A Meta-Active Learning approach exploiting Instance Importance based on Learning Gradient Variation

(Discussion Paper)

Sergio Flesca[1], Domenico Mandaglio[1], Francesco Scala[1] and Andrea Tagarelli[1]

[1] *DIMES Dept., University of Calabria, 87036 Rende (CS), Italy*

Abstract

A major challenge in active learning is to select the most informative instances to be labeled by an annotation oracle at each step. In this respect, one effective paradigm is to learn the active learning strategy that best suits the performance of a meta-learning model. This strategy first measures the quality of the instances selected in the previous steps and then trains a machine learning model that is used to predict the quality of instances to be labeled in the current step. In this paper, we discuss a new approach of learning-to-active-learn that selects the instances to be labeled as the ones producing the maximum change to the current classifier. The key idea is to select such instances according to their importance reflecting variations in the learning gradient of the classification model. Our approach can be instantiated with any classifier trainable via gradient descent optimization, and here we provide a formulation based on a deep neural network model, which has not deeply been investigated in existing learning-to-active-learn approaches. The experimental validation of our approach has shown promising results in scenarios characterized by relatively few initially labeled instances.

Keywords

meta-learning models, model-change framework, learning to active learn, active learning,

1. Introduction

Supervised machine learning methods typically require a large number of training data instances. However, manually labeling training instances is a costly and time consuming process, especially for specialized domains, where a deep expertise is required for correctly labeling data instances. *Active Learning* aims at selecting the data instances to be labeled by an expert, or annotation oracle, in order to train a machine learning model as quickly and effectively as possible. Several strategies have been proposed in the literature [1], which select the instances to be provided to the oracle for annotation using different heuristics; however, none of such heuristics has shown to outperform the others in every scenario of interest. To overcome major limitations, *meta-active learning* approaches have been proposed to automatically detect the best strategy of selection of the instances to be annotated [2, 3, 4].

In this paper, we discuss the main contributions from our earlier study [5], where we introduced a new meta-active learning method whose instance selection step, modeled as a regression problem, exploits the training gradient of a deep neural network model, and in general of any

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy

flesca@dimes.unical.it (S. Flesca); d.mandaglio@dimes.unical.it (D. Mandaglio); francesco.scala@dimes.unical.it (F. Scala); andrea.tagarelli@unical.it (A. Tagarelli)

machine learning model whose training is based on a gradient descent method. Experiments conducted on CIFAR-10 image data, and including a comparison with some baselines, have shown promising results by the proposed approach in terms of percentage increase in accuracy.

2. Related Work

Active learning methods typically fall into one of the following categories: Uncertainty Sampling, Query-By-Committee, Expected Model Change, Expected Error Reduction, Variance Reduction, and Density-Weighted [1].

Uncertainty sampling aims to improve the quality of the labeled dataset by selecting as instances to be labeled those such that the trained classifier is most uncertain in assigning a class label [6, 7]. Among this class of methods, the most popular one is probably least confidence sampling (LCS) [8] which uses as uncertainty measure for an instance the difference between 100% confidence and the most confidently predicted label for the instance. Other approaches use different multi-class uncertainty sampling variants, such as margin sampling [9] or entropy [8].

The query-by-committee approach [10] maintains a set of prediction models, or committee, that are used to predict the label of an instance. The instance over which there is the maximum disagreement on the labels predicted by the models in the committee is regarded as the most informative and hence selected for labeling. Several specializations of the approach have been proposed using different models for the committee members [11, 12, 13, 14].

The expected model change framework [15] aims to define a strategy for selecting the instance that would yield the greatest change to the current model if we knew its label. The strategy computes the expected gradient length and uses it as a measure of the expected change to the model that is associated to the labeling of an instance. The key idea is to prefer instances that are likely to have the greatest influence in changing the model. Theoretical aspects of this framework have been well studied for support vector machines and linear regression [16], although it can be computationally expensive for large feature space and set of labelings.

Expected error reduction aims to select the instance x that yields the maximum reduction of the model generalization error once it is trained using the label of x too. However, since the labels of some instances are not known, the model is usually approximated using the expectation over all possible labels under the current model. This framework has been successfully used with a variety of models such as Naıve Bayes [17], logistic regression [18], and SVM [19].

