A review and experimental analysis of active learning over crowdsourced data

Sayin, Burcu; Krivosheev, Evgeny; Yang, Jie; Passerini, Andrea; Casati, Fabio

DOI
10.1007/s10462-021-10021-3

Publication date
2021

Document Version
Final published version

Published in
Artificial Intelligence Review

Artificial Intelligence Review
https://doi.org/10.1007/s10462-021-10021-3

A review and experimental analysis of active learning over crowdsourced data

Burcu Sayin[1] · **Evgeny Krivosheev**[1] · **Jie Yang**[2] · **Andrea Passerini**[1] · **Fabio Casati**[1,3]

Abstract
Training data creation is increasingly a key bottleneck for developing machine learning, especially for deep learning systems. Active learning provides a cost-effective means for creating training data by selecting the most informative instances for labeling. Labels in real applications are often collected from crowdsourcing, which engages online crowds for data labeling at scale. Despite the importance of using crowdsourced data in the active learning process, an analysis of how the existing active learning approaches behave over crowdsourced data is currently missing. This paper aims to fill this gap by reviewing the existing active learning approaches and then testing a set of benchmarking ones on crowdsourced datasets. We provide a comprehensive and systematic survey of the recent research on active learning in the hybrid human–machine classification setting, where crowd workers contribute labels (often noisy) to either directly classify data instances or to train machine learning models. We identify three categories of state of the art active learning methods according to whether and how predefined queries employed for data sampling, namely fixed-strategy approaches, dynamic-strategy approaches, and strategy-free approaches. We then conduct an empirical study on their cost-effectiveness, showing that the performance of the existing active learning approaches is affected by many factors in hybrid classification contexts, such as the noise level of data, label fusion technique used, and the specific characteristics of the task. Finally, we discuss challenges and identify potential directions to design active learning strategies for hybrid classification problems.

Keywords Active learning · Crowdsourcing · Human in the loop · Classification

✉ Burcu Sayin
burcu.sayin@unitn.it

[1] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

[2] Web Information Systems, Delft University of Technology, Delft, The Netherlands

[3] Servicenow, Inc, Santa Clara, CA, USA

Published online: 30 May 2021

1 Introduction

Despite remarkable advances in machine learning (ML), training data remains a key bottleneck for the successful application of ML techniques. Obtaining a large amount of high-quality training data is usually a long, laborious, and costly process. Active learning (AL) provides an effective means to accelerate the process, by iterating data labeling and model training, and identifying at each iteration which data to label next, to converge more rapidly and effectively to an accurate model. *Crowdsourcing* is often used in conjunction with ML, both as a way to collect labeled data efficiently and as a way to assist trained models for predictions where the model confidence is not deemed sufficient (Callaghan et al. 2018; Krivosheev et al. 2021). Despite their joint usage, the interaction between AL and crowdsourcing has been largely unexplored. This interaction is non-trivial for many reasons: for example, crowdsourcing typically produces rather noisy labels and the impact of such noise on ML algorithm confidence estimation and calibration is still unclear. Furthermore, at every AL iteration, we are faced with several choices, from how to aggregate crowd votes on a label to whether we should ask the crowd to label new data items or verify (reduce the noise on) already labeled items—and these choices may impact AL performance.

This paper reviews existing AL approaches and investigates their performance in the hybrid human–machine classification setting, where crowd workers contribute labels (often noisy) to either directly classify data instances or to train an ML model for classification. Unlike existing surveys (Settles 2010; Aggarwal et al. 2014) that focus on the algorithmic design of AL and the review paper about integrating AL with deep learning (Budd et al. 2019), here we aim at re-evaluating existing AL approaches in terms of cost-effectiveness when data labels are crowdsourced, and so given the constraints of a limited crowdsourcing budget and noisiness of worker-contributed labels.

To this end, we first review existing AL approaches under three categories, based on their reliance on a *strategy*, that is, specific queries to get the sample of interest from the data. These categories are: (i) *fixed-strategy approaches*, that apply a specific item selection method regardless of the data or problem, (ii) *dynamic-strategy* approaches that have a portfolio of strategies and choose one each time they need to sample a batch of items for labeling, based on past performance on that specific problem and data, (iii) *strategy-free* approaches that do not have any apriori selected portfolio of strategies, but rather learn the best strategy from scratch based on the problem, data, and prior experiences. We also review proposals in the literature that discuss how AL can deal with noisy labels. We then report the results of an extensive experimental comparison evaluating the performance of the different AL approaches in human–machine classification.

We evaluate the performance of AL approaches under two different scenarios: (1) *ML only*: this is the "traditional" approach of training a model with AL and then testing its performance on a pool of items. (2) *Hybrid*, where crowd and ML interact also in the classification phase, not just in data labeling. Indeed, many problems we face have a *finite pool*, where the set of items to classify is finite, and there is therefore a trade-off between spending our budget or effort to train an ML model (using AL methods) versus spending that budget to directly classify items in the pool via the crowd, or using a combination of crowd and ML.

To run this comparison, we developed a library of AL approaches collecting implementations provided by the authors when available, and re-implementing them when we could not find existing code. As part of this process, we also created a collection of crowdsourced datasets containing micro-level information (i.e., individual crowd votes), by aggregating

the publicly available ones and adding the ones we collected (note that most available crowdsourced datasets do not make individual votes available). Both the software library[1] and the collection of benchmarking datasets[2] are made freely available to the scientific community. We believe that both the implementations and the crowdsourced micro-data will be an important contribution in their own right given the difficulty we had in obtaining both, despite extensive searches.

An important lesson we learned from the experimentation is that prior conclusions on the performance of AL approaches obtained in non-crowd labeling settings cannot be blindly extended to crowdsourced data. Specifically, strategy-free approaches that have shown to be effective in many contexts do not achieve the best performances across crowdsourced datasets. We speculate that this can be due to the impact of noise in ML model calibration and uncertainty estimation. We also observed that hybrid classification improves the performance of AL approaches over crowdsourced datasets.

In summary, we make the following contributions:

- We identify three categories of AL approaches in the literature and analyze their characteristics and effectiveness.
- We contribute a library of implementations of state-of-the-art AL algorithms and a collection of benchmarking datasets for human–machine classification.
- We report the results of an extensive experimental evaluation, providing insights on the performance of existing AL strategies in hybrid human–machine classification contexts.
- We provide a critical discussion on the main insights that emerged from our analysis, highlighting relevant open challenges and potential future directions to address them.

2 Active learning strategies: a review

AL (Cohn et al. 1996) has been a very lively research field over the last decade. Given a set of items I and an ML algorithm M, AL aims at defining a strategy to progressively sample items from I on which to obtain true labels for, so that M can be trained with a smaller dataset with respect to random item sampling. The underlying assumption is that obtaining training data is costly, and therefore minimizing the size of such dataset for a given target accuracy is highly beneficial (Johnson et al. 2018). In the following, we review existing AL approaches by grouping them in terms of the type of strategy used to choose the sample of interest.

2.1 Fixed-strategy approaches

Pioneering fixed-strategy approaches have been proposed in the 1990s. Seung et al. (1992) proposed the *query by committee (QBC)* approach, which polls a committee of different classifiers trained on the current set of labeled items to predict the label of each unlabeled item. Then, items to label are selected based on the maximum degree of disagreement among the classifiers. They showed that the prediction error decreases exponentially fast

[1] https://tinyurl.com/source-code-data-results.
[2] https://github.com/TrentoCrowdAI/crowdsourced-datasets.

in the number of queries. The approach and experimentation are however limited to parametric learning models with continuously varying weights and cases where learning is perfectly realizable, and the learning algorithm is Gibbs algorithm (Haussler et al. 1991). In Freund et al. (1997), they proved that *QBC* such an exponential decrease is guaranteed for a general class of learning problems. They used two machine classifiers as the "committee", and determined general bounds on both the number of queries and the number of instances to be labeled. Specifically, the paper defines higher and lower bounds for the expected information gain of QBC and proves that if the queries have high expected information gain then the prediction error is guaranteed to decrease rapidly with the number of queries. McCallum and Nigam (1998) follow a similar "committee-based" approach but apply *expectation-maximization (EM)* to determine class probabilities and the extent to which classifiers disagree, and weigh item selection by their *density*, defined as the distance from a document to the others. With this approach, they reduced the required number of labeled documents by 42% over the previous *QBC* approaches.

