Crowd Learning with Candidate Labeling: an EM-based Solution

Iker Beñaran-Muñoz[1], Jerónimo Hernández-González[2], and Aritz Pérez[1]

[1] Basque Center for Applied Mathematics, Al. Mazarredo 14, Bilbao, Spain
Emails: {ibenaran, aperez}@bcamath.org
[2] University of the Basque Country UPV/EHU, P. Manuel de Lardizabal 1, Donostia, Spain. Email: jeronimo.hernandez@ehu.eus

Abstract. Crowdsourcing is widely used nowadays in machine learning for data labeling. Although in the traditional case annotators are asked to provide a single label for each instance, novel approaches allow annotators, in case of doubt, to choose a subset of labels as a way to extract more information from them. In both the traditional and these novel approaches, the reliability of the labelers can be modeled based on the collections of labels that they provide. In this paper, we propose an Expectation-Maximization-based method for crowdsourced data with candidate sets. Iteratively the likelihood of the parameters that model the reliability of the labelers is maximized, while the ground truth is estimated. The experimental results suggest that the proposed method performs better than the baseline aggregation schemes in terms of estimated accuracy.

Keywords: Supervised classification, Crowdsourced labels, Weak supervision, Candidate labeling, Expectation-Maximization based method

1 Introduction

Nowadays, due to the ever-increasing use of the Internet, there is a huge amount of data available. Among the machine learning community, the use of crowdsourcing has become popular as a means of gathering labels at a relatively low cost. In the crowdsourcing context, **crowd labeling** is the process of getting noisy labels for the instances in the training set from a set of various non-expert **annotators (or labelers)** A. In this sense, an annotator $a \in A$ can be seen as a classifier which provides labels with a certain amount of noise. In the traditional crowdsourcing scenario, referred to as **full labeling** throughout this paper, every annotator is asked to select a single label for each instance.

Crowd learning consists of learning a classifier from a dataset with crowdsourced labels. A straightforward approach would separate this learning task into two stages: (i) label aggregation (to determine the ground truth label of each instance of the training set) and (ii) learning (to learn a model using the aggregated labels and standard supervised classification techniques). In this paper, we assume this approach for crowd learning and focus on its first stage.

Probably the most popular aggregation technique is majority voting (MV), which labels every instance with the label that most annotators have selected. In weighted voting [11], the vote of each annotator is weighted according to their reliability. Many aggregation methods that also model the reliability of annotators were derived from the expectation-maximization (EM) strategy [6], which was first applied to crowdsourcing by Dawid and Skene [5], and has been widely used since then [14, 17, 4, 18, 12, 20]. An extensive review of different label aggregation and crowd learning techniques can be found in [19].

In crowd learning, annotators are usually assumed to be non-experts. In this scenario, it may seem reasonable to consider a more relaxed request than to force them to provide a single label. Some approaches are more flexible with labelers by allowing annotators to (i) express how sure they are about the provided labels [9, 15], or (ii) state that they do not know the answer [7, 16, 21]. Recently, **candidate labeling** [2], inspired by weak supervision [10], allows annotators to select a subset of labels, called the **candidate set**, instead of just one. It has been shown that this type of labeling can extract more information from labelers than full labeling, especially with few annotators or difficult instances. It could also lead to faster and/or less costly labeling [1].

In social sciences, a similar problem has been extensively studied under the name of **approval voting** (AV) [3, 8, 13]. Without ground truth, the objective is to identify popular (approved) options. When a single option needs to be selected, aggregation is usually carried out as follows (using machine learning terminology): Given an instance x, the label included in most candidate sets is chosen. This approach is used as a baseline for comparison in this paper.

In this paper, an EM-based technique is proposed to aggregate the candidate labels of crowdsourcedly annotated examples. Firstly, the candidate labeling framework is set. In Section 3, annotator modeling is explained, and maximum likelihood estimates for the parameters are provided. Next, the proposal is presented and its performance is discussed in Section 5 based on experiments with synthesized data. Finally, conclusions are drawn and future work is discussed.

