ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230397

Robust Assignment of Labels for Active Learning with Sparse and Noisy Annotations

Daniel Kałuża[a, b;*], **Andrzej Janusz**[a, b] **and Dominik Ślęzak**[a, b]

[a]Institute of Informatics, University of Warsaw, Poland
[b]QED Software, Poland
ORCiD ID: Daniel Kałuża https://orcid.org/0000-0002-2544-5052,
Andrzej Janusz https://orcid.org/0000-0002-9763-1399, Dominik Ślęzak https://orcid.org/0000-0003-2453-4974

Abstract. The performance of classification models depends on the quality of training data labels. Unfortunately, acquiring good-quality annotations for many tasks is infeasible or too expensive. To address this challenge, active learning algorithms are commonly employed to select only the most relevant data for labeling. However, this is possible only when the quality and quantity of labels acquired from experts are sufficient. In many applications, a trade-off between annotating individual samples by multiple annotators to increase label quality vs. annotating new samples to increase the total number of labeled instances is necessary. In this paper, we address the issue of faulty data annotations in the context of active learning. In particular, we propose two novel annotation unification algorithms that utilize unlabeled parts of the sample space. The proposed methods require little to no intersection between samples annotated by different experts. Our experiments on four public datasets indicate the robustness and superiority of the proposed methods in both, the estimation of the annotator's reliability, and the assignment of actual labels, against the state-of-the-art algorithms and the simple majority voting.

1 Introduction

Supervised learning algorithms are commonly used to create prediction models for various classification tasks. The quality of most of these machine learning models heavily depends on the labeled dataset used during model construction. In real-life scenarios, we often start with no or only a few labeled samples, as the data annotation process is expensive and requires laborious human involvement. To make this process more cost-effective, active learning algorithms are commonly employed [17].

Active learning can be defined as *a set of algorithms that, given a limited labeling budget, try to obtain the best possible model under the assumption that they can iteratively query an oracle (usually human experts) to annotate chosen samples.* In some cases, labels can be obtained in an automated manner, e.g., using computer simulations. However, for many classification problems, such as security alert notifications [6], humans have to manually annotate selected samples. Since it requires considerable domain knowledge and experience, we usually refer to the annotators as experts.

As humans are imperfect by nature, the acquired annotations might contain mistakes, influencing the quality of obtained models. The frequency of those mistakes usually depends on the difficulty of the task itself and the annotators' expertise. If errors occur too often and the quality of acquired labels is insufficient, corrective measures must be used. In this field, there are two dominant approaches: annotation unification algorithms [16] (also known as consensus algorithms), and faulty label identification and removal methods [8].

The first approach involves using input from multiple human experts to create a more accurate label for a given sample. It takes advantage of the fact that some of the experts will provide correct information. However, since multiple experts are typically needed to label each sample, it introduces a trade-off between label quality and the number of labeled samples due to limited resources. The second approach involves identifying and removing mislabeled samples, but this may result in the removal of correctly labeled instances and oversimplification of the model, especially in low-budget annotation scenarios or imbalanced datasets. Therefore, in this work, we focus on the label unification algorithms to improve label accuracy without sacrificing important details about complex data instances.

In this paper, we propose two algorithms based on the Expectation-Maximization (EM) technique and an intuitive idea to augment every expert using a machine learning model. A detailed description of our approach is available in Section 3. The proposed algorithms do not have the major drawback of requiring many annotations per sample to achieve high-quality labels. We compare our methods with baseline reference, i.e., the majority voting and commonly used EM-based algorithm, in experiments on four datasets. Our experiments are described in Section 5. Since two of the datasets used in experiments are highly imbalanced, applying the most commonly used probability cut-off of 0.5 to assign labels leads to poor performance of models according to metrics adjusted for the imbalanced classification, such as the balanced accuracy (BAC). To address this issue, a novel cut-off computation method is proposed in Section 4. The proposed method can be used even without prior knowledge about class distribution, which is suitable for typical active learning scenarios.

2 Related Work

Reaching a consensus among labelers is one of the fundamental issues for active learning research [17]. In this setting, the main objective is to iteratively select the most informative unlabeled samples and request their labels from an oracle, e.g., human annotators or other labeling sources. This approach has been successfully applied

* Corresponding Author. Email: d.kaluza@mimuw.edu.pl

to various classification tasks such as text analysis and classification [19], image classification [3, 18], and medical diagnosis [1, 22]. The most popular approach to the active selection of training instances is the so-called pool-based uncertainty sampling [13]. It assumes that there is an unlabelled pool of data available, from which an active learning algorithm can select the next batch of samples to be annotated by the oracle in the next labeling iteration. Data instances are chosen for labeling based on some estimation of the prediction uncertainty that can be computed using various approaches [4, 21].

Active learning has also been applied to many other types of prediction tasks, such as multi-label classification, where each sample may belong to multiple classes simultaneously [11]. In this case, the annotation process becomes even more complex, since multiple labels must be assigned to each sample. Approaches that have been proposed to address this issue include models investigating correlations between label occurrences or methods that select samples based on the uncertainty of the entire label set. These methods have been shown to be effective in reducing labeling costs and improving the performance of multi-label classification models [15]. However, active learning has also been successful for regression problems [7] and many other ML tasks. A comprehensive survey of active learning applications and sample selection techniques can be found in [17].

