Unsupervised online clustering and detection algorithms using crowdsourced data for Malaria diagnosis

Alba Pagès-Zamora[a,*], Margarita Cabrera-Bean[a], Carles Díaz-Vilor[a]

[a]*SPCOM Group, Universitat Politècnica de Catalunya - BarcelonaTech (UPC), C/Jordi Girona 31, 08034, Barcelona, Spain (e-mail: alba.pages@upc.edu; marga.cabrera@upc.edu).*

Abstract

Crowdsourced data in science might be severely error-prone due to the inexperience of annotators participating in the project. In this work, we present a procedure to detect specific structures in an image given tags provided by multiple annotators and collected through a crowdsourcing methodology. The procedure consists of two stages based on the Expectation-Maximization (EM) algorithm, one for clustering and the other one for detection, and it gracefully combines data coming from annotators with unknown reliability in an unsupervised manner. An online implementation of the approach is also presented that is well suited to crowdsourced streaming data. Comprehensive experimental results with real data from the MalariaSpot project are also included.

Keywords: Crowdsourcing; unreliable annotators; unsupervised method; online EM algorithm; MalariaSpot.

1. Introduction

The term *crowdsourcing* was coined by J. Howe and M. Robinson in 2005 when analyzing how businesses were using internet to outsource work to individuals. In a crowdsourcing methodology, an entity broadcasts an open call for contributions to solve a problem, and individuals submit inputs which become property of the entity. This methodology has enormous potential in science because it allows large data sets to be analyzed in a timely and accurate manner by leveraging a network of human analysts or *annotators* instead of relying on a reduced number of experts. A representative sample of crowdsourcing projects from disciplines as diverse as astronomy, biology, and linguistics, among others, can be found in the *Zooniverse* platform at https://www.zooniverse.org. Typically, in these projects, annotators are asked either to classify images into binary or multiple classes, or to identify specific structures in an image. For instance, the *Snow Spotter* project presents landscape pictures and annotators are

[*] Corresponding author.
Declarations of interest: none.

asked whether there is snow on top of the trees or not, i.e., a binary classification task. An example of multiple classification can be found in the *Notes of Nature* project in which images of labeled butterflies are shown to annotators who transcribe the country handwritten in the label. Instead, in the *Microscope Masters* project, annotators pick out proteins in electron microscopy images for biological molecule reconstruction. Inevitably, crowdsourcing methodology is severely error-prone since annotators are usually non-experts, or may even be malicious, a fact that motivates robust techniques to process the collected data.

This paper focuses on the problem of identifying structures in an image. In particular, it uses crowsourced data of the *MalariaSpot* project [1] as an illustrative application in which annotators are asked to identify malaria parasites in digitized images of blood smears. The gold standard approach to diagnose this infection is microscopic examination of Giemsa-stained thick and thin blood films for counting malaria parasites. Reliable detection of malaria parasites in microscopic images demands trained technicians, resulting in a very expensive and time consuming task. Therefore, automated methods for identification and counting of malaria parasites in an unsupervised manner are highly valued (see [2] for a comprehensive review). Automated processes based on image processing techniques already exist in the literature, e.g., [3, 4, 5, 6, 7], and mostly analyze thin blood films where parasites remain inside red blood cells so that they can be identified more easily. Still, the use of thick blood films is preferred by microscopists since detection and counting of parasites is more reliable due to the higher concentration [8]. However, in general, image processing techniques with thick blood films tend to erroneously identify many artifacts as parasites since these are not inside a blood cell any longer. Still, existing contributions based on image processing techniques using thick blood films can be found in [9, 10] but, unlike the approach proposed in this paper, both of them are supervised methods. The MalariaSpot project advocates a completely different methodology for malaria diagnosis described in [11] and based on algorithms that process crowdsourced data. Through a dedicated on-line gaming platform, the MalariaSpot project offers digitized thick blood images through the web to volunteers who, after a short training period, deliver their inputs to be processed in a centralized manner by a simple algorithm.

In this paper, we propose a robust technique to process crowdsourced data provided by annotators with unknown reliability who are asked to identify specific structures in an image, as in the MalariaSpot project in which annotators spot parasites in images. The proposed technique also rates annotators according to their performance so that data provided by unreliable annotators is judiciously combined, e.g., [12]. The errors made by annotators are basically of two different natures. Some of them are isolated randomly located errors, whereas others correspond to an artifact erroneously tagged by several annotators. With the aim of processing the tags of the annotators while discarding these errors, the proposed approach consists of two steps: an unsupervised clustering stage and a detection stage both based on the Expectation-Maximization (EM) algorithm [13]. In the first stage, the data provided by all annotators is processed in a joint manner so that different clusters are identified and annotators are ranked. Specifically, the probability density function (pdf) of the data provided by annotators is modeled as a mixture of an unknown number of Gaussian components plus a uniformly distributed random variable (rv), which models the annotators' isolated errors as outliers. Unlike previous works for clustering, e.g., [14, 15, 16, 17], the proposed EM-based clustering algorithm not only estimates the number of Gaussian components and the parameters of the Gaussian plus non-Gaussian mixture density, but also annotators' reliability. In the detection stage, a decision is made, on each cluster identified in the clustering step, on whether it corresponds to one of the desired structures or not, taking into account annotators' reliability. When known, the true labels of the clusters are referred to as *ground truth*. The detection algorithm is inspired by [18, 19] which are prominent works on latent variable models applied to crowdsourcing. In summary, the clustering algorithm jointly ranks annotators and discards randomly located errors to cluster the data, whereas the detection stage aims at rejecting artifacts erroneously tagged by several annotators.

The main contributions of this paper are the following. Firstly, an unsupervised algorithm for the clustering stage is presented that is similar to our previous work in [20] albeit updated to deal with real data from the MalariaSpot project. Secondly, the complete procedure of clustering and detection, taking into account annotators' performance, is presented using a harmonized notation, which gracefully enables information from the clustering to the detection stage to be conveyed. Further, an online implementation of the complete procedure of clustering and detection is developed, which is of great interest in crowdsourced projects where streaming data are usually available. Whereas existing online EM algorithms, e.g., [21, 22, 23], assume a fixed set of parameters, in our setup the set of parameters to estimate increases as new data are available which poses an additional challenge. Finally, both the batch and the online proposed techniques are assessed not only with synthetic data but also with comprehensive numerical tests on real data from the MalariaSpot project[1]. Although out of the scope of this paper, the described techniques might also be used to process similar data provided not by annotators but instead by automated individual methods with unknown reliability.

The rest of this paper is organized as follows. Section 2 defines notation and introduces the data model. Section 3 presents the unsupervised clustering algorithm and the associated

[1] The full batch procedure was partially published in [24, 25] without the harmonized notation and with very limited experimental results. A preliminary simpler version of the online implementation of the detection stage with a fixed set of parameters was also included in [25].

annotators rating problem. Simulation results using synthetic data are included in this section for the sake of clarity. Then, Section 4 describes the procedure to transfer the results of the clustering stage to the detection stage, and Section 5 presents the detection algorithm. The online implementation of the whole procedure, i.e., clustering and detection, is presented in Section 6. Section 7 shows results using real data of the MalariaSpot project and Section 8 concludes this paper. For the interested readers, Appendix A includes an illustrative description of the complete procedure with images from the MalariaSpot project.

Notation: Lowercase bold letters, $\boldsymbol{x}$, denote vectors; uppercase bold letters, $\mathbf{X}$, represent matrices; and calligraphic uppercase letters, $\mathcal{X}$, stand for sets. Sets of elements will be denoted with braces; for instance, $\{\boldsymbol{\mu}_m : m = 1, \cdots, M\}$ is the set of vectors $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_M\}$. $\mathbb{R}^D$ stands for the D-dimensional real Euclidean space; $\mathbf{x}^\top$ is the transpose of vector $\mathbf{x}$; $|\mathbf{X}|$ is the determinant of matrix $\mathbf{X}$; and $\mathbb{E}[\cdot]$ stands for expectation.

2. Collected Data Model

The data provided by annotators when identifying structures in an image are modeled statistically as a density mixture as follows. Consider a set of R annotators indexed by $r = 1, \ldots, R$. Each one provides N_r instances of a D-dimensional vector[2], denoted by $\boldsymbol{x}_{r,i} \in \mathbb{R}^D$. The i^{th} instance of annotator r is modeled as

$$\boldsymbol{x}_{r,i} = a_{r,i} \sum_{m=1}^{M} \delta(z_{r,i} - m)\boldsymbol{w}_m + (1 - a_{r,i})\boldsymbol{u} \tag{1}$$

where $\delta(\cdot)$ denotes the Kronecker delta function; for $r = 1, \cdots, R$ and $i = 1, \cdots, N_r$, scalar $a_{r,i} \in \{0, 1\}$ is an i.i.d Bernoulli random variable (rv) $a_{r,i} \sim \text{Bern}(p_r)$ where $p_r \in [0, 1]$, and scalar $z_{r,i} \in \{1, \ldots, M\}$ is an i.i.d discrete rv distributed as $\Pr\{z_{r,i} = m\} = \pi_m$, where $\sum_{m=1}^{M} \pi_m = 1$; vector $\boldsymbol{w}_m \in \mathbb{R}^D$ is an i.i.d. Gaussian rv distributed as $\boldsymbol{w}_m \sim \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, where $\boldsymbol{\mu}_m$ is the mean, $\boldsymbol{\Sigma}_m$ is the covariance matrix, and M is the number of Gaussian components; and $\boldsymbol{u} \in \mathbb{R}^D$ is a random vector with probability density function (pdf) denoted by $f_{\boldsymbol{U}}(\boldsymbol{u})$ and whose components are uniformly distributed as $u(d) \sim \text{Unif}[\mathcal{U}_d^{\min}, \mathcal{U}_d^{\max}]$ for $d = 1, \cdots, D$, [3]. The pdf of $\boldsymbol{w}_m$ is given by

$$f_{\boldsymbol{\Omega}}(\boldsymbol{w}_m; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_m|}} \exp\left\{ -\frac{1}{2} (\boldsymbol{w}_m - \boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{w}_m - \boldsymbol{\mu}_m) \right\} \tag{2}$$

Further, we assume that different instances are independent, and that all rv's in (1) are independent as well.

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

[3] In the described crowdsourcing setup, the support corresponds to the image dimension.

For convenience, we define the set of observed variables $\mathcal{X} := \{\boldsymbol{x}_{r,i}\}$ for $r = 1, \cdots, R$ and $i = 1, \cdots, N_r$, and similarly the sets of latent or hidden variables $\mathcal{A} := \{a_{r,i}\}$ and $\mathcal{Z} := \{z_{r,i}\}$, all three sets with cardinality $N := \sum_{r=1}^{R} N_r$.

The model in (1) is a mixture of M Gaussians plus a uniformly distributed rv with probabilities that depend on the annotator. The Gaussian components account for the clusters and the uniformly distributed rv for annotator errors or outliers. Note that $a_{r,i} = 1$ implies that the i^{th} instance provided by annotator r corresponds to the Gaussian component of the index given by $z_{r,i} \in \{1, \ldots, M\}$. Conversely, when $a_{r,i} = 0$, the instance is deemed to be an outlier and modeled as a uniformly distributed rv. Therefore, probability p_r can be seen as a measure of annotators' *reliability* since the lower p_r is, the higher the probability that annotator r provides an outlier.

The following sections 3-5 present the clustering and detection algorithms based on the Expectation-Maximization (EM) algorithm [13, 26] using a unified notation for the sake of clarity.

3. Robust Clustering of Crowdsourced Data

The objective of the unsupervised clustering stage is, given $\mathcal{X}$ and without knowing the ground truth, to estimate the set of unknown parameters of the model in (1) gathered in vector $\boldsymbol{\theta}$ defined as

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

These parameters are the number of Gaussian components or clusters M; the mean vector of the Gaussian components or cluster centroids $\{\boldsymbol{\mu}_m : m = 1, \cdots, M\}$; the covariance matrices of the Gaussian components $\{\boldsymbol{\Sigma}_m : m = 1, \cdots, M\}$; the probability of each Gaussian component $\{\pi_m : m = 1, \cdots, M\}$; and annotators' reliability $\{p_r : r = 1, \cdots, R\}$. We advocate a maximum likelihood (ML) estimate of $\boldsymbol{\theta}$ and, therefore, we require the likelihood function of the instances $\mathcal{X}$ given by

$$f(\mathcal{X}; \boldsymbol{\theta}) = \prod_{r=1}^{R} \prod_{i=1}^{N_r} \Big(p_r \sum_{m=1}^{M} \pi_m \, f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) + (1 - p_r) f_{\boldsymbol{U}}(\boldsymbol{x}_{r,i}) \Big) \tag{4}$$

where $f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ is the likelihood function of instance $\boldsymbol{x}_{r,i}$ given $z_{r,i} = m$. Since a closed-form maximization of $f(\mathcal{X}; \boldsymbol{\theta})$ is not possible, we resort to a numerical solution based on the so-called Counter-Wise EM (CEM) algorithm proposed in [27], which estimates the parameters of a Gaussian mixture density and the number of Gaussian components. Our approach generalizes the work in [27] to the density mixture in (1), which includes not only Gaussian components but also a uniform distribution, and considers data from multiple annotators with unknown reliability. The purpose is to obtain an algorithm more robust to data errors thanks to the uniform distribution that accounts for outliers.

