Journal of Machine Learning Research 16 (2015) 1607-1627 Submitted 7/14; Published 8/15

Bayesian Nonparametric Crowdsourcing

Pablo G. Moreno PGMORENO@TSC.UC3M.ES
Antonio Artés-Rodríguez ANTONIO@TSC.UC3M.ES
Gregorio Marañón Health Research Institute
Department of Signal Theory and Communications
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganés (Madrid, Spain)

Yee Whye Teh Y.W.TEH@STATS.OX.AC.UK
Department of Statistics
1 South Parks Road
Oxford OX1 3TG, UK

Fernando Perez-Cruz FERNANDO@TSC.UC3M.ES
Gregorio Marañón Health Research Institute FERNANDO.PEREZ-CRUZ@ALCATEL-LUCENT.COM
Department of Signal Theory and Communications
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganés (Madrid, Spain)
Bell Labs (Alcatel-Lucent)
600 Mountain Avenue
New Providence, NJ 07974

Editor: David Blei

Abstract

Crowdsourcing has been proven to be an effective and efficient tool to annotate large data-sets. User annotations are often noisy, so methods to combine the annotations to produce reliable estimates of the ground truth are necessary. We claim that considering the existence of clusters of users in this combination step can improve the performance. This is especially important in early stages of crowdsourcing implementations, where the number of annotations is low. At this stage there is not enough information to accurately estimate the bias introduced by each annotator separately, so we have to resort to models that consider the statistical links among them. In addition, finding these clusters is interesting in itself as knowing the behavior of the pool of annotators allows implementing efficient active learning strategies. Based on this, we propose in this paper two new fully unsupervised models based on a Chinese restaurant process (CRP) prior and a hierarchical structure that allows inferring these groups jointly with the ground truth and the properties of the users. Efficient inference algorithms based on Gibbs sampling with auxiliary variables are proposed. Finally, we perform experiments, both on synthetic and real databases, to show the advantages of our models over state-of-the-art algorithms.

Keywords: multiple annotators, Bayesian nonparametrics, Dirichlet process, hierarchical clustering, Gibbs sampling

1. Introduction

Crowdsourcing services are becoming very popular as a mean of outsourcing tasks to a large crowd of users. The best-known tool is Mechanical Turk (Amazon, 2005), in which *requesters* are able to post small tasks for *providers* registered in the system, who complete them for a monetary payment set by the requester. In machine learning, crowdsourcing allows to distribute the labeling of a data-set among a pool of users, so each user only labels a subset of the instances. The advantage is the ability to gather large data-sets in a short time and, generally, at a low cost. Successful examples include, but it is not limited to, LabelMe (Russell et al., 2008) or GalazyZoo (Lintott et al., 2010).

The quality of the labels retrieved by crowdsourcing is uneven. Unlike traditional ways of gathering a labeled data-set in which labels are provided by a small set of motivated experts, we deal now with a large number of users who are not necessarily experts nor motivated. Further, we might have little or no information about them to perform quality control tests. This motivates the development of statistical models for reliably estimating ground truth from noisy and biased labels provided by users.

Another problem that has received significant attention of late, is the detection of groupings among the labelers (Simpson et al., 2011, 2013). In most crowdsourcing applications we can identify several types of users: experts, novices, spammers and even malicious or adversarial annotators. Identifying these groups of users and learning about their properties is useful to design efficient crowdsourcing strategies that minimize the overall cost, selecting the most suitable users for a labeling task. For example, if we could identify spammers we could ban them from the system and avoid wasting resources. In the same way, if a user is identified as an expert in a particular task, we could reward him by increasing his pay-off or giving him preference over other users when the time to select new tasks comes.

Usually, the detection of grouping of labelers is tackled in a post-processing step, after the ground truth has been estimated from user annotations (see Section 4). In particular Simpson et al. (2011) were the first to tackle the join problem. They estimated the ground truth using a previous model called Independent Bayesian Combination of Classifiers (iBCC) (Kim and Ghahramani, 2012) and then, as a post-processing step they infer the different clusters of users. Therefore, the estimation of the ground truth is done without considering the clustering structure of the users.

In this paper, we propose two unsupervised Bayesian nonparametric models to combine the labels provided by the users in a crowdsourcing scenario, taking into account the presence of clusters of users. Our models jointly solve the problem of the estimation of ground truth and the problem of identification of clusters of users and their properties. The estimation of the ground truth improves the clustering of the users and vice-versa, thus performing better than current state-of-the-art (Kim and Ghahramani, 2012; Simpson et al., 2011). The overall improvement in both tasks is particularly important in the early stages of a crowdsourcing project, when the number of annotations provided by the users is very low. In this case, algorithms that estimate the properties of each user independently, without considering the dependencies among them, tend to provide poor estimates, and may perform worse than majority voting (see Section 5).

In the first model, we propose a clustering structure using a CRP prior (Pitman, 2002) which allows flexible modeling of the number of clusters of users. In this model, all the users that belong to the same cluster share the same parameters governing the way they label instances, and therefore, they have the same behavior. Forcing all the users to share the same exact parameters, is a strong assumption that might lead to groupings with a large number of cluster. Therefore, these groupings

are difficult to interpret and not very useful. To relax this assumption we propose a second model in which users that belong to the same cluster are modeled as having similar parameters, but allows each user to have its own parameters using a hierarchical Bayesian approach.

In this paper, we rely on a Bayesian nonparametric model, because we are not only interesting in having an accurate model but also in having an interpretable one. In the experiments (Section 5) we show that the error rates between the two proposed models are not significantly different. However, the second one is interpretable, in the sense that it perfectly identifies each kind of clusters and it reports the least number of them. The interpretability of the model is principal to us, because we want to use the model to identify the 'good' annotators and be able to reward them accordingly, while other models are not able to provide this information.

The rest of the paper is organized as follows. In Section 2, we present the two new generative models for crowdsourcing that take into account the clustering structure of the users. In Section 3, we propose efficient Markov chain Monte Carlo (MCMC) inference algorithms for estimating the different groups of users as well as the ground truth. In Section 4, we review related literature on crowdsourcing and the identification of user clusters in the context of crowdsourcing. In Section 5, we validate our model on synthetic data and we perform several experiments on real data-sets to show the advantages of our models over state-of-the-art algorithms. Finally, we conclude this paper in Section 6 and present possible extensions for the future.

2. Hierarchical Bayesian Combination of Classifiers

In this section, we propose two different models. Both algorithms receive as input a set of noisy labels $\boldsymbol{Y} \in \{0,...,C\}^{N\times L}$ provided by L users for N instances. The element $y_{i\ell}$ represents the label given by the user ℓ to the instance i and it is 0 if the user did not label the corresponding instance. Notice that this matrix $\boldsymbol{Y}$ is highly sparse in the early stages of a crowdsourcing application. This is known as the *cold start problem* (Schein et al., 2002), i.e. the difficulty of drawing any inferences due to the lack of information. Notice that $\boldsymbol{Y}$ is the only observed variable in the models.

The output of the algorithms is the set of true but unknown labels of the instances $\boldsymbol{z} \in \{1,...,C\}^N$, where z_i indicates the true label estimate of the instance i.