In practice, annotations provided by human labelers quite often contain errors or inconsistencies which can negatively impact the performance of active learning algorithms. A number of research papers have addressed this issue by proposing annotation aggregation methods that can improve the quality of labels. For example, there are methods that combine multiple annotations using majority voting [20] or EM-based algorithms [16]. Some of the recent approaches include learning-based methods that incorporate information about annotator expertise to improve annotation quality [9]. An example of such an approach is the multi-label consensus maximization for ranking (MLCM-r) algorithm proposed by [23]. Another example is the Dawid-Skene model [2]. It assumes that annotators have different error rates for different decision classes and models the probability of a correct label for each sample, given the annotations provided by multiple annotators. Additionally, it uses the EM algorithm to estimate the true labels of the samples and the reliability of each annotator. Several studies have demonstrated the effectiveness of these approaches in reducing the impact of noisy annotations on the performance of active learning algorithms [5] and in scenarios where federated learning techniques were applied [24].

3 Annotations Unification Algorithms

In this section, we delve into the details of proposed algorithms denoted as inferred consensus and simulated consensus algorithms. We consider simulated consensus as a more stable and refined version of the inferred consensus algorithm, however, we present both to comprehensively describe the intuition behind them. Both proposed algorithms have been developed to overcome a major drawback of consensus algorithms, i.e., degradation of performance if many samples are not labeled by multiple experts. We have developed them as extensions of the EM algorithm, as it is the most well-known consensus algorithm, tested in many production implementations. Actually, proposed extensions are independent of the EM itself and can be viewed as metatechniques. We are convinced that they might also be used with other annotation unification algorithms and lead to a refined performance in many circumstances. However, as our experiments cover only the case when they are used together with EM, we will describe them in that context in the rest of this section.

3.1 Expectation-maximization

The application of the EM algorithm to the task of estimation of the labels based on multiple noisy annotations has been originally proposed by Raykar et al. [16]. It was shown to be a robust solution when labels are abundant. Here we briefly paraphrase the theory for binary classification, but it can also be easily used for multi-label scenarios which can be modeled as multiple binary classifications or extended to multi-class problems as shown in the original paper.

Let us denote a true label of the sample i as y_i, a label assigned to this sample by expert j as y_i^j, the representation of this sample as x_i, the number of all samples as N, and the number of all experts as R. As this work focuses on sparse annotations, we denote indices of samples annotated by the expert j as $S^j \subseteq \{1, \ldots, N\}$, and the set of experts that labeled the sample i as $E_i \subseteq \{1, \ldots, R\}$.

This probabilistic algorithm makes the following simplifications:

- Each expert j is modeled by two latent variables measuring expertise for the given class, namely specificity (true negative rate) β^j and sensitivity (true positive rate) α^j.
- Probability that an expert assigns a specific class to the sample depends only on the true hidden label of this sample and latent variables of this expert. In other words, they do not depend on the representation of this sample given the true label. I.e.:
$$P(y_i^j = 1|x_i, y_i) = P(y_i^j = 1|y_i).$$
- Each expert annotates samples independently from other annotators, thus assigned classes are independent given the true labels.
$$P(y_i^j = 1|y_i, y_i^k) = P(y_i^j = 1|y_i) \quad \text{if } j \neq k.$$

The EM algorithm starts by initializing the first estimated probability of true labels with majority voting and then iteratively repeats E and M steps until convergence to stable parameters and probability estimation of true labels.

3.1.1 E-step

We will denote the set of all learned parameters of the algorithm as Θ, containing α, β, and the parameters of the machine learning model if one is used for posterior probability estimation. Then, based on the independence of the annotators given a true label and Bayes' theorem, the probability of a positive class $\mu_i = P(y_i = 1|y_i^1, ...y_i^R, \Theta, x_i)$ can be written as:

$$\mu_i = \frac{P(y_i^1, ...y_i^R|y_i = 1, \Theta) \cdot P(y_i = 1|\Theta, x_i)}{P(y_i^1, ...y_i^R|\Theta, x_i)} \tag{1}$$

$$\propto P(y_i^1, ...y_i^R|y_i = 1, \Theta) \cdot P(y_i = 1|\Theta, x_i). \tag{2}$$

Where $P(y_i = 1|\Theta, x_i)$ is posterior probability and can be modeled with a machine learning model. We denote it as p_i. $P(y_i^1, ...y_i^R|\Theta, x_i)$ does not depend on the label, thus it is of no interest to us and can be handled by normalization of scores to a proper probability distribution. If we define $a_i = P(y_i^1, ...y_i^R|y_i = 1, \alpha)$ and $b_i = P(y_i^1, ...y_i^R|y_i = 0, \beta)$, we can rewrite equation for μ_i as:

$$\mu_i = \frac{a_i p_i}{a_i p_i + b_i(1 - p_i)}, \tag{3}$$

$$a_i = \prod_{j \in E_i} [\alpha^j]^{y_i^j} [1 - \alpha^j]^{(1 - y_i^j)}, \tag{4}$$

$$b_i = \prod_{j \in E_i} [\beta^j]^{(1 - y_i^j)} [1 - \beta_i^j]^{y_i^j}. \tag{5}$$

The last set of equations can be used to efficiently compute the expected probability for the positive class.

3.1.2 M-step

The maximization step is used to update the parameters Θ of the algorithm. The equations resulting from computing the gradient of log-likelihood of estimated labels over the parameters α, β are:

$$\alpha^j = \frac{\sum_{i \in S^j} \mu_i y_i^j}{\sum_{i \in S^j} \mu_i} \quad (6)$$

$$\beta^j = \frac{\sum_{i \in S^j} (1-\mu_i)(1-y_i^j)}{\sum_{i \in S^j} (1-\mu_i)}. \quad (7)$$

An update of the parameters of a machine learning model used for the posterior probability prediction can be done using the regular gradient descent method.