The proposed EM algorithm is an iterative algorithm that regards $\mathcal{X}$ as the *incomplete* observation and the set $\{\mathcal{X}, \mathcal{A}, \mathcal{Z}\}$ as the *complete* one. Upon initialization of the parameters' estimate with $\hat{\boldsymbol{\theta}}^0$, the EM algorithm alternates between an expectation (E) step and a maximization (M) step in an iterative fashion as follows.

At iteration $t+1$ for $t \geq 0$, and given an estimate $\hat{\boldsymbol{\theta}}^t$, the E-step computes the conditional expectation of the log-likelihood function

$$Q_c(\tilde{\boldsymbol{\theta}}; \hat{\boldsymbol{\theta}}^t) := \mathbb{E}_{\mathcal{A},\mathcal{Z}}\{\log f(\mathcal{X}, \mathcal{A}, \mathcal{Z}; \tilde{\boldsymbol{\theta}}) \mid \hat{\boldsymbol{\theta}}^t, \mathcal{X}\} \tag{5}$$

where $\tilde{\boldsymbol{\theta}}$ denotes a 'trial' value of $\boldsymbol{\theta}$, and the complete pdf is

$$\begin{aligned} f(\mathcal{X}, \mathcal{A}, \mathcal{Z}; \tilde{\boldsymbol{\theta}}) = \prod_{r=1}^{R} \prod_{i=1}^{N_r} & \Big(\tilde{p}_r \sum_{m=1}^{\tilde{M}} \delta(z_{r,i} - m) \tilde{\pi}_m \, f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Sigma}}_m)\Big)^{a_{r,i}} \\ & \cdot \Big((1 - \tilde{p}_r) f_{\boldsymbol{U}}(\boldsymbol{x}_{r,i})\Big)^{(1-a_{r,i})} \end{aligned} \tag{6}$$

Recalling that $\mathcal{A}$ and $\mathcal{Z}$ are independent, it holds that

$$\begin{aligned} Q_c(\tilde{\boldsymbol{\theta}}; \hat{\boldsymbol{\theta}}^t) = & \sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \log \tilde{p}_r \sum_{m=1}^{\tilde{M}} \zeta_{r,i,m}^t \log\Big(\tilde{\pi}_m f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Sigma}}_m)\Big) \\ & + \sum_{r=1}^{R} \sum_{i=1}^{N_r} (1 - \alpha_{r,i}^t) \log\left((1 - \tilde{p}_r) f_{\boldsymbol{U}}(\boldsymbol{x}_{r,i})\right) \end{aligned} \tag{7}$$

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

$$\alpha_{r,i}^t = \frac{\hat{p}_r^t \sum_{m=1}^{\hat{M}^t} \hat{\pi}_m^t f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t)}{\hat{p}_r^t \sum_{m=1}^{\hat{M}^t} \hat{\pi}_m^t f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t) + (1 - \hat{p}_r^t) f_{\boldsymbol{U}}(\boldsymbol{x}_{r,i})} \tag{8}$$

and

$$\zeta_{r,i,m}^t = \frac{\hat{\pi}_m^t f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_m^t, \hat{\boldsymbol{\Sigma}}_m^t)}{\sum_{l=1}^{\hat{M}^t} \hat{\pi}_l^t f_{\boldsymbol{\Omega}}(\boldsymbol{x}_{r,i}; \hat{\boldsymbol{\mu}}_l^t, \hat{\boldsymbol{\Sigma}}_l^t)} \tag{9}$$

for $r = 1, \cdots, R$; $i = 1, \cdots, N_r$; and $m = 1, \cdots, \hat{M}^t$. Probability $\alpha_{r,i}^t$ is a *soft* decision at iteration t on whether instance $\boldsymbol{x}_{r,i}$ is an outlier or not, and $\zeta_{r,i,m}^t$ is a *soft* assignment of instance $\boldsymbol{x}_{r,i}$ to the m^{th} Gaussian component.

The M-step follows a Bayesian criterion, so that the estimate $\hat{\boldsymbol{\theta}}^{t+1}$ for the next iteration is obtained solving

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{t+1} \; = \; & \arg\max_{\tilde{\boldsymbol{\theta}}} \; Q_c(\tilde{\boldsymbol{\theta}}\,; \hat{\boldsymbol{\theta}}^t) + \log f(\tilde{\pi}_1, \ldots, \tilde{\pi}_{\hat{M}^t}) \\ & \text{subject to } \tilde{\pi}_m \geq 0; \; \sum_{m=1}^{\hat{M}^t} \tilde{\pi}_m = 1 \end{aligned} \tag{10}$$

except for $\hat{M}^{t+1}$, and where a negative Dirichlet-type prior is assumed

$$f(\tilde{\pi}_1, \ldots, \tilde{\pi}_{\hat{M}^t}) \propto \exp\left\{ -\frac{L}{4} \sum_{m=1}^{\hat{M}^t} \log \tilde{\pi}_m \right\}, \tag{11}$$

where $L = D(D+3)/2$ is the number of parameters per Gaussian component. The negative Dirichlet prior encourages configurations where $\tilde{\pi}_m$ tends to be either 1 or 0. Therefore, this, together with the probability constraint $\sum_{m=1}^{\hat{M}^t} \tilde{\pi}_m = 1$, promotes sparsity in the set $\{\hat{\pi}_m^{t+1} : m = 1, \ldots, \hat{M}^t\}$.

Then, substituting (7) in (10), it can be readily seen that annotators' reliability is updated as

$$\hat{p}_r^{t+1} = \frac{1}{N_r} \sum_{i=1}^{N_r} \alpha_{r,i}^t \tag{12}$$

for $r = 1, \cdots, R$; and the updated mean vectors and covariance matrices of the Gaussian components are given by

$$\hat{\boldsymbol{\mu}}_m^{t+1} = \frac{\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t \, \boldsymbol{x}_{r,i}}{\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t} \tag{13}$$

and

$$\hat{\boldsymbol{\Sigma}}_m^{t+1} = \frac{\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \zeta_{r,i,m}^t (\boldsymbol{x}_{r,i} - \hat{\boldsymbol{\mu}}_m^{t+1})(\boldsymbol{x}_{r,i} - \hat{\boldsymbol{\mu}}_m^{t+1})^\top}{\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t} \tag{14}$$

for $m = 1, \cdots, \hat{M}^t$. Note that the denominator in (13) and (14) is a *soft* count of all non-outlier instances that belong to the m^{th} Gaussian component at iteration $t+1$. Further, the probability of the m^{th} Gaussian component is computed solving the constrained optimization problem in (10), which becomes

$$\hat{\pi}_m^{t+1} = \frac{\max\{0, (\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t) - \frac{L}{4}\}}{\sum_{m=1}^{\hat{M}^t} \max\{0, (\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \, \zeta_{r,i,m}^t) - \frac{L}{4}\}} \tag{15}$$

Interestingly, the impact of (15) on the iterative algorithm is that those Gaussian components of the density mixture with a reduced number of *soft* assigned instances will be eventually annihilated by obtaining a probability equal to 0. It is therefore convenient to select the initial estimated number of Gaussian components such that $\hat{M}^0 \gg M$, which also makes our algorithm more robust to the initial values of the rest of the parameters. Finally, the estimated number of Gaussian components, $\hat{M}^{t+1}$, is set equal to the number of Gaussian components such that $\hat{\pi}_m^{t+1} \neq 0$[4].

The criterion proposed to stop the iterative algorithm is based on the function $\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t)$,

[4] Here, we are assuming that at each iteration t, the indexing of the $\hat{M}^{t+1}$ Gaussian components with $\hat{\pi}_m^{t+1} \neq 0$ is reorganized to become the first $\hat{M}^{t+1}$ components.

defined for $t \geq 0$ as

$$\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t) = Q_c(\hat{\boldsymbol{\theta}}^{t+1}; \hat{\boldsymbol{\theta}}^t) - \frac{L}{4} \sum_{m=1}^{\hat{M}^{t+1}} \log \hat{\pi}_m^{t+1} - \frac{\hat{M}^{t+1}(L+1)+R}{2} \log \left(\sum_{r=1}^{R} \sum_{i=1}^{N_r} \alpha_{r,i}^t \right) \quad (16)$$

where we have the sets $\mathcal{A}^t := \{\alpha_{r,i}^t\}$ and $\mathcal{Z}^t := \{\zeta_{r,i,m}^t\}$ for $r = 1, \cdots, R$; $i = 1, \cdots, N_r$; and $m = 1, \cdots, \hat{M}^{t+1}$. Note that (16) is the M-step cost function in (10) plus a term that penalizes models with a large number of parameters. Following the procedure in [27], when

$$\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t) - \mathcal{L}(\hat{\boldsymbol{\theta}}^{t}, \mathcal{A}^{t-1}, \mathcal{Z}^{t-1}) < \epsilon |\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t)|, \quad (17)$$

the least probable component of the Gaussian mixture is annihilated, i.e., the smallest non-zero $\hat{\pi}_m^{t+1}$ is set to 0, and the algorithm is run again until inequality (17) is satisfied[5]. This procedure is successively applied until $\hat{M}^{t+1} = 1$, or to a lower bound on the number of Gaussian components if known beforehand. The final parameter estimates after the clustering stage, denoted by $\hat{\boldsymbol{\theta}}^c$, are set equal to those that maximize (16), i.e.,

$$\{\hat{\boldsymbol{\theta}}^c, \{\alpha_{r,i}^c\}, \{\zeta_{r,i,m}^c\}\} = \arg\max_{\forall t} \mathcal{L}(\hat{\boldsymbol{\theta}}^{t}, \mathcal{A}^{t-1}, \mathcal{Z}^{t-1}). \quad (18)$$

The algorithm implemented by equations (8), (9), and (12)-(15) is denoted hereafter as the Outlier EM (OEM) algorithm and it is summarized in Alg. 1. The output of the clustering stage is computed in (18) and given by $\hat{\boldsymbol{\theta}}^c$ and the sets of a posterior probabilities $\{\alpha_{r,i}^c\}$ and $\{\zeta_{r,i,m}^c\}$ for $r = 1, \cdots, R$; $i = 1, \cdots, N_r$; and $m = 1, \cdots, \hat{M}^{t+1}$.

3.1. Simulation results with synthetic data

This section shows simulation results with synthetic data to illustrate the performance of OEM. We consider $R = \{11, 21, 25, 31, 41, 51\}$ annotators providing instances according to (1) with $D = 2$ and 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 average number of instances per annotator is 20 and $N_r \in [16, 24]$. Sixty percent of annotators have $p_r = 0.95$, 30% $p_r = 0.75$, and 10% have low reliability with $p_r = 0.25$. The number of Gaussian components is $M = 10$ with $\pi_m = 0.1$ for $m = 1, \cdots, M$. Figure 1 shows a realization with $N = 1000$ instances of (1) with $R = 51$ and it also includes the Gaussian means $\{\boldsymbol{\mu}_m : m = 1, \cdots, M\}$. The covariance matrices of 5 Gaussian components are $\boldsymbol{\Sigma}_m = \text{diag}([0.04, 0.05])$, of 4 Gaussian components $\boldsymbol{\Sigma}_m = \text{diag}([0.08, 0.1])$ and the last one has even larger variances $\boldsymbol{\Sigma}_m = \text{diag}([0.12, 0.15])$. This setup is selected because of its difficulty due to the proximity of Gaussian components with different variances. Results of OEM are averaged using 100 independent realizations. The initial estimated means $\{\hat{\boldsymbol{\mu}}_m^0; \forall m = 1, \cdots, \hat{M}^0\}$ are the centroids obtained by the k-means algorithm [28, 29] with $\hat{M}^0$ equal to 6 times the average

[5] In our experiments, $\epsilon = 1e-5$ and $\mathcal{L}(\hat{\boldsymbol{\theta}}^0, \mathcal{A}^{-1}, \mathcal{Z}^{-1})$ is initialized to $-\infty$.

