TOWARDS PRACTICAL ACTIVE LEARNING FOR CLASSIFICATION

TOWARDS PRACTICAL ACTIVE LEARNING FOR CLASSIFICATION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Tuesday 20, November 2018 at 10:00 o'clock

by

Yazhou YANG

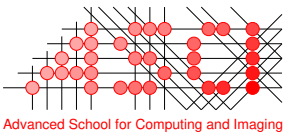
Master of Engineering in Control Science and Engineering,
National University of Defense Technology
born in Henan, China.

This dissertation has been approved by the promotors:

Prof. dr. ir. M. J. T. Reinders and
Prof. dr. M. Loog

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. M. J. T. Reinders,	Delft University of Technology, promotor
Prof. dr. M. Loog,	Delft University of Technology, University of Copenhagen, promotor
Independent members:	
Prof. dr. D. M. Gavrila	Delft University of Technology
Prof. dr. ir. D. de Ridder	Wageningen University & Research Centre
Prof. dr. H. H. Hoos	Leiden University
Dr. C. Sanchez	Radboud University Medical Center
Dr. F. A. Oliehoek	Delft University of Technology
Prof. dr. ir. R. L. Lagendijk	Delft University of Technology, reserve member



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 396.

Keywords: Active learning, machine learning, experimental design, image classification, benchmark

Printed by: ProefschriftMaken `www.proefschriftmaken.nl`

ISBN 978-94-6380-102-7

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

To my family

CONTENTS

1

INTRODUCTION

1.1. WHAT IS ACTIVE LEARNING?

The primary target of machine learning is to learn from existing examples and discover general hidden patterns which can be used to evaluate the unseen future objects. When it comes to the input of machine learning algorithms, usually called training data, one important concern is whether these training data points are labeled, i.e., each sample has corresponding output variables such as class label or other meaningful tags. The majority of machine learning algorithms are supervised learning methods, where all the training samples are labeled [1, 2]. On the contrary, there also exists many unsupervised learning algorithms in which the input data are totally unlabeled [3]. A special case is semi-supervised learning, where a training set typically consists of a limited number of labeled data and a large amount of unlabeled data [4].

In many real-world applications, to obtain satisfactory classification or regression performance, we, in general, first categorize a subset of training data available and then apply supervised machine learning models. However, categorizing a large amount of data is usually time-consuming and expensive. Examples of such challenging labeling tasks are:

- Medical image annotations. In the field of medical image analysis, various annotation tasks have to be conducted by specialists that are highly trained in a particular branch of medicine. Hiring these medical experts for annotation is very expensive, especially when the manual annotations take a long time. For example, Kohli *et al.* [5] show several practical issues of collecting and annotating medical imaging data, which indicates that currently labeling medical images is costly and difficult.

- Large-scale visual database annotations. For some large-scale visual datasets which generally contains a large number of different categories, it is very difficult to conduct manual annotation since human experts have to choose one class label from hundreds of candidates. For example, the ImageNet database [6] contains more

than 1000 categories and over 14 million images. This situation is even worse when human annotators are asked to also provide the exact position of objects like bounding box, e.g. the Youtube-BoundingBoxes dataset [7] consists of about 380,000 videos and 5.6 million bounding boxes.

In these fields where data is abundant but labels are expensive to obtain, it is particularly valuable if we can carefully choose the most useful instances that are to be labeled instead of just randomly selecting a subset of unlabeled instances. Active learning techniques have been proposed to tackle the labeling challenge by selectively querying useful instances for human annotation to achieve similar performance with as few labeled instances as possible. The key hypothesis is that if an active learner can freely choose which instance to query and can learn from the feedback (i.e., usually the true label, obtained from an oracle, e.g. a human expert), it can achieve a good learning performance with less labeled data.

Active learning has been applied to a large number of real-world applications, e.g. image classification [8–14], image retrieval [15–17], remote sensing [18–22], text categorization [23–26], named entity recognition [27, 28], natural language processing [29–31], and recommender systems [32–35]. The aforementioned works have verified that active learning can indeed reduce the labeling cost.

Generally speaking, there are two different scenarios for active learning, depending on how the unlabeled instances are posed to the active learning algorithms [36].