3.2 Inferred consensus

As the performance of the EM algorithm degrades with smaller numbers of annotations for each sample, the main idea of inferred consensus algorithm is to propagate the annotations to unlabelled samples, using the knowledge from the samples that an expert has labeled. The intuition behind this idea is expressed by the following question: "What label do we expect annotator j would have given sample i, which hasn't been annotated by him?" To be able to answer this question and infer the predictions, for every expert a machine learning model is trained on annotations given by this expert.

More formally, for expert j we create a model f^j trained on samples $< x_i, y_i^j >_{i \in S^j}$. Then, this model is used to infer predictions for the whole dataset obtaining new annotations, $y_i'^j = f^j(x_i)$ for $i \in \{1, ..., N\}$ and every expert $j \in \{1, ..., R\}$. As the majority of machine learning models return not only a label but also a probability distribution of classes, we utilize the returned distribution as soft annotations, e.g., an artificial expert says that from its perspective there is a 10% chance that the object has the positive class and 90% chance it belongs to the negative class. Finally, the EM algorithm can be run on inferred annotations y' for all of the samples, potentially leading to a better estimation of the hidden true labels, as we have a full inferred annotation set of size R for every sample.

The algorithm can be presented as the following set of steps:

1. Train the model f^j for each expert using $< x_i, y_i^j >_{i \in S^j}$.
2. Infer predictions $y_i'^j = f^j(x_i)$ for $i \in \{1, ..., N\}$
3. Call EM algorithm using y' instead of original annotations.

This algorithm can be viewed as the creation of a new labeling task, that was annotated by artificial experts derived from the original annotators. The advantage of this task is that it is fully labeled by each annotator, therefore it is more suitable for the EM algorithm, and the downside is that artificially created annotators usually have worse quality than original experts, as they are trained only on the small subset of samples, and dependant on the used machine learning model. Moreover, since we associate real experts with models trained on samples annotated by them, we obtain unreliable estimations of experts' reliability, which changes during the annotation process, as the model usually gets better with the increasing number of samples annotated by the expert.

3.3 Simulated consensus

To fix downsides of the inferred consensus algorithm we have prepared a more mature and refined version called simulated consensus. The schematic illustration of the algorithm can be seen on Figure 1. Once again we start by training a machine learning model for each expert, but now we infer predictions only on samples that has not been annotated by this expert, i.e. were not used in training of this model. Then, we use the predictions (in form of probability distributions) as annotations from a new expert fully separate from the original one. In this way we obtain $2R$ annotators, when first R of them are human experts, and second R are simulated. Finally, the EM algorithm is used on the combined, partially soft, annotations set.

The algorithm works as follows:

1. Train the model f^j for each expert using $< x_i, y_i^j >_{i \in S^j}$.
2. Infer predictions $y_i'^j = f^j(x_i)$ for $i \notin S^j$
3. Create new annotations $\hat{y}$ as concatenation of $y^j|_{j \in \{1,...,R\}}$ and y'
4. Call EM algorithm using $\hat{y}$ instead of original annotations.

This algorithm also leads to performing consensus on a set of R annotations for each sample, therefore tackling the major drawback of the original EM in the case of sparse annotations. Moreover, it has several advantages over the inferred consensus algorithm from the theoretical point of view. First of all, it uses the original annotations of the experts and, as they are fully separated from the artificially created annotators, reliably evaluates their quality. We also believe that the quality of the experts might be better evaluated as there is always a quorum of R annotators participating in voting for each sample. Besides, the algorithm is less prone to errors caused by the poor quality of the created models, because their quality is also separately evaluated in EM (this evaluation is reliable as none of the artificial experts make predictions on their training samples) and if they achieve poor performance, their influence in the voting diminishes.

Intuitively, this algorithm also can be viewed as a new labeling task in the following mind experiment. Let us imagine that we have a joint set of original experts' annotations and another group of slightly worse artificial annotators. In the real world, there might exist a person who would return the same annotations as our artificial annotator. Therefore, those should be perfectly fine annotations from the perspective of an annotation unification algorithm, and if they are of poor quality, the algorithm should evaluate them as such and be only slightly influenced by them.

4 Prediction for imbalanced data

The Expectation-Maximization algorithm results in the estimation of a probability distribution of class labels for each sample. Unfortunately, many models cannot be trained using such soft labels. Moreover, we usually expect a definitive answer on whether a sample should be considered as belonging to a particular class or not. If the considered machine learning task corresponds to a balanced distribution, a standard 0.5 cut-off for binary classification or $\frac{1}{K}$, where K is the number of classes, is a sound solution. However, if we are dealing with an imbalanced classification task and try to optimize metric which assigns the same importance to the recognition of each class, such as balanced accuracy, it is not a good threshold.

In some active learning scenarios with noisy annotations, even an approximated class distribution might not be known a priori, e.g. cybersecurity attacks detection. In such cases, threshold tuning is infeasible, as we do not have a reliable validation set to efficiently evaluate various thresholds. Therefore, a reliable method that does not require prior knowledge or true labels is needed. This is why we propose the following method for approximating the class distribution:

1. Compute a probability distribution from the perspective of the model for all samples available during training, let us call distribution for sample i a $\tilde{y}_i$ and probability of class c for this sample $\tilde{y}_{i,c}$.