Variance reduction methods reduce the generalization error indirectly by minimizing output variance. The early method in [20] was proposed for active learning based on the reduction of the estimated distribution of the model's output for regression. Applications of variance reduction include multi-class image classification [21].

The key idea of density-weighted methods is that informative instances should not only be the uncertain ones, but also those representative of the underlying distribution [11, 22, 23, 8, 24]. Hence, the instances are selected according to both a base selection measure (e.g., LCS) and a density based measure (e.g., the average similarity of an instance w.r.t. the other instances).

Meta-learning algorithms have recently been proposed for the active learning tasks. In [2, 3], several active learning heuristics are combined using a bandit algorithm exploiting a maximum entropy criterion that estimates classification performance without knowing the actual labels.

Rather than combining existing heuristics, the meta-learning approach to active learning in [25] models the active learning task as a regression problem: given a trained classifier and its output for a specific unlabeled instance, it predicts the reduction in generalization error that can be expected by providing the actual label of the instance. Note that the regressor in [25] is required to be trained on a specific set of instance-driven features, such as the variance of the classifier output for the instance or the predicted probability distribution over possible labels for the instance. Our approach does not have the same constraint, since we utilize the raw features of the instances, yet we can in principle exploit instance-driven features. More importantly, for each active learning epoch, [25] requires to perform several training steps of the classifier while we perform just a single training step.

3. Proposed Approach

A classification problem consists in associating every instance taken from a predefined domain $\mathcal{D}$ with a label selected from a fixed domain of labels $\mathcal{L}$. We assume the presence of a set of instance-label pairs $LI \subseteq \mathcal{D} \times \mathcal{L}$ and a set of unlabeled instances $UI \subseteq \mathcal{D}$, where for each pair $\langle x, y \rangle \in LI$, x is an instance in $\mathcal{D}$ and y is the label associated with x.

Algorithm 1 shows the general schema of the proposed approach, named ***L**earning to **A**ctive **L**earn by Instance **I**mportance based **Grad**ient **V**ariation* (LAL-IGradV). LAL-IGradV receives in input a (small) set of labeled instances LI, a set of unlabeled instances UI, a deep neural network model DNN, a regressor model R, the number $epch$ of active learning epochs, and the number k of unlabeled instances to select for oracle labeling at each active learning epoch.

Our proposed approach is comprised of two phases: initialization and an iterative phase. In the initialization phase, the algorithm first trains DNN using LI (line 1), randomly selects k unlabeled instances from UI and asks the oracle to label them, thus obtaining the initial set NLI of oracle-labeled instances (lines 2- 3). In each step of the iterative phase (lines 4-11), the set NLI of newly labeled instances is used to train the classifier together with the set LI (line 5). When retraining the classifier, every instance $x \in NLI$ is associated with its *importance score* r_x. The computation of the importance scores of the instance in NLI is performed using one of the techniques described in Section 3.1. Next, a regressor R is trained on the set $\{(x, r_x)|x \in NLI\}$ and *NLI* instances are added to *LS* (lines 6-7). The regressor R is then applied to the instances in UI so that, given an instance x, it predicts its importance score $\widehat{r_x}$ (line 8). Finally, the top-k instances having the greatest importance score are selected for oracle labeling and, once labeled, they replace the set NLI so to start the next active learning step (lines 9-11).

Following the model change framework [15], the importance score of an instance x measures the impact of having x in the training set for the obtained classifier. That is, the importance score of a (labeled) instance x w.r.t. a set of labeled instances is a measure of the difference between the parameters of the classifier θ trained over LI and the parameters of the classifier $\hat{\theta}$ trained over $LI \cup \{\langle x, y \rangle\}$, where y is the label of x. Unfortunately, in the case of neural network classifiers, for the most commonly used training methods, such as the stochastic gradient, such difference between the parameters of the model (almost) does not exist. To overcome this issue, we define different notions of importance score, as discussed next.