Algorithm 1 OEM clustering

```
Input: $R$, $\mathcal{X}$, $\hat{\boldsymbol{\theta}}^0$, $M_{min}$, $T_{max}$, $\epsilon$
Output: $\hat{\boldsymbol{\theta}}^c, \{\alpha^c_{r,i}\}, \{\zeta^c_{r,i,m}\}$
Set $t \leftarrow -1$ and $\mathcal{L}(\hat{\boldsymbol{\theta}}^0, \mathcal{A}^{-1}, \mathcal{Z}^{-1}) \leftarrow -\infty$
while $\hat{M}^{t+1} > M_{min}$ and $t < T_{max}$ do
    repeat
        $t \leftarrow t + 1$
        E-Step: Compute $\{\alpha^t_{r,i}\}$ and $\{\zeta^t_{r,i,m}\}$ using (8) and (9).
        M-Step: Compute $\hat{\boldsymbol{\theta}}^{t+1}$ using (12)-(15), and set $\hat{M}^{t+1}$ equal to the number of Gaussian components such that $\hat{\pi}^{t+1}_m \neq 0$.
        Calculate $\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t)$ using (16).
    until $\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t) - \mathcal{L}(\hat{\boldsymbol{\theta}}^t, \mathcal{A}^{t-1}, \mathcal{Z}^{t-1}) < \epsilon |\mathcal{L}(\hat{\boldsymbol{\theta}}^{t+1}, \mathcal{A}^t, \mathcal{Z}^t)|$
    Set $\hat{\pi}^{t+1}_{m_0} \leftarrow 0$ where $m_0 = \arg\min_{\{\forall m=1,\cdots,\hat{M}^{t+1}\}} \hat{\pi}^{t+1}_m$
end while
Obtain $\{\hat{\boldsymbol{\theta}}^c, \{\alpha^c_{r,i}\}, \{\zeta^c_{r,i,m}\}\}$ using (18).
```

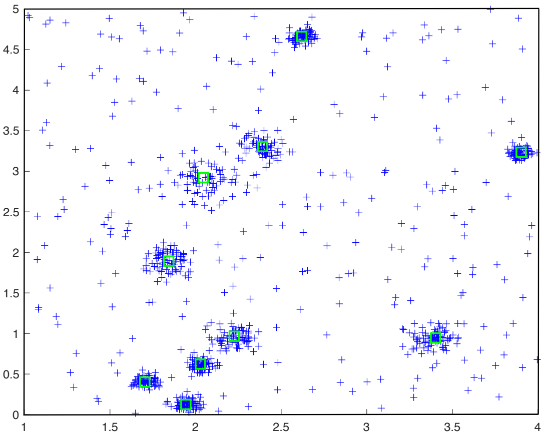

Figure 1: Instances + and true Gaussian means big □ with synthetic data.

number of clicks per annotator, i.e., around 120 in our setup. The initial estimated Gaussian covariance matrices are all set to $\{\hat{\boldsymbol{\Sigma}}^0_m = \boldsymbol{\Sigma}^0 := \frac{\sigma^2_x}{200}\mathbf{I}; \forall m = 1, \ldots, \hat{M}^0\}$, where σ^2_x is the sample variance of the instances. Probabilities are initialized as $\hat{\pi}^0_m = 1/\hat{M}^0$ for all m, and $p_r = p^0 := 0.9$ for all r. OEM is executed until $\hat{M}^t = 1$ or up to $T_{max} = 500$ iterations. For comparison purposes, k-means, the hierarchical agglomerative clustering (HAC) method (see e.g., [29] for details), and CEM of [27] are also evaluated. CEM is initialized exactly as OEM, and k-means is run with a number of centroids twice the average number of clicks per annotator. Results are evaluated in terms of *sensitivity* denoted by S^c and *precision* denoted by P^c which are measured as

$$\begin{aligned} S^c &= \frac{TP^c}{N_p} \\ P^c &= \frac{TP^c}{TP^c + FP^c} \end{aligned} \tag{19}$$

where TP, FP, TN and FN stand for True/False Positives/Negatives; N_p denotes the number of true ground truth elements; and supraindex c denotes after the clustering stage. In this setup with synthetic data, the ground truth are the means of the Gaussian components and, therefore, $N_p = M = 10$. Figure 1 shows the sensitivity and precision obtained applying OEM, CEM, k-means, and HAC. Clearly, HAC performs the worst with lower sensitivity and precision.

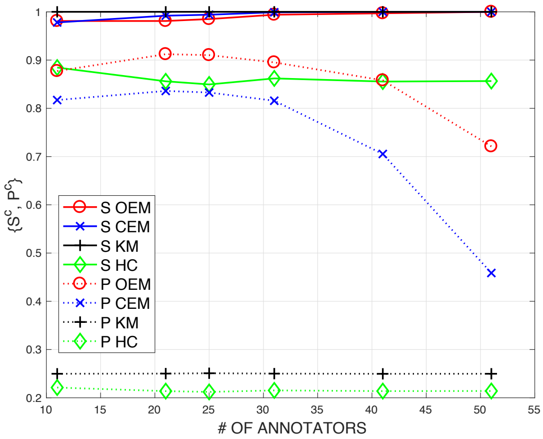

Figure 2: Sensitivity (solid) and precision (dotted) after clustering with OEM o, CEM x, k-means + and HAC ◇.

The other three methods achieve a similar sensitivity close to one, and OEM outperforms the rest with a higher precision. Note that k-means might easily improve precision by reducing the number of centroids, but at the cost of reducing sensitivity as well. Also note that at the clustering stage it is crucial to not miss true positives, i.e., prioritize a high sensitivity, otherwise there would be no option to identify them as positive in the detection stage.

4. Data Processing after Clustering

The information at the end of the clustering stage is computed in (18) and given by the parameter estimate, $\hat{\boldsymbol{\theta}}^c$, and the soft assignment of each instance to the clusters and the outliers set, given respectively by the posterior probabilities $\{\zeta^c_{r,i,m}\}$ and $\{\alpha^c_{r,i}\}$ for $r = 1, \cdots, R$; $i = 1, \cdots, N_r$; and $m = 1, \cdots, \hat{M}^{t+1}$.

The number of identified clusters at the end of the clustering stage is given by $\hat{M}^c$, which is the number of Gaussian components with non-zero probability. Without loss of generality, we assume a reorder of the cluster indexes such that the first $\hat{M}^c$ clusters are those with $\hat{\pi}_m^c \neq 0$. These $\hat{M}^c$ clusters are the (possibly erroneous) structures identified jointly by all annotators. For instance, in the MalariaSpot setup, these clusters become potential parasites and the final objective of the detection stage is to pick out those that correspond to true parasites. The rest of the estimated parameters after the clustering stage include the cluster centroids given by $\{\hat{\boldsymbol{\mu}}_m^c : m = 1, \cdots, \hat{M}^c\}$ and the cluster covariance matrices $\{\hat{\boldsymbol{\Sigma}}_m^c : m = 1, \cdots, \hat{M}^c\}$. Note that covariance matrices are indicative of the cluster size and, in some applications, might be well used to complement the detection stage. Finally, annotators are also ranked according to $\hat{p}_r^c$. These values can be used to initialize $\hat{p}_r^0$ in the clustering stage of other images where any of the current annotators provide instances, and will definitely be used in the online implementation in Section 6.

Before applying the detection stage, results provided by the annotators should be organized according to the identified clusters. Firstly, instances that correspond to an outlier with high probability are discarded, a fact that can be inferred from the posterior probability $\alpha_{r,i}^c$. For convenience, let us define the set $\mathcal{X}^c := \{\boldsymbol{x}_{r,i} : r = 1, \cdots, R; i = 1, \cdots, N_r;$ such that $\alpha_{r,i}^c \geq \delta^c\}$ with $0 \leq \delta^c \leq 1$, as the set of non-outlier instances[6]. Accordingly, we also define $\{\mathcal{X}_r^c : r = 1, \cdots, R\}$ to denote the set of non-outlier instances provided individually by the annotators.

Secondly, instances of $\mathcal{X}^c$ must be assigned to one of the identified clusters and for that we use the soft assignment $\zeta_{r,i,m}^c$. Thus, a hard decision is taken to assign each non-outlier instance to the cluster with higher posterior probability among the $\hat{M}^c$ identified clusters as follows

$$\mathcal{C}(\boldsymbol{x}_{r,i}) = \underset{m \in 1, \cdots, \hat{M}^c}{\arg\max} \, \zeta_{r,i,m}^c$$

for all $\boldsymbol{x}_{r,i} \in \mathcal{X}^c$. That is, $\mathcal{C}(\boldsymbol{x}) \in \{1, \cdots, \hat{M}^c\}$ can be seen as an operator that returns the cluster associated to the generic instance $\boldsymbol{x}$.

Finally, since not all $\hat{M}^c$ identified clusters have been tagged individually by all annotators, we compute the variables $y_{r,m} \in \{0, 1\}$ for all $r = 1, \cdots, R$ and $m = 1, \cdots, \hat{M}^c$ as follows

$$y_{r,m} = \begin{cases} 1 & \text{if } \exists \boldsymbol{x}_{r,i} \subset \mathcal{X}_r^c \text{ s.t. } \mathcal{C}(\boldsymbol{x}_{r,i}) = m \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

and build the set of labels $\mathcal{Y} := \{y_{r,m} : r = 1, \cdots, R; m = 1, \cdots, \hat{M}^c\}$. For convenience, let us define the subset $\mathcal{Y}_m := \{y_{r,m} : r = 1, \cdots, R\}$ that can be seen as the set of individual binary labels given by the R annotators to the m^{th} cluster identified in the clustering stage.

[6] The higher δ^c is, the more conservative the decision about non-outliers.

5. Detection Stage

The problem at hand at this stage is, given annotator labels in $\mathcal{Y}$, to make a binary decision, on each one of the $\hat{M}^c$ identified clusters, on whether it corresponds to a true structure or *true positive*, or not. That is, we face a classification problem of the identified clusters into 2 classes. For instance, in the MalariaSpot setup, we have to decide whether each cluster identified in the clustering stage corresponds to a true parasite or not. We denote the unknown true labels by the set of binary variables $\mathcal{B} := \{b_m : m = 1, \cdots, \hat{M}^c\}$, meaning that $b_m = 1$ when the m^{th} identified cluster corresponds to a true positive, and $b_m = 0$ if it corresponds to a false positive. The elements in $\mathcal{B}$ are modeled as hidden rvs with prior probability of having a true positive equal to $\mu := Pr(b_m = 1)$ for $m = 1, \cdots, \hat{M}^c$. To solve this binary classification problem, we model annotators' labels in $\mathcal{Y}$ as Bernouilli rvs, and apply the EM algorithm proposed in [18] that estimates the unobserved true labels in $\mathcal{B}$ using $\mathcal{Y}$.

For that, we assume that each annotator tags the m^{th} identified cluster as 0 or 1 based on two biased coins. Annotator r flips a coin with bias $\psi_r := Pr(y_{r,m} = 1|b_m = 1)$ if $b_m = 1$, or with bias $\eta_r := Pr(y_{r,m} = 0|b_m = 0)$ if $b_m = 0$. These biases are known respectively as *sensitivity*, or true positive ratio, and *specificity*, or true negative ratio. Subscript r denotes that they may differ from one annotator to another.

