Revisiting Machine Learning from Crowds a Mixture Model for Grouping Annotations

Francisco Mena(✉) and Ricardo Ñanculef(✉)

Federico Santa María University, Valparaíso, Chile
francisco.mena@alumnos.inf.utfsm.cl, jnancu@inf.utfsm.cl

Abstract. Today, supervised learning is widely used for pattern recognition, computer vision and other tasks. In this setting, data need to be explicitly annotated. Unfortunately, obtaining accurate labels can be difficult, expensive and time-consuming. As a result, many machine learning projects rely on labelling processes that involve *crowds*, i.e. multiple subjective and inexpert annotators. Handling this noise in a principled way is an important challenge for machine learning, called learning from crowds. In this paper, we present a model that learns patterns of label noise by grouping annotations. In contrast to previous art, we do not model specific labeling patterns for each annotator but explain the data using a fixed-size mixture model. This approach allows to handle a sparse distribution of labels among annotators and obtain a model with less parameters that can scale better to large-scale scenarios. Experiments on real and simulated data illustrate the advantages of our approach.

Keywords: Learning from crowds · Mixture model · Multiple annotations · Clustering

1 Introduction

In the last years, artificial intelligence has been widely spread into several areas of science and industry. Many of these applications rely on *supervised learning*, i.e., methods capable to realize an input-output mapping from large amounts of data (inputs) annotated with *ground-truth* labels (output). In many real-world tasks however, obtaining accurate labels can be difficult or infeasible. Consider, for instance, the problem of classifying a bio-medical image into a set of clinical conditions of interest. The ground-truth label could only be obtained after performing slow, expensive and invasive experiments in physical labs. Collecting multiple subjective, but possibly inaccurate labels, from annotators of varying levels of expertise, is often more feasible and cheaper [10]. Current crowdsourcing platforms, such as Amazon Mechanical Turk (AMT), are making this procedure more common, especially in computer vision and natural language processing tasks. Unfortunately, as inexpert annotators can be inaccurate, spammer or even malicious, training a traditional supervised model, with annotations

I. Nyström et al. (Eds.): CIARP 2019, LNCS 11896, pp. 493–503, 2019.
https://doi.org/10.1007/978-3-030-33904-3_46

collected in this way, is often ineffective. Similarly, simple aggregation rules such as majority voting, that reduce crowd annotations into a single label, can fail if the expertise of the different annotators vary significantly [12] or if, as usual in crowd-sourcing platforms, the distribution of labels among annotators is sparse. The problem of learning from annotations of varying reliability can be traced back to [2]. Here, Dawid and Skene proposed a method, based on the EM algorithm, that automatically detects the ability of each annotator and estimates a consensus label that can be used to train a standard classifier. Many subsequent methods are extensions of this framework. For instance, Raykar et al. [7] proposed to directly train the ground-truth predictor in the maximization step of the EM algorithm. Kajino et al. [4] proposed to train a separate model for each annotator and then infer a consensus model rather than consensus label. More recently, Albarqouni et al. [1] have proposed the use of deep learning to implement the ground-truth predictor of [7] and Rodrigues et al. [9] introduced more simple training procedures based on back-propagation method.

A common limitation of the current models is that the number of learnable parameters becomes very large as the number of annotators increases, limiting their scalability to massive crowd-sourcing scenarios. In this paper, we present a model for learning from crowds that detects patterns of label noise by grouping/clustering annotations together. In contrast to previous work, we do not model specific labeling patterns for each annotator, but explain the annotations using a fixed-size generative mixture model. As preliminary experiments confirm, this allows to obtain a method that can scale better to large-scale scenarios and can improve the state-of-the-art in sparse annotation scenarios.

The remainder of this paper is organized as follows: Sect. 2 formalizes the problem and introduces the notation used in for the proposed method, that is present in Sect. 3; Sect. 4 provides a discussion of related works; Sect. 5 experimentally compares our approach with baseline methods; finally, Sect. 6 summarizes the conclusions of this work.