- Pool-based active learning. It assumes that a large pool of unlabeled instances is given in advance. An active learner is required to select a single instance or a batch of instances from the unlabeled pool in each iteration, and the chosen samples will be categorized by a human annotator and added into the labeled data set. Figure 1.1 illustrates an example of pool-based active learning.

- Stream-based active learning. In this scenario, the unlabeled instances are sampled from an underlying data distribution. Typically, the unlabeled instances are queried one by one and active learning algorithms have to decide whether or not to ask a human expert to label it. This setting is preferred in many real-world applications that data is presented in a stream, such as in visual surveillance tasks [37], or spam filtering [38].

An advantage of stream-based strategies over pool-based ones is its computational efficiency: there is no need to go through the data pool to query the best sample. But the price of high efficiency is a weaker performance: Ganti and Gray [39] found that stream-based active learning algorithms are likely to perform worse than pool-based methods, for instance, more data points are queried for human annotation in the stream-based setting than that in the pool-based setting. One reason is that in the stream-based setting, active learning algorithms can not go through all the unlabeled data to select the most useful samples. It is likely that before the most informative samples are presented in a stream, the annotation budget is already finished. Most active learning algorithms focus on the pool-based scenario [11, 15, 24, 40–46]. In this thesis, we mainly concentrate on exploring new strategies for this setting as well.

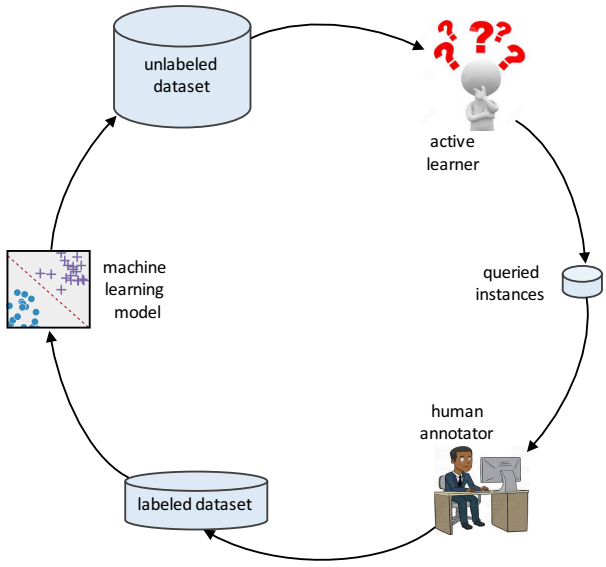


Figure 1.1: Illustration of pool-based active learning. Based on the already labeled data the active learner picks several samples to be annotated from a set of unlabeled samples and asks the oracle (human annotator) to label those. This can be repeated multiple times.

In the remainder of this chapter, we first provide a brief overview of various active learning strategies in the literature, followed by a summary of the contributions of this thesis. Overviews, reviews, and surveys of active learning can be found in the following list of works: [36, 47–51].

1.2. GENERAL ACTIVE LEARNING STRATEGIES

The core of active learning algorithms is the selection criterion which measures the usefulness of unlabeled instances, i.e. for which instances information about the label would result in the largest improvement for the classifier to be built. Generally speaking, active learning methods can be roughly divided into two categories: label-free and label-based approaches, depending on whether these active learning algorithms make use of the true label information obtained from human experts or not. The vast majority of active learning algorithms belong to the label-based category, where the true labels of selected instances must be obtained to choose the subsequent queries.

1.2.1. LABEL-FREE ACTIVE LEARNING

As label-free approaches do not use label information for selecting instances, they can also be seen as unsupervised active learning approaches. Some examples are: optimal experimental design based methods [26, 52–54], dissimilarity-based approaches [55], and density or diversity-based methods [41, 56].

Experimental design methods: Optimal experimental design approaches select the

most representative samples by minimizing some criteria of a statistical model. For example, Yu *et al.* [52] proposed the so-called transductive experimental design (TED) that selects representative instances by minimizing the expected predictive variance on unlabeled data. Cai and He [26] argued that TED fails to consider the manifold structure and proposed to add a manifold regularizer to TED to capture the local geometry of the data. Similar to Cai and He [26], He *et al.* [57] utilized a graph Laplacian regularizer to preserve the intrinsic geometry of the pool of data. As TED can be transformed to a kernelized version, Wang *et al.* [58] proposed a learned kernel to TED via multiple kernel learning [59]. To further increase the diversity of selected instances by TED, Shi and Shen [60] added a diversity regularizer to TED. There are also two graph-based variance minimization methods [53, 54]. The difference between these two methods and TED is that they minimize the predictive variance of a Gaussian random field while TED minimizes the predictive variance of regularized least squares.