Lewis and Gale (1994) presented the idea of *uncertainty sampling*, where the intuition is to sample items on which a model M is more uncertain, and this approach has long been a de-facto standard in the AL literature. They showed that aiming at reducing the uncertainty of M significantly decreases the number of items that must be labeled to achieve the target accuracy. Cohn et al. (1994) proposed *selective sampling*, based on the idea of identifying uncertain regions in a vector space used to represent items, and then on selecting items (points in space) from such uncertain regions to minimize them. In the case of support vector machines, uncertainty sampling is implemented by selecting instances that are closest to the decision boundary (Tsai et al. 2010).

Roy and McCallum (2001) introduced the *error-reduction sampling* approach, aiming at selecting items that will reduce the expected error of the active learner in the next test examples. They computed the expected error rate of an item either by using the entropy of the posterior class distribution (log-loss), or by using the posterior probability of the most likely class (0-1 loss). Mozafari et al. (2014) focused on the same objective and proposed the *MinExpError* approach that uses the theory of non-parametric bootstrap (Efron and Tibshirani 1993) to design generic and scalable sampling strategies. First, bootstraps are created and assigned to different classifiers; then, the expected error of these classifiers for every single item in the unlabeled data is computed; finally, the items that minimize the expected error are selected. The authors showed that the *MinExpError* algorithm requires significantly fewer labeled items than existing approaches back then.

Probabilistic Active Learning (PAL) (Deroski et al. 2014) combined the idea of uncertainty sampling and the expected error reduction with smoothness assumption (Chapelle et al. 2010). The underlying assumption is that if two items are close in the feature space, then their labels should also be close. An item is represented by two attributes; (i) the total number of labeled instances in the neighborhood of the item, and (ii) posterior estimate for the total number of the positive labeled neighbor set. This approach uses probabilistic estimates to investigate the neighborhood statistics of an item (label statistics), and measures the overall gain in classification performance (probabilistic gain) in terms of a user-defined point classification performance (Parker 2011). It then selects items that improve the expected probabilistic gain most within their neighborhood. Its time complexity is comparable to uncertainty sampling, and it provides fast and stable performance.

Saar-Tsechansky and Provost (2004) proposed the *Bootstrap-LV* approach, which detects the variance in the probability estimates of bootstrap samples and uses weighted sampling to find the most informative items. Another weighted-sampling approach is known as *Importance-Weighted Active Learning (IWAL)* (Beygelzimer et al. 2009) which applies an

adaptive rejection sampling to each instance and assigns an importance weight (the inverse probability of being retained) to each retained item. Beygelzimer et al. (2010a) improved *IWAL* by using a rejection threshold based on the importance-weighted error estimates that minimize the prediction error. They showed that this approach improves the label complexity which reflects the intrinsic difficulty of the learning problem (Wang 2011; Yan et al. 2019).

Additional strategies include clustering instances and selecting cluster representatives as the most informative items (Brew et al. 2010), or combining representativeness and informativeness of instances to minimize the maximum possible classification loss (i.e. *Query Informative and Representative Examples (QUIRE)* (Huang et al. 2010)).

Although many of these approaches explicitly target generalization performance improvement, by means of expected error reduction (Roy and McCallum 2001; Zhu et al. 2003; Guo and Greiner 2007; Roy and McCallum 2001; Mozafari et al. 2014), or variance reduction (Schein and Ungar 2007; Settles and Craven 2008; Hoi et al. 2006), fixed-strategy approaches are unlikely to work on all scenarios (Baram et al. 2004; Hsu and Lin 2015). The reason is that they rely on intuitions and heuristics that do not generalize to all datasets and ML problems. For example, even in our experiments, discussed next, uncertainty sampling performs well when false positives and false negatives have the same "cost", but less so when errors, and specifically errors of a specific type are more costly than others. This reveals that fixed-strategy approaches are not able to adapt to the data and the problem at hand.

2.2 Dynamic-strategy approaches

Approaches to dynamic strategy selection are in essence based on progressively learning which AL approach works best for the data at hand. This kind of "learning to learn" approach was first proposed by Baram et al. (2004), who showed that one single strategy cannot perform well on all problems. Their approach, named *COMB*, combines a group of AL strategies and dynamically evaluates them to achieve the best possible performance on the problem at hand. Although the online selection of strategies expedites the AL process, combining multiple strategies and evaluating their performance brings two challenges (Baram et al. 2004). First, as dynamic strategies choose the next action by estimating the performance of each AL approach given the current state and the past observations, the quality of such estimation becomes crucial. However, items selected by the active learner are biased to be the "hard" ones and do not reflect the exact distribution of the items, and as such the quality estimation in absence of a test dataset (which is rarely available in AL) is biased. Second, at each batch only the label of the instances proposed by the selected AL approach are available; there is no way to know the consequences of labeling other instances proposed by other strategies. To handle these challenges, *COMB* (Baram et al. 2004) is designed as an adversarial multi-armed bandit problem (MAB) (Auer et al. 1995, 2003; Audibert and Bubeck 2009) combined with the EXP4 algorithm (Auer et al. 2003), where AL strategies are considered as "experts" and unlabeled items are the "slot machines". Thus, the selection of an item is based on the opinion of all experts.

Hsu and Lin (2015) proposed the *Active Learning by Learning (ALBL)* algorithm as an extension of *COMB*. It represents each bandit machine as an AL approach and uses the EXP4.P algorithm (Beygelzimer et al. 2010b) to select a machine adaptively. The main differences between *COMB* and *ALBL* are as follows: (i) while COMB represents each machine as a single unlabeled instance, a machine corresponds to an AL approach

in *ALBL*, (ii) both *COMB* and *ALBL* adopt the EXP4 algorithm, but *COMB* restricts each machine to being pulled only once, while a machine can be pulled many times in *ALBL*, and (iii) *COMB* uses human-designed evaluation criteria based on entropy, while *ALBL* uses an unbiased estimator of the test accuracy (weighted accuracy) to decide the rewards of the single strategies. Hsu and Lin (2015) showed that *ALBL* gives either comparable or better results than *COMB*. In general, the above papers show that ALBL works better than fixed-strategy approaches when the problem is easier to learn, while it is comparable for harder problems (where strategy selection is also more challenging).

While *ALBL* probabilistically blends the items suggested by different AL strategies to select the most informative one, Chu and Lin (2016) proposed blending the strategies themselves to build an aggregated strategy. Their approach, called *LSA (Linear Strategy Aggregation)*, combines *LinUCB (linear upper-confidence-bound)* (Li et al. 2010), a state-of-the-art MAB approach, with the task at hand. They represent experience as the weights with which to aggregate strategies, and adaptively adjust these weights when tackling a new problem. They aim to transfer this experience learned from the model to other AL tasks through biased regularization. They proved that the transfer of the learned experience is beneficial to achieve better performance.

Although dynamic-strategy approaches use bandits to ensemble multiple strategies in the learning process (Baram et al. 2004; Hsu and Lin 2015; Chu and Lin 2016), they still assume that there is a single best combination in each batch (stationary bandits). In so doing they are not robust to non-stationary cases, where the weighting proportions must be adapted over time in the learning process. Recently, strategy-free approaches have been proposed to overcome these limitations.

2.3 Strategy-free approaches

Strategy-free AL processes use prior experience (meta-data) to learn new tasks (*active meta-learning*). The main difference between *dynamic* and *strategy-free* approaches is that the latter does not rely on any human-designed strategies.

In this category, Konyushkova et al. (2017) devised a novel data-driven AL algorithm, named *Learning Active Learning (LAL)*. *LAL* is formulated as a regression problem that learns how to predict the reduction in the expected generalization error when we add a new label to the training set. It uses Monte-Carlo sampling to correlate the test performance directly with the classifier and item properties. Both the classifier and the items are represented with a set of parameters so that *LAL* can sense any change in the training set. As a result, the learning state is continuously tracked as a vector whose elements depend on the state of the current classifier and the selected item. A drawback of this algorithm is being classifier-specific, which is designed as a random forest regressor.

Woodward and Finn (2017) and Fang et al. (2017) use reinforcement learning (RL) for learning an active learner in a data-driven approach. They adopted a stream-based AL process in which the agent observes the data in sequence and decides whether a single item should be labeled by the agent itself or it should be asked to an oracle. Based on the decision, the agent receives a reward and a prediction model is adopted to be used in new tasks. They improved the performance of models, but they tend to learn only from related datasets and domains (Konyushkova et al. 2018.