As usual in the related literature [18], we also adopt the common assumption that annotators are conditionally independent, i.e., for any pair of different annotators r and q we assume that

$$Pr(y_{r,m}, y_{q,m}|b_m) = Pr(y_{q,m}|b_m) \cdot Pr(y_{q,m}|b_m),$$

meaning in practice that annotators do not communicate among them. Assuming that decisions on each identified cluster are independent, the likelihood function of the complete set $\{\mathcal{Y}, \mathcal{B}\}$ is equal to

$$\begin{aligned} f(\mathcal{Y}, \mathcal{B}; \boldsymbol{\phi}) &= \prod_{m=1}^{\hat{M}^c} f(\mathcal{Y}_m, b_m; \boldsymbol{\phi}) \\ &= \prod_{m=1}^{\hat{M}^c} ((1-\mu)B_0(\mathcal{Y}_m; \boldsymbol{\phi}))^{(1-b_m)} (\mu B_1(\mathcal{Y}_m; \boldsymbol{\phi}))^{b_m} \end{aligned}$$

where $B_0(\mathcal{Y}_m; \boldsymbol{\phi}) := Pr(\mathcal{Y}_m|b_m{=}0)$ and $B_1(\mathcal{Y}_m; \boldsymbol{\phi}) := Pr(\mathcal{Y}_m|b_m{=}1)$ given by

$$B_0(\mathcal{Y}_m; \boldsymbol{\phi}) = \textstyle\prod_{r=1}^{R} \eta_r^{(1-y_{r,m})} (1-\eta_r)^{y_{r,m}} \tag{21}$$

and

$$B_1(\mathcal{Y}_m; \boldsymbol{\phi}) = \textstyle\prod_{r=1}^{R} \psi_r^{y_{r,m}} (1-\psi_r)^{(1-y_{r,m})}. \tag{22}$$

Note that (21) and (22) hold because annotators are conditionally independent. Vector $\boldsymbol{\phi}$ is the parameter vector defined as

$$\boldsymbol{\phi} = [\mu; \psi_1, \cdots, \psi_R, \eta_1, \cdots, \eta_R] \tag{23}$$

and includes the prior probability of the classes, and the sensitivity and specificity of all annotators. Since all these parameters are unknown, the EM algorithm in [18] estimates not only the unobserved true labels, but also the prior probabilities of each class and the sensitivity and specificity of each annotator in a joint manner. After initializing $\hat{\boldsymbol{\phi}}^0$ conveniently, the EM algorithm alternates between an E-step and an M-step in an iterative fashion until convergence. At iteration $k+1$[7], the E-step computes the following expectation of the log-likelihood function

$$Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^k) := \mathbb{E}_{\mathcal{B}}\{\log f(\mathcal{Y}, \mathcal{B}; \tilde{\boldsymbol{\phi}}) | \hat{\boldsymbol{\phi}}^k, \mathcal{Y}\} \tag{24}$$

where $\tilde{\boldsymbol{\phi}}$ denotes a 'trial' value of $\boldsymbol{\phi}$. This step basically requires the computation of the posterior probability of the latent variables that, using Bayes' theorem, are equal to

$$\begin{aligned} \beta_m^k &:= Pr\{b_m = 1 | \hat{\boldsymbol{\phi}}^k, \mathcal{Y}\} \\ &= \frac{\hat{\mu}^k B_1(\mathcal{Y}_m; \hat{\boldsymbol{\phi}}^k)}{\hat{\mu}^k B_1(\mathcal{Y}_m; \hat{\boldsymbol{\phi}}^k) + (1 - \hat{\mu}^k) B_0(\mathcal{Y}_m; \hat{\boldsymbol{\phi}}^k)} \end{aligned} \tag{25}$$

for $m = 1, \cdots, \hat{M}^c$. The M-step updates the parameter estimate by solving

$$\hat{\boldsymbol{\phi}}^{k+1} = \arg\max_{\tilde{\boldsymbol{\phi}}} Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^k). \tag{26}$$

Then, at iteration $k+1$, the prior probability of having a true label is

$$\hat{\mu}^{k+1} = \frac{1}{\hat{M}^c} \sum_{m=1}^{\hat{M}^c} \beta_m^k, \tag{27}$$

and the sensitivity and specificity are, respectively, equal to

$$\hat{\psi}_r^{k+1} = \frac{\sum_{m=1}^{\hat{M}^c} \beta_m^k y_m^r}{\sum_{m=1}^{\hat{M}^c} \beta_m^k} \text{ and} \tag{28}$$

$$\hat{\eta}_r^{k+1} = \frac{\sum_{m=1}^{\hat{M}^c} (1 - \beta_m^k)(1 - y_m^r)}{\sum_{m=1}^{\hat{M}^c} (1 - \beta_m^k)} \tag{29}$$

for $r = 1, \cdots, R$. Equations (25),(27)-(29) are iterated until convergence[8]. The final parameter estimates are given by

$$\hat{\boldsymbol{\phi}}^d := \hat{\boldsymbol{\phi}}^K, \tag{30}$$

[7] For the sake of clarity, we use different iteration indexes to distinguish between the OEM and EM detection algorithm.

[8] In practice, we set a maximum number of iterations given by K_{max}.

Algorithm 2 DEM Algorithm

```
Input: $\mathcal{Y}$, $\hat{M}^c$, $\{\beta_m^0 : m = 1, \cdots, \hat{M}^c\}$, $\epsilon$, $K_{max}$
Output: $\{\beta_m^d : m = 1, \cdots, \hat{M}^c\}$
Set $k \leftarrow -1$ and $Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^0) \leftarrow -\infty$
repeat
    $k \leftarrow k + 1$
    M-Step: Compute $\hat{\boldsymbol{\phi}}^{k+1}$ using (27)-(29).
    E-Step: Compute $\{\beta_m^{k+1}\}$ using (25).
    Calculate $Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1})$ using (24).
until $\left(Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1}) - Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k}) < \epsilon |Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1})|\right)$ or $(k = K_{max})$
Set $\beta_m^d \leftarrow \beta_m^{k+1}$ for $m = 1, \cdots, \hat{M}^c$.
```

where K is the minimum between K_{max} and the iteration in which $Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1}) - Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k}) < \epsilon |Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1})|$ where ϵ is a predefined small positive real. Similarly, the final posterior probabilities are given by

$$\beta_m^d := \beta_m^K \tag{31}$$

for $m = 1, \cdots, \hat{M}^c$. The decision on whether the clusters identified in the clustering step correspond to a true label or not is taken by a hard decision of the final posterior probabilities β_m^d. That is, for $m = 1, \cdots, \hat{M}^c$ we decide the m^{th} identified label is a true label if $\beta_m^d \geq \delta^d$, and a false label otherwise, where $0 < \delta^d < 1$. Upon initialization of β_m^0, the detection EM (DEM) algorithm proceeds, alternating between the M-step and E-step until convergence, as summarized in Alg. 2.

We do not provide results of the DEM algorithm with synthetic data since it has already been widely studied in the literature. Still, the initialization of the algorithm based on the results of the clustering stage is worthy of mention.

5.1. Algorithm Initialization

It is well known that the EM algorithm should be judiciously initialized to guarantee convergence to the ML solution. For DEM, we consider three different options to initialize the posterior probabilities as follows

$$\beta_m^0 = \frac{1}{R} \sum_{r=1}^{R} y_{r,m}; \tag{32}$$

$$\beta_m^0 = \begin{cases} 1 & \text{if } \sum_{r=1}^{R} y_{r,m} \geq \frac{R}{2} \\ 0 & \text{otherwise} \end{cases}; \tag{33}$$

and

$$\beta_m^0 = \frac{\hat{\pi}_m^c}{\max\{\hat{\pi}_1^c, \cdots, \hat{\pi}_{\hat{M}^c}^c\}} \tag{34}$$

for $1, \cdots, \hat{M}^c$. The first and second initialization options given in (32) and (33) are *soft* and *hard* majority voting criteria, respectively. The third initialization in (34) uses results of the clustering stage and normalizes the maximum value of β_m^0 to 1. Recall that the non-zero cluster probabilities $\hat{\pi}_m^c$ in (15) are a *soft* majority voting but weigh each instance by its probability of not being an outlier, which is given in the clustering step by $\alpha_{r,i}^c$. In the experimental results section, we will further comment on the initialization of the detection EM algorithm.

6. Online Implementation

Online implementations of the clustering-detection algorithm are highly advised in crowdsourcing applications because data provided by annotators are usually available in a streaming manner. Moreover, an online approach is more efficient since images can be set aside once results are good enough, and annotators are forwarded to analyze new images.

To implement the complete procedure in an online manner, we need to address both the clustering and detection EM algorithms. Several EM online implementations already exist in the literature, e.g., [21, 22, 23], but most of them use a fixed set of parameters. In our setup, however, the set of parameters to estimate increases as new instances are available since different annotators might come into play and new potential parasites can be identified after clustering. The online algorithm is summarized in Alg. 3. After initialization, the algorithm executes the clustering stage (OEM) followed by the detection stage (DEM) as new instances are available. For clarity, the index for the outer iteration is denoted by s.

6.1. Initialization

The algorithm is initialized by executing the batch clustering and detection algorithms described in Sections 3-5 after $R(0)$ annotators provide data [9]. That is, firstly OEM in Alg. 1 is executed in a batch mode to obtain the estimation of the clustering parameters denoted by $\hat{\boldsymbol{\theta}}^c(0)$, and the posterior probabilities $\{\alpha_{r,i}^c(0)\}$ and $\{\zeta_{r,i,m}^c(0)\}$ for $r = 1, \cdots, R(0)$; $i = 1, \cdots, N_r$; and $m = 1, \cdots, \hat{M}^c(0)$, where $\hat{M}^c(0)$ denotes the number of identified clusters at $s = 0$. Then, the data are processed as in Section 4 to generate $\mathcal{Y}(0)$. Finally, DEM in Alg. 2 is executed in a batch mode initialized with $\mathcal{Y}(0)$, $\hat{M}^c(0)$ and $\{\beta_m^0(0) : m = 1, \cdots, \hat{M}^c(0)\}$ to compute $\{\beta_m^d(0) : m = 1, \cdots, \hat{M}^c(0)\}$.

6.2. Online clustering algorithm

Then, at the outer iteration $s > 0$ we assume that there are new instances given by $\mathcal{X}^{new}(s) = \{\boldsymbol{x}_q^{new}(s); \forall q = 1, \cdots, |X^{new}(s)|\}$, so that a total of $\mathcal{X}(s) = \mathcal{X}(s-1) \cup \mathcal{X}^{new}(s)$

[9] Note that (0) shows dependence of the parameters' estimate at the first outer iteration $s = 0$.

Algorithm 3 Online clustering and detection algorithm

```
Input: $\mathcal{R}(0)$,$\{R^{new}(s) : s = 1, \cdots, S\}$, $\mathcal{X}(0)$, $\{\mathcal{X}^{new}(s) : s = 1, \cdots, S\}$, $\hat{\boldsymbol{\theta}}^0(0)$, $M_{min}$, $T_{max}$, $\epsilon$, $\delta^c$, $\delta^d$
Output: Number and centroids of true labels
Run once OEM-DEM batch algorithm to compute $\{\beta^d_m(0) : m = 1, \cdots, \hat{M}^c(0)\}$ with inputs $\mathcal{R}(0)$, $\mathcal{X}(0)$, $\hat{\boldsymbol{\theta}}^0(0)$.
Set $s \leftarrow 0$
while $s < S$ do
    $s \leftarrow s + 1$
    Given $\mathcal{X}^{new}(s)$, set $\hat{M}^0(s) = \hat{M}^c(s-1) + |\mathcal{X}^{new}_s|$ and compute $\hat{\boldsymbol{\theta}}^0(s)$ as explained in Section 6.2 using (35)-(37).
    Build $\mathcal{R}(s) = \mathcal{R}(s-1) \cup \mathcal{R}^{new}(s)$ and $\mathcal{X}(s) = \mathcal{X}(s-1) \cup \mathcal{X}^{new}(s)$; compute $\hat{\boldsymbol{\theta}}^c(s)$, $\{\alpha^c_{r,i}(s)\}$, $\{\zeta^c_{r,i,m}(s)\}$ using Alg. 1 with inputs $\mathcal{R}(s)$, $\mathcal{X}(s)$, $\hat{\boldsymbol{\theta}}^0(s)$.
    Build $\mathcal{Y}(s)$ using (20) in Section 4 and compute $\{\beta^0_m(s) : m = 1, \cdots, \hat{M}^c(s)\}$.
    Compute $\{\beta^d_m(s) : m = 1, \cdots, \hat{M}^c(s)\}$ using Alg. 2 with inputs $\mathcal{Y}(s)$, $\hat{M}^c(s)$ and $\{\beta^0_m(s) : m = 1, \cdots, \hat{M}^c(s)\}$.
end while
For $m = 1, \cdots, \hat{M}^c(s)$, decide whether cluster $m$ is a true label or not using $\beta^d_m(s) \geq \delta^d$.
```

instances are available at outer iteration s. Further, we denote by $R(s)$ the number of annotators who have provided instances until iteration s so that $R(s) = R(s-1) + R^{new}(s)$, where $R^{new}(s)$ is the number of *new* annotators at iteration s and $R(s-1)$ is the number of *old* annotators. Note that this notation is general enough to cover different cases: (a) the same set of annotators provides new instances, i.e., $R^{new}(s) = 0$ and $\mathcal{X}^{new}(s) \neq \emptyset$; (b) new annotators provide new instances, i.e., $R^{new}(s) > 0$ and $\mathcal{X}^{new}(s) \neq \emptyset$; or (c) both old and new annotators provide new instances.