Algorithm 1: LAL-IGradV

```
Data: $LI$: set of labeled instances, $UI$: set of unlabeled instances, DNN: deep neural network
      model, $R$: importance score regressor, $epch$: maximum number of epochs, $k$: number of
      relevant instances to select
Train DNN on $LI$
$NLI \leftarrow$ Select $k$ instances from $UI$ uniformly at random
The oracle annotates the instances in $NLI$
for $i = 1 \ldots epch$ do
    Train DNN on $LI \cup NLI$ and compute importance score $r_x$, for each $x \in NLI$
    Train $R$ on the set of pairs $\{\langle x, r_x \rangle \mid x \in NLI\}$
    $LI \leftarrow LI \cup NLI$
    Apply $R$ to $UI$ instances to predict importance scores ($\widehat{r_x}$)
    $topK \leftarrow$ Select top-$k$ instances from $UI$ by importance score $\widehat{r_x}$
    The oracle annotates the instances in $topK$
    $NLI \leftarrow topK$
```

3.1. Importance scoring strategies

Let $f(x_i, \theta)$ be the output of a DNN model f characterized by a vector of parameters θ for an input x_i and let $X = \{x_1, \ldots, x_n\}$ be a set of instances used for training f, where each sample $x_i \in X$ is associated to a label y_i.

The training of the DNN f over X requires solving

$$\arg\min_{\theta} \left(\sum_{x_i \in X} \left(L(y_i, f(x_i, \theta)) + reg(\theta) \right) \right),$$

where $L(y_i, f(x_i, \theta))$ is the loss of the model for instance x_i and $reg(\theta)$ is the regularization of the parameters. The training of f is done by iteratively updating the parameters θ, through two steps: (i) computing the change in f w.r.t. all parameters, i.e., the *gradient*, defined as follows

$$\delta(X) = \frac{\partial}{\partial \theta} \sum_{x_i \in X} (L(y_i, f(x_i, \theta)) + reg(\theta)),$$

and (ii) updating θ using $\delta(X)$, i.e., $\theta_{k+1} = \theta_k - \eta \times \delta(X)$, where η is the update step size.

We define four strategies to associate each instance in NLI with its importance score during the training of the DNN classifier. The goal shared by the various techniques is to modify the training of the neural network model by accounting for the importance of the instances in NLI involved in each training step. Each of the proposed techniques makes use of the gradient corresponding to the instances currently in LI and NLI, i.e., $\delta(LI \cup NLI)$, hereinafter simply denoted as δ. The four proposed techniques differ in the way the importance of an instance x in NLI is calculated with respect to the single epoch. We will use symbol δ_x to denote the value of the gradient $\delta(\{x\})$, and $\delta_{\neg x}$ to denote the value of the gradient $\delta(LI \cup NLI \setminus \{x\})$. In the following, we describe our proposed techniques for computing the importance scores.

Direct similarity (*DS*) – given an instance x in NLI, this strategy compares the learning gradient of the neural network at the current epoch, δ, with the gradient calculated with respect to x only, i.e., δ_x. The importance score of x at the current epoch is defined as the cosine similarity between δ and δ_x, i.e., $r_x = cos(\delta, \delta_x)$. The rationale of this strategy is that an instance $x \in NLI$ is likely to be more important for the training of DNN at the current epoch if there is a small difference between the directions of the gradients δ and δ_x, as reflected by a high value of the cosine similarity between the two gradients. That is, the more the learning behavior of the neural network considering the whole training set is similar to the one of the same neural network trained on x only, the higher the importance of x is.

Ranked direct similarity (*RDS*) – this strategy first applies the *DS* technique, then the importance scores of the instances in NLI computed by *DS* are ordered and divided into three bins, which correspond to the top quartile of the importance scores, the bottom quartile, and the union of the second and third quartiles. The instances falling into the top quartile will be associated with score 1, the ones falling into the bottom quartile with score 0, and the other instances with score 0.5.

Leave-one-out distance (*LD*) – given an instance x in NLI, this strategy compares δ with the gradient calculated when leaving out x, i.e., $\delta_{\neg x}$. The importance score of x at the current epoch is defined as the complement of the cosine similarity (i.e., cosine distance) between δ and $\delta_{\neg x}$, i.e., $r_x = 1 - cos(\delta, \delta_{\neg x})$. The rationale of this strategy is that an instance $x \in NLI$ is likely to be more important for the training of DNN at the current epoch if leaving it out will lead to large differences between the learning behavior of the neural network considering the whole training set and the learning behavior of the same neural network trained without x, i.e., a large change in the direction of the gradient $\delta_{\neg x}$ w.r.t. the gradient δ, as reflected by a high value of the cosine distance between the two gradients.

