Scalable Expert Selection when Learning from Noisy Labelers

Chirine Wolley
Aix-Marseille University, CNRS, LSIS UMR 7296
13397, Marseille, France
Email: chirine.wolley@etu.univ-amu.fr

Mohamed Quafafou
Aix-Marseille University, CNRS, LSIS UMR 7296
13397, Marseille, France
Email: mohamed.quafafou@univ-amu.fr

***Abstract*—With the advent of crowdsourcing services, it has become easy and fast to get a dataset labeled by multiple annotators. In a supervised learning context, various methods have been proposed to learning from different labelers. However, very recently, the problem has shifted towards ranking and filtering low-quality annotators, and estimating the consensus labels based only on the remaining experts, i.e, annotators that provide high-quality annotations. In this paper, we propose a novel approach to tackle this issue. Our solution is based on a probabilistic method where a combination of two metrics, a probabilistic score and an entropy measure, are integrated in order to iteratively select the experts and estimate the labels based only on the selected annotators. Extensive experiments on a broad range of datasets validate the effectiveness of our ExpertS algorithm in terms of quality and rapidity, compared to other previous efficient algorithms.**

I. Introduction

Crowdsourcing has become an important process in machine learning, as it is rapidly changing the way datasets are generated. A few years ago, labeling a dataset could be time-consuming, expensive, and even sometimes impossible. With the advent of crowdsourced services such as Amazon Mechanical Turk, it is possible to assign labels to hundreds, even thousands of annotators, and get the results in a couple of hours: labeling huge datasets from multiple annotators has become easy and fast. However, combining knowledge from different sources does not guarantee a good labeling. Indeed, a major drawback of crowdsourced services is that we do not have control over the quality of each labeler. Some annotators may provide random or bad quality labels in the hope they will go unnoticed and still be paid, while others may have good intentions and label the dataset given seriously. In the domain of supervised learning, the use of crowdsourced labels has created a number of interesting problems that has received increasing attention over the past years. For instance, how to adapt conventional supervised learning algorithm when we have multiple annotations instead of the ground truth? how to evaluate the system? how reliable and trustworthy is each annotator?

Recently, several approaches have been proposed for this new setting, including [1], [2], [3]. The common target of this family of work is both to learn a good classifier and to evaluate the trustworthiness of each annotator. Although various successful techniques have been proposed to solve the problem of learning from multiple annotators, a significant obstacle remains: how can we make the best use of the labeling information provided to approximate the hidden true? Since we do not have control over the quality of the annotators, very often the annotations are dominated by spammers. In this paper, we use the term spammer for a low quality annotator, who assigns labels randomly, without looking at the dataset given [4]. On the contrary, an expert is an annotator who provides the correct label for almost all the instances presented.

Spammers can significantly increase the cost of acquiring labels and can degrade the quality of the generated classifier. Therefore, a mechanism that eliminates spammers and keeps only experts is clearly desirable in order to improve the generated model. We here present a novel probabilistic approach to learning from multiple annotators, when the annotations are dominated by spammers. This situation has been previously studied in [4] and [5]. Our ExpertS algorithm, inspired from the previous efficient method of Raykar et al. [2], iteratively evaluates annotators, keeps only experts, and re-estimates the labels based only on the information obtained from the good annotators. The particularity of our ExpertS algorithm is the combination of two metrics in order to select the experts among all the annotators: the entropy measure and a spammer score defined later in the paper. Experimental results show that using this combination to eliminate spammers is clearly easier and faster than other efficient algorithms, without degrading the quality of the learned model.

We organize the rest of the paper as follows. The next section reviews related work in the literature. Section 3 explains in detail the proposed framework, followed by the experimental results in section 4. Finally, Section 5 presents our conclusions.

II. Related Work

The ease with which data can be organised and shared by a large number of labelers using crowdsourced services has created a large number of interesting problems in various areas. In the biostatistics community, authors in [6] studied the error rate estimation problem when conflicted responses are given to various medical questions. In the machine learning community, authors in [2], [3] propose an EM-based algorithm that jointly learns the classifier and the ground truth. However, one drawback with crowdsourcing is that we do not have tight

control over the quality of the annotators. The performance of the annotators can vary widely; they can be experts, novices, biased, malicious, or spammers. Therefore, it is crucial to develop an algorithm that evaluates, ranks and selects the good annotators in order to generate a good classifier.

