Robust Clustering of Data Collected via Crowdsourcing

Alba Pagès-Zamora*, Georgios B. Giannakis†, Roberto López-Valcarce‡ and Pere Giménez-Febrer*
*SPCOM Group, Universitat Politècnica de Catalunya-Barcelona Tech, Spain
†Dept. of ECE and DTC, University of Minnesota, Minneapolis, USA
‡GPSC, Universidade de Vigo, Spain

***Abstract*—Crowdsourcing approaches rely on the collection of multiple individuals to solve problems that require analysis of large data sets in a timely accurate manner. The inexperience of participants or *annotators* motivates well robust techniques. Focusing on clustering setups, the data provided by all annotators is suitably modeled here as a mixture of Gaussian components plus a uniformly distributed random variable to capture outliers. The proposed algorithm is based on the expectation-maximization algorithm and allows for soft assignments of data to clusters, to rate annotators according to their performance, and to estimate the number of Gaussian components in the non-Gaussian/Gaussian mixture model, in a jointly manner.**

***Index Terms*—Crowdsourcing, Gaussian plus non-Gaussian Mixture, Outlier, EM algorithm, Bayesian Information Criterion**

I. Introduction

PARAMETER estimation of mixture distributions has well-documented merits for unsupervised learning tasks encountered in general-purpose clustering applications for various data mining and machine learning applications including image or speech analysis. Clustering algorithms are particularly relevant to applications using a crowdsourcing methodology[1], which leverages multiple individuals having access to large data sets instead of relying on a single expert. In a considerable number of crowdsourcing applications, annotators are asked to click on specific structures of an image. However, the whole process is severely error-prone since annotators are usually non-experts [1]. For instance, in the *MalariaSpot* project [2] annotators are asked to identify malaria parasites in digitized blood smears through an online game for an early malaria diagnosis, but they often mistake parasites with other cells such as leukocytes, for instance; in the *Microscope Masters* project [3], annotators must pick out proteins in electron microscopy images for biological molecule reconstruction but, instead, they mark smudges or proteins that are clumped together. Other erroneous clicks do not correspond to any particular structure, and are just placed on random parts of the image; see e.g. Fig. 2 in [2].

The standard approach to process the unreliable data collected by crowdsourcing applications consists of two steps. First, the data provided by all annotators are clustered to identify labels. Subsequently, since some of the labels may be erroneously identified, a decision is made on each one whether it corresponds to a desired structure or not [1], [4]. When known, the true labels are referred to as the *gold standard*. It is important to remark that the closer the identified labels are to the gold standard, the lower the probability of false detection in the second step. Crowdsourcing approaches also entail rating annotators according to their performance, so that data provided by unreliable annotators in future experiments can be discarded. Interestingly, data is available in a streaming manner at possibly different locations, which calls for distributed online implementation of the solutions.

This paper focuses on clustering and the associated annotators rating problem. The probability density function (pdf) of the collected data is modeled as a mixture of an *unknown* number of Gaussian components plus a uniformly distributed random variable (rv), which captures outliers. Further, the proposed formulation includes a set of latent rv's to denote the annotators' performance. A closed-form approximate maximum likelihood (ML) estimate of the parameters for Gaussian plus non-Gaussian mixtures was given in [5], where the number of Gaussian components is estimated by choosing among a set of pre-estimated candidate models. Instead, here we opt for an approach based on the expectation-maximization (EM) algorithm [6] that solves the overall estimation problem jointly. As a result, the proposed algorithm will allow for (a) soft assignments of data points to clusters; (b) rating of annotators; and, (c) estimating the number of Gaussian components in the mixture model based on the algorithm developed in [7] for a Gaussian mixture only. Relative to prior works in robust clustering [8]–[12], the present contribution accounts for the variable reliability of data to be clustered, which is a distinct feature of crowdsourcing.

The rest of the paper is organized as follows. Sec. II describes the data probabilistic model, and Sec. III develops the EM-based algorithm. Sec. IV presents simulation results, and Sec. V concludes the paper and comments on future work.

II. Data Model

Consider a set of R annotators indexed by $r \in \{1, \ldots, R\}$, who provide instances of a $D \times 1$ vector[2]. Instances of

This work has been funded in part by the "Ministerio de Economía y Competitividad" of the Spanish Government, ERDF funds (TEC2013-41315-R, TEC2015-69648-REDC, TEC2016-75067-C4-2-R,TEC2013-47020-C2-1-R, TACTICA), the Catalan Government (2014 SGR 60 AGAUR), and the Galician Government (AtlantTIC, GRC2013/009, R2014/037).

