Multi-Labeller classification Method based on Mixture of Classifiers and Genetic Algorithm Optimization

Andrés F. Arciniegas M., David E. Imbajoa R.,
Ivan D. Gustin, Mauricio Bolaños L.,
Dario F. Fajardo F.
Universidad de Nariño
Calle 18 No. 42
Pasto, Colombia
E-mail:arciandres@gmail.com,deivy311@hotmail.com
Ivanfly.92@gmail.com, arciandres@gmail.com,
dario@udenar.edu.co

Fredy Guasmayán G.,
María J. Bravo
Universidad Mariana
Calle 18 No. 34-104
Pasto, Colombia
E-mail:
fguasmayan@umariana.edu.co,
mabravo@umariana.edu.co

Diego H. Peluffo-Ordoñez
Universidad Técnica del Norte
Universidad Cooperativa de Colombia
Calle 18 No. 42
Pasto, Colombia
E-mail: dhpeluffoo@unal.edu.co

***Abstract*— This work presents a new method proposal applied to Multi-Labelers scenarios. This is a situation where labelling individuals in a set of data based on certain characteristics in the process of determining labels to individuals in a set of data based on certain characteristics. Our approach consists in processing a Support Vector Machine classifier to each labelers substantiated on his answers. We formulate a genetic algorithm optimization to obtain a set of weights according to their opinion, in order to penalize each panelist. Finally, their resulting mappings are mixed, and a final classifier is generated, showing to be better than majority vote. For experiments, the well-known Iris database is handled, with multiple simulated artificial labels. The proposed method reaches very good results compared to conventional multi-labeler methods, able to assess the concordance among panelists considering the structure data.**

***Abstract*— Multi-labeller, multicriteria optimization, genetic algorithm, Gaussian distribution, support vector machine.**

I. Introduction

Typically, the approaches of pattern recognition, based on supervised classification, require previous knowledge that takes into account a structure of labels, given by an unique expert or assessor. However, there are scenarios where information is assessed by multiple experts. That is to say, the characteristics of the data itself not only require supervision but also a range of opinions, so that its analysis obtains validity [1]. Some examples of these cases might be a group of specialists in the diagnosis of the pathology of a patient with specialized equipment [2] or the evaluation of the academic performance of a student [3]. In this type of scenario, where data is exposed to multiple interpretations, several factors that directly affect the adequate analysis of the information must be addressed. Among them, the inaccurate evaluation provided by the labelers that prevent a correct revision of the information. Therefore, it is necessary to find strategies that enable the reduction of the influence of the mistaken labels in relation to the real ones or the ground truth. The analysis of multiple experts focuses on the compensation of the negative effect of the mistaken labels. The mentioned compensation can improve the learning process in terms of factors of penalization or quantifying the efficiency of the evaluators [1] [4]. In particular, the support vector machines (SVMs) have shown to be a suitable alternative to approach this problem, mainly due to their versatility in regards to supervised classification [1]. In this paper, a new strategy for data classification contained inside an approach of multiple labelers is presented. The final classification is carried out using a variety of classifiers trained through the intervention of each labeler. Our method establishes a vector of decision variables that satisfies the restrictions and optimizes a function of vector whose elements represent the objective functions, as it generates values of decision in relation to the labelers.

The approach suggested initially provides a multi-objective criterion that, based on the functions generated by the labelled sets, estimating their respective optimal values with which the corresponding weighting values are generated. Finally, the combination of classifiers whose properties are established by means of the functions of cost and the factors of weighting aforementioned are carried out. The strategy proposed is evaluated on the database *IRIS* of the UCI learning machine. Label vectors are created entering in them, different noise percentages in relation to the vector designated as ground truth. N different experiments with m iterations were conducted to prove the stability of this approach, our multi-labeler method accomplishes quite good results and stands for an efficient alternative in regards to conventional approaches.

The outline of this paper is as follows: Relevant related works are described in Section II. Section III explains the methods used in the proposed weighted Multi-labeler classification. Experimental results are shown in section IV. Finally, section V draws the conclusions and final remarks.