Ranked leave-one-out distance (*RLD*) – analogously to *RDS* w.r.t. *DS*, the *RLD* strategy adds the same discretization step over the importance scores computed by *LD*.

4. Experimental Evaluation

Data. We used the well-known CIFAR-10 dataset [26], which consists of 60000 instances representing 32x32 colour images, labeled using 10 mutually exclusive classes, with 6000 images per class. The dataset is organized into 50000 instances as the training set and 10000 instances as the test set. The latter contains exactly 1000 randomly-selected images from each class, while the training set is comprised of five training batches, which contain 5000 images from each class. We divided the training set into two parts, the one corresponding to the set of labeled instances (LS), and the other corresponding to the set of unlabeled instances (US).

Baseline methods. We compare the performance of our methods with a Random baseline and the LCS method [8]. The Random baseline, hereinafter denoted as Rnd, simply selects k instances to be annotated at each epoch uniformly at random from the set of unlabeled instances. The LCS method follows an uncertainty sampling approach, therefore the unlabeled instance selection is driven by the uncertainty of the instances. More precisely, given an instance x and a classification model θ, the LCS method measures the uncertainty of x w.r.t. θ ($\phi(x)$) as $\phi(x) = (1 - P_\theta(y^*|x)) \times \frac{m}{m-1}$, where $P_\theta(y^*|x)$ denotes the probability that the model θ assigns to the label y^* for the instance x, y^* is the label for which θ yields the maximum probability on x (i.e., $y^* = \arg\max_y P_\theta(y|x)$), and m is the cardinality of the set of labels. Note that the

Table 1
Performance of our proposed methods: initial and final accuracy, percentage increase w.r.t. Rnd and w.r.t. LCS, and active learning time (sec) averaged over the epochs, for various percentage values of unlabeled instances.

	$A^{(0)}$	*DS*				*RDS*				*LD*				*RLD*			
		A	$\%_{\mathsf{Rnd}}$	$\%_{\mathsf{LCS}}$	time	A	$\%_{\mathsf{Rnd}}$	$\%_{\mathsf{LCS}}$	time	A	$\%_{\mathsf{Rnd}}$	$\%_{\mathsf{LCS}}$	time	A	$\%_{\mathsf{Rnd}}$	$\%_{\mathsf{LCS}}$	time
10%	0.793	0.831	2.32	0.43	186	0.832	2.44	0.54	191	0.831	2.28	0.39	625	0.828	1.90	0.01	769
20%	0.783	0.826	1.90	0.75	178	0.825	1.79	0.65	217	0.824	1.72	0.57	623	0.822	1.46	0.32	796
30%	0.784	0.827	1.95	0.50	170	0.828	2.06	0.61	250	0.826	1.75	0.30	620	0.822	1.46	0.32	827
40%	0.763	0.819	4.01	1.08	170	0.811	3.04	0.13	295	0.811	3.02	0.11	620	0.811	2.96	0.05	872
50%	0.733	0.801	5.97	2.84	162	0.800	5.82	2.70	352	0.799	5.80	2.67	619	0.779	3.07	0.03	1002
60%	0.728	0.801	6.32	3.21	162	0.798	5.96	2.86	423	0.795	5.57	2.48	614	0.777	3.20	0.18	1089
70%	0.708	0.778	6.49	2.50	154	0.778	6.38	2.40	513	0.773	5.82	1.86	607	0.760	4.01	0.12	1190
80%	0.640	0.705	5.39	1.82	139	0.704	5.27	1.71	613	0.700	4.62	1.08	604	0.694	3.78	0.27	1310
90%	0.570	0.644	5.89	2.22	129	0.636	4.60	0.98	732	0.632	3.95	0.35	602	0.636	4.59	0.97	1395

uncertainty function ranges between $[0, 1]$, where 1 is the most uncertain score.

Settings and assessment criteria. In our experimental evaluation, we used 6 Convolutional Neural Network (CNN) 2D layers, with 3 input channels, kernel size 3, stride size 3, padding size 1, ReLU activation function. The CNN module has on top a fully-connected network with an input layer of size 4096, one hidden layer with input size 4096 and output size 1024, another hidden layer with input size 1024 and output size 512, an output layer of size 10 (i.e., number of classes), and a dropout layers with probability 0.1.