More recently, the problem has shifted to how to make the best use of the labeling information provided by multiple annotators. An online approach developed in [7], finds and prioritizes experts when requesting labels, and actively excludes unreliable annotators. In [4], an empirical Bayesian algorithm called SpEM is proposed. It iteratively eliminates the spammers and estimates the consensus labels based only on good annotators. The common strategy of all the above works is to consider at the beginning all the set of available annotators as experts, and to eliminate the spammers iteratively by estimating each time their performance. However, with the use of crowdsourced services, there could be hundreds, even thousands of available annotators, most of them being spammers, and estimating the performance of each labeler can be time-consuming. We believe this last element is a major drawback of the previous algorithms developed, and we propose in this paper a novel approach, called ExpertS, to selecting experts in a context of learning from multiple annotators. Unlike [7], [4], our algorithm, based on the work of Raykar et Al. [2], integrates two metrics, the entropy measure and the spammer score, in order to select experts. The use of the entropy measure ensures an easy and direct way to significantly reduce the dataset with not significant costs. Therefore, combined with the spammer score introduced in [4], we obtain our ExpertS method, which is a simple and direct approach for expert selection in the context of learning from multiple annotators.

III. Method

In this section, we review the Baseline method [2], and we introduce the entropy measure and the spammer score. Based on these works, we propose a new algorithm named ExpertS, which filters the low-quality annotators and estimates the ground truth based only on the good annotators.

A. Baseline Probabilistic Model

Let N be the number of instances x_i and T the number of annotators. Let $\mathcal{D} = \left\{\left(y_i^1, ..., y_i^T\right)\right\}_{i=1}^N$ be the observed annotations from the T annotators, and z_i the corresponding unknown true label of each instance x_i. x_i is a d-dimensional vector and $z_i \in \{0, 1\}$ in binary classification. Let $p = Pr[z_i = 1]$ be the prevalence of the positive class. We define the matrices $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$[1], $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ and $Z = [z_1, ..., z_N]^T$. The Baseline algorithm models the problem with the joint conditional distribution $P(\mathcal{D}|\Theta)$, where Θ is the set of parameters to be estimated (defined later). Assuming the training instances are independently sampled, the log likelihood can be written as:

$$lnPr(\mathcal{D}|\Theta) = \prod_{i=1}^{N} ln\left[a_i p + b_i(1-p)\right] \tag{1}$$

where:

$$a_i = \prod_{t=1}^{T} Pr[y_i^t | z_i = 1, \alpha^t] = \prod_{t=1}^{T} [\alpha^t]^{y_i^t} [1-\alpha^t]^{1-y_i^t}$$

$$b_i = \prod_{t=1}^{T} Pr[y_i^t | z_i = 0, \beta^t] = \prod_{t=1}^{T} [\beta^t]^{1-y_i^t} [1-\beta^t]^{y_i^t}$$

This log-likelihood is then maximized by the Expectation-Maximization (EM) algorithm [8], leading to an estimation of all parameters $\Theta = \left[\alpha^1, \beta^1, ..., \alpha^T, \beta^T, p\right]$ using the following equations:

$$\alpha^t = \frac{\sum_{i=1}^{N} \mu_i y_i^t}{\sum_{i=1}^{N} \mu_i} \tag{2}$$

$$\beta^t = \frac{\sum_{i=1}^{N} (1-\mu_i)(1-y_i^t)}{\sum_{i=1}^{N} (1-\mu_i)} \tag{3}$$

$$p = \frac{\sum_{i=1}^{N} \mu_i}{N} \tag{4}$$

B. Score to Select Experts

1) Spammer Score: Following the previous study in [9], [4], an annotator defined as an expert has both his sensitivity (true positive rate) and specificity (1-false positive rate) close to one. In other words, annotator t is an expert if: $\alpha^t + \beta^t - 1 = 1$. On the other hand, a spammer is defined as an annotator who assigns labels randomly. Hence, the probability that he labels an instance one or zero does not depend on the actual true label. In other words, annotator t is a spammer if:

$$Pr[y^t = 1 | z^t = 1] = Pr[y^t = 1 | z^t = 0] \tag{5}$$

$$\alpha^t + \beta^t - 1 = 0 \tag{6}$$

It corresponds to the diagonal on the ROC curve. To give a concrete example, we simulate 80 spammers and 20 experts for a binary dataset available on the UCI machine Learning Repository (Glass dataset) [10]. We estimate the sensitivity and the specificity of each annotator with the baseline model describes in section III-A. Results can be seen on Figure 1. We notice that all spammers lie on the diagonal of the ROC plot. Therefore, to rank annotators, authors in [9] define the Spammer Score as

$$S^t = (\alpha^t + \beta^t - 1)^2 \tag{7}$$

An annotator is a spammer if S^t is close to zero. In other words, let $A = \{1, ..., T\}$ be the set of annotators and E be the set of experts among the annotators. We have:

$$E = \left\{t \in A | S^t > \phi\right\} \tag{8}$$

where $\phi \in [0, 1]$.