[1]A representative sample of crowdsourcing projects can be found in *Zooniverse* platform at https://www.zooniverse.org.

[2]If instances correspond to clicks on an image, then $D = 2$

annotator r are modeled by the $D\times 1$ random vector

$$\boldsymbol{x}_r = a_r \sum_{m=1}^{M} \delta(z_r - m)\boldsymbol{w}_m + (1-a_r)\boldsymbol{u} \tag{1}$$

where $\delta(\cdot)$ denotes the Kronecker delta function; $\boldsymbol{w}_m \sim \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ is the m^{th} D-dimensional Gaussian rv with mean $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$ for $m \in \{1,\ldots,M\}$; and $\boldsymbol{u}$ is a D-dimensional uniformly distributed rv with independent entries and known pdf[3] denoted by $g_{\boldsymbol{U}}(\cdot)$ with support $[\mathcal{U}_d^{\min}, \mathcal{U}_d^{\max}]$ for $d \in \{1,\ldots,D\}$. Variables $\{a_r; \forall r\} \in \{0,1\}$ are independent Bernoulli with probability $p_r \coloneqq \Pr\{a_r = 1\}$, and $\{z_r; \forall r\} \in \{1,\ldots,M\}$ are independent rv's with probability $\Pr\{z_r = m\} \coloneqq \pi_m$. We further assume that all rv's in (1) are independent among them. The model in (1) is a mixture of M Gaussians plus a uniformly distributed rv with a priori probabilities that depend on the annotator. Note that when $a_r = 1$, the instance provided by annotator r corresponds to one out of M Gaussians, given by z_r. Conversely, when $a_r = 0$, the instance of annotator r is a uniformly distributed rv, and it is thus deemed as being an outlier. Therefore, probability p_r is a measure of the annotators' reliability since the lower p_r is, the higher the probability that annotator r provides an outlier.

Suppose further that each annotator r provides $N_r \in \mathbb{N}$ instances denoted by $\{\boldsymbol{x}_{r,i} \in \mathbb{R}^{D\times 1}; i = 1,..,N_r\}$, which are independent identically distributed (iid) realizations of $\boldsymbol{x}_r$ in (1). Let $\mathcal{X} \coloneqq \{\boldsymbol{x}_{r,i}; r = 1,\ldots,R$ and $i = 1,\ldots,N_r\}$ collect the instances provided by all annotators, with cardinality equal to $\mathcal{N} \coloneqq |\mathcal{X}| = \sum_{i=1}^{R} N_r$. Similarly, collect in $\mathcal{A} \coloneqq \{a_{r,i}; \forall r, i\}$ and $\mathcal{Z} \coloneqq \{z_{r,i}; \forall r, i\}$, both with cardinality N, the set of all iid realizations of a_r and z_r, respectively.

Under the aforementioned independence assumptions, the likelihood function of the provided instances $\mathcal{X}$ is

$$\begin{aligned} f(\mathcal{X};\boldsymbol{\theta}) = \prod_{r=1}^{R}\prod_{i=1}^{N_r} \Big(p_r \sum_{m=1}^{M} \pi_m \, \mathcal{N}(\boldsymbol{x}_{r,i}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \\ + (1-p_r) g_{\boldsymbol{U}}(\boldsymbol{x}_{r,i}) \Big) \end{aligned} \tag{2}$$

where $\mathcal{N}(\boldsymbol{x}_{r,i}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ is the likelihood function of instance $\boldsymbol{x}_{r,i}$ given $z_{r,i} = m$, and vector $\boldsymbol{\theta}$ gathers the set of all unknown parameters, namely

$$\boldsymbol{\theta} \coloneqq [\boldsymbol{\mu}_1; \ldots; \boldsymbol{\mu}_M; \mathrm{vec}(\boldsymbol{\Sigma}_1); \ldots; \mathrm{vec}(\boldsymbol{\Sigma}_M); \pi_1; \ldots; \pi_M; p_1; \ldots; p_R]. \tag{3}$$

The objective is not only to cluster data, but also to estimate the M cluster centroids $\{\boldsymbol{\mu}_m; \forall m\}$, the covariance matrices $\{\boldsymbol{\Sigma}_m; \forall m\}$ which are indicative of the cluster spread, the probability of occurrence of each cluster $\{\pi_m; \forall m\}$, and the annotator's reliability $\{p_r; \forall r\}$. Although out of the scope of this work, all these parameters might be useful in crowdsourcing applications to support the decision whether the identified clusters correspond to a desired structure or not. As a closed-form maximization of $f(\mathcal{X};\boldsymbol{\theta})$ is not possible, we resort to a numerical solution based on the EM algorithm.