2 Candidate labeling

This work deals with multi-class classification problems, where each instance of the training set belongs to one of $r > 2$ possible classes. Two types of random variables are considered: (i) the features X, that take values in the space Ω_X, and (ii) the class variable C, which takes r distinct values in the space $\Omega_C = \{1, \ldots, r\}$. We assume that the multidimensional random variable (X, C) is distributed according to an (unknown) probability distribution $p(X, C)$. In this work it is assumed that there is a single **ground truth** label for each unlabeled instance x, denoted by c_x. A crowd learning problem with candidate labeling [2] is also considered. In this scenario, for each instance x in the training set, each annotator $a \in A$ provides a candidate set, denoted by $L_x^a \subseteq \Omega_C$.

We focus on estimating the ground truth for an unlabeled dataset $\mathcal{D}$ given the candidate sets provided by the labelers. The goal is to maximize the **estimated**

accuracy (or, for the sake of brevity, accuracy) function:

$$acc(\phi) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{1}(c_x = \phi(x)) \tag{1}$$

where ϕ is a classifier, i.e., a function that maps Ω_X into Ω_C, and $\mathbb{1}(b)$ is a function which returns 1 if the condition b is *true* and 0 otherwise.

For estimating the ground truth using data with candidate sets, the candidate voting (CV) function [2] is defined, given an instance x and the set of candidate sets $\mathcal{L}_x = \{L_x^a\}_{a \in A}$ gathered for it, as follows,

$$\omega(\mathcal{L}_x) = \arg\max_c w_x(c), \tag{2}$$

where

$$w_x(c) = \frac{1}{|A|} \sum_{a \in A} \frac{\mathbb{1}(c \in L_x^a)}{|L_x^a|} \tag{3}$$

is the **candidate voting estimate**. CV can be understood as a weighted voting function where the weights depend on the sizes of the provided candidate sets. It works as a generalization of the MV strategy from the full labeling to the candidate labeling context. It can be easily observed that CV behaves as MV when annotations are obtained by means of full labeling.

As the trustworthiness of the labelers is not homogeneous, having information about their reliability can be of great advantage to aggregate the labels that they provide. In the next section, a model of the annotators based on their reliability is proposed and the maximum likelihood estimates of its parameters are obtained.

3 Modeling annotators and maximum likelihood estimate

In order to aggregate the candidate sets gathered through candidate labeling, the contribution of each labeler can be weighted according to their reliability. In this section, a model for the behavior of annotators is described, with parameters that control their reliability and the way the candidate sets are generated. Then, the maximum likelihood estimates are inferred for those parameters. Finally, a procedure for estimating the labels is presented.

In the presented framework, the candidate set L_x^a is assumed to be generated by asking annotator a one question of the kind "Do you consider that the given instance x might belong to class c?" for each $c \in \Omega_C$. Let α_c^a denote the probability that annotator a includes label c in the candidate set for instances which really belong to class c. Let us also define β_c^a as the probability that annotator a includes any label $c' \neq c$ ($c' \in \Omega_C$) in the candidate set when annotating instances which really belong to class c. Note that we assume that, given an instance of a certain class, the rest of class labels have the same probability of being mistakenly selected. The parameters α_c^a and β_c^a provide us insights into the behavior of annotator a when labeling instances that really belong to class c.

Assuming the process of generation of the candidate sets described above, the likelihood of the parameters given a candidate set L_x^a is:

$$\begin{aligned} Pr(L_x^a|\boldsymbol{\alpha},\boldsymbol{\beta},c_x) = (\alpha_{c_x}^a)^{\mathbb{1}(c_x\in L_x^a)} \cdot (1-\alpha_{c_x}^a)^{1-\mathbb{1}(c_x\in L_x^a)} \\ \cdot (\beta_{c_x}^a)^{|L_x^a|-\mathbb{1}(c_x\in L_x^a)} \cdot (1-\beta_{c_x}^a)^{r-|L_x^a|+\mathbb{1}(c_x\in L_x^a)}, \end{aligned} \tag{4}$$

where the set of probabilities for annotators of selecting the (unknown) correct label c_x is $\boldsymbol{\alpha} = \{\alpha_c^a\}_{a\in A, c\in\Omega_C}$ and the set of probabilities for annotators of selecting each incorrect label is $\boldsymbol{\beta} = \{\beta_c^a\}_{a\in A, c\in\Omega_C}$.