[1] $(.)^T$ is the matrix transpose

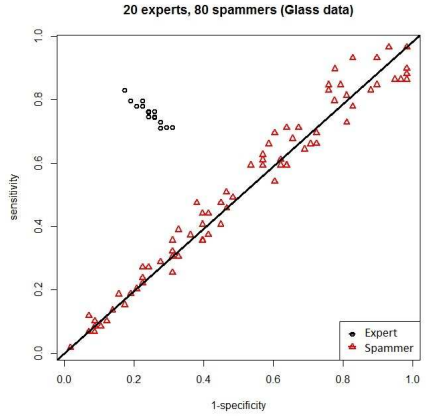

Fig. 1: Representation of the annotators according to their sensitivity and (1-specificity).

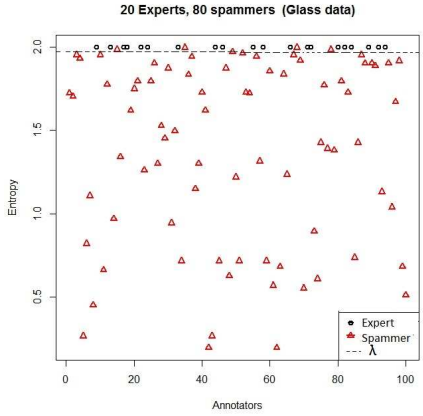

Fig. 2: Representation of the Entropy measure for each annotator simulated.

This evaluation score was recently formalized and proved in the SpEM algorithm proposed in [4]. However, we here believe that estimating the score S for all the annotators is time-consuming, since the set of available annotators can be very large. Instead, we combine the spammer score with the entropy measure, in order to significantly reduce the dataset with no significant costs.

2) Entropy Measure: The concept of entropy has been first introduced by Hartley [11], but was really developed and used in the industrial context by Shannon and Weaver [12]. They proposed a measure of information which is the general entropy of a distribution of probabilities. The key concept of entropy in information theory is that it measures the randomness in a probability distribution, or set of observed data. Among the widely used entropy measures, Shannon's is one of the most popular [13]. It can be defined as follows: Let X be a random variable with n possible outcomes $\{x_1, x_2, ..., x_n\}$. Shannon entropy, denoted by H(x), is computed as follows:

$$H(x) = -\sum_{i=1}^{n} p(x_i) ln(p(x_i)) \tag{9}$$

where $p(x_i)$ is the probability mass function of the outcome x_i. In a probabilistic context, H(x) is viewed as a measure of the randomness carried by x. If the distribution of x is uniform, the entropy measure is high, which indicates a completly random variable according to Shannon's definition, and thus, no information is carried by x. If, on the contrary, the entropy score is low, the variable is more predictable and consequently, gives more information.

In machine learning and more specifically in supervised learning, the entropy measure evaluates the quantity information about an outcome provided by the distribution of the class variables. Consequently, entropy-biased strategies can be used to select smaller units of annotations. However, the entropy measure according to Shannon's definition is not always valid in this context. For instance, in case of balanced dataset, all experts will have a uniform distribution on their labels leading to a high entropy, while a spammer, who have always labeled 1 or 0, will have a lower entropy. On the contrary, if the dataset is unbalanced, experts will have a low entropy while spammers' entropy will be high. Therefore, the entropy measure as defined by Shannon is not always valid, as it assumes that the uniform distribution is the most uncertain distribution, which is not always the case in supervised learning. A novel entropy that possesses more suitable properties has been introduced in [14]: Let H_w be the new entropy. H_w is defined as follows:

$$H_w(N, f_1, f_2, ..., f_n) = \sum_{i=1}^{n} \frac{\lambda_i(1-\lambda_i)}{(-2w_i+1)\lambda_i + w_i^2} \tag{10}$$