2 Problem Statement and Notation

Consider an input pattern $\boldsymbol{x} \in \mathbb{X}$ and a ground-truth label $\boldsymbol{z} \in \mathbb{Z}$ observed with probability distribution $p(\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$ respectively. The goal of a supervised learning algorithm is to estimate the conditional $p(\boldsymbol{z}|\boldsymbol{x})$ from a set of examples of the form $S = \{(\boldsymbol{x}^{(1)}, \boldsymbol{z}^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, \boldsymbol{z}^{(N)})\}$, where $(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}) \sim p(\boldsymbol{x}, \boldsymbol{z}) \ \forall i \in [N]$. More specifically, given a loss function $Q : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$ and a hypothesis space $\mathcal{H} \subset \mathbb{Z}^{\mathbb{X}}$, a supervised learning algorithm attempts to minimize $\mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} \left(Q(f(\boldsymbol{x}), \boldsymbol{z})\right)$ in $\mathcal{H}$ when $p(\boldsymbol{x}, \boldsymbol{z})$ is unknown and only S is given.

In *learning from crowds*, one has the same objective, but the ground-truth labels $\boldsymbol{z}^{(i)}$ corresponding to the input patterns $\boldsymbol{x}^{(i)}$ are not observed. Instead, one is given multiple noisy labels $\mathcal{L}_i = \{\boldsymbol{y}_i^{(1)}, \ldots, \boldsymbol{y}_i^{(T_i)}\}$, $\boldsymbol{y}_i^{(\ell)} \in \mathbb{Z}$ for each training pattern $\boldsymbol{x}^{(i)}$. These labels have been collected from T_i annotators and do not follow the ground-truth distribution $p(\boldsymbol{z}|\boldsymbol{x})$, but are observed according to an unknown labelling process $p(\boldsymbol{y}^{(\ell)}|\boldsymbol{x}, \boldsymbol{z})$, which is another objective to study.

In general, the annotations $\mathcal{L}_i$ for a pattern $\boldsymbol{x}^{(i)}$ come from a subset $\mathcal{A}_i$ of the set of all the annotators $\mathcal{A}$ participating on the labelling process. A common assumption is that $\mathcal{A}_i = \mathcal{A}$ (**dense** scenario). A more challenging problem is the scenario in which a variable number of labels is collected by data point and annotator, i.e. $|\mathcal{A}_i| \neq |\mathcal{A}_j| < |\mathcal{A}| = T$ (the **sparse** scenario). Furthermore, we are interested in the so-called **Global** scenario, in which we are given $\mathcal{L}_i$, but we do not know which annotators provided the labels i.e., we know $|\mathcal{A}_i|$ but not $\mathcal{A}_i$. The opposite scenario, referred to as **Individual** , allows to study the properties of each annotator separately.

Focus. For sake of simplicity, we concentrate in the pattern recognition case, that is, we let $\mathbb{Z}$ be a small set of K categories or classes $\{c_1, c_2, \ldots, c_K\}$.

3 Proposed Method

3.1 Model Specification

As in previous works, we represent the ground-truth label as a latent/hidden variable $\boldsymbol{z}$ with (unknown) probability distribution $p(\boldsymbol{z}|\boldsymbol{x})$. To explain an annotation $\boldsymbol{y} \in \mathbb{Z}$, assigned to an input pattern $\boldsymbol{x}$, we propose a generative finite mixture model (GMM) of the form

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum\nolimits_{m=1}^{M} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{g} = m) \cdot p(\boldsymbol{g} = m \mid \boldsymbol{x}) = \sum\nolimits_{m=1}^{M} p_m(\boldsymbol{y}|\boldsymbol{x}) \cdot \alpha_m, \quad (1)$$

where $p_m(\boldsymbol{y}|\boldsymbol{x})$ represents one of M possible sub-models, $\boldsymbol{g}$ is a categorical random variable with values in $[M] = \{1, 2, \ldots, M\}$ identifying the group/component that generated the observation $\boldsymbol{y}$, and $\alpha_m = p(\boldsymbol{g} = m)$ is the *a-priori* probability that pattern $\boldsymbol{x}$ is annotated according to $p_m(\boldsymbol{y}|\boldsymbol{x})$. Note that we are assuming that the mixing coefficients α_m are independent of $\boldsymbol{x}$. If we relax this assumption, we obtain a mixture of experts model (MOE) with gating functions $\alpha_m(x)$.