II. Related Works and Background

Many approaches to deal with multi-labeler problems are formulated within support-vector-machines (SVM) frame-works, due to its versatility and outstanding performance in several applications. For instance, a bi-class multi-labeler classifier (BMLC) is introduced in [5]. It starts from the simplest formulation for a bi-class or binary SVM-based classifier. Let us define the ordered pair $\{\boldsymbol{x}_i, \bar{y}_i\}$ to denote the i-th sample or data point, where $\boldsymbol{x}_i$its d-dimensional feature vector and $\bar{y}_i \in \{1, -1\}$, is its binary class label. All feature vectors can be gathered into a Nxd data matrix $\boldsymbol{X}$ such that $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^\intercal$, considering a data set of N samples, whereas labels into a labeling vector $\bar{\boldsymbol{y}} \in \mathbb{R}^m$. Consider k labelers or labeling vectors $\{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(k)}\}$ as well. That said, there are some approaches to estimate the labeling vector $\bar{\boldsymbol{y}}$, which is a reference vector to be determined. By calculating the simple average as done in [6], for instance. We assume a latent variable model in the form: $e_i = \boldsymbol{w}^\intercal \boldsymbol{x}_i + b = < \boldsymbol{x}_i, \boldsymbol{w} > +b$ to pose the classifier's objective function, where $\boldsymbol{w}$ is a d-dimensional vector, b is a bias term and notation $<\cdot,\cdot>$ stands for Euclidean inner product. As can be readily noted, vector $\boldsymbol{e} = [e_i, \dots, e_m]$ results from a linear mapping of elements of $\boldsymbol{X}$, which is a hyperplane, from a geometrical point of view, and can thus be seen as a projection vector. By design, if assuming $\boldsymbol{w} \in \mathbb{R}^d$ as an orthogonal vector to the hyperplane, projection vector can be used to encode the class assignment by a decision function in the form $sign(e_i)$. Alternatively, projection vector can be expressed in matrix terms as $\boldsymbol{e} = \boldsymbol{X}\boldsymbol{w} + b\mathbf{1}_m$, being $\mathbf{1}_m$ an m-dimensional all ones vector.

In addition, the distance between the hyperplane and any data point can be constrained to be at least 1 by fulfilling the condition $\bar{y}_i e_i \geq 1$, $\forall_i$, in order to avoid that data points lie in an ambiguity region for the decision making. The distance between hyperplane $\boldsymbol{e}$ and data point $\boldsymbol{x_i}$ can be calculated as: $d(\boldsymbol{e}, \boldsymbol{x}_i) = \bar{y}_i e_i / \|\boldsymbol{w}\|^2$, where $\|\cdot\|$ denotes Euclidean norm. Therefore, since the upper boundary of $d(\boldsymbol{e}, \boldsymbol{x}_i)$ is $1/\|\boldsymbol{w}\|^2$, one expect that $\bar{y}_i \cong e_i$. Then, the classifier objective function to be maximized can be written as: $max_{\boldsymbol{w}} \bar{y}_i e_i / \|\boldsymbol{w}\|^2; \forall_{\boldsymbol{i}}$. Consequentially, we can write the problem, for accounts of minimization, so: $min_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{w}\|^2$, s. t. $\bar{y}_i e_i = 1, \forall_{\boldsymbol{i}}$. Notice that previous formulation is attained under the *hard* assumption that $\bar{y}_i = e_i$, and can then be named as hard-margin SVM. By relaxing it, and by adding slack terms, a soft-margin SVM (SM-SVM) can be written as:

$$\min_{\boldsymbol{w},\xi} f(\boldsymbol{w}, \xi | \lambda) = \min_{\boldsymbol{w},\xi} \frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i^2 \quad s.t. \xi_i \geq 1 - \bar{y}_i e_i, \tag{1}$$

Where λ is a regularization parameter and ξ_i is a slack term associated to data point i.

Binary approach: Aimed at designing a multi-label classifier, in [4], [5], the SM-SVM given in equation 1 is modified by adding penalty factors $\theta_j{}_{j=1}^{k}$ and computing $\bar{\boldsymbol{y}}$ as the average of the set of the labeling vectors. This factor is intended to make f increases when adding wrong labels otherwise f should not or insignificantly decrease. In other words, consider a set of k labelers or panelists who singly provide their corresponding labeling vectors. Then, the j-th panelist's quality is quantified by the penalty factor θ_j. Accordingly, with the penalty factors θ being included, a new binary classification problem is introduced by modifying problem stated in 1, as shown in [5]. As explained, the solution of this problem is accomplished by a primal-dual formulation.