with:

$$\lambda_i = \frac{N f_i + 1}{N + n} \tag{11}$$

f_i is the frequency of each class i, and w_i is a reference distribution that is viewed as of maximal entropy. The above-mentioned entropy no longer considers the uniform distribution as the most uncertain distribution; consequently, it is more suitable for supervised learning problems. When the probability of each class is known, it is consistent to use these a priori probabilities of the classes to determine the reference distribution. Otherwise, it could be estimated from the overall class frequencies in the learning dataset. Therefore, when multiple annotators are available, all experts will have a high entropy score. This can be seen on Figure 2, where we compute the entropy measure H_w for each annotator simulated in section III-B1.

Unlike experts, spammers' entropy measure varies widely. Indeed, a spammer is, by definition, an annotator who labels randomly the dataset given. Consequently, the resulted distribution may be equivalent to the reference distribution. We denote by H_w^t the entropy of annotator t. We have $H_w = \left\{H_w^1, ..., H_w^T\right\}$. We denote by EC the group of Experts Candidates, i.e, the annotators who are the most likely to be experts. We have:

$$EC = \left\{t \in A | H_w^t > \lambda\right\} \tag{12}$$

where $\lambda \in [0 : +\infty[$.

Using the entropy measure is a good way to eliminate a large set of spammers, and reduce the annotations obtained from crowdsourced services. Combined with the spammer score defined in section III-B1, we obtain the ExpertS method, an efficient and rapid algorithm that jointly estimates the annotators accuracy, eliminates the spammers and learns the actual true label of an instance based only on the experts annotations. Next section describes more precisely our ExpertS algorithm.

C. ExpertS Algorithm

The expert selection step is established using a combination of the entropy measure and the spammer score. First, the entropy measure is calculated for all the annotators in order to identity the EC group. Then, the EM algorithm is run on the selected group in order to estimate the performance of each annotator and eliminate the hidden spammers in the group. Therefore, in this paper we define an expert as an annotator who satisfies both equations 8 and 12. Let E_f be the set of final experts selected, we have:

$$E_f = \{t \in (EC \cap E)\} \tag{13}$$

$$= \left\{t \in A | (H_w^t > \lambda) \cap (S^t > \phi)\right\} \tag{14}$$

where $\lambda \in [0 : +\infty[$ and $\phi \in [0, 1]$.

In this study, we favor good annotators who have $S^t > \frac{1}{|A|}$, where $|A|$ is the total number of annotators. Also we call annotators who have $S^t \leq \frac{1}{|A|}$ as spammers, and filter them during the integration process. Therefore, in our experiments we set $\phi = \frac{1}{|A|}$. But how can we set the value of the second threshold λ, as the only information we know is that it is in the interval $[0, +\infty[$? In other words, how to identify the EC group? We propose to apply the 3-steps following method:

1- Ranking Step: We start by calculating the entropy measure H_w^t for each annotator, and we define $(A, \leq)$, a structure that orders the annotators according to their entropy measures in an ascending order. We set the EC group as the top K annotators, i.e, the K annotators with the highest entropies.

2- Evaluation Step: EM algorithm is performed on the EC group in order to estimate the sensitivity and specificity of each annotator.

3- Filter Step: We compute the Spammer Score S^t for each annotator, and we define the final group of experts as follows:

$$E_f = EC - \left\{t \in EC | S^t \leq \phi\right\}$$

We repeat steps 1, 2 and 3 adding in each iteration the next Top K annotators to the final group of experts selected, until no more experts are found. We finally use majority voting to predict the ground truth label for each instance. ExpertS algorithm is summarized in Algorithm.1.

Algorithm 1 ExpertS

1: Imput: X, Y.
2: Initialize $\alpha = 0, \beta = 0$, thresholds ϕ and the number K of annotators to add in each iteration.
3: Set $A = \{1, ..., T\}$.
4: Compute $H_w(A) = \left\{H_w^1, ..., H_w^T\right\}$.
5: Ranking Step: EC $\leftarrow TopK(A, \leq)$.
6: Initialize the estimated ground true label for each instance: $\mu_i = 1/|EC| \sum_{t \in EC} y_i^t$
7: Evaluation Step: $\forall t \in EC$, Update $\{\alpha^t, \beta^t\}$ and prevalence p using EM algorithm and equations 2, 3 and 4.
8: Filter Step: Compute $S^t, \forall t \in EC$. Select the final group of experts:

$$E_f^{old} \leftarrow EC \setminus \left\{t \in EC | S^t \leq \phi\right\}$$

.
9: **repeat**
10: A $\leftarrow$ A$\setminus TopK(A, \leq)$.
11: EC $\leftarrow TopK(A, \leq) \cup E_f^{old}$.
12: Update $\mu_i = 1/|EC| \sum_{t \in EC} y_i^t$
13: EM algorithm: Update $\{\alpha^t, \beta^t, p\}$ with equations 2, 3 and 4.
14: Compute S^t, $\forall t \in EC$. Select the final group of experts:

$$E_f^{new} \leftarrow EC \setminus \left\{t \in EC | S^t \leq \phi\right\}$$

.
15: Bool $\leftarrow (E_{old}^f = E_{new}^f)$.
16: **if** not Bool **then**
17: $E_f^{old} \leftarrow E_f^{new}$
18: **end if**
19: **until** Bool
20: **return** Parameters α, β, p and set of experts E_f^{new}
21: Calculate the ground truth label for each instance of the dataset with majority voting over all the experts selected.

IV. Experiments and Results

We validate the effectiveness of the proposed algorithm on datasets from the UCI Machine Learning Repository [10] for which binary labels are available. Experts were simulated with a sensitivity and specificity between 0.75 and 0.95, and all spammers lie around the diagonal of the ROC plot. We compare our proposed ExpertS approach with three methods: the commonly used Majority Voting (MV), the baseline method of Raykar et Al.[2], and the SpEM algorithm proposed in [4]. Among all the above-mentioned methods, only SpEM have a mechanism to explicity detect spammers. We simulate each model a hundred times using the bootstrap method, and we evaluate our algorithm using the following criterias: the area under the ROC curve (AUC), in order to estimate the accuracy of each model, and the computation time (in seconds) in order to compare their rapidity.

A. Performance of ExpertS Algorithm

We first validate the performance of the proposed algorithm on eight datasets from the UCI: Ionosphere (351,34), Cleveland Heart (297,13), Musk(version 1) (476,167), Glass (214,10), Bupa (345,7), Vertebral (310,6), Spect Heart (267,22) and Haberman (306,3) (with (number of instances,

number of features)). We simulate 100 annotators, in which 15% are experts and 85% are spammers, and we set the threshold K to 0.30 times the number of available annotators. We compute for each dataset and for each method the AUC of the estimated probabilistic ground truth (cf.Table I), and we estimate the computation time in seconds (cf.Table II). Additionally, we compute the sensitivity of spammer detection for both SpEM and ExpertS methods, which is essentially the fraction of spammers correcty detected (cf.Table III). The following observations can be made: Concerning the quality of the learned model, we confirm the superiority of SpEM and Experts algorithms, which are clearly better than the baseline method and the Majority Voting, since their estimated AUC are significantly higher for all the datasets presented (refer Table I). This validates the efficiency of selecting the good annotators when learning the model. Additionally, results in Table III support the fact that ExpertS is as good as SpEM since both have a sensitivity of spammer detection around 99%. Furthermore, The clear advantage of the proposed ExpertS algorithm over SpEM can be seen in Table II, where we can notice that ExpertS is clearly faster in terms of computation time than SpEM (around 163sec for SpEM compared to 2.34sec for ExpertS).

B. Effect of Increasing Annotators

In this section, we test how robust is our method when confronted to a high number of annotators. We first simulate 200 annotators, and we add 200 annotators each time, until we reach 10000 annotators. In each simulation, 15% of the annotators are experts and 85% are spammers. We compute the estimated AUC and the computation time (in seconds) for each model. We plot the results obtained for Glass data in Figures 3. The following observations can be made: for all the models tested, the quality of the estimated labels does not depend on the number of annotators simulated: our proposed ExpertS algorithm and SpEM algorithm are always better in terms of AUC than the baseline method and the Majority Voting. However, unlike ExpertS algorithm, SpEM algorithm have a clear inconvenience since the computation time increases linearly when we extend the number of simulated annotators (ExpertS computation time varies from 1sec to 8sec, whereas SpEM varies from 1sec to almost 5000sec). Therefore, our proposed approach is clearly more practical when confronted to a high number of annotations.

TABLE I: Comparison of the estimated AUC between ExpertS, Majority Voting (M.V), Baseline, and SpEM.