[3]This is a reasonable assumption for the crowdsourcing applications described in Sec. I.

III. EM for Clustering Crowdsourced Data

The proposed approach is to estimate the unknown parameters in (3) using the iterative EM algorithm. The algorithm is developed first when the number of Gaussian components is known; that is, $M_0 = M$.

A. Number of Gaussian components known

We regard $\mathcal{X}$ as the *incomplete* observation and the set $\{\mathcal{X}, \mathcal{A}, \mathcal{Z}\}$ as the *complete* one. Initialized with $\hat{\boldsymbol{\theta}}^0$, at iteration $t+1$ with $t \geqslant 0$, the EM algorithm proceeds as follows.

S1) *E-step:* given an estimate $\hat{\boldsymbol{\theta}}^t$, compute the conditional expectation of the log-likelihood function

$$Q(\tilde{\boldsymbol{\theta}}; \hat{\boldsymbol{\theta}}^t) \coloneqq \mathbb{E}_{\mathcal{A},\mathcal{Z}}\{\log f(\mathcal{X},\mathcal{A},\mathcal{Z};\tilde{\boldsymbol{\theta}}) \mid \hat{\boldsymbol{\theta}}^t, \mathcal{X}\} \tag{4}$$

where $\tilde{\boldsymbol{\theta}}$ denotes a 'trial' value of $\boldsymbol{\theta}$.

S2) *M-step:* obtain the estimate for the next iteration as

$$\hat{\boldsymbol{\theta}}^{t+1} = \arg\max_{\tilde{\boldsymbol{\theta}}} Q(\tilde{\boldsymbol{\theta}}; \hat{\boldsymbol{\theta}}^t). \tag{5}$$

Recalling that $\mathcal{A}$ and $\mathcal{Z}$ are independent, it holds that (cf. (2))

$$\begin{aligned} Q(\tilde{\boldsymbol{\theta}}; \hat{\boldsymbol{\theta}}^t) = & \sum_{r=1}^{R}\sum_{i=1}^{N_r} \alpha_{r,i}^t \sum_{m=1}^{M_0} \zeta_{r,i,m}^t \log\Big(\tilde{p}_r \tilde{\pi}_m \mathcal{N}(\boldsymbol{x}_{r,i}; \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Sigma}}_m)\Big) \\ & + \sum_{r=1}^{R}\sum_{i=1}^{N_r} (1-\alpha_{r,i}^t) \log\left((1-\tilde{p}_r) g_{\boldsymbol{U}}(\boldsymbol{x}_{r,i})\right) \end{aligned} \tag{6}$$

where $\alpha_{r,i}^t \coloneqq \Pr\{a_{r,i} = 1 | \hat{\boldsymbol{\theta}}^t, \mathcal{X}\}$ and $\zeta_{r,i,m}^t \coloneqq \Pr\{z_{r,i} = m | \hat{\boldsymbol{\theta}}^t, \mathcal{X}\}$ are the posterior probabilities of the hidden variables. Then, using Bayes' theorem, in the *E*-step one basically updates these a posteriori values according to

$$\alpha_{r,i}^t = \frac{\hat{p}_r^t \sum_{m=1}^{M_0} \hat{\pi}_m^t \mathcal{N}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t)}{\hat{p}_r^t \sum_{m=1}^{M_0} \hat{\pi}_m^t \mathcal{N}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t) + (1-\hat{p}_r^t) g_{\boldsymbol{U}}(\boldsymbol{x}_{r,i})} \tag{7}$$

and

$$\zeta_{r,i,m}^t = \frac{\hat{\pi}_m^t \mathcal{N}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t)}{\sum_{m=1}^{M_0} \hat{\pi}_m^t \mathcal{N}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t)}. \tag{8}$$