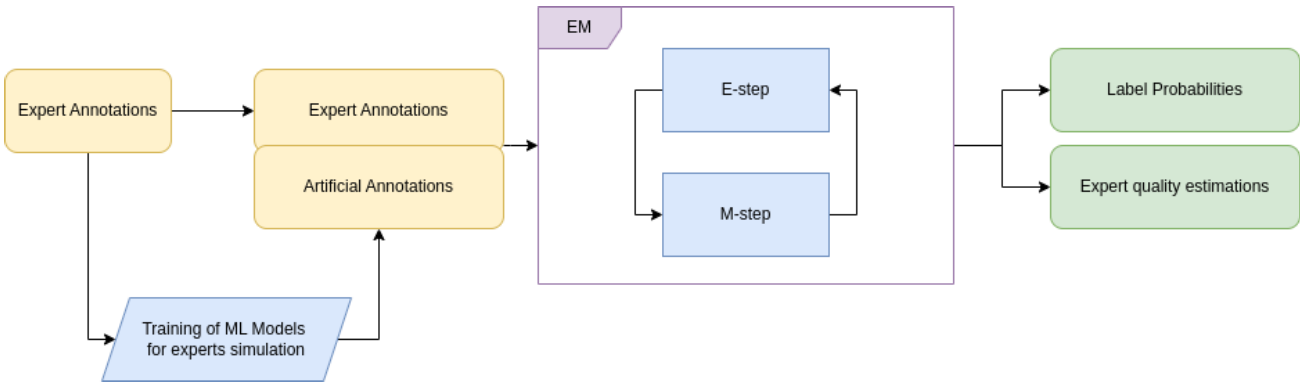

Figure 1. Visualization of Simulated consensus algorithm steps. Algorithm outputs are denoted with green, annotations data yellow and algorithm steps are shown as blue.

2. Compute the average probability for each class across all of the samples. This value will become a threshold adjustment. Let us call it t_c for class c. Formally:

$$t_c = \frac{1}{N} \sum_{i=1}^{N} \tilde{y}_{i,c}.$$

3. For the multi-label classification, we assign class c to the sample i if $\tilde{y}_{i,c} \geq t_c$.
4. For the single-label classification, we choose class c' from the set of all classes C in the following way:

$$c' = \underset{c \in C}{\operatorname{argmax}}\, \tilde{y}_{i,c} - t_c.$$

This method allows us to determine the cut-off without any prior knowledge about the problem. In particular, it can be combined with any EM-based algorithm. In such a case, the considered soft labels correspond to the estimated class probability distributions for each sample. Moreover, it can also be used to choose a threshold for ML models trained using the estimated labels. The probability t_c is computed on all available pool data, using the probabilities predicted by the model. Therefore, no additional computations are needed and the adjustment is independent of the dataset for which predictions are made. When we use this procedure on an unbiased model, it leads to an unbiased estimator of the true class distribution. Moreover, for balanced datasets, it converges with the increasing number of training samples to the regular $\frac{1}{K}$ threshold.

5 Experiments

To properly evaluate proposed algorithms, we have created an experimental setup similar to a real-life active learning scenario. As data labeling by human experts only for the purpose of experiments is too expensive and in real-life scenarios with annotators you usually do not have access to the hidden true labels, we have prepared a randomized procedure for creating annotations based on true labels of known public datasets. The procedure generates a set of binary annotations for a specified number of experts, which is a parameter of the method, in the following way:

1. Number of labeled samples differs for each expert. We model the probability that expert j annotates a sample, denoted as r^j, as a Beta distribution with parameters $\tilde{\alpha} = 1$, $\tilde{\beta} = 20$, therefore average probability is equal to $\frac{1}{21} \approx 0.048$. Intuitively, we can think of it that experts will on average label one per every 21 samples.
2. The fact that an expert j has annotated a sample i is decided by drawing from Bernoulli distribution with a success ratio equal r^j.
3. The hidden true positive and true negative rates of each expert j, denoted as $\hat{\alpha}^j$ and $\hat{\beta}^j$, are drawn from the Beta distribution with parameters $\tilde{\alpha} = 4$, $\tilde{\beta} = 1$, that have the expected value equal 0.8.
4. Classes assigned by the expert j to annotated sample i are drawn from Bernoulli distribution with the probability of success $\hat{\alpha}^j$ if the true label of this sample is positive or is equal to $1 - v$, where v is drawn from Bernoulli with the probability of success equal to $\hat{\beta}^j$ when the true label of the sample i is negative.

We set the number of experts to 15 for our experiments because the randomization of samples annotated by each expert might lead to experts labeling only a few samples (which is consistent with a real-life scenario when somebody leaves a company after a few days of annotation). Using the above procedure, we obtained annotations assigned by diverse artificial experts. Thanks to the fact that it is based on public datasets, we had true labels for all of the samples and the hidden quality of each expert to properly evaluate tested algorithms. As the proposed annotation generation procedure assumes the binary classification task and works in an independent manner for each class, we used one-hot encoded labels of every problem as in a multi-label setting. The evaluation for each dataset was performed five times to obtain the statistical significance of the experiments, each time with a fixed seed creating a different set of expert annotations. The evaluation procedure was as follows:

1. Create expert annotations for a given random seed.
2. Use each consensus method to generate label probabilities and experts' quality estimations.
3. Generate labels using all tested cut-off techniques.
4. Train a machine learning model on the obtained labels.
5. Make predictions with the obtained model on the separate test set. Use the cut-off techniques again to assign labels to test cases.
6. Compute evaluation metrics, both on the consensus results and the model predictions.