Dissimilarity-based methods: Another category of label-free methods is the dissimilarity-based methods. For instance, Elhamifar *et al.* [55] proposed a dissimilarity-based technique which chooses representative samples by minimizing the dissimilarity between the selected instances and all data in the pool.

Diversity/density-based methods: Other alternatives propose to select diverse instances which are far from already labeled instances. One example is Kernel Farthest-First (KFF) [41], which selects the samples that are farthest from currently labeled data. Hu *et al.* [56] considered not only how far the sample is from labeled data (what they refer to as diversity) but also whether the sample is located in a dense region (what they refer to as density).

The advantage of label-free approaches over label-based methods is that they can act independently from the true labels such that all the instances to be labeled can be determined in advance. This means that human experts are not required to be present during the process of selecting instances. They can more freely decide when to carry out the annotation task. However, this also means that these label-free active learning algorithms only concentrate on *exploration* by sampling representative instances and cannot consider *exploitation*, i.e. selecting instances to refine the classification boundary. Brinker [61] found that these pure exploration based methods performed well in the very early stage but failed to outperform exploitation-based models in the later stage.

1.2.2. LABEL-BASED ACTIVE LEARNING

Most active learning algorithms exploit the true labels provided by the oracle and decide which instance to query by training on currently labeled data. The received label information will have an influence on which instance to select. Depending on the number of selected instances in each iteration, label-based active learning can also be categorized into two classes. The first one is myopic active learning, where a single instance is queried at a time. The other one is batch mode active learning in which a batch of samples is selected and labeled simultaneously. Normally, the batch size is equal to or greater than 2, which means that the correlation and redundancy between selected samples should be taken into consideration. We start with an overview of existing myopic active learning algorithms, followed by a short survey of batch mode active learning methods.

MYOPIC ACTIVE LEARNING

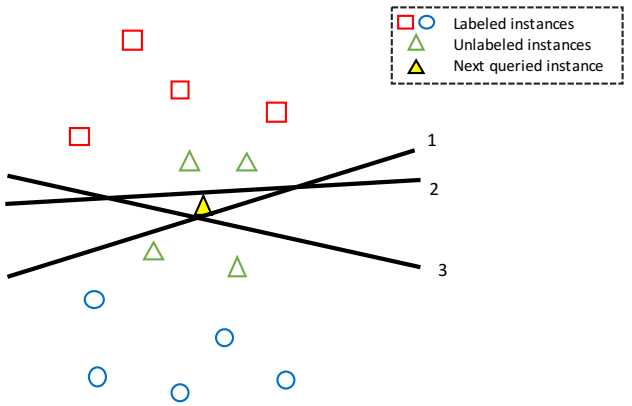
Extensive studies have been undertaken for myopic active learning. Here we classify existing myopic active learning into the following categories according to the criteria which are used to measure the usefulness of unlabeled samples.

Query-by-committee methods: Seung *et al.* [62] first proposed the so-called query by committee (QBC) technique. QBC constructs a committee which consists of a number of different models. Each committee member can vote on the labeling of unlabeled instances. The instance which causes greatest disagreement within the committee will be selected for annotation. For exemplified in Figure 1.2a, if an instance is classified as positive by half of committee members and is categorized as negative by the other half, the committee is most uncertain on the label of this sample. Hence getting information about its label will reduce the uncertainty of the classifier most. Several variants of QBC have been proposed with respects to two aspects: how to generate the committee and how to measure the disagreement. Seung *et al.* [62] used Gibbs sampling to generate hypotheses which are consistent with currently labeled data. Mamitsuka [63] proposed two variants, Query by Bagging and Query by Boosting, which employ the bagging technique [64] and AdaBoost algorithm [65] to construct committees, respectively. McCallum and Nigam [23] trained the classifiers on the same training data but with different parameters. Melville and Mooney [66] generated artificial training data to obtain diverse committee members. There are also some methods which split the feature space to generate different models [15, 67]. In [68], the committee members are the learning models that are trained on currently labeled data and an additional unlabeled instance associated with a possible label. As measure of disagreement, vote entropy [69], Kullback-Leibler (KL) divergence [23], and Jensen-Shannon divergence [70] have been proposed. The variance of class probability can also be used to estimate the disagreement [71].