The components $p_1(\boldsymbol{y}|\boldsymbol{x}), \ldots, p_M(\boldsymbol{y}|\boldsymbol{x})$ in (1) represent different *annotation patterns* that can occur in the labelling process. They may correspond to clusters/groups of annotators that follow similar rules to annotate data or groups of annotations for which similar mistakes were made. The relationship between an annotation $\boldsymbol{y}$ and the ground-truth $\boldsymbol{z}$ for $\boldsymbol{x}$ is obtained as follows

$$\begin{aligned} p_m(\boldsymbol{y} = j \mid \boldsymbol{x}) &= \sum\nolimits_{k=1}^{K} p(\boldsymbol{y} = j, \boldsymbol{z} = k \mid \boldsymbol{x}, \boldsymbol{g} = m) \\ &= \sum\nolimits_{k=1}^{K} p(\boldsymbol{y} = j \mid \boldsymbol{g} = m, \boldsymbol{z} = k) \cdot p(\boldsymbol{z} = k \mid \boldsymbol{x}), \end{aligned} \quad (2)$$

where K is the number of classes and the second line was obtained by assuming that $\boldsymbol{y}$ is conditionally independent of $\boldsymbol{x}$ given $\boldsymbol{z}$, in order to keep a simple model. Indeed, this simplification allow us to parametrize $p_m(\boldsymbol{y}|\boldsymbol{x})$ using only K^2

parameters per sub-model and a single predictive model $f(\boldsymbol{x};\theta)$ that approximates the ground-truth distribution $p(\boldsymbol{z}|\boldsymbol{x})$, as Table 1 summarized. Substituting (2) into (1), we obtain the specification of the proposed model for annotation $\boldsymbol{y}$

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{k=1}^{K} \sum_{m=1}^{M} p(\boldsymbol{y} \mid \boldsymbol{g} = m, \boldsymbol{z} = k) \cdot p(\boldsymbol{z} = k \mid \boldsymbol{x}) \cdot p(\boldsymbol{g} = m) \quad (3)$$

3.2 Learning Objective

We start by introducing a data representation that full-fills the requirements of the **Global** and **sparse** scenarios defined in Sect. 2. We define $\boldsymbol{r}^{(i)}$ to be the K-dimensional vector whose components $\boldsymbol{r}_j^{(i)}$ are the frequencies of label c_j among the annotations $\mathcal{L}_i$ of a pattern $\boldsymbol{x}^{(i)}$. If we assume that those annotations are conditionally independent given $\boldsymbol{x}^{(i)}$, we obtain that $\boldsymbol{r}^{(i)}$ follows a Multinomial distribution with sample size T_i and probabilities $p_{ij} = p(\boldsymbol{y} = j|\boldsymbol{x}^{(i)})$ given by the model parametrization (see Table 1) on Eq. (3). The conditional log-likelihood of the data $G = \{(\boldsymbol{x}^{(i)}; \boldsymbol{r}^{(i)})\}_{i=1}^{N}$ is thus given by

$$\begin{aligned}
\ell(\Theta) = & \sum_{i}^{N} \log p_\Theta(\boldsymbol{r}^{(i)} \mid \boldsymbol{x}^{(i)}) = \sum_{i}^{N} \log \left(\text{const} \cdot \prod_{j=1}^{K} p_\Theta(\boldsymbol{y} = j \mid \boldsymbol{x}^{(i)})^{\boldsymbol{r}_j^{(i)}} \right) \\
& = \text{const} + \sum_{i}^{N} \sum_{j}^{K} \boldsymbol{r}_j^{(i)} \cdot \log p_\Theta(\boldsymbol{y} = j \mid \boldsymbol{x}^{(i)}) \\
& = \text{const} + \sum_{i}^{N} \sum_{j}^{K} \boldsymbol{r}_j^{(i)} \cdot \log \left(\sum_{m,k} \boldsymbol{\beta}_{k,j}^{(m)} \cdot f_k(\boldsymbol{x}^{(i)};\theta) \cdot \alpha_m \right). \quad (4)
\end{aligned}$$

The parameters of the proposed model, called *Crowd Mixture Model* (CMM), can be learnt to maximize $\ell(\Theta)$. Unfortunately, due to the log-sum, this optimization is not straightforward. We address this issue using the EM algorithm [3].

3.3 Training Procedure

By the Jensen inequality, we can consider any bi-variate distribution $q_{ij}(\boldsymbol{g}, \boldsymbol{z})$ assigning annotations among groups and ground-truth categories, to obtain the following lower bound of $\ell(\Theta)$

Table 1. Model parametrization. Entry (k, j) of $\boldsymbol{\beta}^{(m)}$ represents the probability that a pattern of class $\boldsymbol{z} = k$ is annotated as $\boldsymbol{y} = j$ by the group/component $\boldsymbol{g} = m$. The model $f(\boldsymbol{x};\theta)$, used to predict the ground-truth of $\boldsymbol{x}$, may have many parameters $|\theta|$, but this number is independent of the number of annotators T.