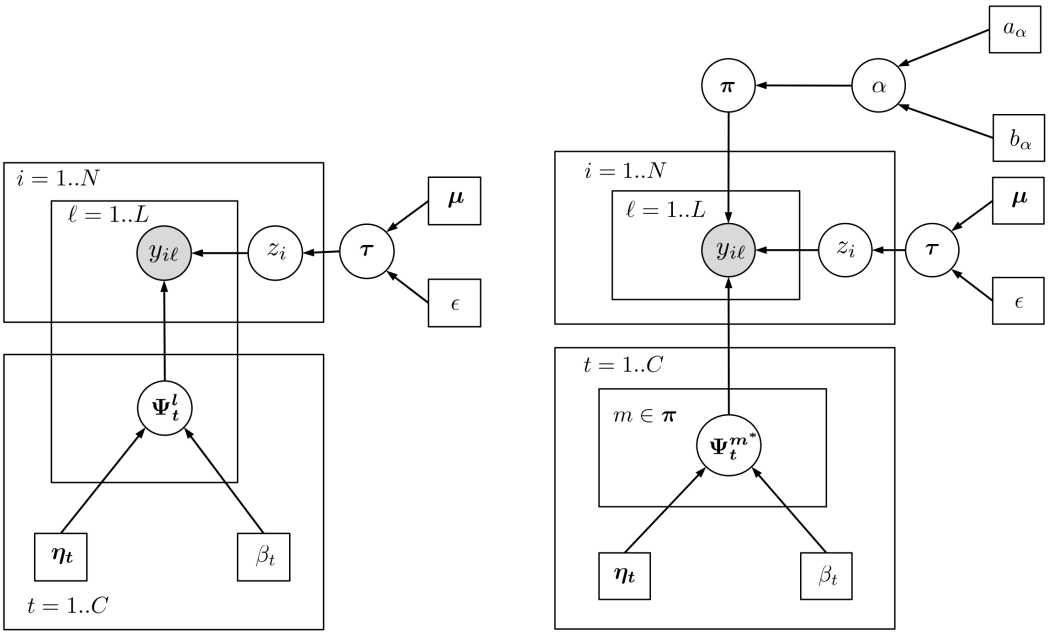
We denote by $[L] = \{1,2,...,L\}$ the set of indices of the users and by $\boldsymbol{\pi}_L$ a partition of $[L]$. A partition is a collection of mutually exclusive, mutually exhaustive and non-empty subsets called clusters. We denote the cluster assignment of the user ℓ with a variable q_ℓ such that $q_\ell = m$ denotes the event that the user ℓ is assigned to cluster $m \in \boldsymbol{\pi}_L$.

2.1 Clustering Based Bayesian Combination of Classifiers

Firstly, we propose a model for users in which they can belong to different clusters. In each cluster all the users have the same properties. We name it Clustering based Bayesian Combination of Classifiers (cBCC) (see Figure 1b) and it has the following observation model

$$y_{i\ell}|z_i, \boldsymbol{\pi}, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}^{q_\ell}_{z_i}),$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

We assume that all the users that belong to cluster $m \in \boldsymbol{\pi}$ share the same properties, i.e. the same confusion matrix $\boldsymbol{\Psi}^m \in [0,1]^{C\times C}$, where Ψ^m_{tc} is the probability that a user allocated in cluster m labels an instance as $y = c$ when the ground truth is $z = t$. We use the notation $\boldsymbol{\Psi}^m_t \in \mathscr{S}^C$ to denote



(a) iBCC model (b) cBCC model

Figure 1: Graphical model representation of the iBCC and cBCC models

the row t of $\boldsymbol{\Psi}^m$, where $\mathscr{S}^C$ is the C-dimensional probability simplex. The component τ_t of $\boldsymbol{\tau} \in \mathscr{S}^C$ is the probability of the ground truth z being equal to $t \in \{1,...,C\}$.

We also need to define priors to complete the Bayesian model. In particular, we choose conjugate priors

$$\boldsymbol{\Psi}_t^m | \boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta_t \boldsymbol{\eta_t}),$$
$$\boldsymbol{\tau} | \boldsymbol{\varepsilon}, \boldsymbol{\mu} \sim \text{Dir}(\boldsymbol{\varepsilon} \boldsymbol{\mu}),$$

where we use a Dirichlet prior on each of the rows of the confusion matrices in which $\boldsymbol{\eta}_t \in \mathscr{S}^C$ is the mean value of $\boldsymbol{\Psi}_t^m$ while $\beta_t \in \mathbb{R}_+$ is related to its precision. Notice that this is an over parametrization of the Dirichlet distribution, which only needs C parameters to be fully determined. However, this decomposition is useful to interpret the results as well as for the development of the inference algorithms in Section 3. Likewise, we set a Dirichlet prior on $\boldsymbol{\tau}$, where $\boldsymbol{\mu} \in \mathscr{S}^C$ is the mean and $\boldsymbol{\varepsilon} \in \mathbb{R}_+$ relates to the precision.

We could use a parametric model in which the cardinality of the partition $M = |\boldsymbol{\pi}|$ is fixed a priori. Unfortunately, in this case the inferences are sensitive to the value of M chosen. In the limiting case $M = 1$ the model is equivalent to majority voting. If M is too large the model does not take advantage of the presence of clusters of users. In the limiting case $M = L$ each user becomes a singleton cluster, and the model does not capture the dependencies among the users.

To find M we could use traditional model selection strategies like cross-validation (Stone, 1974) or Bayesian Information Criterion (Fraly and Raftery, 1998). This approach has two limitations. First, we usually do not have access to a validation set for which $\boldsymbol{z}$ is known. Second, is the high computational complexity. An alternative pathway is to set a prior on the space of partitions.

We denote by $\mathscr{P}_L$ the space of all partitions $\pi_L \in \mathscr{P}_L$. In a Bayesian setting, we have to set the prior without observing the number of users. One option is to set a prior on an infinite number of users, i.e. on partition $\pi \in \mathscr{P}_\infty$. To build such a prior we further assume that the observations are exchangeable and that we deal with consistent random partitions (Pitman, 2002). A (exchangeable and consistent) prior on $\mathscr{P}_\infty$ is the CRP introduced by Blackwell and Macqueen (1973) and that can be seen as the induced distribution over the partition space by a Dirichlet Process (DP) (Ferguson, 1973; Antoniak, 1974). We place a CRP prior over the users' partitions

$$\pi|\alpha \sim \text{CRP}(\alpha). \tag{1}$$

We can generate samples from this prior using the following conditional distributions

$$p(q_\ell = m|\pi^{\neg\ell}, \alpha) \propto \begin{cases} |m|^{\neg\ell}, & m \in \pi^{\neg\ell} \\ \alpha, & m = \emptyset \end{cases},$$

where $|m|$ represents the number of users in cluster m and $|m|^{\neg\ell}$ is equal to $|m|$ excluding user ℓ. We denote by $\pi^{\neg\ell}$ the partition with the user ℓ removed and $q_\ell = \emptyset$ denotes the event that user ℓ is assigned to a new cluster. α is the so called concentration parameter and control the a priori probability of generating new clusters. We further place a gamma prior over the concentration parameter α

$$\alpha|a_\alpha, b_\alpha \sim \text{Gamma}(a_\alpha, b_\alpha). \tag{2}$$

If α tends to infinity, every user is allocated to a singleton cluster. If α tends to 0, all the users share the same confusion matrix and the model produces a majority voting solution

In general, the CRP assigns more mass to partitions with a small number of clusters. This is sensible in our case, because when the number of annotations is scarce, majority voting may perform better than more elaborate algorithms since there is no enough information to estimate the individual properties of the users. In this case, the CRP prior dominates and therefore all users are allocated to the same cluster. When the number of annotations increase, the likelihood term dominates and different clusters of users are created.

Finally, we analyze the correlation structure that it is introduced among the users as a consequence of this clustering. The correlation a priori among two users ℓ and ℓ' is

$$\text{Corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)|z_i = t) = \begin{cases} -\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\sqrt{\frac{\eta_{ta}\eta_{tb}}{(1-\eta_{ta})(1-\eta_{tb})}} & a \neq b \\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right) & a = b \end{cases}. \tag{3}$$

Here, $\mathbb{I}(.)$ represents the indicator function. The proof is in the supplementary material. In Section 4, we show how this model relates to other state-of-the-art algorithms.