In our LAL-IGradV algorithm, the DNN model was trained using cross entropy as loss function and Adam optimizer (with learning rate 1e-4 and weight decay 5e-4), a number of epochs equal to 10 for both the initialization step of training (Line 1) and the training steps in the main loop (Line 5). Also, the maximum number of iterations of the algorithm, i.e., number of epochs in the active learning process ($epch$) was set to 10. Unless otherwise specified, the number k of instances to select from UI was set to 500; the size of LI, resp. UI, was experimentally varied. As the regressor (R), we used two models: the Gradient Boosting Regressor, with least absolute deviations (LAD) loss function and 200 estimators, for the *DS* and *LD* strategies, and the Random Forest Classifier, with maximum depth 5, for the *RDS* and *RLD* strategies.

To simulate the oracle for annotating the instances, we resorted to the availability of class label information for the CIFAR-10 data: whenever an instance was used in the UI set, we masked its actual label during the learning process, and we unveiled the label only if the instance was selected within the $topK$ set of instances to annotate.

To assess the performance of the methods, we considered the accuracy of the classifier during the various training batches, in absolute terms as well as in terms of percentage increase w.r.t. the early accuracy of the classifier itself or the accuracy of a reference method. More precisely, we computed: the *accuracy at the initial step of training* of LAL-IGradV (line 1), denoted as $A^{(0)}$, and the accuracy at the end of the active learning process, denoted as A; the *percentage increase in the accuracy* of LAL-IGradV, which is defined as $100(A - A^{(0)})/A^{(0)}$; the *percentage increase in the accuracy* of LAL-IGradV w.r.t. Rnd, resp. LCS, which is defined as $\%_{\mathsf{Rnd}} = 100(A - A_{\mathsf{Rnd}})/A_{\mathsf{Rnd}}$, resp. $\%_{\mathsf{LCS}} = 100(A - A_{\mathsf{LCS}})/A_{\mathsf{LCS}}$, where A_{Rnd} and A_{LCS} denote the accuracy at the end of the active learning process for Rnd and LCS.

Results. Table 1 reports on the performance of our LAL-IGradV variants corresponding to the four importance scoring techniques, for varying percentages of the set of unlabeled instances (UI). As expected, the accuracy values (i.e., columns corresponding to A and $A^{(0)}$) tend to decrease as the percentage of unlabeled instances gets higher, since the LAL-IGradV method is forced to handle progressively reduced sets of labeled instances on its initial training. More interestingly, the percentage increase of each of the LAL-IGradV variants w.r.t. both Rnd and LCS is always positive — up to 6.5% against Rnd and up to 3.2% against LCS — and it tends to improve with higher percentages of unlabeled instances, with peaks around 70% against Rnd and around 50-60% against LCS. As concerns the impact of the importance scoring technique, we observe that all the LAL-IGradV variants are able to improve upon the accuracy at the initial training step. Moreover, the direct similarity based techniques, i.e., *DS* and *RDS*, reveal to be more efficient[1] as well as more accurate than the leave-one-out distance based techniques, for each percentage of unlabeled set. We tend to ascribe this fact to a higher sensitivity of the approach in capturing the gradient direction change due to the individual contribution of an instance rather than to the masking of a single instance in the training gradient, which would result in a more diluted signal of variation of the training gradient.