Term	Model	# Parameters
$p(\boldsymbol{g})$	Mixing coefficients $\alpha_m = p(\boldsymbol{g} = m)$	$M - 1$
$p(\boldsymbol{y}\|\boldsymbol{g}, \boldsymbol{z})$	Confusion matrix $\boldsymbol{\beta}^{(m)}$ for group m $p(\boldsymbol{y} = j\|\boldsymbol{g} = m, \boldsymbol{z} = k)$	$MK(K-1)$
$p(\boldsymbol{z}\|\boldsymbol{x})$	Neural net $f(\boldsymbol{x};\theta)$	Indep. of T

$$\ell(\Theta) \geq \text{const} + \sum_{i,j} \boldsymbol{r}_j^{(i)} \left[\sum_{m,k} q_{ij}(m,k) \cdot \log \left(\frac{\boldsymbol{\beta}_{k,j}^{(m)} \cdot f_k(\boldsymbol{x}^{(i)};\theta) \cdot \alpha_m}{q_{ij}(m,k)} \right) \right]. \quad (5)$$

The EM algorithm now follows easily. In one step, we improve our estimate of $q_{ij}(\cdot)$ to make the bound tight. Then, we optimize the lower bound in the model parameters Θ. The iteration of these two steps is guaranteed to converge to a local maximum of $\ell(\Theta)$. Exact solutions for our model are provided below.

E-step. For grouping the annotations based on ground-truth, we obtain

$$q_{ij}(m,k) = \tfrac{1}{N_{ij}} \boldsymbol{\beta}_{k,j}^{(m)} f_k(\boldsymbol{x}^{(i)};\theta)\alpha_m, \text{ with } N_{ij} = \sum_{m',k'} \boldsymbol{\beta}_{k',j}^{(m')} f_{k'}(\boldsymbol{x}^{(i)};\theta)\alpha_{m'}.$$

M-step. For the mixing coefficients and confusion matrices, we obtain

$$\alpha_m = \frac{\sum_{ij} \boldsymbol{r}_j^{(i)} \cdot q_{ij}(m,\cdot)}{\sum_{ij} \boldsymbol{r}_j^{(i)}}, \quad \boldsymbol{\beta}_{k,j}^{(m)} = \frac{\sum_i q_{ij}(m,k) \cdot \boldsymbol{r}_j^{(i)}}{\sum_{ij'} q_{ij'}(m,k) \cdot \boldsymbol{r}_{j'}^{(i)}}, \quad (6)$$

where $q_{ij}(m,\cdot) = \sum_k q_{ij}(m,k)$, then confusion matrix is a weighted average of annotations of that group. Now, defining $q_{ij}(\cdot,k) = \sum_m q_{ij}(m,k)$ and $\bar{\boldsymbol{r}}_k^{(i)} = \sum_j q_{ij}(\cdot,k)\boldsymbol{r}_j^{(i)}$, we obtain the following objective to minimize for the neural net:

$$J(\theta) = \sum\nolimits_{i,k} -\bar{\boldsymbol{r}}_k^{(i)} \cdot \log f_k(\boldsymbol{x}^{(i)};\theta) \propto \sum\nolimits_i \mathbb{H}\left(\bar{\boldsymbol{p}}^{(i)}, f(\boldsymbol{x}^{(i)};\theta)\right), \quad (7)$$

where $\mathbb{H}(\cdot,\cdot)$ is the *categorical cross-entropy loss* between the neural net and a "consensus" distribution on the categories, $\bar{\boldsymbol{p}}_k^{(i)} = \bar{\boldsymbol{r}}_k^{(i)} / \sum_{k'} \bar{\boldsymbol{r}}_{k'}^{(i)}$, which has been computed for $\boldsymbol{x}^{(i)}$ considering the confusion matrices and the mixing coefficients.