2.2 Hierarchical Clustering Based Bayesian Combination of Classifiers

In the cBCC model, the users that belong to the same cluster share the same confusion matrix. However, in a practical situation, each user has a behavior that is different from every other user, but it is in some sense similar to the behavior of users that are allocated to its cluster.

To capture this behavior, we propose a hierarchical extension of the cBCC model called hcBCC (hierarchical cBCC) depicted in Figure 2. The observation model is the following

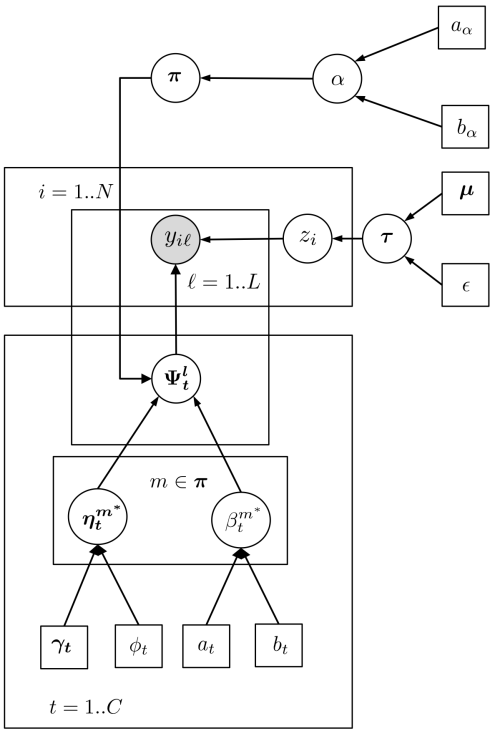


Figure 2: Graphical model representation of the hcBCC model

$$y_{i\ell}|z_i, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}^{\ell}_{z_i}),$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

Now each user has its own confusion matrix $\boldsymbol{\Psi}^{\ell}$ in contrast to the cBCC model where we had a confusion matrix per cluster $\boldsymbol{\Psi}^{m}$. To capture the similarity between users that belong to the same cluster we use the following hierarchical prior:

$$\boldsymbol{\Psi}^{\ell}_t|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta_t^{q_\ell}\boldsymbol{\eta}_t^{q_\ell}),$$
$$\beta_t^m|a_t, b_t \sim \text{Gamma}(a_t, b_t),$$
$$\boldsymbol{\eta}_t^m|\boldsymbol{\phi}, \boldsymbol{\gamma} \sim \text{Dir}(\boldsymbol{\phi}_t\boldsymbol{\gamma}_t),$$
$$\boldsymbol{\tau}|\varepsilon, \boldsymbol{\mu} \sim \text{Dir}(\varepsilon\boldsymbol{\mu}).$$

In this way, the confusion matrices of all users that belong to the same cluster m are generated from the same distribution. In particular, each of the rows of the confusion matrices of all the users that belong to cluster m, i.e. $\{\boldsymbol{\Psi}^{\ell}_t : q_\ell = m\}$, are i.i.d. samples from the same Dirichlet distribution whose parameters are β_t^m and $\boldsymbol{\eta}_t^m$. A Dirichlet prior is set on the vector $\boldsymbol{\eta}_t^m$ while a gamma prior is set on the scalar β_t^m. Finally, for $\boldsymbol{\pi}$ and α we, respectively, use the same priors given by Equations 1 and 2. With this we have a model where we no longer cluster the confusion matrices of the users, but the distributions that generate them.

Notice that the vector $\boldsymbol{\beta}^m$ governs the variability among the users that belong to the same cluster m. The bigger are these values, the lower is the intra-cluster variability. If we make each of the components of $\boldsymbol{\beta}^m$ tend to infinity, then the variability among the users tend to 0 and the model becomes equivalent to the cBCC model. In this way, this model can be seen as a generalization of some state-of-the-art methods (see Section 4).

3. Inference

Computing the posterior distribution of the clusters allocation, the properties of the users and the estimated ground is intractable, so we have to resort to approximate inference. Since the proposal of the DP by Ferguson (1973), approximate inference schemes based on Markov Chain Monte Carlo (MCMC) methods have played a crucial role (Escobar, 1994; MacEachern and Müller, 1998; Neal, 2000; Ishwaran and James, 2001; Walker, 2007; Kalli et al., 2011) among others. In this section we propose to use Gibbs sampling together with the corresponding auxiliary variables whenever it is not possible to compute the conditional distributions due to non-conjugacies.

3.1 CBCC

We use a collapsed Gibbs sampling algorithm where we integrate out the variables $\boldsymbol{\Psi}^m$ and $\boldsymbol{\tau}$, obtaining the following new set of equations

$$p(\boldsymbol{Y}|\boldsymbol{\pi},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_m \prod_t \left[\frac{\Gamma(\beta_t)}{\Gamma(n_{mt}+\beta_t)} \prod_c \frac{\Gamma(n_{mtc}+\beta_t\eta_{tc})}{\Gamma(\beta_t\eta_{tc})} \right],$$
$$p(\boldsymbol{z}|\boldsymbol{\varepsilon},\boldsymbol{\mu}) = \frac{\Gamma(\varepsilon)}{\Gamma(N+\varepsilon)} \prod_t \frac{\Gamma(n_m+\varepsilon\mu_t)}{\Gamma(\varepsilon\mu_t)},$$

where $\Gamma(\cdot)$ denotes the gamma function. We denote $n_{i\ell mtc} = \mathbb{I}(z_i = t, y_{i\ell} = c, y_{i\ell} \neq 0, q_\ell = m)$, and when an index of this variable is omitted we assume it is summed out. For example, n_{mtc} represents the number of annotations equal to c provided by the users of cluster m for the set of instances whose ground truth is equal to t. We use Gibbs sampling to infer the value of the ground truth $\boldsymbol{z}$, the clusters of annotators $\boldsymbol{\pi}$, as well as the hyper parameters of the CRP, conditioned on the observed variables $\boldsymbol{Y}$.

Firstly, to update the cluster assignment of annotator ℓ, we need the conditional distribution of q_ℓ given the rest of the variables

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg\ell} \times \prod_t \frac{\Gamma(n_{mt}^{\neg\ell}+\beta_t)}{\Gamma(n_{mt}^{\neg\ell}+n_{\ell t}+\beta_t)} \prod_t \prod_c \frac{\Gamma(n_{mtc}^{\neg\ell}+n_{\ell tc}+\beta_t\eta_{tc})}{\Gamma(n_{mtc}^{\neg\ell}+\beta_t\eta_{tc})}, & m \in \boldsymbol{\pi}^{\neg\ell} \\ \alpha \times \prod_t \frac{\Gamma(\beta_t)}{\Gamma(n_{\ell t}+\beta_t)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t\eta_{tc})}{\Gamma(\beta_t\eta_{tc})}, & m = \emptyset \end{cases},$$

where $q_\ell = \emptyset$ denotes the event that user ℓ is assigned to a new cluster. The quantities $n_{mt}^{\neg\ell}$ and $n_{mtc}^{\neg\ell}$ are defined in the same way as n_{mt} and n_{mtc} respectively, but excluding the annotator ℓ. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$. To sample the estimate of the ground truth z_i of each instance conditioned on the rest of the variables, the required conditional distribution is

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \varepsilon\mu_t\right) \times \prod_m \left[\frac{\Gamma(n_{mt}^{\neg i}+\beta_t)}{\Gamma(n_{mt}^{\neg i}+n_{im}+\beta_t)} \prod_c \frac{\Gamma(n_{mtc}^{\neg i}+n_{imc}+\beta_t\eta_{tc})}{\Gamma(n_{mtc}^{\neg i}+\beta_t\eta_{tc})} \right].$$

The quantities $n_{mt}^{\neg i}$ and $n_{mtc}^{\neg i}$ again correspond to n_{mt} and n_{mtc} but excluding the instance i. The complexity of updating the $\boldsymbol{z}$ variables is $O(NMTC)$

Finally, we sample the concentration parameter α following the procedure proposed by Escobar (1994).