Multi-class approach: Using a one-against all strategy, another work [7], naturally extends this approach to multiclass scenarios. This approach consists basically of building a number of SVM models -one per class. A multi-class approach is accomplished by applying c times the BMLC approach. In general, in case of using SVM-approaches, class c is compared with the remaining ones in such a way that it is matched with a positive label, meanwhile the others with a negative label [8]; so that a binary labeling vector per each single class is formed. Concretely, the labeling reference vector $\bar{\boldsymbol{y}}^{(\ell)}$ associated to class ℓ is assumed as a binarized version of labeling vector, as explained in [7]. In this sense, the BMCL is generalized to deal with more than two clases. Consequently, the decision hyperplanes are given by $\{e_1^{(\ell)}, \dots, e_i^{(\ell)}\}$, where $\boldsymbol{e}^{(\ell)} = \boldsymbol{X}\boldsymbol{w}^{(\ell)} + b^{(\ell)}\mathbf{1}_m$.

III. PROPOSED MULTI-LABELER CLASSIFICATION APPROACH

Unsupervised analysis covers all methods denominated as discriminative, which does not require a priori knowledge of the classes for classification. They usually require only one initialization parameter, like the number of resulting groups or any other indication about the initial partition. Then, the unsupervised analysis task is grouping homogeneous patterns without any information about the nature of classes present in the dataset. For this reason, the analysis does not generate unsupervised automatic classification, but generate s a homogeneous subset of data from some criterion based on distances, dissimilarities or statistical measures. Hence, the term of unsupervised classification refers to the grouping of data into subsets of similar elements and not some sort of automatic classification. There are several reasons why unsupervised methods are of special interest: converge quickly and they keep good performance if the characteristics change little over time, allowing categorizing items; they are useful when labeling a large set of samples is not feasible, among others. However, the solution generated by an unsupervised analysis system can be affected by factors such as inadequate initial parameters, which might generate a bad convergence, as explained in [9].

A. *Multi-labeler approach*

Our approach may result appealing since it is easy to solve by means of a quadratic programming search, given the form of the dual formulation. However, as BMLC, solution is highly dependent on the chosen reference vector $\bar{\boldsymbol{y}}$ as well as a no new coordinate axis is provided since only one vector α is yielded. Furthermore, to design a multi-labeler approach from this formulation, the quadratic problem should be solved k times (one per labeler). Instead, we propose to perform a mixture of classifiers. Let us define $f^{(j)}(\boldsymbol{X})$ the trained cost function by using the labels given by the labeler$j$. Then, in order to take advantage of the information of the whole set of labelers, we propose a classifier whose cost function is the following mixture:

$$\bar{f}(\boldsymbol{X}) = \sum_{j=1}^{k} \eta_j f^{(j)}(\boldsymbol{X}) \tag{2}$$

Where η_t are the weighting factors to be defined.

B. *Grouping based on centroids*

The general idea of grouping based on centroids, is to minimize or maximize an objective function, which defines how good the solution pooling is. To achieve this, we use a method based on Gaussian Expectation Maximization, commonly used in clustering applications [10]. A generalized way to perform this grouping may be obtained by studying the proportion or degree of belonging of an element to a group, and the influence of each element in the centroid's updating. And the resulting partition for each iteration corresponds to the allocation of the subset elements whose centroid is nearest. Variants of these algorithms consist on changes of the objective function and therefore the update function centroids.

1) *Gaussian Expectation maximization Mixture:* (GEMM)

It is part of clustering methods based on probability density (DBC) and its objective function is the linear combination of Gaussian distributions centered in the centroids of each group, as follows:

$$GEMM_{log}(\mathbf{X},\mathbf{C}) = -\sum_{i=1}^{C} log\left(\sum_{j=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{q}_i)p(\boldsymbol{q}_j)\right), \tag{3}$$

Where $p(\boldsymbol{x}_i|\boldsymbol{q}_i)$ is the probability of $\boldsymbol{x}_i$ since it is generated by a Gaussian distribution centered in $\boldsymbol{q}_j$, $p(\boldsymbol{q}_j)$ is the probability a priori of the group whose centroid is $\boldsymbol{q}_j$.