Assuming that annotators provide the candidate sets independently and that all instances are i.i.d. according to $p(X, C)$, the likelihood given a dataset $\mathcal{D}$ where each instance is annotated with a set of candidate sets is:

$$Pr(\{\mathcal{L}_x\}_{x\in\mathcal{D}}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{x\in\mathcal{D}}\prod_{a\in A} Pr(L_x^a|\boldsymbol{\alpha},\boldsymbol{\beta},c_x) \tag{5}$$

From this expression, the maximum likelihood estimates of both alpha and the beta parameters are:

$$\hat{\alpha}_c^a = \frac{\sum_{x\in\mathcal{D}} \mathbb{1}(c_x = c)\mathbb{1}(c\in L_x^a)}{\sum_{x\in\mathcal{D}} \mathbb{1}(c_x = c)} \tag{6}$$

$$\hat{\beta}_c^a = \frac{\sum_{x\in\mathcal{D}} \mathbb{1}(c_x = c)\cdot(|L_x^a| - \mathbb{1}(c\in L_x^a))}{r\cdot\sum_{x\in\mathcal{D}} \mathbb{1}(c_x = c)} \tag{7}$$

The estimate $\hat{\alpha}_c^a$, given by maximum likelihood, is the number of instances of class c for which annotator a included class label c in the candidate set over the total number of instances of class c. On the other hand, the estimate $\hat{\beta}_c^a$, given by maximum likelihood, is the number of mistaken class labels that annotator a included in the candidate sets of all the instances of class c over the whole set of possible class labels for the total number of instances of class c.

The estimates in Equations (6) and (7) can be computed when the true class labels are known for all instances. Conversely, if the true labels are not known, they can be estimated by means of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters. Using Bayes' Theorem, it follows that:

$$Pr(c|\mathcal{L}_x,\boldsymbol{\alpha},\boldsymbol{\beta}) \propto Pr(c)\cdot Pr(\mathcal{L}_x|\boldsymbol{\alpha},\boldsymbol{\beta},c) \tag{8}$$

Using Eq. (4) for the case that $c_x = c$ and estimating the marginal probability as $Pr(c) = \frac{\sum_{x\in\mathcal{D}}\mathbb{1}(c_x=c)}{|\mathcal{D}|}$, Equation (8) can be rewritten as:

$$\begin{aligned} Pr(c|\mathcal{L}_x,\boldsymbol{\alpha},\boldsymbol{\beta}) \propto \frac{\sum_{x\in\mathcal{D}}\mathbb{1}(c_x=c)}{|\mathcal{D}|} \cdot \prod_{a\in A}\Big((\alpha_c^a)^{\mathbb{1}(c\in L_x^a)}\cdot(1-\alpha_c^a)^{1-\mathbb{1}(c\in L_x^a)} \\ \cdot(\beta_c^a)^{|L|-\mathbb{1}(c\in L_x^a)}\cdot(1-\beta_c^a)^{r-|L|+\mathbb{1}(c\in L_x^a)}\Big) \end{aligned} \tag{9}$$

In this way, the probability that a given instance x belongs to each possible class label can be computed by means of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and the candidate sets. This probability distribution could be considered as an estimate for the ground truth. In practice, neither the true labels nor the values of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are known. A method based on the EM strategy [6] that estimates all of them jointly is proposed.

4 EM-based method for candidate labeling aggregation

The EM strategy attempts to gather maximum likelihood estimates when there is missing data. Two steps are iterated: (i) Expectation (E-step), where the expected values of the missing data are computed using the current parameter estimates and (ii) Maximization (M-step), where the parameters are updated with the new maximum likelihood estimates given the current expected data. This method is guaranteed to converge to a local maximum.

In the crowdsourcing context, the true class labels of the training instances are the missing data. Our method is based on the Dawid-Skene approach [5], which is implemented as follows: First, an initial estimate of the ground truth labels is obtained. After that, the method consists of two steps: (i) **M-step**: The parameters that model the reliability of the annotators are updated with estimates that maximize or, at least, improve the likelihood achieved in the previous E-step; and (ii) **E-step**: Given an estimate of the parameters, the expected values of the ground truth labels are obtained for every instance, given the expected labels. The M and E steps are carried out iteratively until convergence.