3.2 HCBCC

As in the cBCC we start by integrating out the $\boldsymbol{\Psi}^{\ell}$ and $\boldsymbol{\tau}$ variables

$$p(\boldsymbol{Y}|\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_m \prod_{\ell:q_\ell=m} \prod_t \left[\frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t \eta_{tc}^m)} \right],$$
$$p(\boldsymbol{z}|\varepsilon,\boldsymbol{\mu}) = \frac{\Gamma(\varepsilon)}{\Gamma(N+\varepsilon)} \prod_t \frac{\Gamma(n_m+\varepsilon\mu_t)}{\Gamma(\varepsilon\mu_t)}.$$

The variables we need to sample from are $\boldsymbol{\pi}$ and the ground truth estimate $\boldsymbol{z}$. Note however that we cannot marginalize out the cluster parameters $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$, as the Dirichlet prior and the Gamma prior are not conjugate to the likelihoods given above, so that these variables will have to be sampled as well.

The conditional distribution of $p(q_\ell = m|rest)$ when $m \in \boldsymbol{\pi}^{\neg \ell}$ can be computed like in the cBCC model. However, to compute $p(q_\ell = m|rest)$ when $m = \emptyset$ we need to integrate the parameters of the new clusters, i .e . $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$. In this case, due to the non-conjugacy we cannot solve this integral analytically. Instead, we use the recently proposed *Reuse algorithm* (Favaro and Teh, 2012). This algorithm is similar to the well-known Algorithm 8 (Neal, 2000), where the idea is to use a set of h auxiliary empty clusters H_{empty} to approximate the integral. However, the *reuse algorithm* is more efficient as it requires less simulations from the prior over the cluster parameters. For each cluster $m \in \boldsymbol{\pi} \cup H_{empty}$ we keep track of the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$. The conditional distribution of q_ℓ is then

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg \ell} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in \boldsymbol{\pi}^{\neg \ell} \\ \frac{\alpha}{h} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in H_{empty} \end{cases}.$$

If an auxiliary empty cluster is chosen, it is moved into the partition $\boldsymbol{\pi}$, and a new empty cluster is created in its place by sampling from the prior over cluster parameters. If a cluster in $\boldsymbol{\pi}$ is emptied as a result of sampling q_ℓ, it is moved into H, displacing one of the empty clusters (picked uniformly at random). In addition, at regular intervals the parameters of the empty clusters are refreshed by simulating them from their priors, while those in $\boldsymbol{\pi}$ are updated. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$.

Again, due to the non-conjugacy of the Dirichlet and Gamma priors, the conditional distributions of the parameters $\boldsymbol{\eta}^m$ and $\boldsymbol{\beta}^m$ for $m \in \boldsymbol{\pi}$ cannot be computed analytically. To solve this, we use an auxiliary variable method similar to the one proposed by Escobar (1994) and Teh et al. (2003). Specifically, we introduce two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ (see the supplementary material for fur-

ther details), and apply the following Gibbs updates that leave invariant the posterior distribution:

$$\nu_{\ell t} \sim \text{Beta}(\beta_t^{q_\ell}), \qquad s_{\ell tc} \sim \text{Antoniak}(n_{\ell tc}, \beta_t^{q_\ell}\eta_{tc}^{q_\ell}),$$
$$\eta_{t:}^m \sim \text{Dir}\left(\sum_{\{\ell:q_\ell=m\}} s_{\ell t:} + \phi_t \gamma_{t:}\right),$$
$$\beta_t^m \sim \text{Gamma}\left(\sum_{\{\ell:q_\ell=m\}}\sum_c s_{\ell tc} + a_t, b_t - \sum_{\{\ell:q_\ell=m\}} \log(\nu_{\ell t})\right).$$

Here the Antoniak distribution introduced by Antoniak (1974) is simply the distribution of the number of clusters in a partition of $n_{\ell tc}$ items under a CRP with concentration parameter $\beta_t^{q_\ell}\eta_{tc}^{q_\ell}$.

To update z_i, we compute its conditional distribution given the rest of the variables:

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \varepsilon\mu_t\right)\prod_m \prod_{\{\ell:q_\ell=m\}} \frac{\prod_c (n_{\ell tc}^{\neg i} + \beta_t\eta_{tc})^{\mathbb{I}(y_{i\ell}=c)}}{(n_{\ell t}^{\neg i} + \beta_t)^{\mathbb{I}(y_{i\ell}\neq 0)}}.$$

The complexity of updating the $\boldsymbol{z}$ variables is $O(NLTC)$. Finally, we use the same scheme as the one applied in Section 3.1 to update α.

4. Related Work

Dawid and Skene in a seminal work proposed a model in which each user is characterized by a confusion matrix, and they use the EM algorithm to estimate the most likely values of both the parameters governing the behavior of each user and the ground truth (Dawid and Skene, 1979). Similar models have been applied to depression diagnosis (Young et al., 1983) and myocardial infarction (Rindskopf and Rindskopf, 1986), among other areas.

Ghahramani and Kim (2003); Kim and Ghahramani (2012) proposed a Bayesian extension of the method proposed by Dawid and Skene (1979) called Independent Bayesian Combination of Classifiers (iBCC), whose graphical model is shown in Figure 1a. In our cBCC model, if α tends to infinity, every user is allocated in a different cluster, and it becomes equivalent to the iBCC model. We see that in this case, the correlation a priori among two users (Equation 3) is zero. Also, in the hcBCC model, if each component of ϕ tends to infinity, and we also make the quantities a_t and b_t tend to infinity with a fixed $\frac{a_t}{b_t}$ ratio for all t, then we recover the iBCC model with $\eta_t = \gamma_t$ and $\beta_t = \frac{a_t}{b_t}$. If α tends to ∞, then the model is equivalent to the iBCC model, but with additional priors on η and β. To sum up, we can see each the cBCC and the iBCC as particularizations of the hcBCC model, which capture more complex relationships among the users.

Ghahramani and Kim (2003); Kim and Ghahramani (2012) also presented two extensions. The first one uses a latent variable that categorizes the instances in two classes: easy and difficult to classify. The assumption is that the annotators have the same behavior regarding the easy instances, while they are different for the difficult ones. In the second one, they propose a more flexible correlation model based on a factor graph. However, these models do not identify groupings of users.

Simpson et al. (2011) extend the proposal of Ghahramani and Kim (2003); Kim and Ghahramani (2012) in two directions. First, they derived a variational inference algorithm for the iBCC, which is more efficient for large data-sets. Second, they apply community detection algorithms to the estimated confusion matrices to detect clusters of users with the same behavior. Recently, they have

extended the model to the case in which the properties of the users can vary in time (Simpson et al., 2013). In both cases, the detection of groups of users is made in a post-hoc manner and therefore, this information is not used to improve the estimation of the confusion matrices of the users or the ground truth estimate. To the extent of our knowledge, only Kajino et al. (2013) perform the inference of the groups of users and the ground truth at the same time, using convex optimization. However, the performance depends on a constant that controls the strength of the clustering and for tuning this constant, the authors rely on a labeled validation set. Our algorithm, on the other hand, is fully unsupervised and therefore can be apply to the standard problem presented by Ghahramani and Kim (2003); Kim and Ghahramani (2012).