Many other data-driven approaches for pool-based AL processes have been proposed recently. While Bachman et al. (2017) and Pang et al. (2018b) used RL to build the learning model, Liu et al. (2018) formulated learning AL strategies as an *imitation learning*

problem (i.e., the machine is trained to perform a task from demonstrations by learning a mapping between observations and actions), Contardo et al. (2017) and Ravi and Larochelle (2018) applied *few-shot learning* (i.e. classifying a new data having seen only a few training examples), and Pang et al. (2018a) extended the *LSA* approach using non-stationary multi-armed bandit with expert advice.

Active meta-learning approaches have also been developed in application-specific scenarios. Sun-Hosoya et al. (2018) developed (*ActivMetaL*), an active meta-learning recommender system. *ActivMetaL* keeps the scores of multiple AL approaches on given tasks in a sparsely populated collaborative matrix, predicts the performance of each approach for a new task, and then fills the corresponding row of the matrix for this task. In this way, they predict which algorithm will perform best for the new task/dataset.

While capable of generalizing across learning task, these meta-learning approaches still have many limitations, such as being classifier specific (Bachman et al. 2017; Contardo et al. 2017; Ravi and Larochelle 2018), having a greedy approach and missing a long-term reward (Liu et al. 2018) or being limited to specific domains (i.e. imitation learning, or few-shot learning) (Bachman et al. 2017; Fang et al. 2017; Liu et al. 2018; Sun-Hosoya et al. 2018).

To overcome these limitations, (Konyushkova et al. 2018) proposed an RL variant of their LAL algorithm, named *LAL-RL*, that defines AL as a Markov Decision Process and tries to find the optimal and general-purpose strategy. *LAL-RL* is independent of the dataset and ML classifier (contrarily to *LAL* that is designed for Random Forests), and its objective does not depend on a specific performance measure. The authors show how *LAL-RL* can transfer learned strategies across substantially different datasets.

Recently, Desreumaux and Lemaire (2020) tested the performance of LAL-RL on 20 real-world datasets and compared it to *random sampling* and *uncertainty sampling*. Although LAL-RL shows very good performance on average, it is not always better than random sampling, especially in the case of highly unbalanced datasets (Desreumaux and Lemaire 2020). In addition, their results show that the choice of the model is decisive (i.e. random forest classifier gives better results than logistic regression). They also report that LAL-RL is sensitive to the metric used to evaluate the performance, and it requires optimizing many hyper-parameters. This analysis shows that even general-purpose strategy-free approaches have limitations when dealing with real-world problems.

Although the main objective of these meta-learning approaches is to adapt the prediction/learning model to new environments/tasks, they do not consider the characteristics of the target environment in the prediction model. Hence, they are likely to be effective in similar environments only. Vu et al. (2019) proposed a new approach that learns a good policy directly based on the target environment either by using a pre-trained AL model or learning a new policy from scratch considering the budget for human annotation. They showed that this approach is more effective than the previous work (Fang et al. 2017; Liu et al. 2018) when the source task and the target task are different. Rudovic et al. (2019) proposed a deep Q-learning approach that is capable of dealing with multiple environments by learning a multi-modal AL strategy. They focus on the task of engagement estimation from real-world child-robot interactions during autism therapy. They employ an LSTM network to classify the individual modalities into engagement levels (i.e. low, medium, or high) and feed its predictions into the deep RL agent, making it capable of efficiently personalizing the interaction strategy to the target user.

In summary, the state of the art shows that the existing *active meta-learning* approaches can outperform the fixed-strategy and strategy-free approaches, especially when the dataset is not highly unbalanced. However, most of them do not present

comprehensive benchmarks to prove the transferability of the learned policies into real-world tasks (i.e. when we do not have a separate test set) (Desreumaux and Lemaire 2020). While showing a noticeable improvement in terms of generalization ability with respect to previous approaches, meta-learning approaches still cannot cope with all challenges that are to be faced in real-world scenarios.

Specifically, nearly any real-world application we encountered has two aspects that haven't received much attention so far in AL research: the first is that the amount of noise in the labels is much higher with respect to standard settings where labels are provided by domain experts, and in crowdsourcing there are several trade-offs we can make to reduce the noise and to balance budget vs noise trade-offs (for example, we can collect more votes for the same items and aggregate them to get a more reliable label, or we can pay more budget to ask for highly rated workers, or we can instead focus on getting a larger number of items labeled at low cost, although with higher noise). The second is that the cost of false positive and false negatives (and more generally of different types of errors) is rarely the same and that low calibration error is often a key quality of a good ML model. There are however a few contributions on dealing with noisy labels and we discuss them next.

2.4 Dealing with noisy labels

While crowdsourced labels can be made to be very precise via redundancy and aggressive worker selection/crowd testing strategies, the individual votes are often noisy. There indeed exists a line of work in the AL community that explicitly focuses on dealing with noisy labels. Zhao et al. (2011) proposed combining uncertainty and inconsistency (entropy of the label distribution) measures to select instances. The underlying idea is that the learning strategy should select items that are in unexplored regions or near the ones that may have been mislabeled. They also propose relabeling the mislabeled items via crowdsourcing. They show that when the labels are noisy and the aggregated label is not trustful (i.e. the aggregated label is provided by less than 50% of the workers, who annotated the corresponding item, in binary classification setting), then relabeling significantly improves the performance of AL. Bouguelia et al. (2016) proposed a method for identifying and mitigating mislabeling errors, where they derive an informativeness measure to see how much a queried label would be useful if it was corrected. They show that this approach is more efficient in characterizing label noise compared to the commonly used entropy measure. Then, Bouguelia et al. (2018) extended this approach by measuring how much the queried item's label is likely to be wrong, based on disagreement with the current classification model, without relying on crowdsourcing.

A related line of work aims at addressing label noise by explicitly modeling the uncertainty of annotators. Yan et al. (2010) introduce a model that jointly learns a classifier and infers annotators' reliability, in an active learning setting that involves both data instance and annotators selection. Fang et al. (2012) consider the case when annotators can learn from one another to improve their annotation reliability. Zhong et al. (2015) model a scenario where workers can explicitly express their annotation confidence, by allowing them to choose an unsure option. Yan et al. (2016) further consider labeler properties such as consistency. Yang et al. (2018) extend the problem to enabling deep active learning from crowds, i.e., enabling deep neural networks to actively learn from crowd workers. However, none of them tried to actively estimate the noise level of data during the learning process.

3 Experimental work

As we have seen, the near totality of existing AL approaches (i) assume that oracles provide the gold (ground truth) labels (ii) strive for accuracy as a metric, and (iii) assume classification is done by ML only. In reality, the situation and needs are different, especially when we have access to a large set of human annotators of different reliability. First, labels can be noisy; second, the trade-off between the benefit of a correct classification and cost of an error varies greatly by application, and achieving a low calibration error may be as important as achieving high accuracy; finally, in many scenarios, we do have the option of relying on human classification when ML is uncertain, and in those contexts it is important that ML learns to know when it doesn't know.

In this section, we examine the behavior of AL approaches in crowdsourcing settings. Specifically, we focus on problems where we start from a blank slate, have a pool of items to classify and a crowd at our disposal, and need not only to choose/assess AL approaches but also to assess if the crowd is leveraged only to get labeled data for training or also to perform classification at inference time, as done in hybrid classification contexts (Krivosheev et al. 2018a; Callaghan et al. 2018).

3.1 Problem formulation

We focus on hybrid binary classification problems where classification is accomplished via the combined contribution of humans and machines. The problem can be formulated as follows. We are given a tuple of (I, M, Q, B), where:

- I is a pool of unlabeled items to be classified.
- M is an untrained machine learning classifier.
- Q is an active learning query strategy.
- B is the budget available (expressed in terms of the total number of crowd votes we can ask).

Our aim is to classify all items I via the ML classifier M or/and crowd workers, assuming we do not have any training data to start with. To achieve this, we apply AL strategy Q for querying the most informative items $T \subset I$ that will be annotated by crowd workers on money B. The annotated items T will be used as training data for machine M and finally, I will be classified based on M and the feedback collected from the crowd. Our hybrid AL workflow is described in detail in Algorithm 1.