Assuming that a total of $\mathcal{X}(s)$ instances from $R(s)$ annotators are available at outer iteration s, the initial value of parameters $\hat{\boldsymbol{\theta}}^0(s)$ of the OEM algorithm is computed as follows. Firstly, the initial number of Gaussian components of the clustering algorithm is set equal to the number of clusters identified in the previous stage, i.e., $\hat{M}^c(s-1)$, plus the number of new instances, i.e., $\hat{M}^0(s) = \hat{M}^c(s-1) + |\mathcal{X}^{new}_s|$. The mean vector of these new clusters is initialized to be equal to the new instances whereas the mean vector of the first $\hat{M}^c(s-1)$ Gaussian components is equal to the values obtained at the end of the clustering of the previous round, i.e.,

$$\hat{\boldsymbol{\mu}}^0_m(s) = \begin{cases} \hat{\boldsymbol{\mu}}^c_m(s-1) & m = 1, \cdots, \hat{M}^c(s-1) \\ \boldsymbol{x}^{new}_q(s) & m = \hat{M}^c(s-1) + 1, \cdots, \hat{M}^0(s) \end{cases} \tag{35}$$

Similarly, the covariance matrix of the first $\hat{M}^c(s-1)$ Gaussian components is equal to the values obtained at the end of the clustering of the previous outer iteration, and the covariance matrix of the new clusters is initialized to $\mathbf{\Sigma}^0$ as follows

$$\hat{\mathbf{\Sigma}}_m^0(s) = \begin{cases} \hat{\mathbf{\Sigma}}_m^c(s-1) & m = 1, \cdots, \hat{M}^c(s-1) \\ \mathbf{\Sigma}^0 & m = \hat{M}^c(s-1)+1, \cdots, \hat{M}^0(s) \end{cases} \tag{36}$$

Further, the probability of the new clusters is initialized to one-eighth of the minimum among $\{\hat{\pi}_m^c(s-1) : m = 1, \cdots, \hat{M}^c(s-1)\}$; afterwards, cluster probabilities at iteration s, i.e.,$\{\hat{\pi}_m^c(s) : m = 1, \cdots, \hat{M}^0(s)\}$, are normalized to sum up to 1. Finally, we assume that the reliability of new annotators is set to p^0, as follows

$$\hat{p}_r^0(s) = \begin{cases} \hat{p}_r^c(s-1) & r = 1, \cdots, R(s-1) \\ p^0 & r = R(s-1)+1, \cdots, R(s) \end{cases} \tag{37}$$

After $\hat{\boldsymbol{\theta}}^0(s)$ is obtained, the OEM algorithm in Alg. 1 is run. Note that the number of iterations until convergence of the clustering algorithm is expected to be much shorter for $s > 0$ than for $s = 0$, since most of the Gaussian components are already identified with good initialization of the mean vector. The outputs of OEM at iteration s are denoted by $\hat{\boldsymbol{\theta}}^c(s)$ for the parameter estimate, and by $\{\alpha_{r,i}^c(s) : r = 1, \cdots, R(s);$ and $i = 1, \cdots, N_r\}$ and $\{\zeta_{r,i,m}^c(s) : r = 1, \cdots, R(s); i = 1, \cdots, N_r;$ and $m = 1, \cdots, \hat{M}^c(s)\}$ for the posterior probabilities.

6.3. Online data processing after clustering

The intermediate data is processed similarly to Section 4 to generate the set $\mathcal{Y}(s) := \{y_{r,m}(s) : r = 1, \cdots, R(s);$ and $m = 1, \cdots, \hat{M}^c(s)\}$ with the individual binary labels given by the annotators to the clusters identified in the clustering stage. It is important to note that the number of identified clusters at outer iteration s, denoted by $\hat{M}^c(s)$, might be different from those identified at the previous iteration, $\hat{M}^c(s-1)$. Therefore, elements in $\mathcal{Y}(s)$ might be different to those in $\mathcal{Y}(s-1)$, not only because new annotators might come into play at iteration s, but also because binary tags of old annotators to the clusters identified at iteration s might have changed. Therefore, we need to build the set $\mathcal{Y}(s)$ from scratch following the procedure described in Section 4. That is, first the set of non-outlier instances is built as $\mathcal{X}^c(s) := \{\boldsymbol{x}_{r,i}(s) : r = 1, \cdots, R_s;$ and $i = 1, \cdots, N_r;$ such that $\alpha_{r,i}^c(s) \geq \delta^c\}$. Then, we assign each non-outlier instance to one of the clusters by computing

$$\mathcal{C}(\boldsymbol{x}_{r,i}) = \underset{m=1,\cdots,\hat{M}^c(s)}{\arg\max}\ \zeta_{r,i,m}^c(s)$$

for all $\boldsymbol{x}_{r,i}(s) \in \mathcal{X}^c(s)$. Finally, we build the set $\mathcal{Y}(s) := \{y_{r,m}(s) : r = 1, \cdots, R(s);$ and $m = 1, \cdots, \hat{M}^c(s)\}$ as follows

$$y_{r,m}(s) = \begin{cases} 1 & \text{if } \exists \boldsymbol{x}_{r,i}(s) \subset \mathcal{X}^c_r(s) \text{ s.t. } \mathcal{C}(\boldsymbol{x}_{r,i}(s)) = m \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

where $\mathcal{X}^c_r(s) := \{\boldsymbol{x}_{r,i}(s) : i = 1, \cdots, N_r;$ such that $\alpha^c_{r,i}(s) \geq \delta^c\}$.

6.4. Online detection and stopping

Finally, the DEM algorithm in Alg 2 is run. The posterior probabilities for all clusters identified in the clustering stage are initialized as follows

$$\beta^0_m(s) = \begin{cases} \beta^d_m(s-1) & m = 1, \cdots, \hat{M}^c(s-1) \\ \beta'_m & m = \hat{M}^c(s-1) + 1, \cdots, \hat{M}^c(s) \end{cases} \tag{39}$$

where β'_m indicates one of the three initialization options (32), (33) or (34) presented in Section 5.1. Hence, posterior probabilities of the clusters identified in the previous round $(s-1)$ remain the same, and the posterior probabilities of the new identified clusters, if any, are initialized as explained in Section 5.1. Note that if $\hat{M}^c(s) < \hat{M}^c(s-1)$, it is not necessary to compute the posterior probability for the annihilated clusters. The output of the DEM algorithm at outer iteration s is given in (31) by the posterior probabilities $\{\beta^d_m(s) : m = 1, \cdots, \hat{M}^c(s)\}$.

At this point, a hard decision is taken to decide the true labels by $\beta^d_m(s) \geq \delta^d$, where $0 < \delta^d < 1$. The online algorithm is summarized in Alg. 3 assuming S outer iterations. In a practical implementation, however, the online algorithm might be stopped when this hard decision does not change throughout several consecutive outer iterations.

7. Experimental results with real data

In this section, results of the proposed approach for 10 digitized images tagged by volunteers through the MalariaSpot platform [11] are presented. These digitized smears, referred to hereafter as Image 1 to Image 10, are from the Health Investigation Centre of Manhiça in Mozambique. For the acquisition of the images, Image 1 to Image 5 were taken with a conventional light microscope (Zeiss, model AX05COP2) attached to a Nokia Xperia Z2 cellphone using a market plastic adapter that aligns the cellphone camera to the ocular lens of the microscope. Image 6 to Image 10 were taken using the standard technology for a clinical image using a camera mounted on the microscope. Figure 3 and Figure 4 show Image 3 and Image 10, respectively. It is important to remark that the use of mobile phones to capture smears is a very appealing technology for working in the field, specially in countries with limited resources. However, the quality of the image is worse compared to that using the standard technology, a

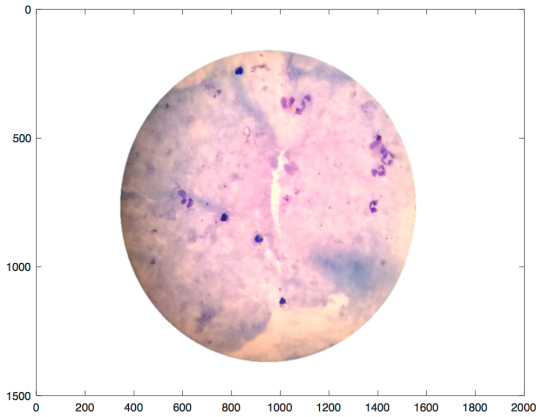

Figure 3: Image 3 taken with a microscope attached to a cellphone camera.

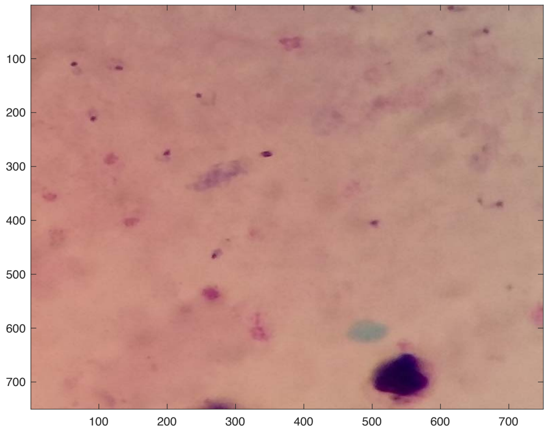

Figure 4: Image 10 taken with standard technology.

fact that adds an extra challenge but also interest to our work. All digitized smears have been analyzed by non-expert volunteers and the ground truth has been identified by experts. Figure 5 and Figure 6 show the instances provided by $R = 25$ annotators selected at random from the data set for Image 3 and Image 10, respectively, and the ground truth. As in Section 3.1, results are evaluated in terms of sensitivity, i.e., the fraction of ground truth that is identified as parasites and denoted by S, and precision, i.e., the fraction of potential parasites that are positive and denoted by P. Sensitivity and precision are both computed after clustering and detection stages as $S = \frac{TP}{N_p}$ and $P = \frac{TP}{TP+FP}$, where TP, FP, TN and FN denote true/false positives/negatives, respectively; and N_p denotes the number of true parasites. Supraindex c denotes sensitivity and precision computed after the clustering stage, and supraindex d after the detection stage. For instance, S^c is sensitivity after clustering and P^d precision after detection. Unless otherwise stated, results are given averaging a total of 300 independent realizations

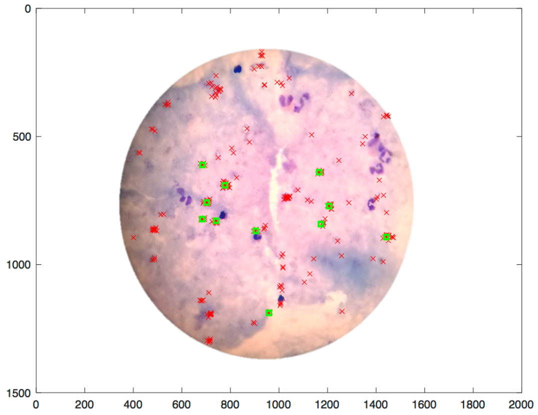

Figure 5: Instances provided by $R = 25$ annotators ($\times$) and ground truth ($\square$) for Image 3.

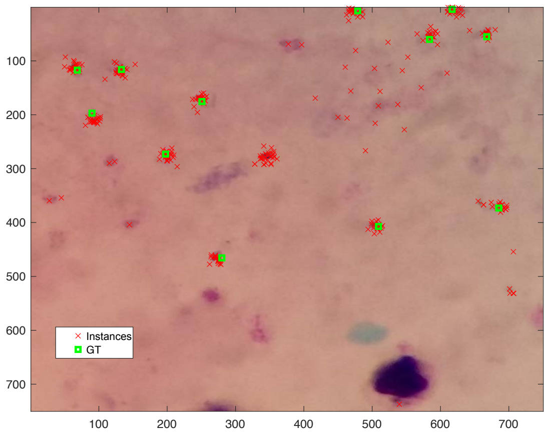

Figure 6: Instances provided by $R = 25$ annotators ($\times$) and ground truth ($\square$) for Image 10.

selecting datum from the MalariaSpot dataset where each image was tagged more than $5,000$ times by non-expert volunteers. The following sections show results after the clustering stage and after the detection stage. Afterwards, we show results obtained with the online algorithm of Section 6.