Dataset	M.V	Baseline	SpEM	ExpertS
Cleveland	0.922	0.934	0.991	0.998
Ionosphere	0.901	0.951	0.995	0.994
Musk	0.930	0.952	1.000	0.998
Glass	0.860	0.907	0.995	0.997
Bupa	0.822	0.876	0.989	0.990
Vertebral	0.900	0.930	0.998	0.996
Spect Heart	0.865	0.904	0.996	0.992
Haberman	0.721	0.875	0.997	1.000
Mean	0.865	0.916	0.995	0.996

TABLE II: Comparison of the computation time (in seconds) between ExpertS, Majority Voting (M.V), Baseline, and SpEM.

Dataset	M.V	Baseline	SpEM	ExpertS
Cleveland	0.86	41.35	152.25	1.66
Ionosphere	1.02	97.77	137.83	3.31
Musk	1.26	157.87	203.63	5.07
Glass	0.59	20.56	62.70	1.56
Bupa	0.86	41.35	152.25	1.63
Vertebral	0.91	107.96	201.41	1.92
Spect Heart	0.91	67.30	134.69	1.72
Haberman	0.89	42.18	258.92	1.81
Mean	0.91	72.04	162.96	2.34

TABLE III: Comparison of the fraction of spammers correctly detected between SpEM and ExpertS.

Dataset	SpEM	ExpertS
Cleveland	0.998	0.997
Ionosphere	0.996	0.995
Musk	0.995	0.996
Glass	0.999	0.989
Bupa	0.987	0.987
Vertebral	0.991	0.997
Spect Heart	0.987	0.999
Haberman	0.989	0.988
Mean	0.993	0.994

C. Effect of Increasing Spammers

We here study the effect of increasing the number of spammers among the annotators. We simulate 5 experts for Glass data and keep adding spammers from a pool of 100 spammers. We compare the evolution of the estimated AUC, and we plot the sensitivity for spammer detection which is essentially the fraction of spammers correctly detected. Additionally, we plot the actual number of annotators used across all the iterations for SpEM and ExpertS. Results are reported in Figures 4. On one hand, we notice that SpEM and ExpertS achieve similar performance and are much more robust than M.V and the baseline method when confronted to a high number of spammers. This can be easily explained by the fact that Both SpEM and ExpertS algorithms iteratively eliminate the spammers and estimate the ground truth without the spammers. On the other hand, the number of annotators used across all the iterations in our proposed algorithm is significantly lower than the number of annotators used in SpEM. This result explains the reduction of cost time for the ExpertS algorithm in comparison with SpEM, and validates our approach.

V. CONCLUSIONS

In this paper, we propose a probabilistic approach for classification when labels given by multiple noisy annotators are available but no gold standart. By integrating two metrics, the entropy measure and the spammer score to the EM algorithm, we obtain the proposed ExpertS algorithm, which eliminates annotations provided by spammers without using any true labels, and estimates the ground truth based only on higher quality annotations provided by experts. Experiments on a broad range of datasets show that our approach is better in

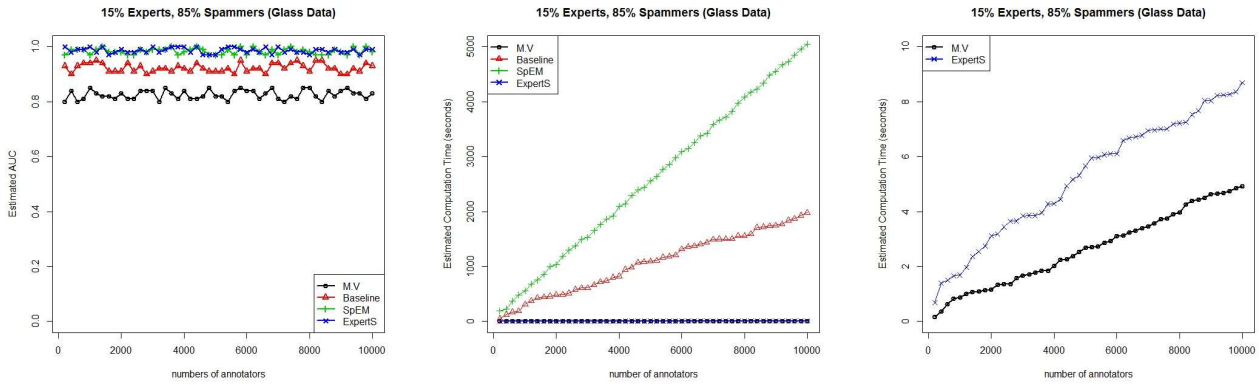

Fig. 3: Effect of Increasing Annotators. Comparison of the evolution of the estimated AUC (Left) and of the computation time (Center) between M.V, the Baseline Method, SpEM and ExpertS. (Right) We plot only the computation time for M.V and ExpertS in order to see more precisely their evolution.