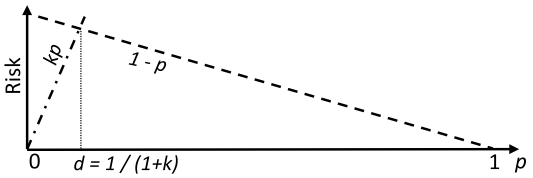

Fig. 1 *Risk score* versus p

Algorithm 1 Hybrid AL workflow

Input: I, M, Q, B

$LI, UI \leftarrow \{\}, I$ # labeled and unlabeled items
$T \leftarrow \{\}$ # training dataset, {(item, label), ..}
$VotesAll \leftarrow \{\}$ # set of crowd votes on items, {(item, worker, vote), ..}
$b \leftarrow 0$ # budget spent
while $b < B$ **do**
  $batch \leftarrow$ select $batchSize$ items from $I (i.e., LI \cup UI)$ via Q
  $VotesBatch \leftarrow$ collect crowd votes for items $batch$
  $VotesAll \leftarrow VotesAll \cup VotesBatch$
  $T \leftarrow AggregateVotes(VotesAll)$ # build training dataset
  train M on T
  test M on I
  update UI, LI, b
end while
classify I based on M and $VotesAll$
return M, *Classified Items*

3.2 A new approach: *Block Certainty*

Most of the AL approaches do not consider cost-sensitive learning scenarios, in which different types of errors can be associated with different costs. This is the typical scenario in medical screening tests for instance, where we try to uncover the presence of a disease. A *false negative* (FN) error (or missed alarm) is in this case extremely critical and much more costly than a *false positive* (FP) one (false alarm). The same is true in many enterprise contexts such as the ones some of the authors face daily, where companies are ok if a customer request is not understood and has to be routed to an agent, but not ok if ML predicts the wrong intent and gives the wrong answer to the customer. An effective ML classifier for this scenario should be trained using a cost-sensitive loss, that potentially trades *precision* for *recall*. We thus propose a simple cost-sensitive AL approach, that we name "Block Certainty", which intrinsically considers the relative harm of FN over FP errors during the AL process.

Let k indicate the relative harm of FN over FP errors, i.e., a single FN error is k times more costly than a FP one. Let p be the probability of an item being positive according to the machine M. Let d be the decision threshold for M. We first investigate how to set d so as to minimize the risk of the classifier M given k. Assuming $p > d$ (item classified as positive), we define the *risk score* for this item as $risk = 1 - p$. Similarly, the probability of an item being negative is $1 - p$. Assuming $p \leq d$ (item classified as negative), the *risk score* for this item will be $risk = k \cdot p$. Figure 1 shows the relation between p and the *risk score*. The optimal classification threshold d is given by the value of p for which the lines $risk = k \cdot p$ and $risk = 1 - p$ intersect, i.e., $d = 1/(1 + k)$.

Algorithm 2 Block Certainty Sampling

Input: I, M, $batchSize$, k

```
p ← M.predictProba(I)
d ← 1/(1+k)
predPos ← [p[i]>d for i in I]
predNeg ← [p[i]≤d for i in I]
PosId ← argmax(predPos, size=d·batchSize)
NegId ← argmax(predNeg, size=(1-d)·batchSize)
return PosId + NegId
```

Then, in Algorithm 2 we define the "Block Certainty" sampling, where at every AL iteration we query predicted positive and negative items with the lowest *risk score* (according to k) for further annotation.

3.3 AL approaches and ML classifier

We examine the following seven AL approaches: (i) *random* (*R*) (items are randomly sampled), (ii) *uncertainty* (*UC*) (Lewis and Gale 1994), (iii) *certainty* (*C*) (the reverse of uncertainty sampling, i.e., the most certain items are sampled), (iv) *block certainty* (*BC*) (our proposal for cost-sensitive sampling), (v) *QUIRE* (*Q*) (Huang et al. 2010), (vi) *MinExpError* (*MinExp*) (Mozafari et al. 2014), and (vii) *meta-learning* (*LAL*) (Konyushkova et al. 2017). We used *R*, *UC*, *C*, *Q*, and *MinExp* as the state of the art fixed-strategies that have been used in most of the comparative analyses in the literature. Since Konyushkova et al. (2017) already proved that *LAL* approach outperforms the state of the art dynamic strategy approach *ALBL* (Hsu and Lin 2015), we tested only *LAL* to have an intuition about adaptive approaches.

Since the *LAL* (Konyushkova et al. 2017) approach is specifically designed to work with the random forest classifier, we chose random forests as the underlying classifier for all AL approaches. We used the implementation provided by the *scikit-learn* library (Pedregosa et al. 2011) with the following parameters: n_estimators = 100, criterion = "gini", max_depth = None, bootstrap = True, class_weight = "balanced", and random_state = 2020. To evaluate the effect of the choice of the classifier on the performance, we additionally evaluated a subset of the AL approaches (excluding *LAL*) using a support vector machine as the underlying classifier.

3.4 Crowdsourcing and evaluation

3.4.1 Crowdsourcing scenarios

We consider the following two crowdsourcing scenarios in our experiments:

- *Unlimited votes* Every single item can be selected an unlimited number of times for a crowdsourced vote, as long as we have an available budget and voters.
- *Limited votes* There is a maximum number of votes *maxVote* that every single item can receive (*maxVote* = 3 in the experiments). Upon reaching this limit, the corresponding item is removed from I and cannot be queried anymore. In principle, this limit could be automatically inferred via meta-learning approaches, but this out of the scope of this paper.

3.4.2 Evaluation scenarios

We evaluate the results under two different cases:

- *ML (only)*: We use the trained M to predict the label of each item in the pool I and evaluate its performance compared to the ground truth labels.
- *ML+C*: We take crowdsourced labels for items LI and use M to predict the label for the unlabelled items in UI. We evaluate the performance of this combined crowd-machine classifier by comparing its predictions with the ground truth labels.

3.4.3 Label fusion methods

Since we use crowd answers instead of gold labels in the learning phase, it is important to consider that they may yield low-quality or noisy labels (Zheng et al. 2017). In the crowdsourcing literature, this problem is addressed by assigning each task to multiple crowd workers and then aggregating the votes (answers) to obtain the correct label. This process is called truth inference and relies on an aggregation strategy to combine labels. Several label fusion strategies has been investigated in the literature (Aydin et al. 2014; Demartini et al. 2012; Callison-Burch 2009; Fan et al. 2015; Li et al. 2014; Liu et al. 2012; Ma et al. 2015).

Because of its simplicity and effectiveness, the most popular label fusion method is *Majority Voting* (MV) (Tu et al. 2019), which selects the label voted by the majority of the workers as the correct answer (Franklin et al. 2011; Parameswaran et al. 2012; Marcus et al. 2011). We thus use majority voting as the label fusion strategy in our experimental evaluation. A known limitation of majority voting is the fact that it assumes that all workers provide the same quality of answers. To measure whether the results depend on this simplifying assumption, we also ran an additional experimental investigation using a more refined label fusion method (Sect. 3.7).

3.4.4 Metrics

We use F1, F3, Accuracy, and Loss metrics to evaluate the performance of the machine classifier (*ML*) and of the hybrid crowd-machine classifier (*ML+C*). We define the Loss of classification as the following (Nguyen et al. 2015; Krivosheev et al. 2018b):

$$Loss = \frac{1}{|I|} \cdot (k * FNN + FPN), \tag{1}$$

where k denotes how much the cost of a false negative outweighs the one of a false positive, FNN is the number of false negatives, FPN is the number of false positives, and $|I|$ is the number of items in I. The Loss summarizes the subjective perspective of the risks for False Positive/Negative errors. This is especially common in real-world applications, such as potential credit card fraud, identifying tweets linked to criminal activities, or literature reviews where screening out a relevant paper is considered to be a serious error affecting the quality of the review, while a falsely included paper just requires some extra work by the authors.

Table 1 Properties of the datasets

	Datasets							
	RTE	Emotion	Amazon-1	Amazon-2	Crisis-1	Crisis-2	Crisis-3	Exergame
Tasks	800	100	1000	998	1948	1948	949	93
Workers	164	38	263	263	79	79	93	38
Total votes	8000	1000	4908	4873	6000	6000	4003	286
Min. vote	10	10	2	2	3	3	3	2
Data (+/-)	400/400	4/96	612/388	99/899	347/1601	883/1065	516/433	53/40
Batch count	40	5	50	50	90	90	50	5
Batch size	20	20	20	20	20	20	20	20
Exp. count	20	20	20	20	20	20	20	20

3.5 Datasets

Crowdsourced datasets can either include the answer (label) of each crowd worker to each item or just provide the aggregated labels (a single discrete label for each item, determined by combining votes from multiple crowd workers). The latter type of datasets is less informative and doesn't allow to test strategies that involve queries to individual workers. However, most of the available crowdsourced datasets are of this type. We thus created an open repository[2] of the available crowdsourced datasets with individual crowd votes; we also added the datasets we collected. We provide a standard format for accessing the datasets so that they can be used in the experiments without any workload in preprocessing. Researchers can benefit from this repository for hybrid human–machine classification and ranking tasks, truth discovery based on crowdsourced data, estimation of the crowd bias, and active learning.