3.4 Group Assignment

Our model allows to cluster annotations and annotators, even outside the training data. Given any set of annotations $\mathcal{L} = \{\mathcal{L}_i\}$ for a pattern $\boldsymbol{x}^{(i)}$, we can compute the probability that these annotations were generated by the component p_m in our model as

$$p\left(\boldsymbol{g} = m|\mathcal{L}, X\right) = \frac{p\left(\mathcal{L}|\boldsymbol{g} = m, X\right) p(\boldsymbol{g} = m|X)}{\sum_{m'} p\left(\mathcal{L}|\boldsymbol{g} = m', X\right) p(\boldsymbol{g} = m'|X)} = \frac{p_m(\boldsymbol{y}_i^{(\ell)}|\boldsymbol{x}^{(i)})\alpha_m}{\sum_{m'} p_{m'}(\boldsymbol{y}_i^{(\ell)}|\boldsymbol{x}^{(i)})\alpha_{m'}}.$$

The probability $p(\boldsymbol{g} = m|\boldsymbol{a})$ that an annotator $\boldsymbol{a}$ belongs to the group m can be estimated with all her annotations. In addition, we can estimate the confusion matrix of an annotator as $\boldsymbol{\beta}_{\boldsymbol{a}} = \sum_m p(\boldsymbol{g} = m|\boldsymbol{a}) \cdot \boldsymbol{\beta}^{(m)}$.

4 Related Work

Existing methods to deal with multiple annotations can be grouped as follows.

Simple Aggregation Methods. These methods use simple summary statistics to reduce the crowd annotations into a single label that can be accepted by standard classifiers. The most used technique of this type is *Majority Voting* (MV), which has two versions in classification problems [8]: *hard-MV*, that selects the most frequent class among the annotations, and *soft-MV*, that defines the output of prediction as the relative frequency of the classes. As shown in [12], the accuracy of MV methods is limited if the annotators have very different levels of accuracy or in cases in which data points do not have many annotations.

Methods Without Predictive Model. These techniques also reduce crowd annotations into a single label and train a predictive model in a separate step. However, they devise specialized techniques to deal with annotators of varying expertise. A pioneer method is the algorithm of Dawid and Skene (DS) [2]. Here, the ability of each annotator is represented using a confusion matrix that can be learnt, together with the ground-truth of the training data, using the EM algorithm. Recently, [13] proposed an initialization method for the EM algorithm that allows to speed-up DS.

Methods with Predictive Model. These methods learn the ground-truth of the training data and the predictive model f approximating $p(\boldsymbol{z}|\boldsymbol{x})$ jointly, which avoids a second learning stage and allows the model f to learn labelling patterns that depend on $\boldsymbol{x}$. For instance, Raykar et al. [7] extended DS, using a logistic regression model to implement $p(\boldsymbol{z}|\boldsymbol{x})$. Almost simultaneously, Yan et al. [11] proposed to use a logistic model to predict the ability of the annotators. A method that avoids the use of the EM algorithm is presented by Kajino et al. [4]. It trains a logistic model for each annotator and then creates a consensus model. Unfortunately the complexity of [4,11] is increased considerably due to the large number of parameters per annotator. Addressing this issue, [8] proposed to change the latent variable of [6], modeling the reliability of each annotator. The main assumption is that an annotator provides completely random labels or annotates data according to a common baseline model. Unfortunately, this assumption represents two extreme possibilities that rarely take place in practice.

Deep Learning. Recent works have proposed the use of neural network models to implement $p(\boldsymbol{z}|\boldsymbol{x})$. For example, [1] extends [7] using a convolutional net and applies the model to a real cancer detection problem. [5] presents two methods that avoid the effect of *label noise* in neural network training. Unfortunately, a single confusion matrix is considered and it needs to be known before training. It is not evident how to use this method with multiple inaccurate annotators. Rodrigues et al. [9] encode the confusion matrices as additional weights of the neural network, avoiding the use of the EM algorithm. Unfortunately, the size of the so-called "crowd layer" grows linearly in the number of annotators.

Discussion. As pointed out in [14], nowadays there is no method that is superior to the others in all the cases, because different assumptions have to be fulfilled

to achieve good results. However, algorithms using a confusion matrix, as our method, to represent the ability of the annotators perform experimentally better than the others [14]. However, while almost all methods focus on modeling each annotator separately (**Individual** scenario), we propose a model with a fixed number of components, into which annotations and annotators can be allocated. As shown in Table 1 this makes the number of parameters independent of T. Besides *computational efficiency*, grouping annotations allows to increase *statistical efficiency*, especially in scenarios where annotators provide a small number of labels and so the estimation of the confusion matrices has to be performed using very few data.

5 Experiments

We evaluate our method on real and simulated scenarios, comparing it against four baselines from the state-of-the-art: *DL-DS* [2], *DL-EM* ([1] and generalized in [9]), and both versions of MV [8]: *hardMV* and *softMV*. We also include the upper bound performance of a model trained with the ground-truth, referred to as *Ideal*. In the vein of latest works, all the methods employ neural networks to implement the ground-truth predictor. All our code is made publicly available[1].