In the *M*-step, the parameters are updated to maximize (6). Thus, at iteration t, the annotators' reliability is updated as

$$\hat{p}_r^{t+1} = \frac{1}{N_r} \sum_{i=1}^{N_r} \alpha_{r,i}^t, \quad \forall r \quad ; \tag{9}$$

and the probability of the m^{th} Gaussian component becomes

$$\hat{\pi}_m^{t+1} = \frac{\sum_{r=1}^{R}\sum_{i=1}^{N_r} \alpha_{r,i}^t \zeta_{r,i,m}^t}{\sum_{r=1}^{R}\sum_{i=1}^{N_r} \alpha_{r,i}^t} \quad , \tag{10}$$

which must satisfy $\sum_{m=1}^{M_0} \hat{\pi}_m = 1$. Interestingly, the denominator in (10) is a *soft* count of all non-outliers instances and, similarly, the denominator in (11) is a *soft* count of instances that belong to the m^{th} Gaussian component at iteration $t+1$. Further, the mean vectors and covariance matrices of the Gaussian components are given by

$$\hat{\boldsymbol{\mu}}_m^{t+1} = \frac{\sum_{r=1}^{R}\sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t \, \boldsymbol{x}_{r,i}}{\sum_{r=1}^{R}\sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t} \tag{11}$$

and

$$\hat{\boldsymbol{\Sigma}}_m^{t+1} = \frac{\sum_{r=1}^{R}\sum_{i=1}^{N_r}\alpha_{r,i}^t\zeta_{r,i,m}^t(\boldsymbol{x}_{r,i}-\hat{\boldsymbol{\mu}}_m^{t+1})(\boldsymbol{x}_{r,i}-\hat{\boldsymbol{\mu}}_m^{t+1})^H}{\sum_{r=1}^{R}\sum_{i=1}^{N_r}\ \alpha_{r,i}^t\,\zeta_{r,i,m}^t} \tag{12}$$

respectively, $\forall m = \{1,\dots,M_0\}$. As proved in [6], the EM iterates will converge at least to a stationary point (local optimum) of the ML objective in (2).

B. Estimating the number of Gaussian components

In the previous section, the number of Gaussian components M_0 is assumed known. To deal with a more practical setting where M is unknown, we modify the EM algorithm presented in Sec. III by adapting the so-called CEM method in [7] to our Gaussian plus non-Gaussian mixture model in (2). First, we assume a Dirichlet-type prior for the $\{\pi_m; m = 1,\dots,M_0\}$ with $M_0 \gg M$ as follows

$$f(\pi_1,\dots,\pi_{M_0}) \propto -\frac{L}{2}\sum_{m=1}^{M_0}\log\pi_m \tag{13}$$

where $L \coloneqq D(D+3)/2$ is the number of parameters per Gaussian component. The negative exponent of the Dirichlet-type prior encourages π_m to be equal either to 0 or to 1, and therefore, since $\sum_{m=1}^{M_0}\hat{\pi}_m = 1$, this prior promotes sparsity in the distribution mixture. Then, the probability of the m^{th} Gaussian component at iteration t is computed as the solution of the following maximum a posteriori (MAP) problem subject to some constraints.

$$\begin{aligned}\hat{\pi}_m^{t+1} \;=\; & \arg\max_{\tilde{\pi}_m}\ Q(\tilde{\boldsymbol{\theta}}\,;\hat{\boldsymbol{\theta}}^t) + \log f(\tilde{\pi}_1,\dots,\tilde{\pi}_{M_0}) \\ & \text{subject to } \tilde{\pi}_m \geqslant 0 \\ & \sum_{m=1}^{M_0}\tilde{\pi}_m = 1 \end{aligned} \tag{14}$$

The proposed algorithm proceeds as follows. The *E*-step remains the same and computes the a posteriori probabilities as in (7) and (8). The *M*-step is modified so that, instead of (10), the probability of the m^{th} Gaussian component becomes the solution of (14) given by

$$\hat{\pi}_m^{t+1} = \frac{\max\{0,(\sum_{r=1}^{R}\sum_{i=1}^{N_r}\ \alpha_{r,i}^t\,\zeta_{r,i,m}^t) - \frac{L}{2}\}}{\sum_{m=1}^{M_0}\max\{0,(\sum_{r=1}^{R}\sum_{i=1}^{N_r}\ \alpha_{r,i}^t\,\zeta_{r,i,m}^t) - \frac{L}{2}\}} \tag{15}$$