Table 1 shows the properties of each dataset we used in our experiments; the number of tasks, number of workers, total number of votes, minimum vote count per item, data proportion in terms of the number of positives and negatives, batch count (a batch is a predefined number of instances, where the batch count is the number of batches that defines the total number of iterations), size of a batch, and the number of experiments (repetitions) for each dataset. We show the distribution of labels for each dataset in Fig. 2.

The task in the *Recognizing Textual Entailment* (RTE) dataset (Snow et al. 2008) is to identify whether a given hypothesis sentence is implied by the information in the given text[3].

The *Emotion*[3] dataset (Snow et al. 2008) is about rating the emotion ("anger") of a given text. Each rating is a value between 0 and 100, and we converted them to binary form (0 if *rating* ≤ 49, else 1).

The *Amazon Sentiment-1*[4] dataset (Krivosheev et al. 2018a) includes annotations about deciding whether the given product review belongs to a book or not. Similarly, the *Amazon Sentiment-2*[4] dataset (Krivosheev et al. 2018a) includes annotations about whether the given product review has a negative or positive sentiment.

[3] https://sites.google.com/site/nlpannotations/.

[4] https://tinyurl.com/AmazonSentiment.

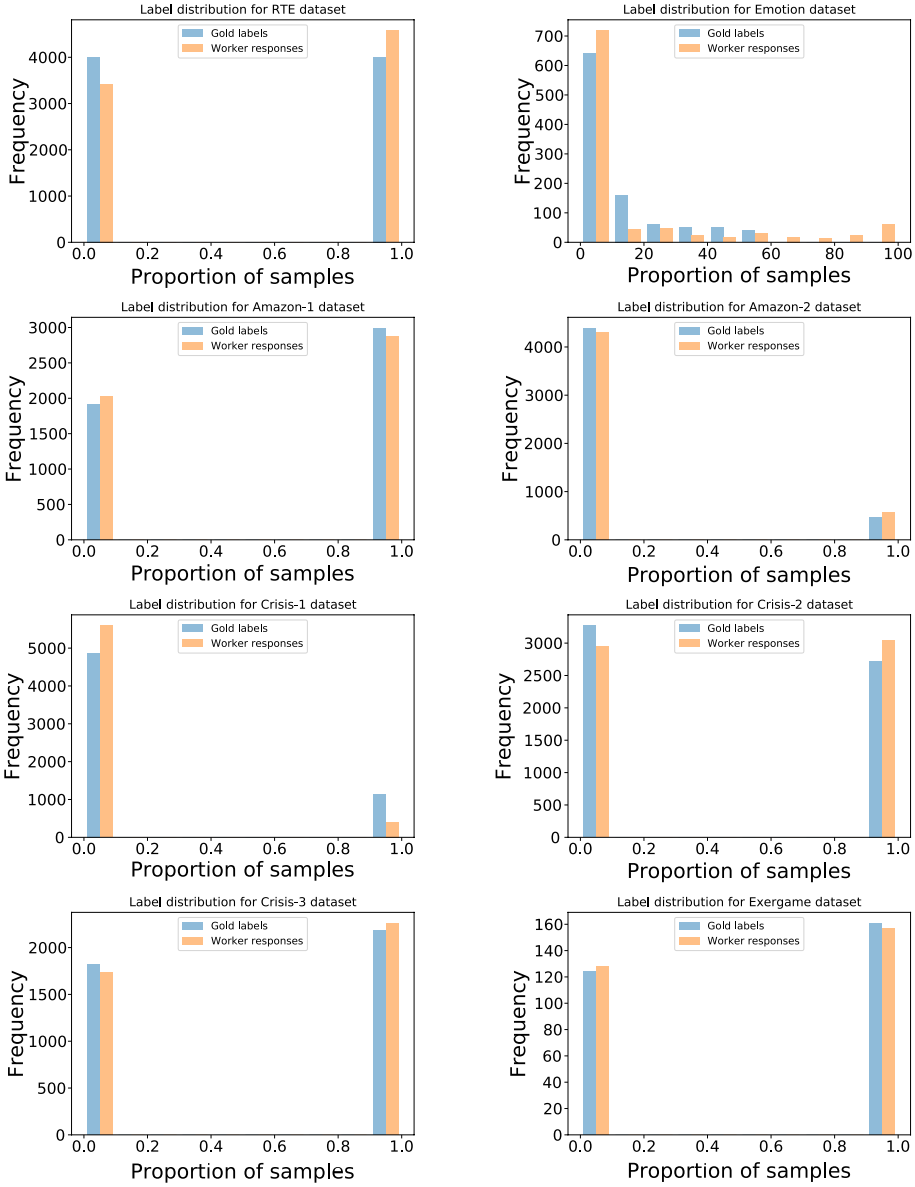

Fig. 2 Label distributions of datasets

The *Crisis-1*[5] dataset (Imran et al. 2013) consists of human-labeled tweets collected during the 2012 Hurricane Sandy and the 2011 Joplin tornado. The task is to decide whether the author of the tweet seems to be an eyewitness of the event. Similarly, *Crisis-2*[5] dataset

[5] https://crisisnlp.qcri.org/.

Table 2 F1 Scores in percentage (Scenario 1: Unlimited votes)

Dataset	Evaluation	AL approach							
		R	UC	C	BC (k=1)	Q	LAL-R	LAL-I	MinExp
RTE	ML	66.3	**69.7**	55.4	60	67.2	64	62.9	64.8
	ML+C	68.4	**72.1**	56.7	61.1	67.4	66.7	65.5	66.8
Emotion	ML	32.8	**53**	44.8	51.7	32.7	42.4	44	33.8
	ML+C	42.2	68.4	60.2	**70**	48.5	60.2	56.6	42.7
Amazon-1	ML	93.9	**96.2**	58.6	74.3	70	92.5	92.7	92.4
	ML+C	93.9	**96.3**	59.1	74.5	70.4	92.6	92.8	92.5
Amazon-2	ML	55	63.5	35	36.4	26	**67.6**	65.7	32.1
	ML+C	60.7	70.1	40.2	42.5	31	**72.5**	71.4	34.2
Crisis-1	ML	**31.4**	7.5	14.9	5.9	7.7	27	28.5	11.8
	ML+C	**38.5**	10.3	17.9	9.4	8.2	30.6	32.4	14.2
Crisis-2	ML	80.8	**85.5**	61.8	66.9	15.7	66.1	61.5	79.7
	ML+C	81	**86**	62	67	16.1	66.3	61.6	79.9
Crisis-3	ML	85.2	**89.5**	69.8	68.7	47.6	69.4	73.1	84.1
	ML+C	85.7	**90.2**	69.9	68.7	47.8	69.6	73.3	84.3
Exergame	ML	79.4	**81.5**	74.9	73.4	79.4	77.8	77.4	78.5
	ML+C	80.3	**81.7**	75	73.8	79.4	78.1	77.8	78.7

(Imran et al. 2013) contains annotations about deciding the type of the message (tweet). The task in *Crisis-3*[5] dataset (Imran et al. 2013) is to analyze hurricane-related tweets and decide whether the tweet is informative or not.

Finally, the *Exergame* dataset includes annotations about whether the given paper describes a study that uses an exergame. An exergame is a form of interactive gaming where people do physical activities while playing a video game, that is, physical exercises by way of video games.

3.6 Results

The elaborated experiment results, all datasets, and the source code for reproducing the experiments are available online[1]. In addition, we present the visual experiment results in a notebook[6], while summarizing the important outcomes below.

Tables 2 and 3 show F1 scores of each AL approach in Scenario 1 and Scenario 2, respectively (bold cells show the best performing AL strategies). We observed that *ML+C* prediction outperforms the *ML* prediction on each dataset, regardless of the AL approach. When we compare F1 and F3 scores, the best AL approach remains the same. That is why we present F1 scores here, while F3 scores can be seen in the results sheet.[7]

Comparing the Loss with different k values, we noticed that (i) when the harm of FP and FN is the same ($k = 1$) uncertainty sampling performs 35% better in average than other approaches, (ii) when the problem is characterized by high k value ($k \geq 10$)

[6] https://tinyurl.com/ALExperimentResults.
[7] https://tinyurl.com/ALResultSheet.