Another alternative to compute the objective function is by using an exponential operator:

$$GEMM_{log}(\mathbf{X},\mathbf{C}) = -\sum_{i=1}^{C} exp\left(\sum_{j=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{q}_i)p(\boldsymbol{q}_j)\right), \tag{4}$$

When the estimated set of these probabilities present an elevated dispersion, the method with the logarithm function is used, as it segments the extraction of classification results, through the estimation of each distribution peak, maximizing the plausibility of each annotator; When the dispersion is low, the method with the exponential function should be used, which is a more aggressive method, as it segments the evaluation results in a more specific region. The minus sign is fixed in order to set a minimization operation with the objective function. The respective membership functions of each element are:

$$m_{GEMM}(\boldsymbol{q}_j|\boldsymbol{x}_i) = \frac{p(\boldsymbol{x}_i|\boldsymbol{q}_i)p(\boldsymbol{q}_j)}{p(\boldsymbol{x}_i)}, \tag{5}$$

Notice that the membership function is a probability value, thus Bayes' rule can be used to calculate its value, considering $p(\boldsymbol{x}_i)$ as evidence:

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{q}_j)p(\boldsymbol{q}_j) \tag{6}$$

$p(\boldsymbol{x}_i|\boldsymbol{q}_j)$ factor can be obtained easily with:

$$p(\boldsymbol{x}_i|\boldsymbol{q}_j) = \frac{1}{\det(\Sigma_j)^{\frac{1}{2}}}(2\pi)^{-\frac{d}{2}}e^{-\frac{1}{2}(\boldsymbol{x}_i-\mu)\Sigma_j^{-1}(\boldsymbol{x}_i-\mu)^{\intercal}}, \tag{7}$$

Where μ is the centroid ($\mu = \boldsymbol{q}_j$) , d is dimension, Σ represent the covariance and $\det(\cdot)$ denotes the matrix determinant argument.

Objective functions to be minimized are given by:

$$\boldsymbol{F}_{exp} = -\sum_{\boldsymbol{i=1}}^{C} \exp(p(\boldsymbol{x}_i)), \tag{8}$$

$$\boldsymbol{F}_{log} = -\sum_{\boldsymbol{i=1}}^{C} \log(p(\boldsymbol{x}_i)), \tag{9}$$