We have included a quality assessment of resulting machine learning models, trained on the obtained labels, as this is usually the ultimate result of an active learning system. If a sophisticated consensus method led to a better estimation of labels but would not lead to a better machine learning model, there would be no advantage in using this method in a production environment. To reduce the computational complexity, a machine learning model used for posterior

probability distribution prediction inside EM-based algorithms (that has to be retrained with every iteration) was a dummy model predicting always the class prior probability estimated on the training set regardless of the passed sample. Nevertheless, a regular machine learning model chosen for each task was trained on top of computed labels for the evaluation.

5.1 Evaluation metrics

The evaluation metrics used in our experiments can be divided into three groups. In the first group, there are metrics computed on the probabilities from the annotations unification algorithms. All of these metrics are computed on the set of samples that were annotated by at least one expert in the experiment. We considered the following measures: area under the receiver operating characteristic curve (AUC) with the macro average on the probabilities returned by the algorithms, and balanced accuracy (BAC) on the labels generated by each of the cut-off methods.

The second group contains evaluations of the estimated quality of experts by the compared algorithms. Metrics used in the comparison: the mean absolute error of the true positive rate estimation (MAE), Pearson correlation, and the Spearman rank correlation between the estimated true positive rates and the hidden true positive rates.

The third group contains evaluations of the machine learning models trained on estimated labels. A separate model is trained for each consensus method and each cut-off technique. For each model, BAC score on the test set is reported. As BAC requires labels and the considered models return probability distributions, the same cut-off method as the one used to generate training labels is applied.

5.2 Datasets

We have used four datasets for the purpose of evaluation.

- MNIST [10] - A dataset of handwritten digits, one of the most widely used benchmark datasets in machine learning research.
- firefighters [12] - A dataset with measurements from wearable inertial sensors placed on fire-fighters during various fire and rescue-related activities from the *AAIA'15 Data Mining Competition: Tagging Firefighter Activities at a Fire Scene* organized at the KnowledgePit.ai platform.
- cybersec [6] - A dataset describing cybersecurity network logs with a prediction task to identify events that should be notified as suspicious. This dataset was originally published in a competition *IEEE BigData 2019 Cup: Suspicious Network Event Recognition* on the KnowledgePit.ai platform.
- credit-fraud [14] - A public dataset of transactions made by European credit card holders, fully anonymized via PCA transformation. The dataset is publicly available both in the OpenML repository and on the Kaggle competition platform. The prediction task is to detect fraudulent transactions.

Those datasets were chosen to diversify both, domains and class distributions used to evaluate our methods. MNIST is a balanced dataset with ten classes, firefighters data have five classes with slightly imbalanced distribution, cybersec is a binary classification task and has imbalanced distribution with less than 6% of positive samples, and credit-fraud is a binary and highly imbalanced dataset with less than 1% fraud examples. Moreover, all of these datasets required human annotations at some point to create the labels for the corresponding tasks. We cannot be sure whether there are errors in the labels, but such investigation remains outside of the scope of this study. For both MNIST and credit-fraud, test sets for evaluation were created by a stratified split with 40% of all available samples, whereas for cybersec and firefighters, splits from the corresponding data science competitions were used. Moreover, for the cybersec and firefighters datasets, the same preprocessing as described in the referenced competition papers was performed. Additionally, each dataset was min-max scaled.

Model architectures with hyper-parameters used for evaluation are shown in Table 1. For MNIST and firefighters, a logistic regression model with default parameters was used. For cybersec and credit-fraud, the XGBoost classifier was used. Since those are highly imbalanced datasets, an appropriate scaling parameter with a value equal to the ratio of negative and positive samples was used for training the models.

5.3 Consensus methods and cut-off threshold

In our experiments we have evaluated the following consensus methods:

- Simulated consensus - a refined version of the proposed algorithm generating additional annotations for each sample with machine learning models described in Section 3.3.
- Inferred consensus - the first revision of the proposed algorithm, substituting expert annotations with machine learning models described in detail in Section 3.2.
- EM - the original expectation-maximization algorithm.
- Majority voting - the regular majority voting algorithm with a slight modification to make it more comparable with other methods. The modification is as follows - it returns a distribution of votes for individual classes instead of just indicating the class with the highest number of votes.

In both, inferred consensus and simulated consensus, models representing experts had exactly the same architecture and hyperparameters as the final model used in the evaluation. The parameter values are given in Table 1.

Table 1. Machine learning models and their relevant hyperparameters used for each of the machine learning tasks. Default hyperparameters have been omitted, XGBoost library in version 1.6.2 and scikit-learn 0.24.2 were used to train the models.

Dataset	Model	Hyperparameters
MNIST	Logistic Regression	max_iter=500, n_jobs=10
firefighters	Logistic Regression	max_iter=500, n_jobs=10
cybersec	XGBClassifier	neg_pos_ratio=$\frac{\#neg}{\#pos}$, n_estimators=300, max_depth=3, learning_rate=0.05, n_jobs=10
credit-fraud	XGBClassifier	neg_pos_ratio=$\frac{\#neg}{\#pos}$, n_estimators=300, max_depth=3, learning_rate=0.05, n_jobs=10

The following cut-off thresholding techniques were used:

- Default - Default 0.5 threshold used in the majority of machine learning frameworks.
- GT-prior - A threshold computed using true labels from the training pool. This threshold represents the ratio of samples having a particular class to all of the samples in the pool.
- Model-posterior - The proposed thresholding technique that uses the probability distribution predicted for the whole available training data pool, as described in Section 4. Keep in mind that for each

model, the prediction was done over all available samples from the pool, not only those which were annotated by experts.