7.1. Results after clustering

In this section, we show results of S^c and P^c after the clustering stage using OEM, CEM, k-means and HAC algorithms and with $R = \{11, 21, 25, 31, 41, 51\}$. The initialization of parameters of OEM and CEM is exactly the same as in Section 3.1. The initial estimated means $\{\hat{\boldsymbol{\mu}}_m^0; \forall m = 1, \cdots, \hat{M}^0\}$ are the centroids obtained by the k-means algorithm with $\hat{M}^0$ equal to 6 times the average number of clicks per annotator, i.e., around 120 in our setup; the initial estimated Gaussian covariance matrices are all set to $\{\hat{\boldsymbol{\Sigma}}_m^0 = \boldsymbol{\Sigma}^0 := \frac{\sigma_x^2}{200}\mathbf{I}; \forall m = 1, \ldots, \hat{M}^0\}$, where σ_x^2 is the sample variance of the instances. Probabilities are initialized as $\hat{\pi}_m^0 = 1/\hat{M}^0$

for all m, and $p_r = p^0 := 0.9$ for all r. OEM and CEM are executed a minimum of 10 iterations until $\hat{M}^t = 1$ or a maximum of 500 iterations.

Figure 7 shows results for Images 1–5 with a cellphone camera (in red, blue, magenta, cyan and green, respectively) of sensitivity (in $\circ$, $\diamond$, $\square$, $\triangleleft$ and $\triangleright$, respectively for each image) and precision (in $*$, $+$, $\times$, $\star$ and $\bullet$, respectively) obtained after clustering with OEM (solid line) and CEM (dotted line). Similarly, Figure 8 shows sensitivity and precision for Images 6–10 with standard technology for OEM and CEM. For the sake of comparison, Figure 9 and Figure 10 show precision and sensitivity after clustering with k-means (solid line) and HAC (dotted line) for the two sets of images, respectively.

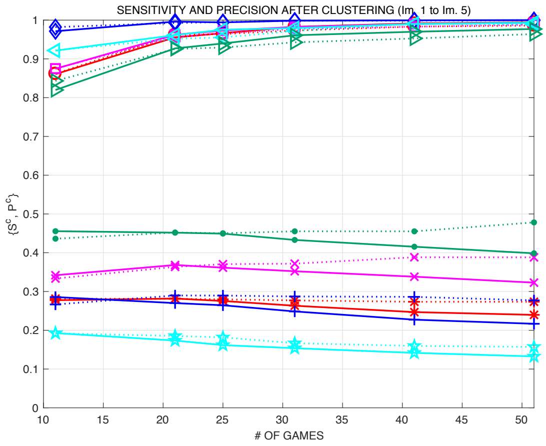

Figure 7: Sensitivity and precision after the clustering stage with OEM (solid) and CEM (dotted) for images taken with a cellphone camera. Im. 1 (in red, S^c $\circ$ and P^c $*$), Image 2 (in blue, S^c $\diamond$ and P^c $+$), Image 3 (in magenta, S^c $\square$ and P^c $\times$), Image 4 (in cyan, S^c $\triangleleft$ and P^c $\star$), and Image 5 (in green, S^c $\triangleright$ and P^c $\bullet$).

As observed, OEM and CEM perform similarly and better than k-means and HAC. Still, in Figure 7, OEM provides overall better sensitivity results and slightly worse precision. In Figure 8, both methods achieve very similar sensitivity (except in Image 9 where OEM is better) and OEM achieves overall better precision. Therefore, and since at the clustering stage it is convenient to prioritize high sensitivity, we may conclude that OEM outperforms CEM with these real datum.

If we compare sensitivity and precision after the clustering stage between the images taken with a cellphone camera (Figure 7, Images 1-5) and the ones taken with standard technology (Figure 8, Images 6-10), we may conclude that the results of images with standard technology are overall better since both sensitivity and precision are higher. For instance, precision with datum obtained from images from a cellphone camera take values between 0.1 and below 0.5, whereas precision with data from images using standard technology increases to the range of $[0.4, 0.6]$ and Image 9 is even higher than 0.9 with OEM. Notably, sensitivity with data from

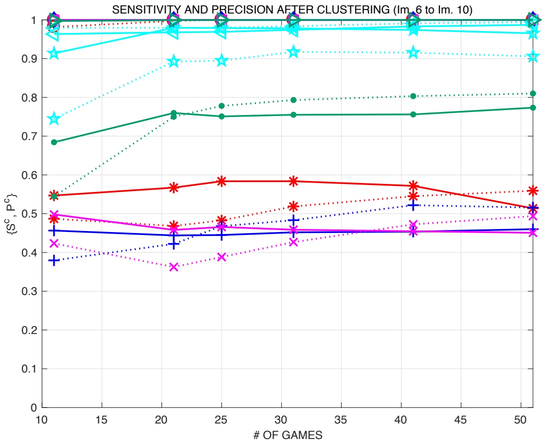

Figure 8: Sensitivity and precision after the clustering stage with OEM (solid) and CEM (dotted) for images taken with standard technology. Image 6 (in red, S^c $\circ$ and P^c $*$), Image 7 (in blue, S^c $\diamond$ and P^c $+$), Image 8 (in magenta, S^c $\square$ and P^c $\times$), Image 9 (in cyan, S^c $\triangleleft$ and P^c $\star$), and Image 10 (in green, S^c $\triangleright$ and P^c $\bullet$).

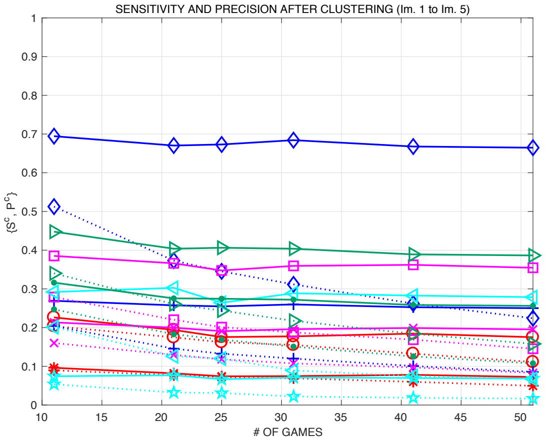

Figure 9: Sensitivity and precision after the clustering stage with k-means (solid) and HAC (dotted) for images taken with a cellphone camera. Image 1 (in red, S^c $\circ$ and P^c $*$), Image 2 (in blue, S^c $\diamond$ and P^c $+$), Image 3 (in magenta, S^c $\square$ and P^c $\times$), Image 4 (in cyan, S^c $\triangleleft$ and P^c $\star$), and Image 5 (in green, S^c $\triangleright$ and P^c $\bullet$).

images using standard technology is very close to 1 for all images. Further, we observe that clustering of clicks for images from a cellphone camera needs at least $R = 25$ games to reach stable sensitivity values although precision does not improve, whereas for images taken with standard technology a lower value of games is needed, i.e., $R = 11$, is sufficient.

435 *7.2. Results after detection*

In this section, we present sensitivity and precision results after the detection stage using the same images as in Section 7.1. Figure 11 shows sensitivity and precision after the detection stage for Images 1–5 taken with a cellphone, and Figure 12 for Images 6–10 taken using standard

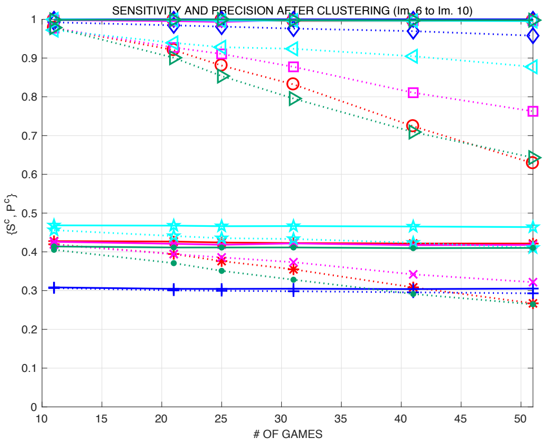

Figure 10: Sensitivity and precision after the clustering stage with k-means (solid) and HAC (dotted) for images taken with standard technology. Image 6 (in red, S^c $\circ$ and P^c $\ast$), Image 7 (in blue, S^c $\diamond$ and P^c $+$), Image 8 (in magenta, S^c $\square$ and P^c $\times$), Image 9 (in cyan, S^c $\triangleleft$ and P^c $\star$), and Image 10 (in green, S^c $\triangleright$ and P^c $\bullet$).

technology, both using clustering results obtained with the OEM algorithm. Therefore, in these figures, solid lines are the results obtained with the detection EM (DEM) algorithm proposed in Section 5, and dashed lines are results obtained with Majority Voting (MV), which is a straightforward procedure for the detection stage.

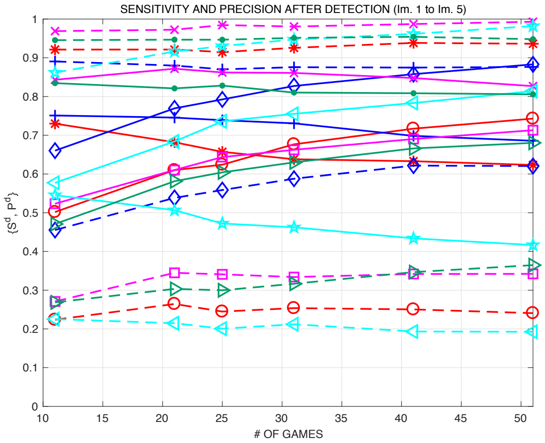

Figure 11: Sensitivity and precision after the detection stage with DEM (solid) and Majority Voting (dashed) for images taken with a cellphone camera. Image 1 (in red, S^d $\circ$ and P^d $\ast$), Image 2 (in blue, S^d $\diamond$ and P^d $+$), Image 3 (in magenta, S^d $\square$ and P^d $\times$), Image 4 (in cyan, S^d $\triangleleft$ and P^d $\star$), and Image 5 (in green, S^d $\triangleright$ and P^d $\bullet$). Clustering with OEM.

Interestingly, for each realization, we run DEM as in Alg. 2 twice initialized with different posterior probabilities using (32) and (33). The one with the highest final value of the objective function $Q_d(\tilde{\phi}; \hat{\phi}^{k+1})$ is kept to make the decision on each potential parasite, i.e., we decide

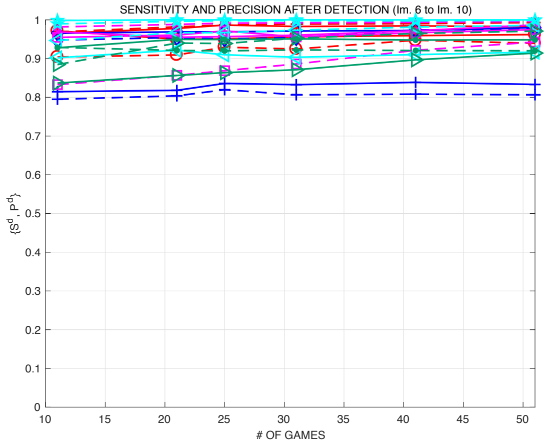

Figure 12: Sensitivity and precision after the detection stage with DEM (solid) and Majority Voting (dashed) for images taken with standard technology. Image 6 (in red, S^d $\circ$ and P^d $*$), Image 7 (in blue, S^d $\diamond$ and P^d $+$), Image 8 (in magenta, S^d $\square$ and P^d $\times$), Image 9 (in cyan, S^d $\triangleleft$ and P^d $\star$), and Image 10 (in green, S^d $\triangleright$ and P^d $\bullet$). Clustering with OEM.

cluster m is positive if $\beta_m^d > \delta^d := 0.5$ and negative otherwise. This approach provides the best results for these real data compared to using only one of the initializations given in (34)-(33). Further, the overall computational cost is not significant because convergence of DEM is very fast; usually just $10 - 15$ iterations are required.

In Figure 11, we observe that sensitivity with MV decreases to the range of $[0.2, 0.4]$, except for Image 2 to the range of $[0.45, 0.65]$. Instead, DEM is capable of keeping sensitivity higher within the range of $[0.5, 0.75]$ and up to 0.9 for Image 2. Conversely, precision is higher with MV than with DEM. Regarding Figure 12, the detection stage both with DEM and with MV increases performance, that is precision is significantly higher than after clustering without sacrificing sensitivity.