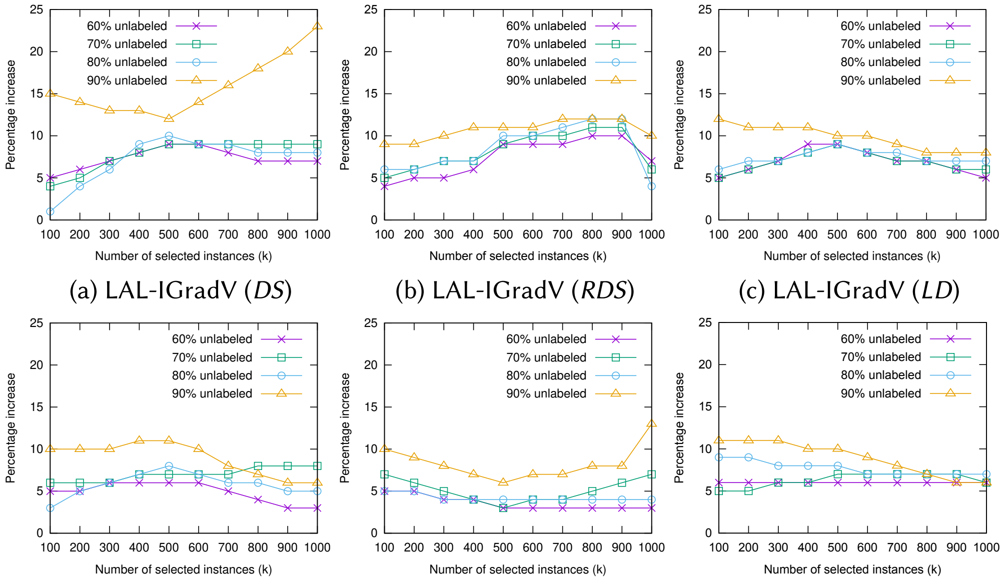
We analyzed the percentage increase in accuracy that each active learning method achieves by varying the fraction of unlabeled instances. As expected due to the advantage of performing an active learning task, the percentage increase values (results not shown) tend to improve for higher fractions of unlabeled instances. The trends are steeper for our LAL-IGradV methods (around 10% increase), particularly for *DS* and *RDS*, followed by LCS. Indeed, it is worth emphasizing that our LAL-IGradV methods achieve the best performance gain against the two baselines as the fraction of labeled instances becomes smaller.

In Fig. 1, we delve into the trends of accuracy percentage-increase obtained by a particular active learning method, for varying k, i.e., number of unlabeled instances to be selected at each epoch of the active learning process. At a first glance, in each of the plots, we notice that the curve of the percentage increase values over k is more likely to change for larger fractions of the set of unlabeled instances, with the most evident changes corresponding to 90%.

A few interesting remarks can be drawn from Figs. 1(a)-(d). When portions of UI below 90% are selected, we observe a relatively small range of variation of the percentage increase values (approximately from 5% to 10%), with peaks around $k = 500$ for the *DS* and *LD* variants, and around $k = 900$ for the *RDS* and *RLD* variants. This would hint at higher requirements (i.e., higher k) needed for the importance scoring strategies that compute discretized importance scores. Another remark is on the curves corresponding to the use of 90% of the set of unlabeled instances: compared to the cases with lower fractions of UI, the percentage increase values are higher on average, and the trends are quite different, especially for the *DS* variant where we observe a minimum (rather than a maximum) for $k = 500$. Apart from this exception, it is worth noticing that better percentage increase of accuracy do not necessarily correspond to a higher number k of selected instances. This might be explained since the more unlabeled instances are selected for labeling, the more the method is less likely to make a correct choice for changing the most the current model, as the latter is being trained only on few instances, thus lacking full knowledge on the class distribution of all the instances for available training.

Concerning the baseline methods, two different situations occur between the Rnd plot

[1] Experiments were carried out on an Intel Core i7 CPU @2.90GHz, 32GB RAM, with NVIDIA GeForce RTX 2070 Super GPU



(a) LAL-IGradV (*DS*)

(b) LAL-IGradV (*RDS*)

(c) LAL-IGradV (*LD*)

(d) LAL-IGradV (*RLD*)

(e) Rnd

(f) LCS

Figure 1: Percentage increase due to active learning, by varying the number of selected instances (k) and the percentage of labeled instances [5].

(Fig. 1(e)) and the LCS plot (Fig. 1(f)). The former shows a decreasing trend until mid values of k (i.e., around 500) followed by a rising trend, which sheds light on the divergent behavior of a random selection of the unlabeled instances w.r.t. all the other instance selection methods. Also, the LCS plot shows curves that tend to monotonically decrease, resp. remain substantially unchanged, for larger, resp. smaller, fractions of UI, which again puts in evidence how our LAL-IGradV variants behave differently from an uncertainty sampling approach like LCS.