Those cut-off thresholds were used to generate labels in the same way as described in Section 4. For the purpose of multi-label model training, a probability distribution was compared with the corresponding threshold to determine whether a class should be assigned to the sample. For the BAC estimation, a difference between the maximal predicted probability and the threshold value was used.

6 Results

6.1 Annotations quality

A summary of annotation quality results can be found in Table 2. The simulated consensus algorithm has obtained significantly better results than all other methods on all datasets but firefighters in both ROC AUC and BAC metrics. On the firefighters dataset, the inferred consensus obtained slightly better ROC AUC than the simulated consensus, which turned out to be the second for this metric. Moreover, we computed the one-sided Wilcoxon signed rank test to check the statistical significance of these results. Scores obtained by the simulated consensus turned out to be significantly greater than the scores of the EM algorithm for all of the datasets in terms of both AUC and BAC-model-posterior with a p-value of 0.03125, which we consider a good result taking into account the limited expressiveness of Wilcoxon test. These results show the robustness and superiority of the proposed annotation unification algorithm.

Noteworthy are also the results of the BAC-model-posterior cut-off, which obtained comparable performance for the balanced datasets and better results than the default threshold for most of the consensus methods on imbalanced dataset combinations. For some imbalanced cases (cybersec and credit-fraud for the inferred consensus and the EM method), it led to good quality labels even when all other cut-off strategies failed. Tested using the one-sided Wilcoxon singed rank test against the default cut-off method, it obtained p-values: 0.026, and 0.001 for the cybersec and credit-fraud datasets, respectively. It suggests that this technique, which does not require a priori knowledge about label distribution, is the safest choice for new active learning scenarios.

6.2 Expert's reliability estimation

Results of experts' true positive rate estimations are shown in Table 3. As suspected, the proposed inferred consensus method leads to distortion of expert reliability estimation. Therefore, it obtains larger mean absolute errors than the regular EM algorithm. Interestingly, the inferred consensus still results in greater correlations for the MNIST and firefighters datasets, which might be caused by better estimation of actual labels.

Nevertheless, the refined version of our algorithm, i.e. simulated consensus, achieves highly superior scores in all three metrics for all of the datasets. The p-values of the one-sided Wilcoxon rank test were: 0.03125, 0.06250, 0.31250, and 0.03125 for MNIST, firefighters, cybersec, and credit-fraud, respectively. The same p-values were obtained for both correlation metrics. Similar results were obtained for MAE: 0.03125, 0.03125, 0.09375, and 0.03125 for the corresponding datasets. Therefore, leading to statistically significant differences in two datasets for correlations and for three datasets for MAE. Noteworthy is the fact that due to the relatively small number of experiment repetitions, the expressiveness of the statistical test was severely limited. However, it still shows the potential of our method considering the fact that the MAE metric on MNIST and credit-fraud datasets was two times smaller on average than for other methods.

6.3 Quality of trained models

Results of model-related metrics can be found in the supplementary materials *Appendix A*, available in the pre-print version of our paper[1]. Our methods led to better machine learning models on the MNIST and firefighters datasets. The simulated consensus model achieved BAC of $0.878(\pm 0.003)$ with model-posterior cut-offs technique on the MNIST dataset and the inferred consensus model achieved BAC of $0.791(\pm 0.012)$, also with model-posterior cut-offs, on the firefighters dataset. This finding is consistent with the label quality results. Surprisingly, on both imbalanced datasets classical majority voting with the default 0.5 threshold achieved better performance than any other model, i.e., $0.773(\pm 0.015)$, and $0.758(\pm 0.019)$ for the cybersec and credit-fraud datasets, respectively. This result is interesting, as other methods have obtained distinctively better label quality estimations on those datasets. The 0.5 threshold on model predictions looks sound from our perspective, as those models were trained with scaled weights for each class to balance the training data, however, we do not have a good explanation for why this threshold is also good for assigning labels for the majority voting algorithm. Therefore, as there is no clear correlation between labels and the resulting model's quality for the imbalanced datasets, this remains a topic for future research.

7 Conclusions

In this paper, we have addressed the issue of faulty data annotations in the context of active learning for classification. We proposed two novel annotation unification algorithms based on Expectation-Maximization (EM) and machine learning models, which require little to no intersection between samples annotated by different experts. Our experiments on four public datasets showed that the proposed methods outperform the state-of-the-art algorithms and simple majority voting, both in terms of the estimation of annotator reliability and the assignment of actual labels. We also proposed a novel cut-off method to tackle the challenge of imbalanced datasets, which can be used even without prior knowledge about class distribution. This approach can be useful in many active learning scenarios where the distribution of classes is unknown or changes over time.

In conclusion, our proposed methods offer an effective solution to the issue of faulty data annotations. By utilizing unlabeled parts of the sample space and incorporating machine learning models, we can improve the quality of labeled datasets and ultimately enhance the performance of supervised classification algorithms. We hope that our work will contribute to further advancements in this field and encourage more research on consensus algorithms for data labeling.

Moreover, our research opens new, as far as we know, yet unexplored topics. Namely, one may ask why for some datasets labels quality does not clearly correlate with the quality of trained machine learning models. As this has a strong influence on all actively annotated machine learning tasks, it requires additional investigation in the future. Of course, this observation might be a result of a relatively small number of experiments, therefore to properly confirm the findings of this paper additional experiment repetitions and validation on new datasets are needed.