Table 3 F1 Scores in percentage (Scenario 2: Limited votes)

Dataset	Evaluation	AL approach							
		R	UC	C	BC (k=1)	Q	LAL-R	LAL-I	MinExp
RTE	ML	65.4	**70.2**	61.5	64.9	57.6	63.9	63.7	65.6
	ML+C	67.9	**72.5**	63.5	66.7	58.7	66.7	66.4	67.7
Emotion	ML	35.4	44.2	35.8	**44.9**	38.8	41.7	41.1	35.7
	ML+C	44.5	54.6	49.1	**59.8**	51.4	56.9	53.4	47
Amazon-1	ML	93.7	**96.2**	65.8	84.6	80.2	94.3	92.9	94
	ML+C	93.8	**96.3**	66.2	84.8	80.7	94.5	92.9	94.1
Amazon-2	ML	56.6	**70.3**	43.6	63.9	42.8	67.1	66.6	52.9
	ML+C	62.6	**76.7**	50.3	70.7	49.8	72.8	72	59
Crisis-1	ML	**32.2**	22.6	31	21.6	18.8	30.3	28.5	20.9
	ML+C	**39.6**	28.6	36.7	27.1	24.5	35.5	34.5	25.2
Crisis-2	ML	81	**86**	71.1	69.2	73.1	73.7	73.2	80.7
	ML+C	81.1	**86.3**	71.2	69.4	73.3	73.9	73.4	80.8
Crisis-3	ML	85.1	**89.5**	74.5	74.2	68	77.5	77.4	85.2
	ML+C	85.3	**90.1**	74.7	74.6	68.3	77.8	77.5	85.5
Exergame	ML	79.8	**84**	75.3	76.1	80	78.7	78.4	78
	ML+C	79.9	**84.5**	75.4	76.1	80.3	79.5	79.1	78.3

Table 4 Loss ($k = 100$) (Scenario 2: Limited votes)

Dataset	AL approach							
	R	UC	C	BC (k=1)	Q	LAL-R	LAL-I	MinExp
RTE	628.08	427	747.46	**158**	884.11	702.72	712.82	621.86
Emotion	6.24	5.2	5.55	**4.74**	5.3	5.39	5.39	6.14
Amazon-1	274.29	**169.1**	1465.4	1522.11	859.26	260.98	292.06	265.15
Amazon-2	256.16	**176.7**	335.31	347.77	342.4	188.39	189.2	291.47
Crisis-1	**1400.06**	1513	1406.86	1529.07	1553.88	1423.26	1446.3	3062.11
Crisis-2	756.29	**460.36**	1278.99	1454.45	1046.7	1287.1	1302.9	1543.61
Crisis-3	391.57	**240.9**	735.25	597.34	1113.8	685.46	667.1	393.94
Exergame	59.2	45.9	75.97	**42.32**	58.2	66.16	63.23	62.28

then block certainty sampling provides 25% better performance in average on five datasets, while uncertainty sampling provides 36% better performance in average across all datasets, and (iii) block certainty sampling approach outperforms others on five datasets with small k values ($k \leq 0.1$) with an improvement of 34.5% with respect to the average performance. We only present the results for $k = 100$ here (see Table 4), while keeping others in the notebook (see footnote 6).

When we analyze the performance of the approaches individually, we draw the following conclusions:

Table 5 Size of training set (Scenario 2: Limited votes)

AL approach	Dataset							
	RTE	Emotion	Amazon 1	Amazon 2	Crisis 1	Crisis 2	Crisis 3	Exergame
R	523.6	**70.5**	649.55	649	**1199.9**	1197.9	638.6	72.3
U C	**645.2**	58.45	**806.85**	**742.95**	1129.3	**1478.8**	**768.25**	**84.9**
C	351.65	51.05	404.9	491.6	837.45	772.3	421.65	57.95
BC (k=1)	356.95	50.45	416.75	582.25	976	779.4	423.75	58.7
Q	283.7	52.05	357.85	358.8	622.65	622.65	354.55	61.75
LAL-R	495.65	55.5	629.2	714.25	1191.3	786.05	441	63.8
LAL-I	496.45	59.35	581.5	710.25	1183.6	794.15	434.85	65.15
MinExp	510.2	59	640	474.9	747.4	1181.7	630.15	73.8

1. Block certainty has an outstanding performance with very big and very small k values ($k \geq 100$ and $k \leq 0.01$) on RTE, Emotion, and Exergame datasets; so it can be used in domains where the harm of a false negative and false positive is very different, such as in literature reviews, or medicine.
2. Certainty and QUIRE (Huang et al. 2010) approaches did not show a promising performance over crowdsourced data in terms of accuracy, F1, and F3 scores.
3. Random sampling performed comparable or sometimes better (i.e. in the Crisis-1 dataset) over imbalanced data with more negatives.
4. Although it is claimed that LAL (Konyushkova et al. 2017) approach outperforms uncertainty sampling (Lewis and Gale 1994), results show that uncertainty sampling outperforms others in most cases for the finite-pool hybrid classification over crowdsourced data.
5. Table 5 shows that uncertainty sampling and random sampling created the biggest number of training sets (note that we may relabel the training data and we may end up with a different number of votes per item; this affects the size of training data after the learning process ends).
6. Limited votes scenario increased the size of training sets for all cases. This shows that sampling strategies may be stuck at some items if we do not limit the maximum number of votes per item.

3.7 Further analysis

The previous experimental analysis was run using random forests as the underlying classifier and majority voting as the label fusion strategy. In this section, we investigate whether changing the classifier or fusion strategy affects the overall picture. Since the LAL (Konyushkova et al. 2017) approach is specifically designed to be used with random forests, we omit it from the following analysis. We also omit minExpError (Mozafari et al. 2014) as its complexity is very high and it did not show to be competitive with less expensive approaches. Hence, we focused on the following AL approaches: (i) *random* (R), (ii) *uncertainty* (UC) Lewis and Gale (1994), (iii) *certainty* (C), (iv) *block certainty* (BC), and (v) *QUIRE* (Q) (Huang et al. 2010).

Table 6 Best performance results in ML + C prediction

Dataset	F	Metric		
		Accuracy	F1	Loss (K=100)
RTE	MV	*RF,UC* **0.69**	*RF,UC* **0.725**	*SVM,R* **134**
	DS	*RF,UC* **0.701**	*RF,UC* **0.701**	*SVM,Q* **184**
Emotion	MV	*SVM,Q* **0.945**	*SVM,UC* **0.599**	*RF,BC* **4.7**
	DS	*RF,BC* **0.96**	*RF,BC* **0.96**	*RF,BC* **4.8**
Amazon-1	MV	*SVM,UC* **0.961**	*SVM,UC* **0.968**	*SVM,UC* **152**
	DS	*RF,UC* **0.954**	*SVM,UC* **0.959**	*RF,UC* **185**
Amazon-2	MV	*SVM,UC* **0.955**	*RF,UC* **0.767**	*RF,UC* **177**
	DS	*RF,UC* **0.956**	*RF,UC* **0.956**	*SVM,UC* **145**
Crisis-1	MV	*RF,R* **0.866**	*RF,R* **0.396**	*RF,R* **1731**
	DS	*SVM,R* **0.859**	*RF,R* **0.858**	*RF,C* **1441**
Crisis-2	MV	*RF,UC* **0.87**	*SVM,UC* **0.865**	*SVM,C* **172.3**
	DS	*SVM,UC* **0.88**	*RF,UC* **0.872**	*SVM,UC* **492**
Crisis-3	MV	*SVM,UC* **0.901**	*SVM,UC* **0.911**	*SVM,UC* **241**
	DS	*SVM,UC* **0.904**	*SVM,UC* **0.914**	*RF,BC* **173**
Exergame	MV	*RF,UC* **0.825**	*RF,UC* **0.845**	*SVM,BC* **33.2**
	DS	*RF,UC* **0.796**	*SVM,UC* **0.812**	*SVM,BC* **35.2**

We repeated all experiments using an SVM classifier, utilizing an implementation provided by the scikit-learn library (Pedregosa et al. 2011) with the following parameters: class_weight ='balanced' and C=0.1. As an alternative to majority voting, we used the *Dawid-Skene* (DS) (Dawid and Skene 1979) strategy, a popular label fusion method that models each worker as a confusion matrix and uses an expectation-maximization approach to decide the correct label.