Simulated Scenario. To compare the methods on a controlled scenario, we simulated a crowd-sourcing process with annotators of varying expertise. Following [4,8], we simulated M levels of ability, by training a neural net on the ground-truth and randomly perturbing its weights with different levels of noise. As we use a confusion matrix to represent the ability of annotators, the matrix of each perturbed model was first calculated. Then, we created T annotators by selecting one of the M ability levels according to a probability distribution $p(\boldsymbol{g})$. To simulate sparse annotations, each data point is labelled by a random subset of the annotators T_i such that, in average, we obtain $\bar{T}_i$ annotators per point and a density of $D_t \approx N \cdot \bar{T}_i/T$ labels per annotator. Each annotator provides a label based on the ground-truth and her ability, i.e, the confusion matrix of the group. This annotation process is applied in two different flavors. In Setup (1), we simulate three uncorrelated isotropic Gaussians (representing classes), with 1000 data points each, centered on $(-0.5; 0)$, $(0.5; 0)$ and $(0; 0.5)$, with $\sigma^2 = 0.4^2$ (homocedasticity). We set $\bar{T}_i = 5$, $M = 3$ (experts, inexperts, spammers), and $p(\boldsymbol{g}) = (0.25; 0.55; 0.20)$. In Setup (2): we use the well-known CIFAR-10 dataset, composed of 60000 real images, classified into 10 categories, and set $\bar{T}_i = 3$, $M = 4$ (experts, inexperts, highly inexpert, spammers), $p(\boldsymbol{g}) = (0.20; 0.45; 0.15; 0.20)$.

Real Data. To evaluate the methods on a real crowd-sourcing scenario, we followed the setup of [9] on the LabelMe dataset. It contains 2688 images of 256×256 resolution, labelled into 8 possible classes by $T = 59$ annotators on Amazon Mechanical Turk. Each image has $\bar{T}_i = 2.6$ annotations in average, which leads to a density of $D_t = 43.2$ labels per annotator.

[1] https://github.com/FMena14/MixtureofGroups.

Table 2. Test accuracy of the different methods on a simulated crowd-sourcing scenario for values of T (columns) ranging from $T = 100$ to $T = 10000$. Marker † represents that the method could not be executed due to insufficient memory (16 GB available).

	Setup (1)						Setup (2)					
Method	100	500	1500	3500	6000	10000	100	500	1500	3500	6000	10000
softMV	69.34	66.21	66.87	68.48	67.00	66.49	63.35	65, 90	63.59	60.07	63.21	64.20
hardMV	79.57	82.49	80.51	81.57	74.30	79.07	71.09	69, 50	68.48	69.09	70.08	66.01
DL-DS	94.66	93.89	92.28	90.00	89.69	85.13	71.33	68, 49	68.08	66.86	†	†
DL-EM	93.97	93.99	92.18	88.27	76.47	67.01	81.38	80, 42	77.81	69.81	†	†
CMM	90.53	91.07	91.66	90.45	90.26	90.46	78.83	78, 36	79.35	77.92	78.45	78.96
Ideal	94.75						83.77					

Table 3. Performance of the different methods in a real crowd-sourcing scenario (LabelMe). Marker ⋄ represents no change with respect to the **Individual** setting. Acc. stands for Accuracy. Iters stands for iterations to converge.

	Individual setting					Global setting			
Method	Iters	Train Acc.	Test Acc.	I-JS	G-JS	Iters	Train Acc.	Test Acc.	G-JS
softMV	9.2	83.32	81.69	0.216	0.024		⋄	⋄	⋄
hardMV	11.8	80.34	79.95	0.225	0.035		⋄	⋄	⋄
DL-DS	10.6	84.30	83.57	0.153	0.036	4.1	12.63	14.08	0.473
DL-EM	3.9	85.18	83.07	0.295	0.259	3.0	78.02	75.92	0.467
CMM	7.2	84.58	83.10	0.234	0.054		⋄	⋄	⋄
Ideal	8	97.90	92.09				⋄	⋄	