Recently, a paper on the inconsistency of the DP Mixture Model to estimate the true number of components was published (Miller and Harrison, 2013). However, we are not interested in estimating the "true" number of users' clusters, specially since this is not a well defined measure in a real crowdsourcing application. Instead, we look for identifying a clustering of the users that improves the performance and helps us to better understand the different types of users that are present in the crowdsourcing application.

Another research line that is related to the problem is relaxing the assumption of the existence of one single gold standard, which is a limiting assumption when the tasks involved in the crowdsourcing problem are subjective and accept multiple reasonable answers (Wauthier and Jordan, 2011; Tian and Zhu, 2012). In this paper, we focus on a crowdsourcing scenario with a well defined gold standard that we aim to predict.

5. Experiments

In this section, we firstly use synthetic data-sets based on different assumptions to validate our models. In the second part we use publicly available real data-sets to compare our two models with state-of-the-art algorithms highlighting their advantages.

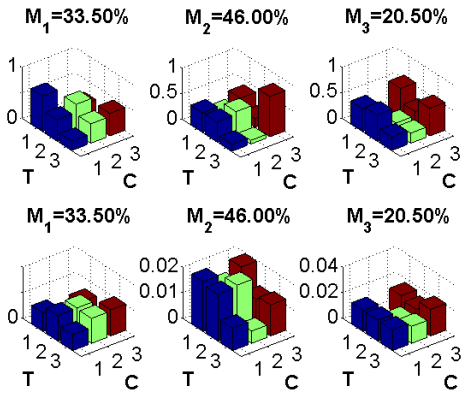
5.1 Synthetic Data-sets

We generate three different data-sets following respectively the assumptions of the hcBCC, cBCC and iBCC models. In order to analyze the properties of the algorithms, we apply each of them to each of the generated data-sets.

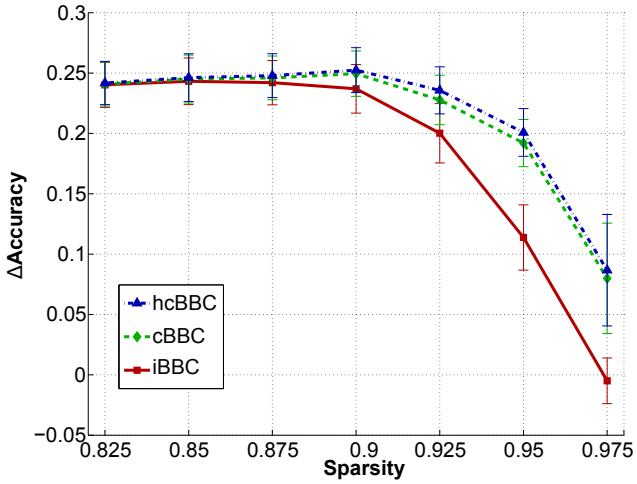
Firstly, we generate a synthetic database called data-set1 following the generative model for the hcBCC model. This data-set has 500 labeled instances provided by 200 users. The number of categories is $C = 3$. These users belong to 3 clusters with properties shown in Figure 3a, where we can see the mean of each cluster, their variances and the percentage of users allocated to each of them.

We analyze the behavior of the different algorithms with respect to the sparsity of the input matrix $\boldsymbol{Y}$. In particular, we randomly erase a percentage of the entries from 82.5% missing entries to 97.5% in steps of 2.5%. This high sparsity levels are typical in crowdsourcing applications, where the idea is to distribute the load of labeling a data-set among many users, and therefore each of them only labels a small subset of the data-set.

In the iBCC, the diagonal elements of $\boldsymbol{\eta}$ are 0.7 while the off diagonal are 0.3, which reflect our prior belief that users perform better than random. All the elements of $\boldsymbol{\beta}$ are 3. In the cBCC model, the hyper parameters of $\boldsymbol{\alpha}$ are $a_\alpha = 1$ and $b_\alpha = 10$. This values agree with our prior belief that if the annotations are very scarce, simpler algorithm like majority voting are more suitable and therefore,



(a) Users' properties for data-set1. (Upper row) Mean of the clusters. (Lower row) Variance of the clusters.



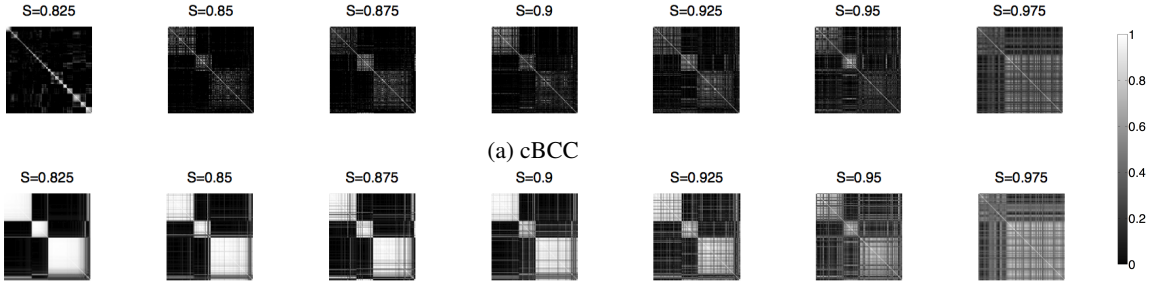
(b) Performance for data-set1

Figure 3: Results for data-set1. a) Characteristics of the users' clusters present in data-set1. M_i denotes the percentage of users allocated to cluster i, T is the ground truth label, and C is the user label. b) Results for data-set 1. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels.

we should favor partitions with a small number of clusters. For these parameters, in the limiting case when the sparsity of $\boldsymbol{Y}$ tends to 100%, the average number of clusters tends to 1. Finally, in the hcBCC model, we set γ and ϕ to the values of $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ in the cBCC model respectively. All the components of a_t are set to 30 while all the components of b_t are set to 2. This reflects our prior belief that the variability among the users inside the clusters should be less than the variability across clusters.

We run the MCMC for 10,000 iterations. After the first 3,000 we collect 7,000 samples to compute $\boldsymbol{z}$ and $\boldsymbol{\pi}$. In the cBCC and hcBCC, we set to five the number of iterations used to sample α following the algorithm proposed by Escobar (1994). In the hcBCC we fix the number of auxiliary clusters used by the *Reuse Algorithm* to $h = 10$.

The increment in accuracy of ours proposals and the iBCC algorithm with respect to majority voting is shown in Figure 3b. The two proposed models outperform iBCC as expected. This improvement of both methods cBCC and hcBCC is particularly significant when the level of sparsity is high, which is a situation that we face in the early stages of a crowdsourcing project. In this case there is not enough information to accurately estimate the confusion matrix of every user independently. We can see that the performance of iBCC drops below the performance of the majority voting algorithm, which assumes all users are similar. Therefore, identifying a clustering structure that allows to share some parameters among the users helps to increase the accuracy of the estimates. Notice that the performance obtained by Simpson et al. (2011) would be equal to the performance of the iBCC model given that it identifies the users' clusters after the ground truth has been estimated, so it does not affect the performance of the algorithm.



(a) cBCC

(b) hcBCC

Figure 4: Co-occurrence matrices of the users

To further analyze the cluster structure identified by the algorithms, in Figure 4 we represent the co-occurrence matrix of the users. The position (ℓ, ℓ') is the probability of ℓ and ℓ' belonging to the same cluster. We can see that the clusters identified by the hcBCC are more useful than the ones extracted by the cBCC, because in a practical situation we are not normally interested in finding users with exactly the same behavior, but users with similar characteristics. For example, we can see that when 82.5% of the annotations are missing, the hcBCC algorithm identifies the 3 main groups of users while the cBCC algorithm identifies instead a much larger number of groups because of the constraint that all users of a cluster must have the same properties. So, although both algorithms' performance is similar, the clustering provided by the hBCC is easier to interpret and gives a simpler explanation of the data.