Our proposal is an adaptation of this strategy to the candidate labeling scenario. Firstly, let us define $q(c|x)$ as the estimate of the probability $Pr(c|\mathcal{L}_x, \boldsymbol{\alpha}, \boldsymbol{\beta})$ described in Eq. (8), that is, the probability that x belongs to class c. In equations (4), (6) and (7), the $q(c|x)$ estimates can substitute the expression $\mathbb{1}(c_x = c)$, switching from two discrete values (0 or 1) to any possible value in the continuous interval $[0, 1]$. Note that the true label c_x is unknown and this modification allows this approach to work with the probabilistic estimates of the ground truth. Our method works in the following way:

After a first step where the estimates $q(c|x)$ are initialized for all $x \in \mathcal{D}$ and $c \in \Omega_C$, the M and E steps of the proposed method are as follows:

- **M-step.** For every $a \in A$ and $c \in \Omega_C$, the estimates $\hat{\alpha}_c^a$ and $\hat{\beta}_c^a$ are computed given q by means of Equations (6) and (7), using the estimates $q(c|x)$ instead of $\mathbb{1}(c_x = c)$.
- **E-step.** For every $x \in D$ and $c \in \Omega_C$, Equation (9) is used to compute the probability distributions $q(c|x)$ given $\hat{\alpha}_c^a$ and $\hat{\beta}_c^a$. As in the E-step, the terms $\mathbb{1}(c_x = c)$ are substituted by the previous estimates $q(c|x)$.

In the next section, the performance of the previously described method is tested using artificial data.

5 Experiments

In this section, the performance of the presented method is evaluated in different scenarios. In order to have insights into its performance: (i) the accuracy is computed for different scenarios, varying the numbers of annotators, classes, and instances, and the values of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters, (ii) the method is compared with candidate voting [2], approval voting [3] and the **privileged aggregation** (where all $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters are known), and (iii) the evolution is observed through each iteration of the method.

5.1 Experimental setting

To the best of our knowledge, there is not any publicly available dataset for crowd learning with candidate labeling. Thus, artificial data has been used as a means of obtaining experimental results. Simulated data is also useful to control the settings and explore different scenarios.

In order to generate different situations, the following experimental parameters are set to different values: number of instances (n), number of annotators (m), number of classes (r), minimum and maximum values of the $\boldsymbol{\alpha}$ parameters ($\underline{\alpha}$ and $\overline{\alpha}$) and minimum and maximum $\boldsymbol{\beta}$ parameters ($\underline{\beta}$ and $\overline{\beta}$). The parameters $\underline{\alpha}$ and $\overline{\beta}$ have both been fixed to 0.5, so that there always can be annotators of minimum expertise and adversarial annotators are not generated.

The method itself has two additional parameters:

- The convergence threshold δ. If $\frac{|\overline{\boldsymbol{\alpha}}_{(it)}-\overline{\boldsymbol{\alpha}}_{(it-1)}|}{\overline{\boldsymbol{\alpha}}_{(it-1)}} < \boldsymbol{\delta}$ or $\frac{|\overline{\boldsymbol{\beta}}_{(it)}-\overline{\boldsymbol{\beta}}_{(it-1)}|}{\overline{\boldsymbol{\beta}}_{(it-1)}} < \boldsymbol{\delta}$, where $\overline{\boldsymbol{\alpha}}_{(it)}$ ($\overline{\boldsymbol{\beta}}_{(it)}$) is the mean value of $\boldsymbol{\alpha}_{(it)}$ ($\boldsymbol{\beta}_{(it)}$) at iteration it, it is considered that the EM has converged. It has been set to $\delta = 0.05$.
- The smoothing parameter γ. There are two factors that lead to undesirable results, such as the likelihood equal to 0: (i) There is a large number of parameters to be estimated ($2 \cdot m \cdot r$) and there is not always sufficient information, and (ii) sometimes, the parameter estimates can get close to 0 or to 1, leading to error. An additive smoothing is used for the $\hat{\alpha}_c^a$ estimates:

$$\hat{\alpha_c^a} = \frac{\gamma + \sum_{x \in \mathcal{D}} \mathbb{1}(c_x = c)\mathbb{1}(c \in L_x^a)}{2 \cdot \gamma + \sum_{x \in \mathcal{D}} \mathbb{1}(c_x = c)} \tag{10}$$

  In this way, all possible values are reached at least once, that is, there is at least one instance of class c such that $c \in L_x^a$ and another instance of class c such that $c \in L_x^a$. In these experiments, Equation (10) is used instead of Equation (6) with $\gamma = 1$.