[1] https://arxiv.org/abs/2307.14380

Table 2. Results of annotation quality metrics, each dataset has a separate subsection. Each row features the results of one annotation unification method for the corresponding dataset. The first column named AUC denotes the area under the ROC curve computed between obtained probabilities and true labels for annotated samples. The rest of the columns denote the balanced accuracy between labels obtained with the thresholding method indicated in the column name and true labels for annotated samples. Standard deviations across the experiments are shown in brackets next to each value. Bold values indicate the largest value in the AUC column and across all BAC columns for each of the datasets.

Method	AUC	BAC-default	BAC-GT-prior	BAC-model-posterior
MNIST				
Simulated consensus	$\mathbf{0.988(\pm 0.002)}$	$\mathbf{0.911(\pm 0.010)}$	$0.908(\pm 0.010)$	$0.907(\pm 0.011)$
Inferred consensus	$0.978(\pm 0.001)$	$0.870(\pm 0.004)$	$0.868(\pm 0.004)$	$0.867(\pm 0.005)$
EM	$0.882(\pm 0.018)$	$0.588(\pm 0.033)$	$0.589(\pm 0.034)$	$0.590(\pm 0.034)$
Majority Voting	$0.801(\pm 0.008)$	$0.405(\pm 0.030)$	$0.419(\pm 0.020)$	$0.459(\pm 0.024)$
firefighters				
Simulated consensus	$0.979(\pm 0.009)$	$0.872(\pm 0.046)$	$0.874(\pm 0.042)$	$\mathbf{0.875(\pm 0.042)}$
Inferred consensus	$\mathbf{0.985(\pm 0.003)}$	$0.845(\pm 0.051)$	$0.840(\pm 0.047)$	$0.842(\pm 0.047)$
EM	$0.875(\pm 0.027)$	$0.647(\pm 0.053)$	$0.681(\pm 0.040)$	$0.687(\pm 0.036)$
Majority Voting	$0.798(\pm 0.016)$	$0.581(\pm 0.034)$	$0.569(\pm 0.041)$	$0.573(\pm 0.037)$
cybersec				
Simulated consensus	$\mathbf{0.909(\pm 0.022)}$	$0.635(\pm 0.054)$	$\mathbf{0.887(\pm 0.020)}$	$0.873(\pm 0.019)$
Inferred consensus	$0.784(\pm 0.131)$	$0.500(\pm 0.001)$	$0.556(\pm 0.073)$	$0.729(\pm 0.022)$
EM	$0.876(\pm 0.021)$	$0.821(\pm 0.027)$	$0.515(\pm 0.029)$	$0.827(\pm 0.022)$
Majority Voting	$0.797(\pm 0.023)$	$0.805(\pm 0.025)$	$0.789(\pm 0.041)$	$0.789(\pm 0.041)$
credit-fraud				
Simulated consensus	$\mathbf{0.869(\pm 0.045)}$	$0.538(\pm 0.035)$	$0.683(\pm 0.178)$	$\mathbf{0.838(\pm 0.057)}$
Inferred consensus	$0.747(\pm 0.151)$	$0.500(\pm 0.000)$	$0.628(\pm 0.159)$	$0.769(\pm 0.145)$
EM	$0.801(\pm 0.061)$	$0.616(\pm 0.059)$	$0.500(\pm 0.000)$	$0.781(\pm 0.052)$
Majority Voting	$0.807(\pm 0.045)$	$0.810(\pm 0.041)$	$0.804(\pm 0.051)$	$0.804(\pm 0.051)$

Table 3. Results of experts' quality estimation metrics. Each dataset has a separate subsection. Each row features results for one annotation unification method for the corresponding dataset. The first column, named MAE, denotes the mean absolute error across estimations of true positive rates for experts. Pearson and Spearman indicate values of the corresponding correlation coefficients between the estimated true positive rates and the ground truths assigned during the experiment setup. Standard deviations across the experiments are shown in brackets next to each value. Bold values indicate the smallest MAE or the largest correlation for each of the datasets.

Method	MAE	Pearson	Spearman
MNIST			
Simulated consensus	$\mathbf{0.045(\pm 0.009)}$	$\mathbf{0.902(\pm 0.044)}$	$\mathbf{0.894(\pm 0.051)}$
Inferred consensus	$0.175(\pm 0.009)$	$0.775(\pm 0.063)$	$0.763(\pm 0.065)$
EM	$0.090(\pm 0.020)$	$0.757(\pm 0.077)$	$0.671(\pm 0.114)$
Majority Voting	NA	NA	NA
firefighters			
Simulated consensus	$\mathbf{0.083(\pm 0.013)}$	$\mathbf{0.689(\pm 0.109)}$	$\mathbf{0.700(\pm 0.096)}$
Inferred consensus	$0.179(\pm 0.012)$	$0.608(\pm 0.088)$	$0.677(\pm 0.056)$
EM	$0.122(\pm 0.028)$	$0.567(\pm 0.149)$	$0.566(\pm 0.190)$
Majority Voting	NA	NA	NA
cybersec			
Simulated consensus	$\mathbf{0.065(\pm 0.015)}$	$\mathbf{0.756(\pm 0.129)}$	$\mathbf{0.713(\pm 0.220)}$
Inferred consensus	$0.275(\pm 0.073)$	$0.358(\pm 0.230)$	$0.408(\pm 0.280)$
EM	$0.101(\pm 0.032)$	$0.689(\pm 0.226)$	$0.634(\pm 0.175)$
Majority Voting	NA	NA	NA
credit-fraud			
Simulated consensus	$\mathbf{0.126(\pm 0.053)}$	$\mathbf{0.456(\pm 0.261)}$	$\mathbf{0.448(\pm 0.199)}$
Inferred consensus	$0.268(\pm 0.053)$	$0.164(\pm 0.178)$	$0.221(\pm 0.147)$
EM	$0.250(\pm 0.025)$	$0.211(\pm 0.162)$	$0.253(\pm 0.122)$
Majority Voting	NA	NA	NA

Acknowledgements

This research was co-funded by Smart Growth Operational Programme 2014-2020, financed by European Regional Development Fund, in frame of project POIR.01.01.01-00-0213/19, operated by National Centre for Research and Development in Poland.