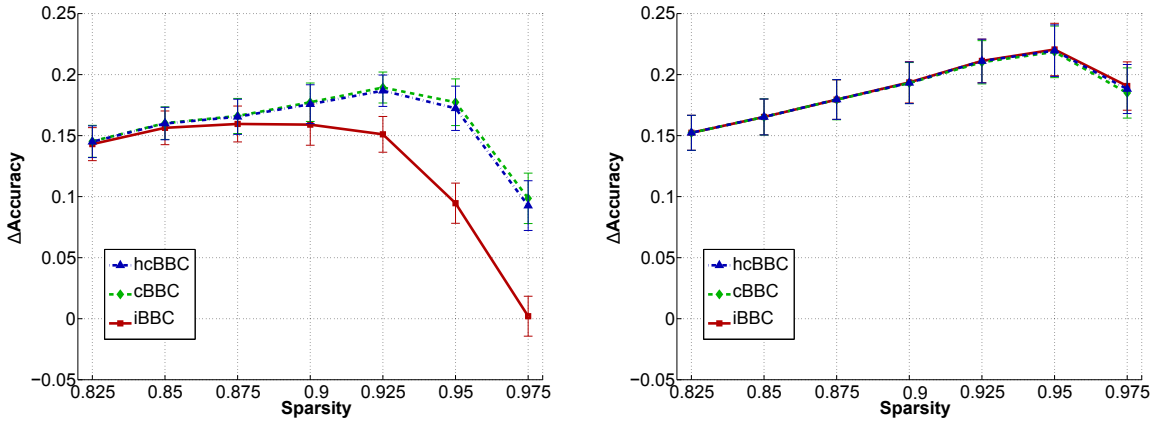
Finally, we test with data-sets that are generated following the iBCC and cBCC models. First, we create a new data-set (data-set2) in which the mean confusion matrix of each cluster is the same as in data-set1 which is shown in Figure 3a. However, in this case the variability of the confusion matrices inside each cluster is zero. Therefore, this new data-set follows the assumptions made by the cBCC model. Again, the performance of the cBCC and the hcBCC models outperforms iBCC as expected (see Figure 5a). However, even though data is generated from the cBCC which is a simpler model than the hcBCC, hcBCC is able to discover the underlying structure of the users and gets a performance which is on par with the cBCC. The hcBCC does not degrade the solution although it is more flexible.

In the last database called data-set3, we generate all the instances from the same clustering (M_2 in Figure 3a). In this case there is no different clusters of users and each of them has its own confusion matrix. Therefore, this data-set fulfill the assumptions of the iBCC. In Figure 5b we see that the performance of the two proposed models is identical to iBCC. To sum up, we see that the performances of cBCC and hcBCC dominate iBCC under all conditions tested.

5.2 Real Data-sets

In this Section, we use 4 publicly available crowdsourced data-sets with $C = 2$ whose principal characteristics are described in Table 1 (Raykar and Yu, 2012).

To choose the hyper-parameters we follow the reasoning of Section 5.1. Specifically, in the iBCC the diagonal elements of $\boldsymbol{\eta}$ are 0.7 while the off diagonal are 0.3, and all the elements of $\boldsymbol{\beta}$ are set to 3. In this way, we incorporate our prior belief that users are imperfect but perform better



(a) Performance for data-set2

(b) Performance for data-set3

Figure 5: Results for data-set2 and data-set3. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

than chance. In the cBCC model we use the same value for η and β so that the comparison is fair. Finally for the hcBCC model, γ is set to the same value used for η in the previous models. All the components of a_t are set to 20 while all the components of b_t are set to 2, reflecting our belief that the variability inside clusters should be lower than the variability across clusters. We fix $a_\alpha = 1, b_\alpha = 10$ in both, cBCC and hcBCC. We run the MCMC for 10,000 iterations and we discard the first 3,000 to compute the posterior distribution of z and π.

In Table 2, we see the performance of the different algorithms in terms of accuracy predicting the ground truth. In particular, we see that the performances of the cBCC and the hcBCC are better than that of the iBCC in the last three data-sets, i.e. rte, temp and valence. On the other hand, in the bluebird data-sets the iBCC performs better. Notice again that the performance of the algorithm described by Simpson et al. (2011) would be exactly equal to the one of the iBCC, given that the communities of users are inferred after the ground truth is inferred and therefore, it does not affect the accuracy in any way.

The performance difference between the cBCC and the hcBCC is only significant in the valence data-set. However, the main advantage of the hcBCC model over the cBCC is clear when we represent the average number of clusters (See Figure 6 and Table 2). Even though the cBCC model correctly captures the clustering structure of the users, forcing all users of a cluster to share the same confusion matrix translates into a large number of clusters, some of them with very similar properties.

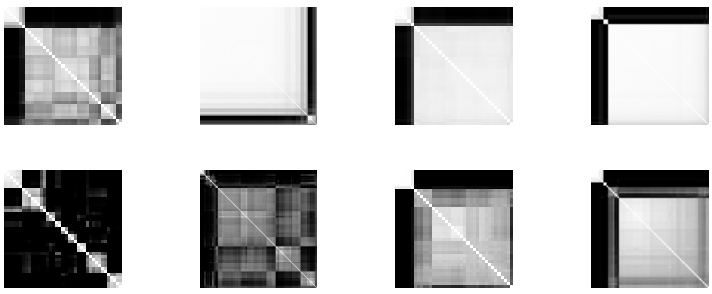
The hcBCC identifies a smaller number of clusters that are much more interpretable, in the sense that it perfectly identifies each kind of clusters thanks to its additional flexibility. We are not interested in identifying clusters of users with the exact same behavior, but what we really want is to find clusters of users that behave in a similar way, so we can establish strategies to boost the overall performance of the crowdsourcing system, i.e. by rewarding the most efficient labelers, avoiding spammers or by better defining the description of the task based on the biases identified in the clusters of users.

data-set	N	L	μ_n	μ_l	Sparsity (%)	Brief Description
bluebird	108	39	108	39	0	Identify whether there is a Indigo Bunting or Blue Grosbeak in the image
rte	800	164	49	10	93.90	Identify wether the second sentence can be inferred from the first
valence	100	38	26	10	73.68	Identify the overall positive or negative valence of the emotional content of a headline
temp	462	76	61	10	86.84	Users observe a dialogue and two verbs from the dialogue and have to identify whether the first verb event occurs before or after the second

Table 1: Description of the real data-sets. N and L denotes the number of instances and users respectively. μ_n stand for the mean number of instances labeled by a user and μ_l designate the mean number of users that label an instance.