Datasets are simulated as follows: The ground truth class labels are distributed uniformly among all instances, that is, there are $\frac{n}{r}$ instances belonging to each class. Next, the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters are generated. In order to have annotators with different types of knowledge, a maximum ($\overline{\alpha}$) value of $\boldsymbol{\alpha}$ and a minimum value for $\boldsymbol{\beta}$ ($\underline{\beta}$) are set. All the parameters are sampled uniformly from the intervals $[0.5, \overline{\alpha}]$ and $[\underline{\beta}, 0.5]$. By means of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters, candidate sets are generated following the interpretation explained at the beginning of Section 3. That is, given an instance that belongs to class c, annotator a includes class c in the candidate set with probability α_c^a and each of the classes $c' \neq c$ with probability β_c^a.

Once the candidate sets are generated, 4 different schemes are used to aggregate them: (i) our EM-based method, (ii) CV (Eq. (2)), (iii) AV and (iv) privileged aggregation (PA). The PA is obtained by computing the estimate from Eq. (9), using the original parameters and the ground truth class labels.

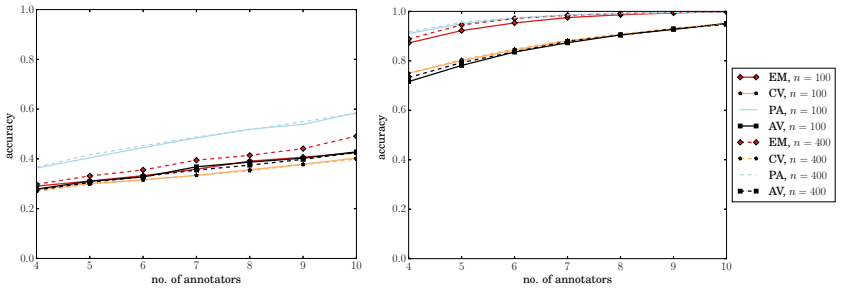
As mentioned above, EM is ensured to converge to a local maximum, so various initializations should be carried out to achieve desirable results. In order to obtain different initializations from the same candidate sets, we initialize the estimates $q(c|x)$ for each instance x in the following way: First, the candidate voting estimates $w_x(c)$ (Eq. (3)) are computed for all $c \in \Omega_C$, using an additive smoothing of $\frac{1}{r}$ for each one. The $q(\cdot|x)$ are normalized so that $0 \leq q(c|x) \leq 1$ and $\sum_{c\in\Omega_C} q(c|x) = 1$. Next, to initialize $q(\cdot|x)$, a Dirichlet distribution with hyper-parameters $r \cdot w_x(c_1), \ldots, r \cdot w_x(c_r)$ is sampled: $q(\cdot|x) \sim Dir(r \cdot w_x(c_1), \ldots, r \cdot w_x(c_r))$.

30 initializations are carried out and the values of the final $q(c|x)$ estimates that maximize the likelihood are used to infer the labels: each instance x takes the class label c that maximizes $q(c|x)$. The process is repeated 30 times and the expected accuracy is approximated by computing the mean of the obtained accuracy estimates.

5.2 Experimental results

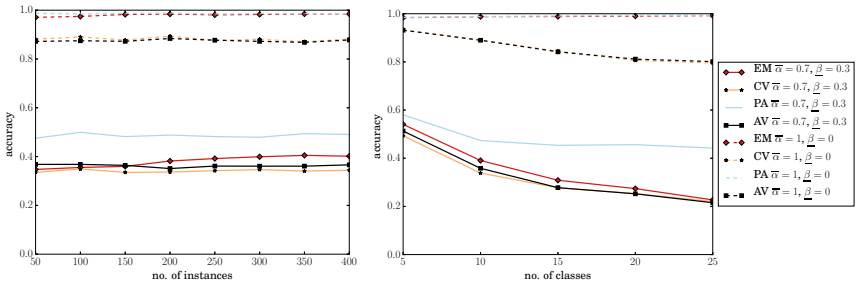
Experiments with artificial data have been performed, varying a number of parameters to compare our method and previous approaches in different scenarios.

Except for the graphics where their evolution is examined, standard values have been chosen for the parameters. The number of annotators varies from 4 to 10, although it is fixed to its standard value ($m = 7$) in different experiments. The numbers of instances used are $n = \{100, 400\}$. In the case $n = 100$, $r = \{5, 10\}$ class labels are considered, and in the case $n = 400$, $r = \{10, 20\}$ class labels are considered. Regarding the expertise of annotators, two scenarios have been studied: (i) $\underline{\beta} = 0.3$ and $\overline{\alpha} = 0.7$, where the average expertise is low, and (ii) $\underline{\beta} = 0$ and $\overline{\alpha} = 1$, where the expertise of the annotators ranges from minimum to maximum values. Due to space limitations, only the results of a representative subset of experiments are shown in this paper.