Training and Evaluation Details. All the methods are trained until convergence (change in loss or parameters below a threshold) up to a maximum of 50 iterations. To obtain more significant results, we perform 20 runs of each experiment and average the results. The initialization of the EM algorithm is done with *softMV* and, for our model, a K-means clustering is previously done over annotations. Multiple restarts (20) was applied for *DL-EM* and our method. In the M step, the neural nets are executed one epoch using the Adam optimizer. To implement the predictive model $f(\boldsymbol{z}; \theta)$, we choose an architecture that, according to previous works, is known to be appropriate for each dataset. As, for all the datasets, the ground-truth is available, we evaluate the methods measuring the **Accuracy** of the predictive model on the test set. To evaluate the ability of the method to estimate the confusion matrices, on the train set, we compute the *Jensen-Shannon* divergence in two variants. We measure the **I-JS**, the average divergence between the real and the predicted matrices of each annotator, as well as **G-JS**, the divergence between the real and predicted global matrices, that represent the behavior of all the annotators/annotations in the labelling process. On real dataset, the M chosen is the one with the highest log-likelihood.

Table 4. Metrics on confusion matrices found on LabelMe dataset.

Group	α_m	I_{sim}	$\mathbb{H}$
1	0.99	0.91	0.48
2	0.01	0.02	2.08
3	0.00	0.03	2.08

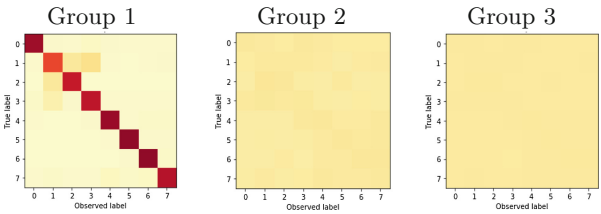


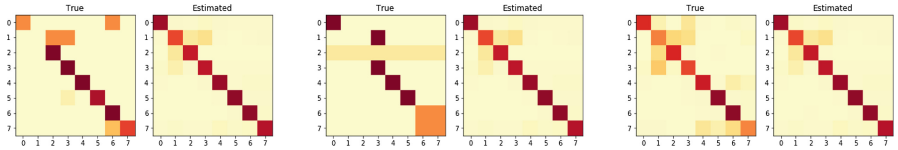
Fig. 1. Confusion matrices found on LabelMe dataset.

Results on the Simulated Data. Table 2 shows the accuracy obtained by the different methods in the simulated scenario, as we vary the number of annotators T. Consistent with previous results [14], we observe that learning-based methods, can significantly improve on simple aggregation techniques such as *MV*. It can also be seen that, as T grows and thus the number of labels per annotator ($D_t \propto \bar{T}_i/T$) decreases, the methods *DL-EM* and *DL-DS* suffer a sharp fall in performance. In contrast, in both setups, the accuracy of our method is more robust to a change in the density of annotations. We attribute this result to the fact that *DL-EM* and *DL-DS* need to estimate a separate sub-model for each annotator (confusion matrix) and thus require that D_t keeps high in order to maintain their accuracy. In contrast, the number of estimated components in our method is independent of T. When the number of annotators is small, our method is competitive, but it is outperformed by more complex models. However, when T is greater than some threshold, in this case 3500, our method achieves the best performance. In some extreme cases, existing learning-based methods cannot be executed due to the large number of parameters in the formulation.

Results on Real Data. We report the results of the LabelMe dataset (using $M = 3$) in Table 3. We experiment with the **Individual** and **Global** settings introduced in Sect. 2. In the first scenario, we know which annotators provided which labels, thus having a quite dense setting. In the second scenario, we do not have that information and thus the annotations are treated independently, leading to a density of $D_t = 1$ (where T grows to 2547 and $\bar{T}_i$ keeps). In the denser case, all the learning-based methods achieve a similar test accuracy (∼83%). In the sparse setting however, the accuracy of *DL-EM* and *DL-DS* suffers an important decrease (∼78% and ∼13% respectively), while the accuracy of our method and *MV* is robust to this change. This shows the disadvantage of methods that model each annotator separately compared to methods based on a **Global** representation that can group annotations together.

Groups Analysis. We visualize in Fig. 1 the confusion matrices found by our method in the LabelMe dataset. We also show in Table 4 the entropy of the confusion matrices $\mathbb{H}$, their similarity I_{sim} with respect to the identity matrix (computed as 1 minus the normalized JS divergence, to obtain a number in $[0, 1]$) and the value of the mixing coefficients α_m (prevalence of each group). We conclude that the method found a group of annotators with a quite expert behavior (high I_{sim}, low $\mathbb{H}$) with a presence of 99%, and a group of spammers

(quite high entropy) with a prevalence of 1%. The third component has an insignificant presence in the mixture ($\alpha_m = 0$ rounding at two decimals) which shows that the method can easily adapt if the number of real groups in the data is lower than those specified into the model. Figure 2 presents visual examples of good and bad predictions of the confusion matrix corresponding to individual annotators (see formulae in Sect. 3.4) and the global confusion matrix.