and the parameters $\{\hat{\boldsymbol{\mu}}_m^{t+1}, \hat{\boldsymbol{\Sigma}}_m^{t+1}\}$ are computed as in (11) and (12), but only for those $m \in \{1,\dots,M_0\}$ such that $\hat{\pi}_m^{t+1} \neq 0$. Parameters $\{\hat{p}_r^{t+1}; \forall r\}$ are updated as in (9). For convenience, let $\hat{M}^t$ denote the number of Gaussian components for which $\hat{\pi}_m^t \neq 0$. Note that the impact of (15) on the iterative algorithm is that some of the components of the Gaussian mixture will be eventually annihilated. It is therefore convenient to select $M_0 \gg M$, but also because it reduces the sensitivity of the algorithm to the initial values of the remaining parameters. Additionally, as pointed out by [7], at each iteration our algorithm calculates the Bayesian information criterion (BIC), namely

$$\mathcal{L}(\mathcal{X},\hat{\boldsymbol{\theta}}^t,\hat{M}^t) = -Q(\tilde{\boldsymbol{\theta}};\hat{\boldsymbol{\theta}}^t) + \frac{L\hat{M}^t}{2}\log\left(\sum_{r=1}^{R}\sum_{i=1}^{N_r}\alpha_{r,i}^t\right) \tag{16}$$

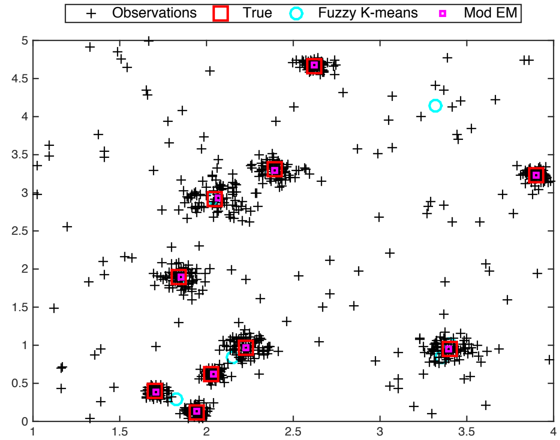

Fig. 1. Instances in + (black), true Gaussian means in big red □, centroids estimated by Fuzzy c-means in cyan ◯ and estimated by the modified EM in small magenta □.

where the double summation inside the $\log$ function is the soft count of non-outlying instances at iteration t. Overall, the BIC criterion is used to terminate the EM iterations, and also once convergence is reached, to check if larger values of $\mathcal{L}(\mathcal{X},\hat{\boldsymbol{\theta}}^t,\hat{M}^t)$ are achieved by setting to zero one by one those components not annihilated by (15). Specifically, the procedure is the following one. First, the presented algorithm is run until (16) does not vary substantially from one iteration to the next. Once convergence is reached, the least probable component of the Gaussian mixture, i.e. the one with smallest non-zero $\hat{\pi}_m^t$, is annihilated and the algorithm is run until convergence again. This last step is iterated until $\hat{M}^t = 1$ or equal to the minimum number of Gaussian components if known. The final estimates, denoted by $\{\boldsymbol{\theta}^{final}, \hat{M}^{final}\}$, are those $\{\boldsymbol{\theta}^t, \hat{M}^t\}$ among all t that maximize (16).

IV. Simulations

Simulations are shown to illustrate the performance of the novel algorithm. We consider $R = 20$ annotators providing instances with $D = 2$ according to (1) confined to a rectangular area of dimensions $\mathcal{U}_1^{\min} = 1$, $\mathcal{U}_1^{\max} = 4$, $\mathcal{U}_2^{\min} = 0$ and $\mathcal{U}_2^{\max} = 5$. The total number of instances is $N = 850$ with $N_r \in [36, 48]$. Fifteen annotators have a high reliability with $p_r = 0.95$, three have $p_r = 0.75$, and two have low reliability with $p_r = 0.25$. The density mixture consists of $M = 10$ Gaussians with equal probability, $\pi_m = 0.1$. As an example, Fig. 1 shows a realization with $N = 850$ instances, the Gaussian means $\{\boldsymbol{\mu}_m; \forall m\}$, the centroids estimated with the fuzzy *clustering*-means (fcm) function of MATLAB using the true number of Gaussians, and the centroid means estimated with our algorithm. In this setup, the covariance matrices of five Gaussian components are $\mathrm{diag}\{\boldsymbol{\Sigma}_m\} = [0.04, 0.05]$, four Gaussian components $\mathrm{diag}\{\boldsymbol{\Sigma}_m\} = [0.08, 0.1]$ and a single Gaussian component has even larger variances $\mathrm{diag}\{\boldsymbol{\Sigma}_m\} = [0.12, 0.15]$.