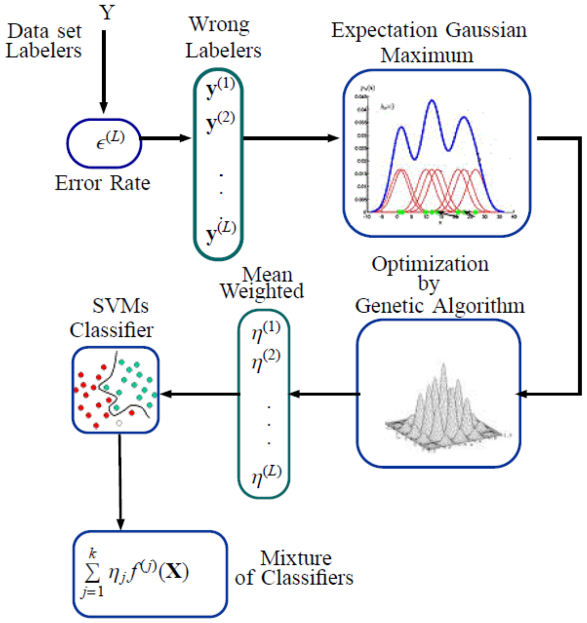


Figure 1: Process Diagram of Multi-labeler approach

C. *Genetic algorithm for weights estimation*

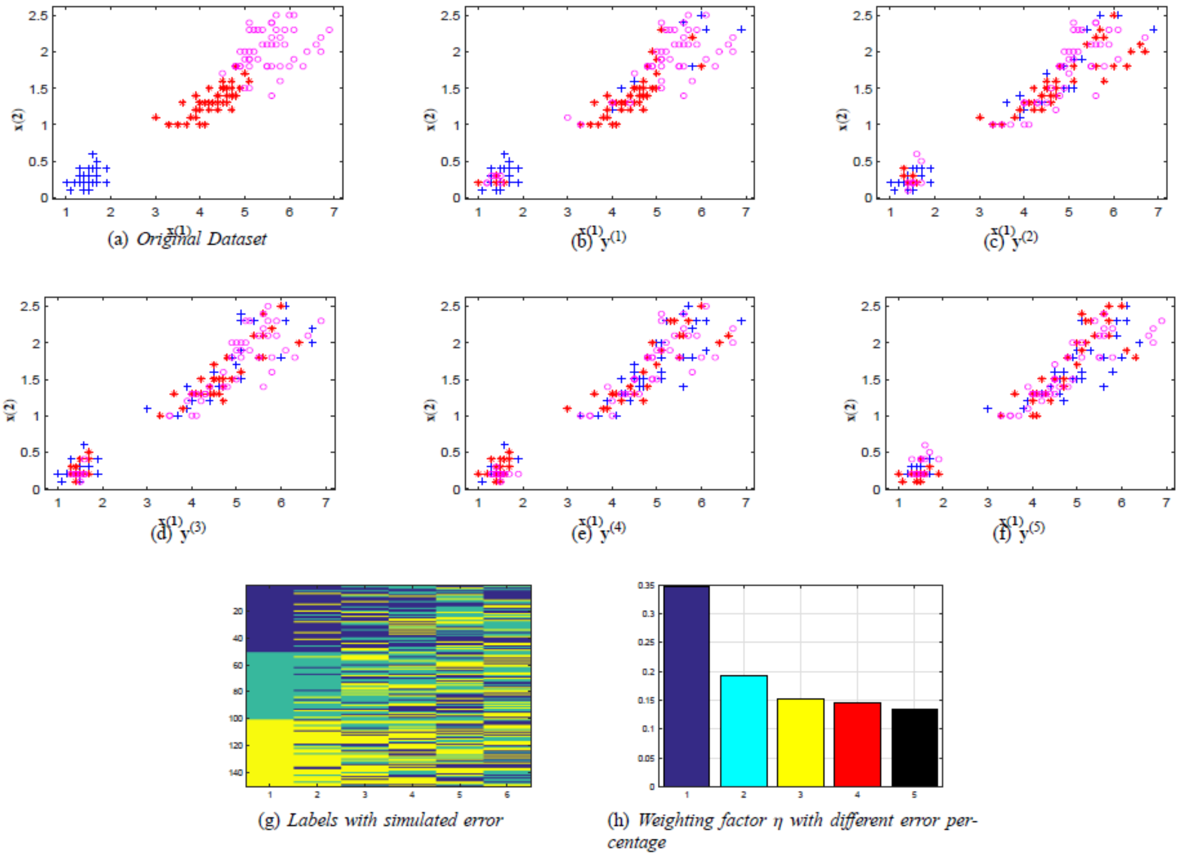
For estimation of weights for pattern classification, a genetic algorithm is chosen. It solves iteratively the optimum values for the weights in a multi-criteria objective function, depicted in 6, which consists in maximizing the sum of the Gaussian distribution for each labeler in every object to be classified, and also taking into account the probability a priori of each Gaussian distribution. For this purpose, the Pareto optimization method is used [11]. A Gaussian distribution function is generated for each class, labelled by each expert. And for each resulting classification, an objective function is generated to be optimized. In Figure 1, the process diagram of the proposed approach is depicted. To achieve better results, according to the dispersion of each classifier's outcome, the objective functions in 8 and 9, are subject to:

$$\sum_{\boldsymbol{j=1}}^{k} \eta_j = 1, \tag{10}$$

Where $\eta_j\ \in [0,1]$

IV. RESULT AND DISCUSSION

Database: Open Iris flower database, extracted from UCI repository [12] is considered for experiments. Three different types of flowers are contained, with fifty samples each: Versicolor, Virginica and Setosa. For each sample, four characteristics were registered: width and length of petal and sepal. Moreover, there is



(a) *Original Dataset*

(b) $y^{(1)}$

(c) $y^{(2)}$

(d) $y^{(3)}$

(e) $y^{(4)}$

(f) $y^{(5)}$

(g) *Labels with simulated error*

(h) *Weighting factor η with different error percentage*

Figure 2: Generated Data and artificial Labelers for Experiment No. 5. Feature 1 is Petal length, and Feature 2 is Petal width. Scatter plots are displayed for *Ground Truth* and the five labelers.