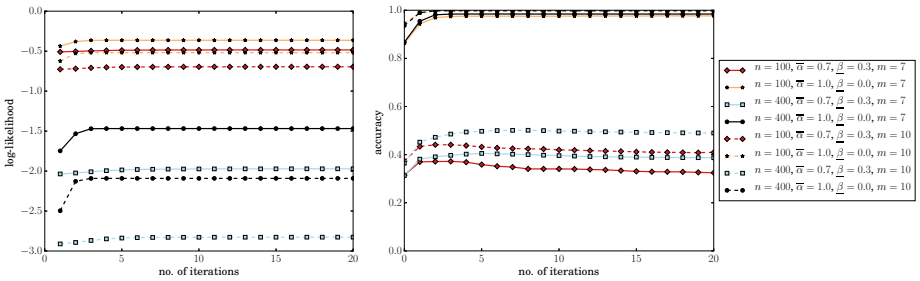
(a) $r = 10,\ \overline{\alpha} = 0.7,\ \underline{\beta} = 0.3$ (b) $r = 10,\ \overline{\alpha} = 1,\ \underline{\beta} = 0$

Fig. 1. Graphical description of the accuracy obtained by annotations simulated with different numbers of annotators.



(a) $m = 7,\ r = 10$ (b) $m = 7,\ n = 400$

Fig. 2. Graphical description of the accuracy obtained by annotations simulated with different numbers of instances and classes.



(a) Log-likelihood (scale: 10^4) (b) Accuracy

Fig. 3. Graphical description of the log-likelihood and the accuracy obtained throughout different iterations, with $r = 10$.

In Figures 1 and 2, the accuracy of the presented method (EM) is compared to that of the CV, the AV and the PA, in scenarios where the number of annotators (m, Fig 1), the number of instances (n, Fig 2a) and the number of classes (r, Fig 2b) are varied. The experimental results suggest that, in general, EM outperforms CV and AV in terms of the accuracy (Eq. (1)). The accuracies are similar only in the case where the average expertise is low and the number of classes is high with respect to the number of instances (see Figure 2 with $\underline{\beta} = 0.3$ and $\overline{\alpha} = 0.7$). Moreover, in the case that $\underline{\beta} = 0$ and $\overline{\alpha} = 1$ (Fig 1b), the proposed method reaches the accuracy of the PA. In other words, in the presence of annotators that are experts in a subset of classes, our EM-based strategy can reach the highest possible accuracy. Note as well that the accuracy of the EM approach decreases at a smoother pace than that of CV or AV as the number of annotators is reduced.

As can be seen in Figure 2a, the number of instances (n) does not seem to affect the differences between the accuracies of the different methods, when it ranges between 100 and 400 (Fig 2a). On the other hand, the number of classes

(r) has a negative effect on the accuracy of all the methods (Fig 2b). The only exception is that when the expertise of the annotators ranges from minimum to maximum values (Fig 2b, $\overline{\alpha} = 1$, $\underline{\beta} = 0$), our EM approach outperforms the baselines.

The evolution of the log-likelihood and the accuracy in each iteration of the EM can be seen in Figure 3. In Figure 3b, the accuracy in iteration number 0 is the one reached using the initial q estimates. As could be expected, generally, the log-likelihood increases monotonically and remains stable after some point (Fig 3a). The accuracy increases in the first iterations as well, and then remains stable in most cases (Fig 3b), but decreases in one case ($n = 100$, $\overline{\alpha} = 0.7$, $\underline{\beta} = 0.3$). This decline may be due to overfitting, since scarce data (each annotator labels 100 instances) is used to estimate many parameters (20 per annotator).

To sum up, according to the experiments, EM seems to outperform CV and AV in most scenarios, especially when the expertise of the annotators is varied. In favorable settings, EM can reach a high accuracy - as if the real $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters were known (PA).

6 Conclusions and future work

In this work, a crowd learning problem is approached with candidate labeling. A model for the reliability of the annotators is proposed. An EM-based method is presented, as an extension to the traditional methods for aggregating crowdsourced labels into the candidate labeling scenario. Experimental results obtained with artificial data suggest that the presented method has an enhanced performance, in terms of estimated accuracy, compared with the baseline methods. Particularly, it stands out when few annotators are available and when they show different levels of expertise.