The experiment proceeds as follows. The EM-based algorithm in Sec. III-B is run for $K = 500$ independent realizations

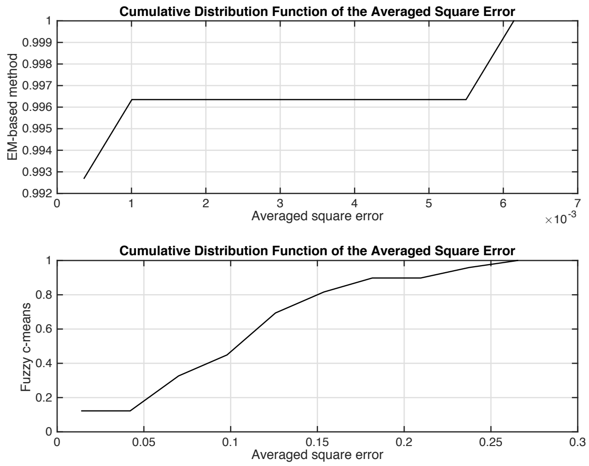

Fig. 2. Cumulative distribution function of ASE for successful realizations using (a) the proposed EM-based algorithm, and (b) fuzzy c-means

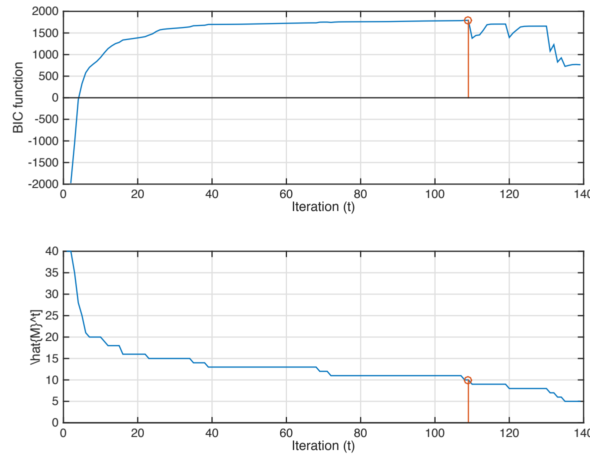

Fig. 3. Evolution of (a) $\mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\theta}}^t, \hat{M}^t)$ in (16), and (b) $\hat{M}^t$ in a single realization. The maximum of $\mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\theta}}^t, \hat{M}^t)$ is marked in (a) with $\circ$, and the corresponding $\hat{M}^t = 10$ in (b).

using the same Gaussian plus non-Gaussian density mixture of Fig. 1. The parameters are initialized as follows. The initial estimated centroids $\{\hat{\boldsymbol{\mu}}_1^0, \ldots, \hat{\boldsymbol{\mu}}_{M_0}^0\}$ are the centroids estimated by the K-means algorithm [13] with $M_0 = 40$; the initial estimated Gaussian covariance matrices are all set to $\{\hat{\boldsymbol{\Sigma}}_m^0 = \text{diag}\{[0.15 \;\; 0.25]\}; \forall m = 1, \ldots, M_0\}$. The algorithm is executed until $\hat{M}^t < 6$ or up to 200 iterations. For comparison purposes, the fuzzy c-means (fcm) function of MATLAB with $M = 10$ clusters is also tested. We decide a realization is successful if it estimates correctly the number of Gaussian components, i.e. $\hat{M}^{final} = M$, and a one-to-one correspondence can be established between the estimated centroids and the true Gaussian means according to a minimum distance criterion. Our algorithm correctly succeeds in 94% of the 500 realizations whereas fcm only succeeds in 47%.