A linearly separable class, and the two classes left are overlapped. The results presented here take into account petal length and width only. Different labels from several annotators are simulated and built using this database, in order to show the method effects and characteristics. Data matrix is normalized so that its maximum value per column be 1, before carrying out the classification procedures.

Methods: As reference methods, we consider the average and the majority vote of the given labeling vectors.

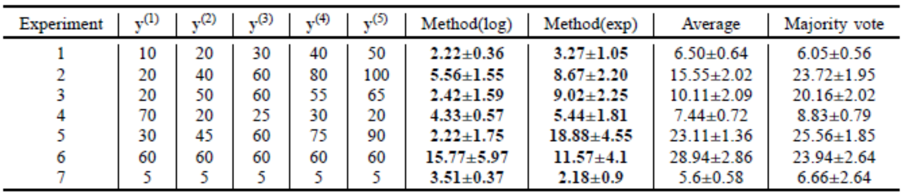
Parameter settings: To perform our multi-labeler approach, we use a multi-objective optimization with a genetic algorithm, with a random weights restricted between 0 and 1, as shown in equations (10)(11); a size of 20 individuals, and stopping criterion given by a tolerance of 0.01% error; tournament selection and 5% mutation rate.

Performance measures: Conventional measures are used to quantify the performance of the considered multi-labeler approaches, such as: standard error, statistic mean and margin of error. Cohen's Kappa Index is also used in this work to evaluate the agreement relation between annotators. It is calculated considering the equal labeled individuals by the experts, where a total agreement equals a Kappa index of 100%, and no agreement at all, a Kappa index of 0%.

Experiments: Simulated annotators with different percentages of error in their labels are generated to evaluate the efficiency of the method. The process is iterated thirty time s in order to reach the stability of the approach. Five cases are depicted, with different induced error rates in the annotator. Noise of data in labeling vectors are completely random, and, in order to try the accuracy of the method, error rates in each annotator were chosen in several different quantities. The assigned weights η are presented in Table II. These values were used in the experiments below, and are associated to the 'Proposed method' column in Table I, where the general results are depicted.

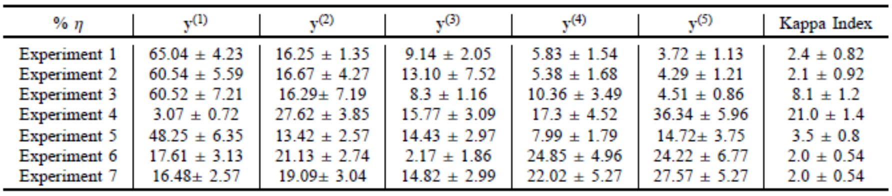
The agreement between labelers in Table II are shown by Kappa Index calculations. In experiment 4, although the error rate is the same for all labelers, it is hardly expected that they share labels in common, so the Kappa index is very low. In experiment 2, as all labelers have a low error rate unlike fourth one, it is very likely that they have many choices in common. Thus, Kappa index is higher respect to the other cases. When the index presents a high value, it is expected that the standard deviation between the η values for an experiment is low, and the annotators will be probably right, at least most of them, as shown in experiment 2. Nevertheless, a low value of this index implicates a higher level of disagreement between labelers, and that does not give much information about the accuracy of the method. This index should be taken into account only if presents high values.

Table I: Performance results in terms of error percentage ∈ of wrong classifications.