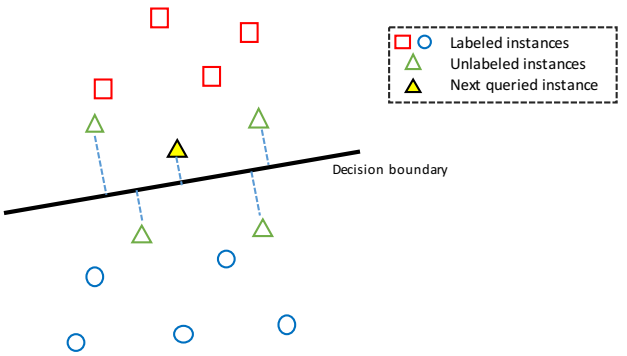
Uncertainty sampling methods: Another popular active learning method is uncertainty sampling. The main idea is that we should query the instance that lies closest to the decision boundary as knowledge about its label will help us most to refine the decision boundary. There are many different ways to define "uncertainty", such as the least confidence [72], the entropy [73], the smallest margin [9, 74], the nearest distance to the decision boundary [24]. The last criterion is illustrated in Figure 1.2b. It first trains SVMs on currently labeled data and then selects instances which are nearest to the decision boundary. Querying those instances helps us the most ot obtain information about the true decision boundary according to this criterion.

Expected error reduction methods: The motivation of expected error reduction (EER) is to select samples which can reduce the future generalization error once labeled. Since the future test data is not available, Roy and Mccallum [40] used the expected error on the remaining unlabeled instances as an approximation of the generalization error. An optimistic variant of EER was proposed by Guo and Greiner [75], who consider the minimal error that can be obtained instead of the expected error. One issue of EER approaches is that the estimated error is an approximated error of the future error. This approximated error may be distant from the true one, especially when the number of labeled instances is limited such that a reliable learning model can not be obtained. Inaccurately estimated error would mislead us to select uninformative samples.

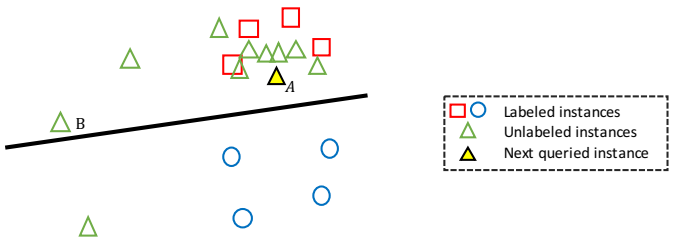
Expected variance reduction methods: Schein and Ungar [76] proposed a technique



(a) Query-by-committee. Three straight lines represent three different classifiers, which are on behalf of three committee members. The yellow triangle point is chosen since it causes the maximum disagreement among three committee members.



(b) Uncertainty sampling based on SVMs. Labeling the yellow point can help us refine the classifier to the utmost extent.



(c) Density/diversity based methods. The instance A is selected as the next queried instance, though it is less uncertain than the instance B. The reason is that the instance A is representative of other unlabeled instances.

Figure 1.2: Illustration of three types of active learning algorithms. One square and circle represent the labeled instances from two classes while one triangle represents an unlabeled instance.

called Expected Variance Reduction (EVR). Instead of minimizing the future generalization error like EER, EVR selects the sample that can reduce the expected variance of the model's output. A disadvantages of EVR is that it suffers from a very high computational complexity, which makes it impractical on large-scale datasets.

Maximum model change methods: Another approach is to select the instance which can cause maximum change of the current learning model once added to the training set. The intuition is that if one instance can impart greatest changes to the model, this instance is more informative than other instances. Settles *et al.* [77] selected the instance which results in the maximum gradient of the objective function once labeled. Freytag *et al.* [78, 79] proposed several approaches to efficiently calculate the expected model changes for Gaussian process regression models. Similarly, Cai *et al.* [45] approximated the model change with the expected gradient of the empirical risk when a new sample is annotated and adapted their principle to two popular classifiers, SVM and logistic regression.