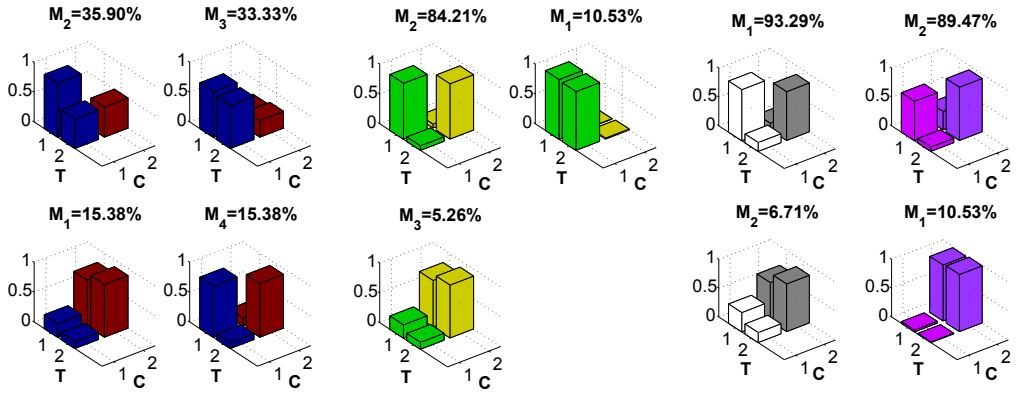
data-set	Accuracy(%)				Average number of clusters	
	Majority	iBCC	cBCC	hcBCC	cBCC	hcBCC
bluebird	75.93	**89.81**	88.89	88.89	11.32 ± 0.04	3.31 ± 0.09
rte	91.88	92.88	**93.12**	**93.12**	7.70 ± 0.07	2.30 ± 0.06
valence	80.00	85.00	88.00	**89.00**	3.5 ± 0.04	2.25 ± 0.02
temp	93.94	94.35	**94.37**	**94.37**	6.20 ± 0.03	3.2 ± 0.02

Table 2: Results for the real data. Mean accuracy of the different algorithms [2]. Average number of clusters (mean $\pm$ one standard deviation).



(a) bluebird (b) rte (c) valence (d) temp

Figure 6: Co-occurrence matrix of the users. (Upper row) hcBCC. (Lower row) cBCC.



(a) bluebird (b) temp (c) rte (d) valence

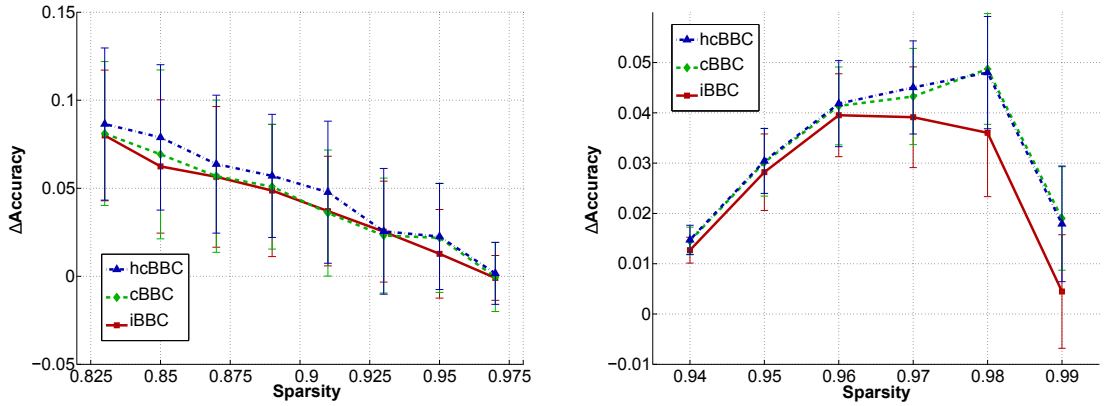
Figure 7: Mean confusion matrices of the user's clusters identified by hcBCC. M_i denotes the percentage of users allocated to cluster i, T is the ground truth label, and C is the user label.

In Figure 7 we show as an example the mean confusion matrix of the hcBCC clusters in the data-sets. It shows very interpretable clusters that are useful for the modeler. In the bluebird data-set we can clearly identify a small subset of experts ($M_4 = 15.38\%$) who shows a high performance labeling the bird images. In addition, we find that the biggest cluster ($M_2 = 35.90\%$) corresponds to users whose accuracy is high when the real class is $z = 1$ (images of Blue Grosbeak) but performs poorly when the class is $z = 2$ (images of Indigo Bunting). Finally, we have two clusters of spammers. In the first cluster ($M_1 = 15.38\%$) users tend to label all images as belonging to class $z = 2$ and in the second ($M_3 = 33.33\%$) users tend to label all images as $z = 1$. In the temp data-set, we can observe that the majority of the users ($M_2 = 84.21\%$) are experts, but there are again two small clusters of spammers. As for the rte data-set, most of the users have a good performance ($M_1 = 93.29\%$). The remaining users are bias toward labeling instances as belonging to class $z = 2$. Finally, in the valence data-set we can see that the majority of the users ($M_2 = 89.47\%$) are very accurate identifying instances belonging to class $z = 2$ and have a medium performance when $z = 1$. In addition we find a small cluster of users that have labeled almost every instance as $z = 2$. All this information about the underlying clustering structure of the users in the data-sets can be used in a real crowdsourcing application to develop efficient strategies to minimize the cost of a crowdsourcing project maximizing the performance.

To conclude this Section, we evaluate the performance of the algorithms in the real data-sets for different levels of sparsity. Following the procedure in Section 5.1, we create 50 random databases for each level of sparsity. We do that in such a way that every instance has at least one label and every user provides at least one label. The results are shown in Figure 8.

In the data-set bluebird and temp, we observe that finding clusters of users does not have a significant effect in terms of accuracy. However, the cBCC and the hcBCC models do not degrade the performance and give us some insight about the users in the crowdsourcing application (See

2. The standard deviations are less than 10^{-4} and are not shown



(a) Performance for bluebird

(b) Performance for rte

(c) Performance for valence

(d) Performance for temp

Figure 8: Results for real data-sets. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

Figure 7). In the rte and valence data-sets, the inference of the clustering structure of the users also translates into an improvement in terms of accuracy. In the rte data-set, this improvement is not significant for the original sparsity level, but it becomes more significant when the sparsity is increased. What happens is that when the sparsity is very high, there are very few annotations provided by each user, and the iBCC algorithm fails to infer the properties of each user separately.

In the valence data-set, we can even see that the performance of the iBCC model drops below the performance of a simple majority voting algorithm when the sparsity is increased. However, the cBCC and hcBCC outperform the majority voting algorithm for every sparsity level. Again the iBCC model does not have enough information to infer the properties of each user and a simpler model like majority voting, which assume that all users have the same level of expertise, performs better. Actually, what is happening is that the the CRP prior used in the cBCC and the hcBCC

models favors partitions with a small number of clusters. When the input matrix $\boldsymbol{Y}$ is very sparse, the prior term dominates over the likelihood and all users tend to be grouped in the same cluster.

6. Conclusions

We have proposed two new Bayesian nonparametric models to merge the information provided by the users in a crowdsourcing system. In addition, the algorithms detect clusters of users that have similar behaviors and use this information to improve the ground truth estimate. In the cBCC model, we have used a CRP to infer the partitioning of the users such that users in the same cluster are constrained to have the same properties. In the hcBCC model, we have used a hierarchical structure to increase the flexibility. In particular, each user has its own properties, but users assigned to the same cluster have similar properties. In this way, it finds smaller number of clusters that are easy to interpret.

We have shown how these new models relate to the iBCC model and analyzed the correlation structure among the users as a consequence of the clustering. We have proposed MCMC methods to infer the parameters of both models and performed several experiments with synthetic and real databases, which have shown that the algorithms outperform the current state-of-the-art.

Finally, we comment possible extensions. The ground truth estimated by the proposed algorithms, can be used to train a supervised learning algorithm. Raykar et al. (2010); Groot et al. (2011); Welinder et al. (2010) propose to train a classifier directly with the noisy labels provided by the users. It would be interesting to extend the models following this line. Also, the models assume consistent users, i.e the users have the same properties across the whole instance space. An extension would be considering users with nonuniform behavior (Zhang and Obradovic, 2011; Yan et al., 2010), i.e. a user can be an expert for a subset of the instances while can act as a novice in another subset. Also, a future research line is to propose new inference schemes that improve the scalability of the methods.