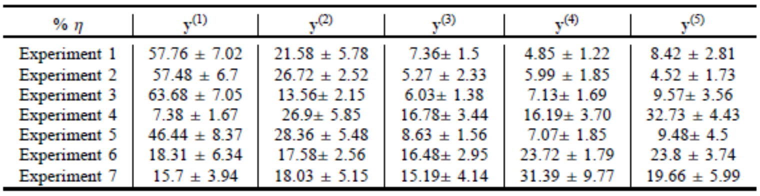
Experiment	$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$y^{(5)}$	Method(log)	Method(exp)	Average	Majority vote
1	10	20	30	40	50	**2.22±0.36**	**3.27±1.05**	6.50±0.64	6.05±0.56
2	20	40	60	80	100	**5.56±1.55**	**8.67±2.20**	15.55±2.02	23.72±1.95
3	20	50	60	55	65	**2.42±1.59**	**9.02±2.25**	10.11±2.09	20.16±2.02
4	70	20	25	30	20	**4.33±0.57**	**5.44±1.81**	7.44±0.72	8.83±0.79
5	30	45	60	75	90	**2.22±1.75**	**18.88±4.55**	23.11±1.36	25.56±1.85
6	60	60	60	60	60	**15.77±5.97**	**11.57±4.1**	28.94±2.86	23.94±2.64
7	5	5	5	5	5	**3.51±0.37**	**2.18±0.9**	5.6±0.58	6.66±2.64

Table II: Weight η values (exp)



% η	$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$y^{(5)}$	Kappa Index
Experiment 1	65.04 ± 4.23	16.25 ± 1.35	9.14 ± 2.05	5.83 ± 1.54	3.72 ± 1.13	2.4 ± 0.82
Experiment 2	60.54 ± 5.59	16.67 ± 4.27	13.10 ± 7.52	5.38 ± 1.68	4.29 ± 1.21	2.1 ± 0.92
Experiment 3	60.52 ± 7.21	16.29± 7.19	8.3 ± 1.16	10.36 ± 3.49	4.51 ± 0.86	8.1 ± 1.2
Experiment 4	3.07 ± 0.72	27.62 ± 3.85	15.77 ± 3.09	17.3 ± 4.52	36.34 ± 5.96	21.0 ± 1.4
Experiment 5	48.25 ± 6.35	13.42 ± 2.57	14.43 ± 2.97	7.99 ± 1.79	14.72± 3.75	3.5 ± 0.8
Experiment 6	17.61 ± 3.13	21.13 ± 2.74	2.17 ± 1.86	24.85 ± 4.96	24.22 ± 6.77	2.0 ± 0.54
Experiment 7	16.48± 2.57	19.09± 3.04	14.82 ± 2.99	22.02 ± 5.27	27.57 ± 5.27	2.0 ± 0.54

Table III: Weight η values (log)



% η	$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$y^{(5)}$
Experiment 1	57.76 ± 7.02	21.58 ± 5.78	7.36± 1.5	4.85 ± 1.22	8.42 ± 2.81
Experiment 2	57.48 ± 6.7	26.72 ± 2.52	5.27 ± 2.33	5.99 ± 1.85	4.52 ± 1.73
Experiment 3	63.68 ± 7.05	13.56± 2.15	6.03± 1.38	7.13± 1.69	9.57± 3.56
Experiment 4	7.38 ± 1.67	26.9± 5.85	16.78± 3.44	16.19± 3.70	32.73 ± 4.43
Experiment 5	46.44 ± 8.37	28.36 ± 5.48	8.63 ± 1.56	7.07± 1.85	9.48± 4.5
Experiment 6	18.31 ± 6.34	17.58± 2.56	16.48± 2.95	23.72 ± 1.79	23.8 ± 3.74
Experiment 7	15.7 ± 3.94	18.03 ± 5.15	15.19± 4.14	31.39 ± 9.77	19.66 ± 5.99

Figure 2 depicts generated data used for the second experiment. Figure 2(a) shows the original labels in the three classes. From Figure 2(b) to 2(f), the corrupt individual error data is shown for the five annotators. The misplaced labels can be noticed based on the colors of the classes. In Figure 2(g), contaminated labels are shown in a clearer way. Figure 2(h) shows the values of _ for each annotator, representing the associated weight value. The classification accuracy in terms of percentage of wrong classifications is presented in Table I. The error rate is decreased for all cases using the proposed method. As the approach assigns different weights to the annotators based on their certainty, the result error rate, compared with the other methods, is lower as the variation of the error in the labelers increases.

Experiment 1: In this case, the classifiers mixture with error percentage in the range of 10 percent in upward way until 50 percent is shown. In Tables II and III, weights for each labeler seen in Experiment 1 are corresponding with the error percentage shown in the Table I.