Results[8] confirmed that the combination of humans and the machine (*ML + C*) outperforms the machine-only case (*ML*). For this reason, in the following, we discuss the results of the *ML + C* case only.

Table 6 shows which *(classifier, AL approach)* pair performs best on each dataset in terms of *Accuracy*, *F1*, and *Loss (K=100)* metrics. We evaluate results in two

[8] https://tinyurl.com/ComparisonOfAggTech.

different conditions; when we aggregate votes using (i) majority voting (MV), and (ii) Dawid&Skene (DS). Results show that this pair may change even on the same dataset with respect to the metric used. For example, when we look at the Crisis-1 dataset random sampling with SVM classifier and DS label fusion method performs best in terms of accuracy while uncertainty sampling with RF classifier is the best in terms of F1 score.

Summing up, these results suggest that the performance that AL strategies exhibit in standard settings cannot be directly transferred to hybrid classification problems. Many factors that may affect the behavior of an AL approach, such as the machine classifier being used, the amount of labeling noise in the data, the characteristics of the problem, and the label fusion method. For example, when we look at the F1 scores in Tables 2 and 3, we see that uncertainty sampling is not the best method for Emotion, Amazon-2 (in Scenario 1), and Crisis-1 datasets. These three datasets are the most unbalanced datasets we used. In addition, the noise level in Emotion and Crisis-1 datasets are very high (please see Fig. 2). These observations show that uncertainty sampling performs best when the dataset is balanced and the noise level of the data is low.

4 Conclusions and open issues

In this paper, we first reviewed the existing AL approaches under three categories: (i) fixed-strategy approaches, (ii) dynamic-strategy approaches, and (iii) strategy-free approaches. We then investigated the performance of a set of representative approaches for different strategies in the hybrid human–machine classification setting. The aim was to discover if and how the performance of the existing approaches can be transferred into the hybrid classification context.

Experimental results showed that as expected, hybrid human–machine classification always improves over purely machine-based classification. When comparing different AL approaches, however, no clear winner emerges. Even a state-of-the-art meta-learning approach like *LAL* fails to show consistent improvements over the alternatives when evaluated across different datasets. We observed that if we have a finite pool classification problem with noisy crowd labels (i.e. RTE, Emotion, Crisis-1, and Crisis-2 datasets), then we can simply start by picking the RF classifier, UC approach, and DS label fusion method to achieve an acceptable performance (see Table 6). In addition, if the cost of FN and FP errors are very different for the problem at hand, then we can consider using a BC approach instead of UC (see Table 4, where BC improved the performance on RTE, Emotion, and Exergame datasets with a big k value).

Hybrid crowd-machine classification is promising but needs more investigation. Most of the existing AL approaches assume that enough high-quality labeled data exist. However, gathering high-quality labeled data is challenging and labels, especially if crowdsourced, can be noisy. For this reason, we analyzed what happens if we use noisy labels instead of gold data in finite-pool hybrid classification problems. We have shown that in the presence of noisy data the conclusions from the state of the art need to be revisited and cannot be taken at face value when data is crowdsourced. This points to the need for further work in the community on what is the role of noise in the success of specific AL algorithms and how much noise they can tolerate before the assumptions and intuitions on which they are based do not hold any longer. The community also needs to analyze how different types of noise (i.e. random noise or biased crowdsourced answers) in data affect the performance of AL, and how the effect of such noise can be smoothed. Indeed, label smoothing can be a

promising direction to pursue as it helps to improve the calibration of ML models, which is central for many AL algorithms.

In addition, our analysis highlights several open issues and challenges that we believe can shape future research on this topic, and specifically: (i) what are the trade-offs between having a smaller but accurately labeled dataset vs a larger but more noisy one? (ii) Would a dynamic assessment of crowd accuracy help strategy-free approaches? and, (iii) How do we operate at the start of an AL process when we have no idea of the crowd accuracy?

Another direction is that of balancing the AL and human vs machine contribution in hybrid crowd-ML classification problems where the pool of items to classify is finite: here the challenge is deciding when to stop spending our budget for collecting samples to optimize our AL process and obtain a stronger ML model, and spend the budget for classifying the remaining items in the pool, by applying the current ML model and resorting to humans for items on which the ML model is unsure about.

In summary, this paper has pointed to several interesting research paths that we need to undertake if we want to address problems that companies face when developing their ML solutions and that are needed to make AL a mainstream part of ML pipelines.

Funding Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement.

References

Aggarwal CC, Kong X, Gu Q, Han J, Yu PS (2014) Chapter 22 active learning: a survey

Audibert JY, Bubeck S (2009) Minimax policies for adversarial and stochastic bandits. In: Proceedings of the 22nd annual conference on learning theory (COLT), pp 217–226

Auer P, Cesa-Bianchi N, Freund Y, Schapire RE (1995) Gambling in a rigged casino: the adversarial multi-armed bandit problem. In: Proceedings of IEEE 36th annual foundations of computer science, pp 322–331

Auer P, Cesa-Bianchi N, Freund Y, Schapire RE (2003) The nonstochastic multiarmed bandit problem. SIAM J Comput 32(1):48–77

Aydin BI, Yilmaz YS, Li Y, Li Q, Gao J, Demirbas M (2014) Crowdsourcing for multiple-choice question answering. In: Proceedings of the twenty-eighth AAAI conference on artificial intelligence, pp 2946–2953

Bachman P, Sordoni A, Trischler A (2017) Learning algorithms for active learning. In: Proceedings of the 34th international conference on machine learning, vol 70, pp 301–310

Baram Y, El-Yaniv R, Luz K (2004) Online choice of active learning algorithms. J Mach Learn Res 5:255–291

Beygelzimer A, Dasgupta S, Langford J (2009) Importance weighted active learning. In: Proceedings of the 26th annual international conference on machine learning, pp 49–56

Beygelzimer A, Hsu D, Langford J, Zhang T (2010a) Agnostic active learning without constraints. In: Proceedings of the 23rd international conference on neural information processing systems, vol 1, pp 199–207

Beygelzimer A, Langford J, Li L, Reyzin L, Schapire R (2010b) An optimal high probability algorithm for the contextual bandit problem. CoRR arXiv:1002.4058
Bouguelia MR, Belaïd Y, Belaïd A (2016) Identifying and mitigating labelling errors in active learning. In: Pattern recognition: applications and methods, vol Lecture Notes in Computer Science. Springer, p 17
Bouguelia MR, Nowaczyk S, Santosh KC, Verikas A (2018) Agreeing to disagree: active learning with noisy labels without crowdsourcing. Int J Mach Learn Cybern 9:1307–1319
Brew A, Greene D, Cunningham P (2010) Using crowdsourcing and active learning to track sentiment in online media. In: Proceedings of the 19th European conference on artificial intelligence, pp 145–150
Budd S, Robinson EC, Kainz B (2019) A survey on active learning and human-in-the-loop deep learning for medical image analysis. ArXiv arXiv:1910.02923
Callaghan W, Goh J, Mohareb M, Lim A, Law E (2018) Mechanicalheart: a human–machine framework for the classification of phonocardiograms. In: Proceedings of ACM Human–Computer Interaction 2(CSCW)
Callison-Burch C (2009) Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 1, pp 286–295
Chapelle O, Schlkopf B, Zien A (2010) Semi-supervised learning, 1st edn. The MIT Press, Cambridge
Chu HM, Lin HT (2016) Can active learning experience be transferred? In: 2016 IEEE 16th international conference on data mining (ICDM), pp 841–846
Cohn D, Atlas L, Ladner R (1994) Improving generalization with active learning. Mach Learn 15:201–221
Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. J Artif Int Res 4(1):129–145
Contardo G, Denoyer L, Artières T (2017) A meta-learning approach to one-step active-learning. In: International workshop on automatic selection, configuration and composition of machine learning algorithms, vol 1998, pp 28–40
Dawid AP, Skene AM (1979) Maximum likelihood estimation of observer error-rates using the EM algorithm. J R Stat Soc Ser C Appl Stat 28(1):20–28
Demartini G, Difallah DE, Cudré-Mauroux P (2012) Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st international conference on world wide web, pp 469–478
Deroski S, Panov P, Kocev D, Todorovski L (2014) Probabilistic active learning: towards combining versatility, optimality and efficiency. In: Proceedings of the 17th international conference on discovery science (DS)
Desreumaux L, Lemaire V (2020) Learning active learning at the crossroads? Evaluation and discussion. arXiv:2012.09631
Efron B, Tibshirani RJ (1993) An introduction to the bootstrap. No. 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton
Fan J, Li G, Ooi BC, Tan Kl, Feng J (2015) Icrowd: an adaptive crowdsourcing framework. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, pp 1015–1030
Fang M, Zhu X, Li B, Ding W, Wu X (2012) Self-taught active learning from crowds. In: 2012 IEEE 12th international conference on data mining, pp 858–863
Fang M, Li Y, Cohn T (2017) Learning how to active learn: a deep reinforcement learning approach. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 595–605
Franklin MJ, Kossmann D, Kraska T, Ramesh S, Xin R (2011) Crowddb: answering queries with crowdsourcing. In: Proceedings of the 2011 ACM SIGMOD international conference on management of data, pp 61–72
Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. Mach Learn 28:133–168
Guo Y, Greiner R (2007) Optimistic active learning using mutual information. In: Proceedings of the 20th international joint conference on artifical intelligence, pp 823–829
Haussler D, Kearns M, Schapire R (1991) Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. In: Proceedings of the fourth annual workshop on computational learning theory, COLT '91. Morgan Kaufmann Publishers Inc., San Francisco, pp 61–74
Hoi SCH, Jin R, Lyu MR (2006) Large-scale text categorization by batch mode active learning. In: Proceedings of the 15th international conference on world wide web, pp 633–642
Hsu WN, Lin HT (2015) Active learning by learning. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 2659–2665
Huang SJ, Jin R, Zhou ZH (2010) Active learning by querying informative and representative examples. In: Proceedings of the 23rd international conference on neural information processing systems, vol 1, pp 892–900