Fig. 2 depicts the cumulative distribution function (cdf) for the means of evaluating the average square error (ASE), namely the square error between the true Gaussian means $\{\boldsymbol{\mu}_m; m = 1, \ldots, M\}$ and the final estimated centroids $\{\hat{\boldsymbol{\mu}}_m^{final}; m = 1, \ldots, M\}$ averaged over the M Gaussian components, that is

$$ASE \coloneqq \frac{1}{M} \sum_{m=1}^{M} ||\hat{\boldsymbol{\mu}}_m^{final} - \boldsymbol{\mu}_m||_2^2. \tag{17}$$

Note that only successful realizations are considered in (17). The proposed algorithm performs much better, since the ASE is less than 6×10^{-3} in all successful realizations (i.e, 94%), whereas the ASE is much higher for fcm. The following figures of merit are further attained by the proposed algorithm in estimating the remaining parameters.

$$\begin{aligned} \tfrac{1}{\mathrm{K}} \textstyle\sum_{k=1}^{K} \tfrac{1}{M} \sum_{m=1}^{M} |\hat{\pi}_m^{final} - \pi_m|^2 &= 2.3 \times 10^{-5} \\ \tfrac{1}{\mathrm{K}} \textstyle\sum_{k=1}^{K} \tfrac{1}{R} \sum_{r=1}^{R} |\hat{p}_r^{final} - p_r|^2 &= 2.7 \times 10^{-3}. \end{aligned} \tag{18}$$

Again only successful realizations are taken into account in (18).

Finally, Fig. 3.a and Fig. 3.b show the evolution of $\mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\theta}}^t, \hat{M}^t)$ and $\hat{M}^t$ in a single realization, respectively. In this particular realization, $\mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\theta}}^t, \hat{M}^t)$ increases due to the annihilation of Gaussian components performed in (15) until iteration $t = 109$, where BIC is stable. After this point, it decreases gradually each time the Gaussian component with lower probability is annihilated. The algorithm stops at iteration $t = 139$ because $\hat{M}^{139} = 5$. The final estimated values of the parameters $\{\boldsymbol{\theta}^{final}, \hat{M}^{final}\}$ used in (17) and (18) are those for which the maximum of $\mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\theta}}^t, \hat{M}^t)$ is attained, marked with a circle at iteration $t = 109$, and corresponding to $\hat{M}^{final} = 10$.

V. Conclusions

This paper formulates and solves a clustering and estimation problem for data adhering to a Gaussian mixture model in the presence of outliers, that are modeled as a uniformly distributed rv. The work fits nicely in the context of crowdsourcing applications, where observations are often provided by different annotators, each with *unknown* expertise. The proposed algorithm jointly estimates the density parameters of the Gaussian plus non-Gaussian mixture, the number of Gaussian components, and the reliability of annotators. Both the data model and the proposed algorithm are broad enough to be of interest in other general-purpose clustering applications. Our future research agenda includes generalizations to kernel-based crowdsourcing approaches to allow for clustering high-dimensional or nonlinearly separable datasets, as well as thorough testing and comparisons on real datasets provided e.g., by contaminating the MINST datasets to account for the variable reliability present in crowdsourcing collections.

REFERENCES

[1] 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*, vol. 11, no. Apr, pp. 1297–1322, 2010.

[2] M. A. Luengo-Oroz, A. Arranz, and J. Frean, “Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears,” *Journal of Medical Internet Research*, vol. 14, no. 6, p. e167, 2012.

[3] https://www.zooniverse.org.

[4] E. Simpson, S. Roberts, I. Psorakis, and A. Smith, “Dynamic Bayesian combination of multiple imperfect classifiers,” in *Decision Making and Imperfection*. Springer, 2013, pp. 1–35.

[5] J. D. Banfield and A. E. Raftery, “Model-based Gaussian and non-Gaussian clustering,” *Biometrics*, pp. 803–821, 1993.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

[7] M. A. T. Figueiredo and A. K. Jain, “Unsupervised learning of finite mixture models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[8] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar, “A review of robust clustering methods,” *Advances in Data Analysis and Classification*, vol. 4, no. 2-3, pp. 89–109, 2010.

[9] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, “Gaussian mixture density modeling, decomposition, and applications,” *IEEE Trans. on Image Processing*, vol. 5, no. 9, pp. 1293–1302, 1996.

[10] D. W. Scott, “Outlier detection and clustering by partial mixture modeling,” in *COMPSTAT 2004?Proceedings in Computational Statistics*. Springer, 2004, pp. 453–464.

[11] D. Nguyen, L. Chen, and C. Chan, “An outlier-aware data clustering algorithm in mixture model,” in *Proc. 7th Int. Conf. Information, Communication and Signal Processing*, 2009, pp. 1–5.

[12] P. A. Forero, V. Kekatos, and G. B. Giannakis, “Robust clustering using outlier-sparsity regularization,” *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4163–4177, 2012.

[13] S. Lloyd, “Least-squares quantization in pcm,” *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.