Acknowledgments

Pablo G. Moreno is supported by an FPU fellowship from the Spanish Ministry of Education (AP2009-1513). This work has been partly supported by Ministerio de Economía of Spain ('COMONSENS', id. CSD2008-00010, 'ALCIT', id. TEC2012-38800-C03-01, 'COMPREHENSION', id. TEC2012-38883-C02-01) and Comunidad de Madrid (project 'CASI-CAM-CM', id. S2013/ICE-2845). This work was also supported by the European Union 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861. Yee Why Teh's research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617411. Finally, the authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the quality of this paper

Appendix A: Correlations in the CBCC Model

In the iBCC model, the joint probability of two users given the ground truth is

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times$$
$$\times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})} = p(y_{i\ell}|z_i = t) \times p(y_{i\ell'}|z_i = t).$$

Therefore

$$\text{corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = 0.$$

In the cBCC model, we have the following expression for the joint distribution of $y_{i\ell}$ and $y_{i\ell'}$

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \left(\frac{1}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} \neq 0))} \times \right.$$
$$\left. \times \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right] + \left(\frac{\alpha}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \times \right.$$
$$\left. \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right].$$

We can now compute the covariance in the following way

$$\text{cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\mathbb{I}(y_{i\ell'} = b)\} - \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\}\mathbb{E}\{\mathbb{I}(y_{i\ell'} = b)\} =$$
$$= \left(\frac{1}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0) + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(a = c, a \neq 0) + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t \eta_{tc})} - \right.$$
$$\left. - \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(a = c, a \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right].$$

Assuming that $a \neq 0$ and $b \neq 0$, and considering the cases where $a = b$ and $a \neq b$ we obtain the following equation for the covariance

$$\text{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i'\ell'} = b)|z_i = t) = \begin{cases} -\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}\eta_{tb} & a \neq b \\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}(1-\eta_{ta}) & a = b \end{cases}.$$

Here we have taken into account that $\Gamma(x+1) = x\Gamma(x)$. Once we get the expression of the covariance, we divide it by the square root of the variances to get the correlation

$$\text{Corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \frac{\text{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b))}{\sqrt{\text{Var}(\mathbb{I}(y_{i\ell} = a))\text{Var}(\mathbb{I}(y_{i\ell'} = b))}}.$$

It is straightforward to see that $\text{Var}(\mathbb{I}(y_{i\ell} = a)) = \eta_a(1-\eta_a)$, getting the expected result.

Appendix B: Inference Details of the HCBCC Model

The posterior distribution of the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$ is proportional to the following expression.

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) \propto p(\boldsymbol{Y}|\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{\pi}) \times p(\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_{\ell}\prod_{t}\left[\frac{\Gamma(\beta_t^{q_\ell})}{\Gamma(n_{\ell t}+\beta_t^{q_\ell})}\prod_{c}\frac{\Gamma(n_{\ell tc}+\beta_t^{q_\ell}\eta_{tc}^{q_\ell})}{\Gamma(\beta_t\eta_{tc}^{q_\ell})}\right]\times$$
$$\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m).$$

We cannot compute an analytic expression for $p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})$ because the prior on $p(\boldsymbol{\eta},\boldsymbol{\beta})$ is no longer conjugate of the likelihood of the observations. The idea is to include two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that we can compute the joint distribution $p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})$. To do so, we use the following relation between the gamma function and the Stirling numbers of the first kind denoted by S

$$\frac{\Gamma(x+n)}{\Gamma(x)} = (x)_n = \sum_{s=0}^{n} S(n,s)(x)^s.$$

Here $(x)_n$ denotes the Pochhammer symbol. Taking into account also the definition of the beta distribution we reach the following expression

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) \propto \prod_{\ell}\prod_{t}\left[\int_0^1 \nu^{\beta_t^{q_\ell}-1}(1-\nu)^{n_{\ell t}-1}d\nu\prod_{c}\sum_{s=0}^{n_{\ell tc}}S(n_{\ell tc},s)(\beta_t^{q_\ell}\eta_{tc}^{q_\ell})^s\right]\times$$
$$\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m).$$

And therefore we can introduce a set of auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that the joint distribution is given by

$$\begin{aligned} p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) &\propto \prod_{\ell}\prod_{t}\left[\nu_{\ell t}^{\beta_t^{q_\ell}-1}(1-\nu_{\ell t})^{n_{\ell t}-1}\prod_{c}S(n_{\ell tc},s_{\ell tc})(\beta_t^{q_\ell}\eta_{tc}^{q_\ell})^{s_{\ell tc}}\right]\times \\ &\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m). \end{aligned} \tag{4}$$

and such that

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) = \int p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})d\boldsymbol{\nu}d\boldsymbol{s}.$$

From Equation 4 it is straightforward to compute the necessary conditional distributions to implement the Gibbs sampler (See Section 3.2).

References

Amazon. Amazon mechanical turk. `http://www.mturk.com`, 2005.

C. E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

D. Blackwell and J. B. Macqueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society*, 28(1):pp. 20–28, 1979.

M. D. Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):pp. 268–277, 1994.

S. Favaro and Y. W. Teh. Mcmc for normalized random measure mixture models. *Preprint*, 2012.

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

C. Fraly and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Z. Ghahramani and H-C. Kim. Bayesian classifier combination. *Technical Report*, 2003.

P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with gaussian processes. In *International Conference on Artificial Neural Networks*, pages 159–164. Springer-Verlag, 2011.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):pp. 161–173, 2001.

H. Kajino, Y. Tsubo, and H. Kashima. Clustering crowds. In *Association for the Advancement of Artificial Intelligence*, pages 1120–1127, 2013.

M. Kalli, J. E. Griffin, and S. G. Walker. Slice sampling mixture models. *Statistics and Computing*, 21(1):93–105, January 2011.

H-C. Kim and Z. Ghahramani. Bayesian classifier combination. *Journal of Machine Learning Research*, 22:619–627, 2012.

C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. Nichol, J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy zoo 1 : data release of morphological classifications for nearly 900,000 galaxies. 2010.

S. N. MacEachern and P. Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):pp. 223–238, 1998.

J. W. Miller and M. T. Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. In *Neural Information Processing Systems*, 2013.

R. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

J. Pitman. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, U.C. Berkeley, 2002. Lecture notes for St. Flour course.

V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, pages 491–518, 2012.

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

D. Rindskopf and W. Rindskopf. The value of latent class analysis in medical diagnosis. *Statistics in Medicine*, 5(1):21–27, 1986.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Special Interest Group on Information Retrieval*, Special Interest Group on Information Retrieval, pages 253–260, New York, NY, USA, 2002.

E. Simpson, S. J. Roberts, A. Smith, and C. Lintott. Bayesian combination of multiple, imperfect classifiers. In *Neural Information Processing Systems*, pages 1–8, Oxford, 2011. University of Oxford.

E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision making and imperfection*, volume 474 of *Studies in Computational Intelligence*, pages 1–35. 2013.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2003.

Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In *Knowledge Discovery and Data Mining*, pages 226–234, 2012.

S. G. Walker. Sampling the dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, 2007.

F. L. Wauthier and M. I. Jordan. Bayesian bias mitigation for crowdsourcing. In *Neural Information Processing Systems*, pages 1800–1808, 2011.

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

Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, and J. Dy. Modeling annotator expertise: learning when everybody knows a bit of something. *International Conference on Artificial Intelligence and Statistics*, 2010.

M.A. Young, R. Abrams, M.A. Taylor, and H.Y. Meltzer. Establishing diagnostic criteria for mania. *The Journal of Nervous and Mental Disease*, 171(11):676–682, 1983.

P. Zhang and Z. Obradovic. Learning from inconsistent and unreliable annotators by a gaussian mixture model and bayesian information criterion. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 553–568, 2011.