Experiment 2: This is a similar case to the experiment 1, but in a wider range. The proposed method gives more importance in the mixture of classifiers to that labelers whose error rates are lower. These are $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$, so the improvement respect to other methods is shown.

Experiment 3: The first annotator $\boldsymbol{y}^{(1)}$ has the lower error rate in this case, unlike the others, surpassing the first one for at least 30%. In this case, it is observed that the weight η associated to that first annotator is higher, so his opinion will be more relevant in the mixture process. Thus, the performance of the mixture of classifiers will improve in the proposed method as the worse annotators are not as considered as the first one.

Experiment 4: Unlike last experiment, the first annotator $\boldsymbol{y}^{(1)}$ has now the higher error rate, with a 70%. The other labelers have a maximum of 30%. It is expected that the weight η of $\boldsymbol{y}^{(1)}$ is lower, so his opinion will be proportionally ignored in the mixture process.

Experiment 5: Ascending error rate values are evaluated in this case, from 30% to 90%, 15 by 15. As a general high error is presented among the annotators, it is expected that the final error is relatively high as well. An outstanding performance of the proposed method is evidenced, where a total error avoidance is accomplished.

Experiment 6: The case of same error rates in all labelers is assessed. The weight η is the same for each annotator, so the improvement in the results is slightly better.

Experiment 7: This is a similar case to the last one, but with a lower general error rate. The method with the exponential function, shown in equation (4), presents better results than the rest of the methods, as expected.

An accomplishment of this work is the clear recognition of the best labelers, with only the natural structure of data. A general improvement is evidenced in the quality measures in every experiment performed, compared with the other conventional methods. We highlight the relevance of assigning weight to the opinions of the experts.

V. CONCLUSION AND FUTURE WORK

We proved experimentally that the proposed approach is capable of quantifying the confidence of a set of reliable labels, taking into account the given information by a group of experts and the variation in the natural structure of the data. In general, the use of this multi-labeler strategy provides a significant improvement in the classifiers design in comparison to the single-labeler approach. In addition, the proposed method has the capability of reducing the influence of wrong labelers establishing penalties and punishing to these bad experts, keeping a good performance in comparison with conventional methods. For future work, we are aiming to explore different alternatives for optimization procedures, to find more suitable penalty values that allow to identify bad annotators in a clearer way, and reduce their relevance. We are aiming also to explore different data sets and multi-labeler cases to apply and improve the algorithm.

REFERENCES

[1] O. Dekel and O. Shamir, "Good learners for evil teachers," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 233–240.

[2] Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy, "Active learni ng from crowds," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 1161–1168.

[3] O. Dekel, C. Gentile, and K. Sridharan, "Selective sampli ng and active learning from single and multiple teachers," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2655–2697, 2012.

[4] S. Murillo, D. Peluffo, and G. Castellanos, "Support vec tor machine-based approach for multi-labelers problems," in *European Symposium on Artificial Neural Networks, Computational Inteligence a nd Machine Learning*, 2013.

[5] S. Murillo-Rendon,´ D. Peluffo-Ordonez,´ J. D. Arias-Londono, and C. G. Castellanos-Dom´ınguez, "Multi-labeler analysis for bi-class problems based on soft-margin support vector machines," in *Natural and Artificial Models in Computation and Biology*. Springer, 2013, pp. 274–282.

[6] O. Dekel, "From online to batch learning with cutoff-ave raging," in *Advances in Neural Information Processing Systems*, 2009, pp. 377–384.

[7] D. H. Peluffo-Ordo´nez,˜ S. M. Rendon,´ J. D. Arias-Londono,˜ and G. Castellanos-Dom´ınguez, "A multi-class extension for multi-labeler support vector machines." in *ESANN*, 2014.

[8] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multicl ass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.

[9] D. H. Peluffo Ordonez, J. L. Rodriguez Sotelo, and C. G. Castel-lanos Dominguez, "Estudio comparativo de m etodos´ de seleccion´ de caracter´ısticas de inferencia supervisada y no supervisada," 2009.

[10] G. Hamerly and C. Elkan, "Alternatives to the k-means algo rithm that find better clusterings," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 600–607.

[11] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[12] M. Lichman, "UCI machine learning repository," 2013. [On line]. Available: http://archive.ics.uci.edu/ml