Imran M, Elbassuoni S, Castillo C, Diaz F, Meier P (2013) Practical extraction of disaster–relevant information from social media. In: Proceedings of the 22nd international conference on world wide web, pp 1021–1024

Johnson M, Anderson P, Dras M, Steedman M (2018) Predicting accuracy on large datasets from smaller pilot data. In: ACL, pp 450–455

Konyushkova K, Sznitman R, Fua P (2017) Learning active learning from data. In: Advances in neural information processing systems, vol 30, pp 4225–4235

Konyushkova K, Sznitman R, Fua P (2018) Discovering general-purpose active learning strategies. CoRR arXiv:1810.04114

Krivosheev E, Casati F, Baez M, Benatallah B (2018a) Combining crowd and machines for multi-predicate item screening. In: Proceedings of ACM Human–Computer Interaction 2

Krivosheev E, Casati F, Benatallah B (2018b) Crowd-based multi-predicate screening of papers in literature reviews. In: Proceedings of the 2018 world wide web conference, pp 55–64

Krivosheev E, Casati F, Bozzon A (2021) Active hybrid classification. Computing Research Repository arXiv:2101.08854

Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval, pp 3–12

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on world wide web, WWW '10. Association for Computing Machinery, New York, pp 661–670. https://doi.org/10.1145/1772690.1772758

Li Q, Li Y, Gao J, Su L, Zhao B, Demirbas M, Fan W, Han J (2014) A confidence-aware approach for truth discovery on long-tail data. Proc VLDB Endow 8(4):425–436

Liu M, Buntine W, Haffari G (2018) Learning how to actively learn: a deep imitation learning approach. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1874–1883

Liu Q, Peng J, Ihler A (2012) Variational inference for crowdsourcing. In: Proceedings of the 25th international conference on neural information processing systems, vol 1, pp 692–700

Ma F, Li Y, Li Q, Qiu M, Gao J, Zhi S, Su L, Zhao B, Ji H, Han J (2015) Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 745–754

Marcus A, Wu E, Madden S, Miller R (2011) Crowdsourced databases: query processing with people. In: CIDR, pp 211–214

McCallum A, Nigam K (1998) Employing EM and pool-based active learning for text classification. In: Proceedings of the fifteenth international conference on machine learning, pp 350–358

Mozafari B, Sarkar P, Franklin MJ, Jordan MI, Madden S (2014) Scaling up crowd-sourcing to very large datasets: a case for active learning. Proc VLDB Endow 8:125–136

Nguyen AT, Wallace BC, Lease M (2015) Combining crowd and expert labels using decision theoretic active learning. In: Proceedings of the third AAAI conference on human computation and crowdsourcing (HCOMP)

Pang K, Dong M, Wu Y, Hospedales T (2018a) Dynamic ensemble active learning: a non-stationary bandit with expert advice. In: ICPR, pp 2269–2276

Pang K, Dong M, Wu Y, Hospedales TM (2018b) Meta-learning transferable active learning policies by deep reinforcement learning. CoRR arXiv:1806.04798

Parameswaran A, Park H, Garcia-Molina H, Polyzotis N, Widom J (2012) Deco: Declarative crowdsourcing. In: Proceedings of the 21st ACM international conference on information and knowledge management, pp 1203–1212

Parker C (2011) An analysis of performance measures for binary classifiers. In: 2011 IEEE 11th international conference on data mining, pp 517–526

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Research 12:2825–2830

Ravi S, Larochelle H (2018) Meta-learning for batch mode active learning. In: 6th international conference on learning representations, ICLR 2018, workshop track proceedings

Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: ICML, pp 894–905

Rudovic O, Zhang M, Schuller BW, Picard RW (2019) Multi-modal active learning from human data: A deep reinforcement learning approach. CoRR arXiv:1906.03098

Saar-Tsechansky M, Provost F (2004) Active sampling for class probability estimation and ranking. Mach Learn 54:153–178

Schein AI, Ungar LH (2007) Active learning for logistic regression: an evaluation. Mach Learn 68:235–265

Settles B (2010) Active learning literature survey, vol 52. University of Wisconsin, Madison

Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing, pp 1070–1079

Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the fifth annual workshop on computational learning theory, pp 287–294

Snow R, O'Connor B, Jurafsky D, Ng A (2008) Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In: Proceedings of the 2008 conference on empirical methods in natural language processing, pp 254–263

Sun-Hosoya L, Guyon I, Sebag M (2018) Activmetal: algorithm recommendation with active meta learning. In: IAL 2018 workshop, ECML PKDD, poster

Tsai M, Ho C, Lin C (2010) Active learning strategies using SVMs. Wiley Int Rev Data Min and Knowl Disc 313–326

Tu J, Yu G, Domeniconi C, Wang J, Xiao G, Guo M (2019) Multi-label crowd consensus via joint matrix factorization. Knowl Inf Syst 62:1341–1369

Vu TT, Liu M, Phung D, Haffari G (2019) Learning how to active learn by dreaming. In: Proceedings of the 57th annual meeting of the Association for Computational Linguistics, pp 4091–4101

Wang L (2011) Smoothness, disagreement coefficient, and the label complexity of agnostic active learning. J Mach Learn Res 12:2269–2292

Woodward M, Finn C (2017) Active one-shot learning. In: NIPS 2016, deep reinforcement learning workshop

Yan S, Chaudhuri K, Javidi T (2016) Active learning from imperfect labelers. In: Proceedings of the 30th international conference on neural information processing systems, pp 2136–2144

Yan S, Chaudhuri K, Javidi T (2019) The label complexity of active learning from observational data. In: 33rd conference on neural information processing systems (NeurIPS 2019)

Yan Y, Rosales R, Fung G, Schmidt M, Hermosillo G, Bogoni L, Moy L, Dy J (2010) Modeling annotator expertise: Learning when everybody knows a bit of something. In: Proceedings of the 13th international conference on artificial intelligence and statistics (AISTATS), vol 9, pp 932–939

Yang J, Drake T, Damianou A, Maarek Y (2018) Leveraging crowdsourcing data for deep active learning an application: learning intents in Alexa. In: Proceedings of the 2018 World Wide Web conference, pp 23–32

Zhao L, Sukthankar GR, Sukthankar R (2011) Incremental relabeling for active learning with noisy crowdsourced annotations. In: 2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing, pp 728–733

Zheng Y, Li G, Li Y, Shan C, Cheng R (2017) Truth inference in crowdsourcing: Is the problem solved? Proc VLDB Endow 10(5):541–552

Zhong J, Tang K, Zhou ZH (2015) Active learning from crowds with unsure option. In: Proceedings of the 24th international conference on artificial intelligence, pp 1061–1067

Zhu X, Lafferty J, Ghahramani Z (2003) Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, pp 58–65