For the purpose of comparison, Figure 13 shows sensitivity and precision after the detection stage for Images 1–5 taken with a cellphone, and Figure 14 for Images 6–10 taken using standard technology, both using clustering results obtained with the CEM algorithm. No significant differences are observed compared to the results obtained clustering with OEM shown in Figure 11 and Figure 12. A measure that takes into account the trade-off between sensitivity and precision is the balanced F$_\beta$-score defined as

$$F_\beta = (1 + \beta^2)\frac{S \cdot P}{S + \beta^2 \cdot P}, \tag{40}$$

such that the closer to one the better. Typical values for β are 0.5, 1 and 2; we select the value of $\beta = 2$ to penalize low sensitivity values. Table 1 lists values of the F$_2$-score measurement for all images using $R = 31$ games after the clustering stage with OEM or CEM, and after the detection stage with MV and with DEM. For comparison purposes, results achieved with

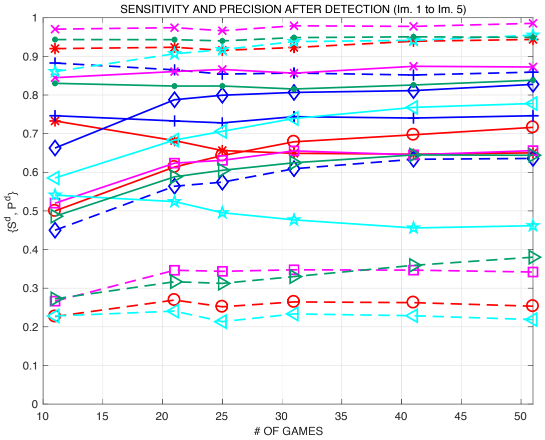

Figure 13: Sensitivity and precision after the detection stage with DEM (solid) and Majority Voting (dashed) for images taken with a cellphone camera. Image 1 (in red, S^d $\circ$ and P^d $*$), Image 2 (in blue, S^d $\diamond$ and P^d $+$), Image 3 (in magenta, S^d $\square$ and P^d $\times$), Image 4 (in cyan, S^d $\triangleleft$ and P^d $\star$), and Image 5 (in green, S^d $\triangleright$ and P^d $\bullet$). Clustering with CEM.

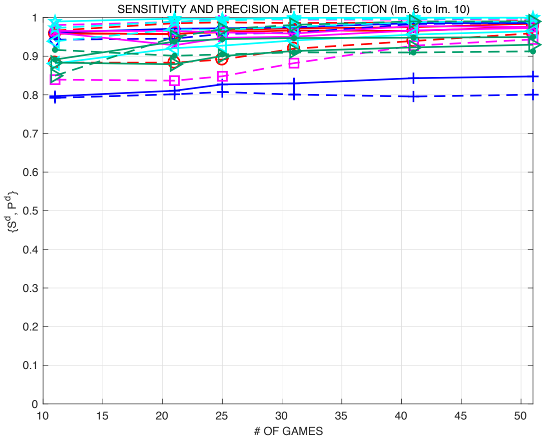

Figure 14: Sensitivity and precision after the detection stage with DEM (solid) and Majority Voting (dashed) for images taken with standard technology. Image 6 (in red, S^d $\circ$ and P^d $*$), Image 7 (in blue, S^d $\diamond$ and P^d $+$), Image 8 (in magenta, S^d $\square$ and P^d $\times$), Image 9 (in cyan, S^d $\triangleleft$ and P^d $\star$), and Image 10 (in green, S^d $\triangleright$ and P^d $\bullet$). Clustering with CEM.

clustering with k-means, and detection DEM are also included.

As can be observed, in general, F_2-score values increase after the detection stage. Regarding the first set of Images 1–5 taken with a cellphone camera, the proposed approach of OEM-DEM provides higher values of the F_2-score for all images except for Image 4 and Image 1. Regarding the second set of Images 6–10 taken using standard technology, the three methods provide similar acceptable results but the combination that works better is k-means for clustering and DEM for detection. Therefore, it may be concluded that the proposed approach of OEM for

Clustering	OEM	CEM	OEM	CEM	OEM	CEM	KM
Detection	–	–	MV	MV	DEM	DEM	DEM
Im. 1	0.6196	0.6267	0.2915	0.3033	0.6558	**0.6586**	0.572
Im. 2	0.6144	0.6501	0.625	0.6418	**0.7953**	0.7840	0.7935
Im. 3	0.7096	0.7186	0.381	0.3952	**0.6855**	0.6781	0.5702
Im. 4	0.4472	0.4551	0.242	0.2658	0.6343	0.6261	**0.6663**
Im. 5	0.7617	0.7588	0.3614	0.3754	**0.6546**	0.6506	0.5812
Im. 6	0.849	0.8083	0.9335	0.9278	0.9587	0.9651	**0.9921**
Im. 7	0.785	0.8002	0.9167	0.9199	0.9373	0.9391	**0.9398**
Im. 8	0.7753	0.7517	0.9011	0.8980	0.952	0.9519	**0.9967**
Im. 9	0.9742	0.9674	0.9778	**0.9802**	0.9202	0.9510	0.9443
Im. 10	0.9284	0.9466	0.9447	0.9652	0.8828	0.9175	**0.9658**

Table 1: F$_2$-score with $R = 31$ games computed after clustering with OEM and with CEM; after detection with Majority Voting and with DEM; and after detection with DEM and clustering with k-means (KM).

clustering and DEM for detection shows a good performance with both types of images, and significantly better results for images of lower quality taken with the cellphone camera. These results are promising because the proposed approach is well suited to process tags provided by annotators on images of worse quality but taken with low-cost technology available to many more people worldwide.

7.3. Results with online algorithm

This section includes results of the online algorithm presented in Section 6 and summarized in Alg. 3. Results for Image 3 and Image 10 are shown. Figure 15 and Figure 16 plot sensitivity and precision as a function of $\mathcal{R}(s)$ after clustering and after detection, respectively. A solid line is used for batch results and a dashed-dotted line for online results. In this case, results are obtained averaging 100 independent realizations. Regarding the online algorithm, the number of annotators starts with $R(0) = 11$ and increases in steps of 1 until 51, i.e., $R(s) \in [11, 51]$. The initial values for $s = 0$ are selected as in Section 7.1 using $\mathcal{R}(0)$ and $\mathcal{X}(0)$. That is, $\hat{M}^0(0)$ is equal to 6 times the average number of clicks per annotator; $\{\hat{\mathbf{\Sigma}}^0_m(0) = \mathbf{\Sigma}^0 = \frac{\sigma_x^2}{200}\mathbf{I}; \forall m = 1, \ldots, \hat{M}^0\}$ where σ_x^2 is the average of the variance of the instances; probabilities are initialized as $\hat{\pi}^0_m(0) = 1/\hat{M}^0(0)$ for all m; and $p_r(0) = p^0 = 0.9$ for all r. Threshold parameters are set equal to $\delta^c = 0.5$ and $\delta^d = 0.5$.

Initialization of $\{\beta^0_m; \forall m = 0, \cdots, \hat{M}^c(s)\}$ at each outer iteration is different for $s = 0$ and $s > 0$. At $s = 0$, we proceed as for the batch DEM (that is, Alg. 2 is run twice initialized

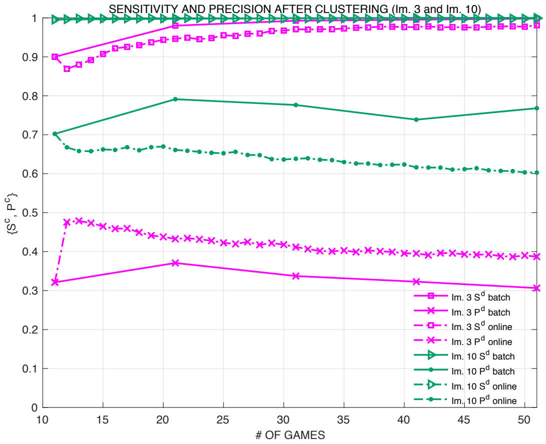

Figure 15: Sensitivity and precision after the clustering stage with the online algorithm (dashed-dotted) and batch (solid)) for Image 3 (in magenta, S^d □ and P^d ×) and Image 10 (in green, S^d ▷ and P^d ●).

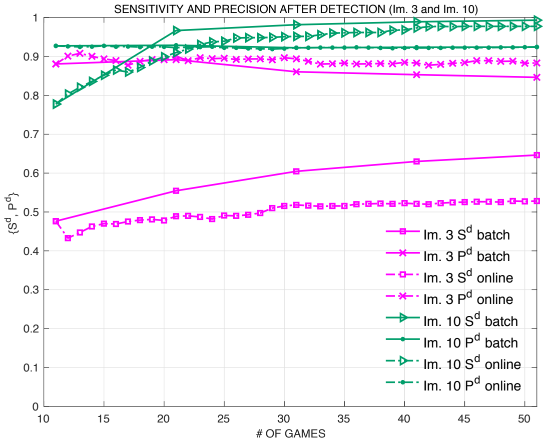

Figure 16: Sensitivity and precision after the detection stage with the online algorithm (dashed-dotted) and batch (solid)) for Image 3 (in magenta, S^d □ and P^d ×) and Image 10 (in green, S^d ▷ and P^d ●).

with (32), and (33); the one with the highest final value of $Q_d(\tilde{\boldsymbol{\phi}}; \hat{\boldsymbol{\phi}}^{k+1})$ is kept). For $s > 0$, $\{\beta_m^0; \forall m = 0, \cdots, \hat{M}^c(s)\}$ is computed with the initialization (32) or (33) chosen at $s = 0$.

As can be seen, sensitivity and precision obtained with the online implementation follow the path of the batch implementation.

8. Conclusions

An unsupervised approach to detect specific structures in an image tagged by non-expert annotators in a crowdsourcing application has been presented. The procedure consists of two stages, namely a clustering stage followed by a detection stage, both based on the EM algorithm. The method is robust to unreliable annotators thanks to the density mixture model that accounts for outliers, and it gracefully combines their responses in a blind manner. Further, a novel online implementation of the method is presented that is suited to crowdsourced applications in which data are available in a streaming manner. Comprehensive experimental results with real data of the MalariaSpot project, in which annotators are asked to identify parasites in thick blood smears, are included to illustrate and support both the batch and the online approach. Good results are obtained not only with high quality images taken with an expensive microscope, but also with images taken with low-cost technology that attaches a cheaper microscope to a cellphone camera. Even though annotators are more error-prone due to the lower quality of the images, the approach still provides acceptable results. Therefore, worldwide Malaria diagnosis may benefit from the presented procedure since it makes the MalariaSpot platform more accessible to countries and organizations with scarce resources.

Acknowledgements

This work was supported by the Ministerio de Economía y Competitividad of the Spanish Government and ERDF funds (TEC2016-77148-C2-1-R, TEC2016-75067-C4-2-R,TEC2015-69648-REDC), and Catalan Government funds (2017 SGR 578 AGAUR).

We would like to express our very great appreciation to Miguel Luengo-Oroz, Maria Postigo-Camps, Daniel Cuadrado-Sanchez and the rest of the MalariaSpot team for early discussions and sharing the crowdsourcing data from MalariaSpot players and the malaria images analyzed in the experiments.

Appendix A. Illustrative example of the two-stage procedure

This appendix illustrates the two-stage approach proposed in this paper using data and images of the MalariaSpot project but without including algorithmic details of the clustering and the detection stages, which are presented in Sections 3–5. Remarkably, note that the approach is general enough to be used not only with any crowdsourced data in which annotators are asked to identify specific structures in images, but also with similar data provided instead by different automated techniques with unknown reliability.

The MalariaSpot project offers digitized images of thick blood samples through an on-line game to volunteers who, after a short training period, identify malaria parasites in the images. For further details about this project, visit [1]. Figure A.17 includes two different examples of such images. The left image is taken with a conventional light microscope (Zeiss, model AX05COP2) attached to a Nokia Xperia Z2 cellphone using a market plastic adapter that aligns the cellphone camera to the ocular lens of the microscope; the right image is taken with the standard technology for a clinical image using a camera mounted on the microscope.

