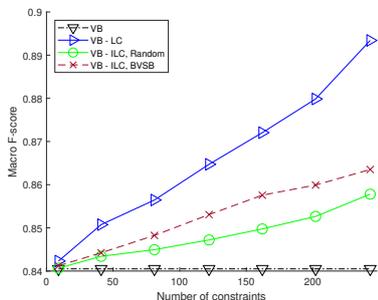
Figs. 1-4 show classification performance when *VB-LC* and *VB-ILC* are provided with  $N_C$  randomly selected constraints, that is  $N_C$  label constraints for *VB-LC* and  $N_C$  instance-level constraints for *VB-ILC*. *VB-LC* and *VB-*



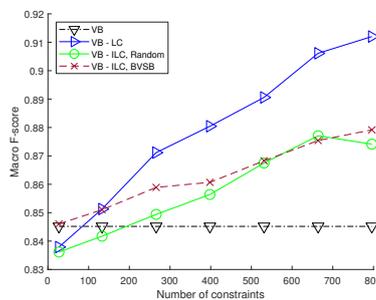
**Fig. 5.** Macro F-score for the RTE dataset, with random and uncertainty based constraint selection for *VB-ILC*



**Fig. 6.** Macro F-score for the Sentence polarity dataset, with random and uncertainty based constraint selection for *VB-ILC*

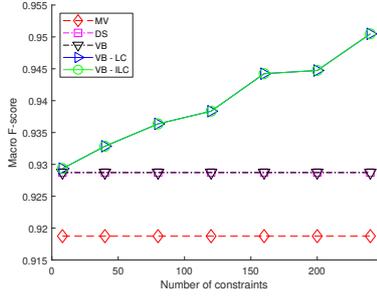


**Fig. 7.** Macro F-score for the Dog dataset, with random and uncertainty based constraint selection for *VB-ILC*

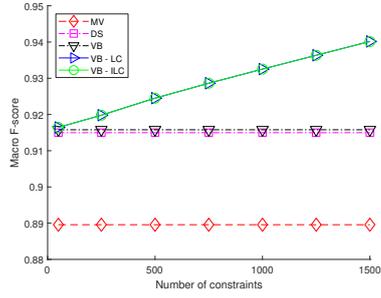


**Fig. 8.** Macro F-score for the Web dataset, with random and uncertainty based constraint selection for *VB-ILC*

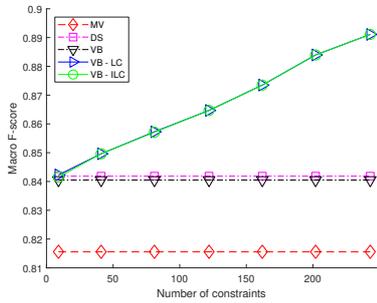
*ILC* exhibit improved performance as the number of constraints increases. As expected, *VB-LC* outperforms *VB-ILC*, when  $K > 2$ , since label information is stronger than instance-level information. Interestingly, for  $K = 2$ , *VB-ILC* exhibits comparable performance to *VB-LC* in the Sentence polarity dataset, and outperforms *VB-LC* in the RTE dataset. Nevertheless, *VB-ILC* outperforms its unsupervised counterparts as  $N_C$  increases, corroborating that even relatively weak constraints are useful. The performance of the constraint selection scheme of Sec. 4.3 is evaluated in Figs. 5-8. The uncertainty sampling based variant of *VB-ILC* is denoted as *VB-ILC, BVSB*. Uncertainty sampling for selecting label constraints is clearly beneficial, as it provides performance gains compared to randomly selecting constraints in all considered datasets. Figs. 9-12 show the effect of providing *VB-ILC* with constraints derived from  $N_C$  label constraints. When provided with  $N_C$  label constraints from  $K$  classes, and with  $z_k$  denoting the proportion of constraints from class  $k$ , the resulting number of must-link constraints is  $N_{ML} = \sum_{k=1}^K \binom{N_C z_k}{2}$ , since for each class every two points must be connected. The number of cannot-link constraints is  $N_{CL} = \sum_{k=1}^K \sum_{k'=1}^{K-1} N_C^2 z_k z_{k'+1}$ , as



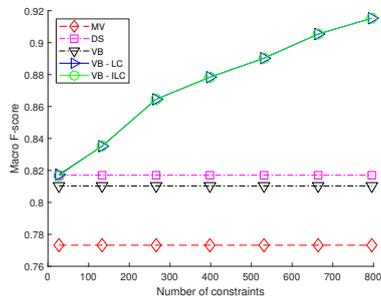
**Fig. 9.** Macro F-score for the RTE dataset, with label derived constraints for *VB-ILC*



**Fig. 10.** Macro F-score for the Sentence Polarity dataset, with label derived constraints for *VB-ILC*



**Fig. 11.** Macro F-score for the Dog dataset, with label derived constraints for *VB-ILC*



**Fig. 12.** Macro F-score for the Web dataset, with label derived constraints for *VB-ILC*

for each class every point must be connected to all points belonging to other classes. In this case, *VB - ILC* almost matches the performance of *VB - LC*. This indicates that when provided with an adequate number of instance-level constraints, *VB - ILC* performs as well as if label constraints had been provided.

## 6 Conclusions

This paper investigated constrained crowdsourcing with pairwise constraints, that encode relationships between data. The performance of the proposed algorithm was analytically and empirically evaluated on popular real crowdsourcing datasets. Future research will involve distributed and online implementations of the proposed algorithm, and other types of constraints alongside semi-supervised crowdsourcing with dependent annotators, and dependent data.