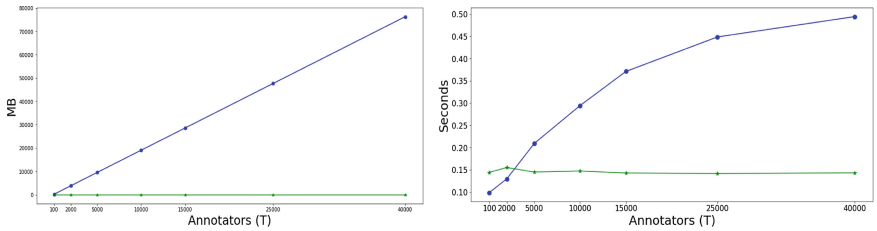


a) Individual: I-JS = 0.150, $p(\boldsymbol{g}|a) = (1; 0; 0)$.

b) Individual: I-JS = 0.232, $p(\boldsymbol{g}|a) = (1; 0; 0)$.

c) Global: G-JS = 0.047, $p(\boldsymbol{g}) = (0.99; 0.01; 0.00)$.

Fig. 2. Examples of confusion matrices (True vs Estimated) on LabelMe dataset.



Memory consumption

Execution time per iteration

Fig. 3. Comparison by increasing T on simulated data setup (1). *DL-EM* is presented in blue and our (*CMM*) is presented in green. (Color figure online)

Computational Efficiency. In Fig. 3, we compare the execution time and memory consumption of *CMM* and *DL-EM* in the simulated setup (1) scenario. In contrast to *CMM*, the computational complexity of *DL-EM* increases monotonically with the value of T. This shows that our method can scale better to scenarios with a large number of annotators as expected from its formulation.

6 Conclusions

We presented a model for learning from crowds that, in contrast to existing methods, does not represent annotators separately but has a fixed number of components into which annotations can be grouped together. Our experiments show that this model achieves competitive accuracy in scenarios with several labels per annotator, but can outperform the baselines when the distribution of labels is sparse. The method is more scalable than other approaches in cases with large number of annotators, also adapts naturally when the amount cannot be determined because the individual annotations are not present. In future work, we plant to extend our method in order to avoid the use of the EM algorithm.

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 Trans. Med. Imaging **35**(5), 1313–1321 (2016)
2. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. JSTOR: Ser. C (Appl. Stat.) **28**(1), 20–28 (1979)
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. JSTOR: Ser. B **39**(1), 1–22 (1977)
4. Kajino, H., Tsuboi, Y., Kashima, H.: A convex formulation for learning from crowds. Trans. Jpn. Soc. Artif. Intell. **27**, 133–142 (2012)
5. Patrini, G., Rozza, A., Menon, A.K., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of IEEE Conference Computer Vision Pattern Recognition (CVPR), pp. 2233–2241 (2017)
6. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. JMLR **13**(Feb), 491–518 (2012)
7. Raykar, V.C., et al.: Learning from crowds. J. Mach. Learn. Res. **11**, 1297–1322 (2010)
8. Rodrigues, F., Pereira, F., Ribeiro, B.: Learning from multiple annotators: distinguishing good from random labelers. Pattern Recogn. Lett. **34**, 1428–1436 (2013)
9. Rodrigues, F., Pereira, F.C.: Deep learning from crowds. In: The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI) (2018)
10. 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 Conference on Empirical Methods in Natural Language Processing, pp. 254–263 (2008)
11. Yan, Y., et al.: Modeling annotator expertise: learning when everybody knows a bit of something. In: Proceedings of the XXX AISTATS, pp. 932–939 (2010)
12. Zhang, J., Wu, X., Sheng, V.S.: Imbalanced multiple noisy labeling. IEEE Trans. Knowl. Data Eng. **27**(2), 489–503 (2015)
13. Zhang, Y., Chen, X., Zhou, D., Jordan, M.I.: Spectral methods meet EM: a provably optimal algorithm for crowdsourcing. JMLR **17**(1), 3537–3580 (2016)
14. Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R.: Truth inference in crowdsourcing: is the problem solved? Proc. VLDB Endow. **10**(5), 541–552 (2017)