During the game, players tag wherever they spot a malaria parasite in the image. As an example, Figure A.18 and Figure A.19 show the tags of 51 players or annotators with red × in the two images, respectively. For these images, the ground truth identified by experts with green □ is also included for testing purposes. As observed, players make mistakes wherever tags do not coincide with the ground truth. Some of the errors are isolated randomly located tags, meaning that very few players erroneously identified a parasite there, while other errors are tagged by several players. In order to circumvent these erroneous tags, the procedure to identify the true parasites given the tags of all annotators consists of a clustering stage followed by a detection stage. For the clustering, tags are modeled as instances of a density mixture model of an unknown number of Gaussians plus a uniform r.v., which models the isolated tags. Using this density mixture model, the data is clustered using an EM-based algorithm so that a number of clusters and their corresponding centroids are obtained after the clustering stage. Besides, the algorithm also rates annotators according to their performance. Figure

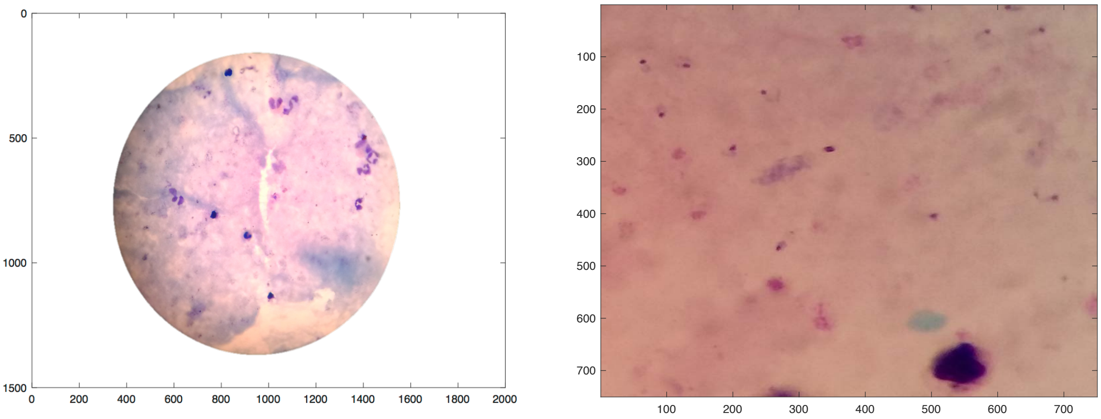

Figure A.17: Digitized Images of blood samples taken with different technologies.

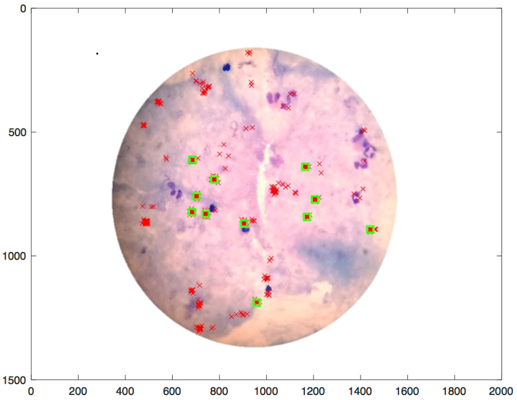

Figure A.18: Tags provided by 51 annotators (×) and ground truth (□).

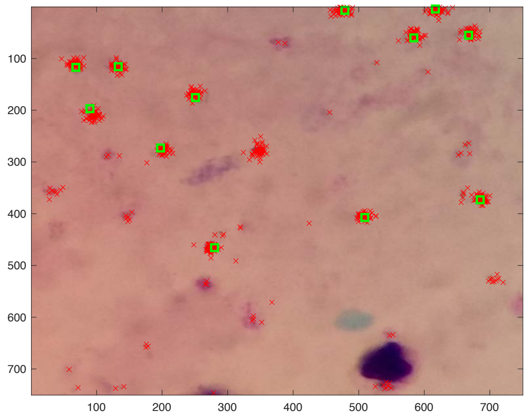

Figure A.19: Tags provided by 51 annotators (×) and ground truth (□).

A.20 and Figure A.21 show the centroids of the clusters identified after the clustering stage for both images, respectively. As observed, the clustering stage performs differently depending on the image. Annotators tend to make more mistakes with the image in Figure A.18, which is taken with less advanced technology, than with the image in Figure A.19. Therefore, the clustering identifies more clusters in the image in Figure A.20 than in the image in Figure A.21. Clearly, in Figure A.21, artifacts erroneously tagged by a significant number of players remain as an additional cluster whereas isolated errors do not affect the clustering. Therefore, the detection stage is responsible for assessing whether a cluster corresponds to a true parasite or not. A different EM-based technique is also used for the detection stage so that both the number of annotators who tag a particular cluster and their reliability are taken into account for the decision. Figure A.22 and Figure A.23 show the centroids detected as parasites after

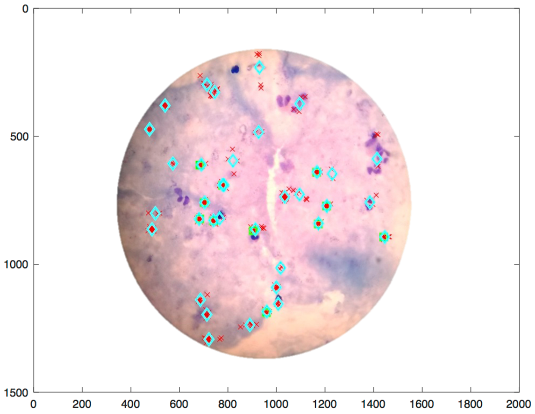

Figure A.20: Tags provided by 51 annotators (×), centroids of the identified clusters after the clustering stage (⋄), and ground truth (□)

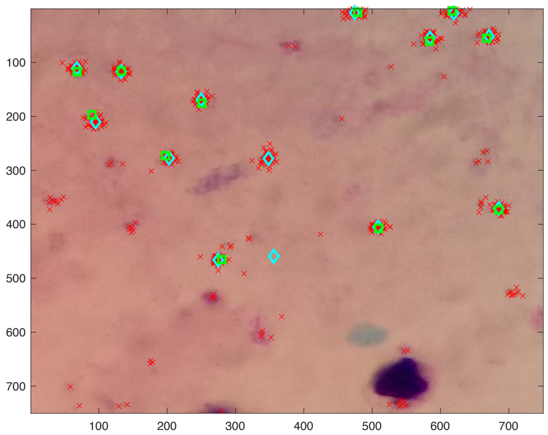

Figure A.21: Tags provided by 51 annotators (×), centroids of the identified clusters after the clustering stage (⋄), and ground truth (□)

the detection stage in blue (+) for images in Figure A.18 and Figure A.19, respectively.

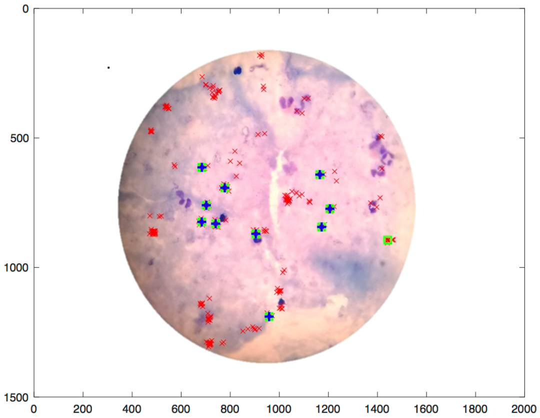

Figure A.22: Tags provided by 51 annotators (×), centroids of the identified clusters after the detection stage (+), and ground truth (□)

In this particular example, precision and sensitivity after detection with the image in Figure 5 560 are $P^d = 0.9091$ and $S^d = 0.9091$, and with the image in Figure 6 are $P^d = 1$ and $S^d = 1$.

References

[1] http://malariaspot.org, 2012.

[2] S. Nag, N. Basu, S. Bandyopadhyay, Different Methods for Diagnosing Malaria Disease, IJCMPR 2 (1) (2016) 197–201.

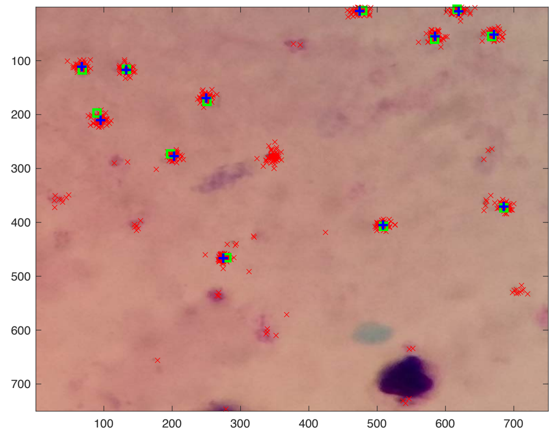

Figure A.23: Tags provided by 51 annotators (×), centroids of the identified clusters after the detection stage (+), and ground truth (□)

[3] Y. Purwar, S. L. Shah, G. Clarke, A. Almugairi, A. Muehlenbachs, Automated and unsupervised detection of malarial parasites in microscopic images, Malaria journal 10 (1) (2011) 1–10.

[4] S. Savkare, S. Narote, Automatic detection of malaria parasites for estimating parasitemia, International Journal of Computer Science and Security (IJCSS) 5 (3) (2011) 310.

[5] P. Suradkar, Detection of malarial parasite in blood using image processing, International Journal of Engineering and Innovative Technology (IJEIT) 2 (10).

[6] S. Raviraja, Geethanjali, Chethana, Kanthesh, The Classification and Recognition of Plasmodium Parasite.., IJARCSSE 5 (7) (2015) 863–886.

[7] M. I. Razzak, Malarial parasite classification using recurrent neural network, International Journal of Image Processing (IJIP) 9 (2) (2015) 69.

[8] S. Kaewkamnerd, C. Uthaipibull, A. Intarapanich, M. Pannarut, S. Chaotheing, S. Tongsima, An automatic device for detection and classification of malaria parasite species in thick blood film, Bmc Bioinformatics 13 (17) (2012) 1.

[9] M. Elter, E. Haslmeyer, T. Zerfas, Detection of malaria parasites in thick blood films, in: Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, IEEE, 5140–5144, 2011.

[10] L. Rosado, J. M. C. da Costa, D. Elias, J. S. Cardoso, Automated detection of malaria parasites on thick blood smears via mobile devices, Procedia Computer Science 90 (2016) 138–144.

[11] M. A. Luengo-Oroz, A. Arranz, J. Frean, Crowdsourcing Malaria Parasite Quantification: An Online Game for Analyzing Images of Infected Thick Blood Smears, J Med Internet Res 14 (6) (2012) e167.

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

[13] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the royal statistical society. Series B (methodological) (1977) 1–38.

[14] M.-S. Yang, C.-Y. Lai, C.-Y. Lin, A robust EM clustering algorithm for Gaussian mixture models, Pattern Recognition 45 (11) (2012) 3950 – 3961.

[15] Z. Zhang, C. Chen, J. Sun, K. L. Chan, EM algorithms for Gaussian mixtures with split-and-merge operation, Pattern Recognition 36 (9) (2003) 1973 – 1983.

[16] J. N. Myhre, K. Ø. Mikalsen, S. Løkse, R. Jenssen, Robust clustering using a kNN mode seeking ensemble, Pattern Recognition 76 (2018) 491 – 505.

[17] F. de Morsier, D. Tuia, M. Borgeaud, V. Gass, J.-P. Thiran, Cluster validity measure and merging system for hierarchical clustering considering outliers, Pattern Recognition 48 (4) (2015) 1478 – 1489.

[18] A. P. Dawid, A. M. Skene, Maximum likelihood estimation of observer error-rates using the EM algorithm, Applied statistics (1979) 20–28.

[19] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, Journal of Machine Learning Research 11 (Apr) (2010) 1297–1322.

[20] A. Pagès-Zamora, G. B. Giannakis, R. López-Valcarce, P. Giménez-Febrer, Robust clustering of data collected via crowdsourcing, in: 2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 4014–4018, 2017.

[21] D. M. Titterington, Recursive parameter estimation using incomplete data, Journal of the Royal Statistical Society. Series B (Methodological) (1984) 257–267.

[22] O. Cappé, E. Moulines, On-line expectation–maximization algorithm for latent data models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71 (3) (2009) 593–613.

[23] F. Saki, N. Kehtarnavaz, Online frame-based clustering with unknown number of clusters, Pattern Recognition 57 (2016) 70 – 83.

[24] M. Cabrera-Bean, A. Pagès-Zamora, C. Díaz-Vilor, M. Postigo-Camps, D. Cuadrado-Sánchez, M. A. Luengo-Oroz, Counting malaria parasites with a two-stage EM based algorithm using crowsourced data, in: 2017 39th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC), 2283–2287, 2017.

[25] C. Díaz-Vilor, EM based algorithms for Malaria diagnose via crowdsourcing, Bachelor's Thesis, Universitat Politècnica de Catalunya, Spain, 2017.

[26] G. McLachlan, D. Peel, Finite mixture models, John Wiley & Sons, 2004.

[27] M. A. T. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on pattern analysis and machine intelligence 24 (3) (2002) 381–396.

[28] S. Lloyd, Least-squares quantization in PCM, IEEE Trans. on Information Theory 28 (2) (1982) 129–137.

[29] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.