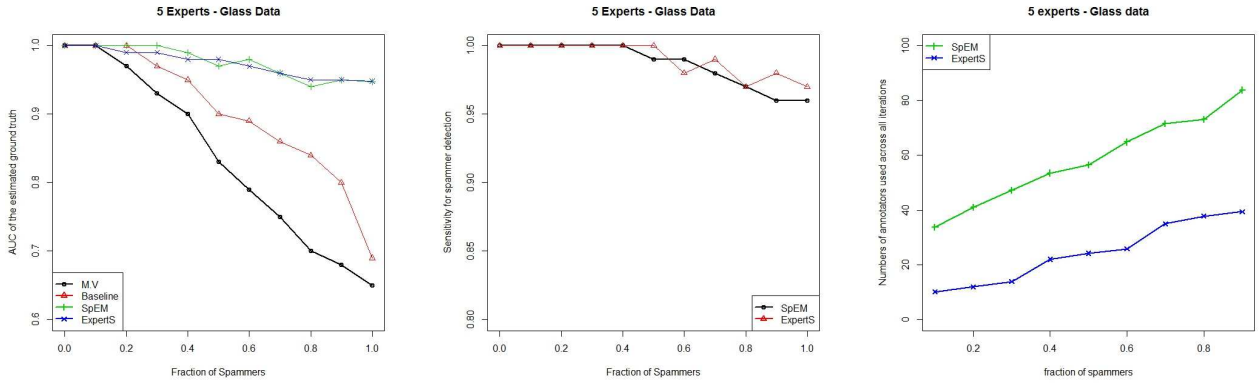

Fig. 4: Effect of Increasing Spammers. (Left) Comparison of the evolution of the estimated AUC as a function of the fraction of spammers. (Center) Comparison of the fraction of spammers correctly detected. (Right) Comparison of the number of annotators used across all the iterations.

terms of accuracy than previous methods that do not eliminate the spammers. More importantly, our approach is significantly better is terms of computation time compared to other methods integrating annotators selection, without degrading the model. Due to its accuracy and rapidity, ExpertS algorithm is expected to be more practical especially in the age of the Internet.

REFERENCES

[1] Y. Tian and J. Zhu, "Learning from crowds in the presence of schools of thought," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 226–234.

[2] V. C. Raykar, S. Yu, L. H. Zhao, and G. H. Valadez, "Learning from crowds," *In Journal of Machine Learning Research 11 - MIT Press*, pp. 1297–1322, 2010.

[3] Y. Yan, G. Hermosillo, R. Rosales, L. Bogoni, G. Fung, L. Moy, M. Schmidt, and J. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," *In Proceedings of the* 13^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, 2010.

[4] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *J. Mach. Learn. Res.*, vol. 13, pp. 491–518, 2012.

[5] P. Zhang and Z. Obradovic, "Integration of multiple annotators by aggregating experts and filtering novices," in *BIBM'12*, 2012, pp. 1–6.

[6] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied Statistics*, vol. 28, pp. 20–28, 1979.

[7] P. P. P. Welinder, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in *CVPR*, 2010.

[8] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data," *J. of the Royal Statistical Society (B)*, vol. 39(1), 1977.

[9] V. C. Raykar and S. Yu, "Ranking annotators for crowdsourced labeling tasks." in *NIPS*, 2011, pp. 1809–1817.

[10] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[11] R. V. L. Hartley, "Transmission of information," *Bell Syst. Tech. Journal*, vol. 7, pp. 535–563, 1928.

[12] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, Chicago, and London: University of Illinois Press, 1949.

[13] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, july, october 1948.

[14] D. A. Zighed, G. Ritschard, and S. Marcellin, "Asymmetric and sample size sensitive entropy measures for supervised learning." in *Advances in Intelligent Information Systems*. Springer, 2010, vol. 265, pp. 27–42.