5. Conclusions

We proposed a learning-to-active-learn approach whose key novelty is twofold: the integration of a regression-based meta-learning approach within a maximum model-change framework, and the definition of policies for scoring the instance importance based on the amount of change in the learning gradient of a deep neural network model. Our experimental evaluation has shown that our proposed LAL-IGradV outperforms both a random baseline and the LCS method, especially when the number of initially available labeled instances gets smaller. As a future work, we plan to evaluate the impact of measuring the importance of an instance not only in terms of its own contribution to the model change but also w.r.t. other instances according to some instance locality principle.

References

[1] B. Settles, Active Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

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

[3] W. Hsu, H. Lin, Active learning by learning, in: Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2659–2665.

[4] S. Ebert, M. Fritz, B. Schiele, RALF: A reinforced active learning formulation for object class recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3626–3633.

[5] S. Flesca, D. Mandaglio, F. Scala, A. Tagarelli, Learning to active learn by gradient variation based on instance importance, in: 2022 26th International Conference on Pattern Recognition (ICPR), 2022, pp. 2224–2230. doi:10.1109/ICPR56361.2022.9956039.

[6] D. D. Lewis, W. A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3–12.

[7] D. D. Lewis, J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in: Proc. of the Eleventh International Conference on Machine Learning, 1994, pp. 148–156.

[8] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing, 2008, pp. 1070–1079.

[9] T. Scheffer, C. Decomain, S. Wrobel, Active hidden markov models for information extraction, in: Proc. of the 4th International Conference on Advances in Intelligent Data Analysis, volume 2189 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 309–318.

[10] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proc. of the Fifth Annual ACM Conference on Computational Learning Theory, 1992, pp. 287–294.

[11] A. McCallum, K. Nigam, Employing EM and pool-based active learning for text classification, in: Proc. of the Fifteenth International Conference on Machine Learning, 1998, pp. 350–358.

[12] I. Dagan, S. P. Engelson, Committee-based sampling for training probabilistic classifiers, in: Proc. of the Twelfth International Conference on Machine Learning, 1995, pp. 150–157.

[13] P. Melville, R. J. Mooney, Diverse ensembles for active learning, in: Proc. of the Twenty-first International Conference on Machine Learning, 2004. doi:10.1145/1015330.1015385.

[14] R. Gilad-Bachrach, A. Navot, N. Tishby, Query by committee made real, in: Proc. of the Neural Information Processing Systems, 2005, pp. 443–450.

[15] B. Settles, M. Craven, S. Ray, Multiple-instance active learning, in: Proc. of the Twenty-First Annual Conference on Neural Information Processing Systems, 2007, pp. 1289–1296.

[16] W. Cai, Y. Zhang, Y. Zhang, S. Zhou, W. Wang, Z. Chen, C. Ding, Active learning for classification with maximum model change, ACM Trans. Inf. Syst. 36 (2017).

[17] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proc. of the Eighteenth International Conference on Machine Learning, 2001, pp. 441–448.

[18] Y. Guo, R. Greiner, Optimistic active-learning using mutual information, in: Proc. of the

20th International Joint Conference on Artificial Intelligence, 2007, pp. 823–829.

[19] R. Moskovitch, N. Nissim, D. Stopel, C. Feher, R. Englert, Y. Elovici, Improving the detection of unknown computer worms activity using active learning, in: Proc. of the 30th Annual German Conference on Artificial Intelligence, volume 4667 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 489–493.

[20] D. A. Cohn, Neural network exploration using optimal experiment design, Neural Networks 9 (1996) 1071–1083.

[21] A. J. Joshi, F. Porikli, N. P. Papanikolopoulos, Scalable active learning for multiclass image classification, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 2259–2273.

[22] H. T. Nguyen, A. W. M. Smeulders, Active learning using pre-clustering, in: Proc. of the Twenty-first International Conference on Machine Learning, 2004.

[23] Z. Xu, R. Akella, Y. Zhang, Incorporating diversity and density in active learning for relevance feedback, in: Proc. of the 29th European Conference on Information Retrieval, volume 4425 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 246–257.

[24] S. Huang, R. Jin, Z. Zhou, Active learning by querying informative and representative examples, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 1936–1949.

[25] K. Konyushkova, R. Sznitman, P. Fua, Learning active learning from data, in: Proc. of the Annual Conference on Neural Information Processing Systems, 2017, pp. 4225–4235.

[26] K. Alex, H. Geoffrey, Learning multiple layers of features from tiny images (2009).