References

[1] Rafael S Bressan, Pedro H Bugatti, and Priscila TM Saito, 'Breast cancer diagnosis through active learning in content-based image retrieval', *Neurocomputing*, **357**, 1–10, (2019).

[2] A. P. Dawid and A. M. Skene, 'Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm', *Journal of the Royal Statistical Society Series C: Applied Statistics*, **28**(1), 20–28, (12 2018).

[3] Yarin Gal, Riashat Islam, and Zoubin Ghahramani, 'Deep bayesian active learning with image data', in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1183–1192. JMLR.org, (2017).

[4] Yarin Gal, Riashat Islam, and Zoubin Ghahramani, 'Deep bayesian active learning with image data', in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1183–1192. JMLR.org, (2017).

[5] Shahana Ibrahim, Xiao Fu, Nikos Kargas, and Kejun Huang, *Crowdsourcing via Pairwise Co-Occurrences: Identifiability and Algorithms*, Curran Associates Inc., Red Hook, NY, USA, 2019.

[6] Andrzej Janusz, Daniel Kałuza, Agnieszka Chądzyńska-Krasowska, Bartek Konarski, Joel Holland, and Dominik Ślęzak, 'Ieee bigdata 2019 cup: Suspicious network event recognition', in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5881–5887, (2019).

[7] Daniel Kaluza, Andrzej Janusz, and Dominik Slezak, 'EVEAL - expected variance estimation for active learning', in *IEEE International Conference on Big Data, Big Data 2022, Osaka, Japan, December 17-20, 2022*, eds., Shusaku Tsumoto, Yukio Ohsawa, Lei Chen, Dirk Van den Poel, Xiaohua Hu, Yoichi Motomura, Takuya Takagi, Lingfei Wu, Ying Xie, Akihiro Abe, and Vijay Raghavan, pp. 6222–6231. IEEE, (2022).

[8] Davood Karimi, Haoran Dou, Simon K. Warfield, and Ali Gholipour, 'Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis', *Medical Image Analysis*, **65**, 101759, (2020).

[9] Ashish Kulkarni, Narasimha Raju Uppalapati, Pankaj Singh, and Ganesh Ramakrishnan, 'An interactive multi-label consensus labeling model for multiple labeler judgments', in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, (2018).

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, **86**(11), 2278–2324, (1998).

[11] Xin Li and Yuhong Guo, 'Active learning with multi-label svm classification', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, p. 1479–1485. AAAI Press, (2013).

[12] Michał Meina, Andrzej Janusz, Krzysztof Rykaczewski, Dominik Ślęzak, Bartosz Celmer, and Adam Krasuski, 'Tagging firefighter activities at the emergency scene: Summary of aaia'15 data mining competition at knowledge pit', in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 367–373, (2015).

[13] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier, 'How to measure uncertainty in uncertainty sampling for active learning', *Mach. Learn.*, **111**(1), 89–122, (jan 2022).

[14] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi, 'Calibrating probability with undersampling for unbalanced classification', in *2015 IEEE Symposium Series on Computational Intelligence*, pp. 159–166, (2015).

[15] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang, 'Correlative multi-label video annotation', in *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, p. 17–26, New York, NY, USA, (2007). Association for Computing Machinery.

[16] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy, 'Learning from crowds', *Journal of Machine Learning Research*, **11**(43), 1297–1322, (2010).

[17] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang, 'A survey of deep active learning', *ACM Comput. Surv.*, **54**(9), (oct 2021).

[18] Łukasz Rączkowski, Marcin Możejko, Joanna Zambonelli, and Ewa Szczurek, 'Ara: Accurate, reliable and active histopathological image classification framework with bayesian deep learning', *Scientific Reports*, **9**, 14347, (2019).

[19] Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar, 'Deep active learning for named entity recognition', in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 252–256, Vancouver, Canada, (August 2017). Association for Computational Linguistics.

[20] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis, 'Get another label? improving data quality and data mining using multiple, noisy labelers', in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, p. 614–622, New York, NY, USA, (2008). Association for Computing Machinery.

[21] Jiaxi Wu, Jiaxin Chen, and Di Huang, 'Entropy-based active learning for object detection with progressive diversity constraint', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9397–9406, (June 2022).

[22] Xing Wu, Cheng Chen, Mingyu Zhong, Jianjia Wang, and Jun Shi, 'Covid-al: The diagnosis of covid-19 with deep active learning', *Medical Image Analysis*, **68**, 101913, (2021).

[23] Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, and Philip S. Yu, 'Multilabel consensus classification', in *2013 IEEE 13th International Conference on Data Mining*, pp. 1241–1246, (2013).

[24] Bixiao Zeng, Xiaodong Yang, Yiqiang Chen, Hanchao Yu, and Yingwei Zhang, 'Clc: A consensus-based label correction approach in federated learning', *ACM Trans. Intell. Syst. Technol.*, **13**(5), (jun 2022).