## References

1. Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., Navab, N.:

- Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging* **35**(5), 1313–1321 (2016). <https://doi.org/10.1109/TMI.2016.2528120>
2. Atarashi, K., Oyama, S., Kurihara, M.: Semi-supervised learning from crowds using deep generative models. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (2018)
  3. Basu, S., Davidson, I., Wagstaff, K.: *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 1 edn. (2008)
  4. Beal, M.J.: *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, University of London, University College London (United Kingdom) (2003)
  5. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*. The MIT Press, 1st edn. (2010)
  6. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. John Wiley & Sons (2012)
  7. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* pp. 20–28 (1979)
  8. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. pp. 248–255 (2009)
  9. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. *J. of Knowl. and Infor. Systems* **35**(2), 249–283 (2013)
  10. Hammersley, J.M., Clifford, P.E.: *Markov random fields on finite graphs and lattices. Unpublished manuscript* (1971)
  11. Howe, J.: The rise of crowdsourcing. *Wired Magazine* **14**(6), 1–4 (2006)
  12. Ibrahim, S., Fu, X., Kargas, N., Huang, K.: Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms. In: *Advances in Neural Information Processing Systems*. vol. 32, pp. 7847–7857. Curran Associates, Inc. (2019)
  13. Jaffe, A., Nadler, B., Kluger, Y.: Estimating the accuracies of multiple classifiers without labeled data. In: *AISTATS*. vol. 2, p. 4. San Diego, CA (2015)
  14. Kajino, H., Tsuboi, Y., Sato, I., Kashima, H.: Learning from crowds and experts. In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012)
  15. Kim, H.C., Ghahramani, Z.: Bayesian classifier combination. In: *Proceedings of Machine Learning Research*. vol. 22, pp. 619–627. PMLR, La Palma, Canary Islands (21–23 Apr 2012)
  16. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: *Proc. of SIGCHI Conf. on Human Factors in Computing Systems*. pp. 453–456. ACM, Florence, Italy (2008)
  17. Liu, M., Jiang, L., Liu, J., Wang, X., Zhu, J., Liu, S.: Improving learning-from-crowds through expert validation. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. pp. 2329–2336 (2017). <https://doi.org/10.24963/ijcai.2017/324>
  18. MATLAB: version 9.7.0 (R2019b). The MathWorks Inc., Natick, Massachusetts (2019)
  19. Moreno, P.G., Artés-Rodríguez, A., Teh, Y.W., Perez-Cruz, F.: Bayesian nonparametric crowdsourcing. *J. Mach. Learn. Res.* **16**(1), 1607–1627 (Jan 2015)
  20. Nguyen, A.T., Wallace, B.C., Li, J.J., Nenkova, A., Lease, M.: Aggregating and Predicting Sequence Labels from Crowd Annotations. *Proc Conf Assoc Comput Linguist Meet* **2017**, 299–309 (2017)
  21. Powers, D.M.W.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Intl. Journal of Machine Learning Technology* **2**, 37–63 (2011)

22. 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 (Aug 2010)
23. Rodrigues, F., Pereira, F., Ribeiro, B.: Sequence labeling with multiple annotators. *Machine Learning* **95**(2), 165–181 (May 2014). <https://doi.org/10.1007/s10994-013-5411-2>
24. Rodrigues, F., Pereira, F.: Deep learning from crowds. *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1) (Apr 2018)
25. Rodrigues, F., Pereira, F., Ribeiro, B.: Gaussian process classification and active learning with multiple annotators. In: *Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 32, pp. 433–441. PMLR, Beijing, China (22–24 Jun 2014)
26. Ruiz, P., Morales-Álvarez, P., Molina, R., Katsaggelos, A.: Learning from crowds with variational Gaussian processes. *Pattern Recognition* **88**, 298–311 (April 2019)
27. Shi, W., Sheng, V.S., Li, X., Gu, B.: Semi-supervised multi-label learning from crowds via deep sequential generative model. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 1141–1149. KDD '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3394486.3403167>
28. Simpson, E., Roberts, S., Psorakis, I., Smith, A.: *Dynamic Bayesian Combination of Multiple Imperfect Classifiers*, pp. 1–35. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
29. Snow, R., O’Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, p. 254–263. EMNLP '08, Association for Computational Linguistics, USA (2008)
30. Tang, W., Lease, M.: Semi-supervised consensus labeling for crowdsourcing. In: *ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR)*, 2011 (2011)
31. Traganitis, P.A., Pagès-Zamora, A., Giannakis, G.B.: Blind multiclass ensemble classification. *IEEE Transactions on Signal Processing* **66**(18), 4737–4752 (Sept 2018). <https://doi.org/10.1109/TSP.2018.2860562>
32. Traganitis, P.A.: Blind ensemble classification of sequential data. In: *IEEE Data Science Workshop*. Minneapolis, MN (June 2019)
33. Traganitis, P.A., Giannakis, G.B.: Unsupervised ensemble classification with dependent data (2019)
34. Venanzi, M., Guiver, J., Kazai, G., Kohli, P., Shokouhi, M.: Community-based bayesian aggregation models for crowdsourcing. In: *Proc. of 23rd Intl. Conf. on World Wide Web*, pp. 155–164. ACM (April 2014)
35. Yan, Y., Rosales, R., Fung, G., Dy, J.: Modeling multiple annotator expertise in the semi-supervised learning scenario. In: *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, p. 674–682. UAI'10, AUAI Press, Arlington, Virginia, USA (2010)
36. Zhang, Y., Chen, X., Zhou, D., Jordan, M.I.: Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In: *Advances in Neural Information Processing Systems*, pp. 1260–1268 (2014)
37. 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 25*, pp. 2195–2203. Curran Associates, Inc. (2012)