For future work, more realistic data could be used, not following the assumption that, given an instance of a certain class, the rest of labels can be selected with the same probability. Also, a real-world dataset with candidate sets could be gathered in order to test the presented method, as well as other aggregation and learning schemes. The presented method could also be refined by reducing the number of parameters or, similar to [14], learning a classification model from data as the crowd-modeling parameters are estimated.

Acknowledgments

IBM and AP are both supported by the Spanish Ministry MINECO through BCAM Severo Ochoa excellence accreditation SEV-2013-0323 and the project TIN2017-82626-R funded by (AEI/FEDER, UE). IBM is also supported by the grant BES-2016-078095. AP is also supported by the Basque Government through the BERC 2014-2017 and the ELKARTEK programs, and by the MINECO through BCAM Severo Ochoa excellence accreditation SVP-2014-068574. JHG is supported by the Basque Government (IT609-13, Elkartek BID3A) and the MINECO (TIN2016-78365-R).

References

1. A. Banerjee, S.O., Gurari, D.: Let's agree to disagree: A meta-analysis of disagreement among crowdworkers during visual question answering. In: GroupSight Workshop at AAAI HCOMP. Quebec City, Canada (2017)
2. Beñaran-Muñoz, I., Hernández-González, J., Pérez, A.: Weak Labeling for Crowd Learning. ArXiv e-prints (2018)
3. Brams, S.J., Fishburn, P.C.: Approval voting. Am. Polit. Sci. Rev. 72(3), 831–847 (1978)
4. Côme, E., Oukhellou, L., Denoeux, T., Aknin, P.: Learning from partially supervised data using mixture models and belief functions. Pattern recognition 42(3), 334–348 (2009)
5. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. J. Royal Stat. Soc. Series C 28(1), 20–28 (1979)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. Series B 39(1), 1–38 (1977)
7. Ding, Y.X., Zhou, Z.H.: Crowdsourcing with unsure option. Mach. Learn. 107(4), 749–766 (2018)
8. Falmagne, J.C., Regenwetter, M.: A random utility model for approval voting. J. Math. Psychol. 40(2), 152–159 (1996)
9. Grady, C., Lease, M.: Crowdsourcing document relevance assessment with mechanical turk. In: NAACL HLT 2010 workshop. pp. 172–179 (2010)
10. Hernández-González, J., Inza, I., Lozano, J.A.: Weak supervision and other non-standard classification problems: A taxonomy. Pattern Rec. Lett. 69, 49–55 (2016)
11. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: NIPS. pp. 1953–1961 (2011)
12. López-Cruz, P.L., Bielza, C., Larrañaga, P.: Learning conditional linear gaussian classifiers with probabilistic class labels. In: Conference of the Spanish Association for Artificial Intelligence. pp. 139–148. Springer (2013)
13. Procaccia, A.D., Shah, N.: Is approval voting optimal given approval votes? In: NIPS. pp. 1801–1809 (2015)
14. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. J Mach Learn Res 11, 1297–1322 (2010)
15. Smyth, P., Fayyad, U.M., Burl, M.C., Perona, P., Baldi, P.: Inferring ground truth from subjective labelling of venus images. In: Proc of NIPS 7. pp. 1085–1092 (1994)
16. Venanzi, M., Guiver, J., Kohli, P., Jennings, N.R.: Time-sensitive bayesian information aggregation for crowdsourcing systems. J. Artif. Intell. Res. 56, 517–545 (2016)
17. Welinder, P., Branson, S., Belongie, S., Perona, P.: The multidimensional wisdom of crowds. In: Proc of NIPS 23. pp. 2424–2432 (2010)
18. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.R.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Proc. of NIPS 22. pp. 2035–2043 (2009)
19. Zhang, J., Sheng, V.S., Wu, J., Wu, X.: Multi-class ground truth inference in crowdsourcing with clustering. IEEE Trans. Knowl. Data Eng. 28(4), 1080–1085 (2016)
20. Zhang, Y., Chen, X., Zhou, D., Jordan, M.I.: Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In: Advances in neural information processing systems. pp. 1260–1268 (2014)
21. Zhong, J., Tang, K., Zhou, Z.H.: Active learning from crowds with unsure option. In: Proc. of 24th IJCAI. pp. 1061–1068 (2015)