Minimum loss increase methods: These methods directly measure the loss function of some learning models (e.g. SVM and logistic regression) when a new instance is to be labeled. They will query the sample which is most likely to result in a minimum increase of the loss. This idea was first proposed by Hoi *et al.* [80], who presented a technique called min-max view active learning since it considers the worst-case scenario. Their technique proposed to query the instance that leads to the smallest value of the objective function of SVMs. Yang and Loog [81] considered a variant of this method for logistic regression. Huang *et al.* [82] presented the so-called QUIRE method which shares a similar idea with [80]. The key difference is that QUIRE considers a regularized least square model.

Density/diversity based methods: There are many works which incorporate a density or diversity measure into active learning [11, 25, 27, 83–86]. The intuition behind using a density measure is that active learning should select the samples which are located in dense regions of the input space. The diverseness principle expresses that the queried samples should be different from each other, i.e. the redundancy between selected data should be small. Many studies try to combine the density or diversity measure with some uncertainty measures. This combination is usually implemented by using a weighted sum or weighted product. For example, Settles and Craven [84] proposed a density-weighted technique where the density is measured by the overall similarity between queried instance and all remaining unlabeled instances. Their method calculated a weighted product of their density criterion and an entropy criterion. Zhu *et al.* [25, 87] proposed to combine uncertainty and density, in which the density is measured based on the cosine similarity. Gu *et al.* [85] presented a technique that jointly maximizes the density and minimizes the redundancy (in other words, enhances the diversity). Figure 1.2c gives an example of density/diversity based methods. The unlabeled point B is closer to the decision boundary than point A, which means that B is more uncertain than A. However, this algorithm chooses to label point A because querying A is likely to give us more information about the labels of the remaining unlabeled instances.

Disagreement-based methods: The aforementioned strategies are built based on a variety of heuristics. Though they empirically perform well in many real-world applications, most of them do not have any theoretical guarantee on their performance. There is

a particular active learning technique, called disagreement-based active learning, which behaviour has been theoretically analyzed in various settings. It was first studied by Cohn *et al.* [88] in the realizable scenario where the data is linearly separable. The idea is that we maintain a set of candidate hypotheses that are consistent with all labels revealed so far, check the unlabeled instances one by one, and select the instances whose labels cause disagreements among at least two hypotheses. Balcan *et al.* [89] extended that method to the agnostic scenario in which label noise and model misspecification exist. There are a series of variants of disagreement-based methods that provided theoretical guarantees under various conditions [90–96]. Generally, we see an apparent divide between heuristics-based active learning algorithms and the theoretical ones: the former depicts promising empirical performance without any mathematical guarantee whereas the latter has solid theoretical guarantees without empirical evidences that show it works well in practice. Finding concrete directions toward bridging this divide can lead to a deeper understanding of how to design good active learning strategies, theoretically and practically.

BATCH MODE ACTIVE LEARNING

The advantage of batch mode active learning over myopic active learning is that it does not need to train the model many times during a single selection step and is more suitable on some parallel labeling platforms. However, there also exists several challenges for batch mode active learning. The first one is that selecting k samples from a pool of n instances may lead to computational complications as the number of possible choices can be very large. The second challenge lies in the formulation of an appropriate criterion to measure the overall information carried by a batch of samples. Brinker [61] found that simply using a myopic selection criterion often leads to poor performance since it disregards the redundancy among selected instances.

Existing batch mode active learning algorithms can be roughly divided into three categories: clustering-based methods, exploration-exploitation approaches, and the remaining algorithms which formulate batch selection as some combinatorial optimization problems.

Clustering-based methods: Clustering-based methods typically first select the top m instances ($m > k$), based for instance on some myopic criterion. Then they partition these candidates into k clusters and eventually choose one instance from each cluster [97, 98]. Figure 1.3 illustrates the main idea of these approaches (the batch size $k = 2$). In this example, $m = 13$ unlabeled instances are selected based on the uncertainty scores and are divided into two clusters. The instance which is near to the centroid of one cluster is queried for labeling. A shortcoming of these approaches is that their performances are sensitive to the parameter m, which is hard to set. For instance, in Figure 1.3, there is a cluster of unlabeled instances at the upper right corner that are not selected for clustering. Including these instance for clustering is likely to change the result of batch selection.

Exploitation-exploration methods: The second class of methods first balances a measure of exploitation (i.e. uncertainty) and a measure of exploration (e.g. diversity or density) via a trade-off parameter and then selects the top k instances based on the combined criterion [44, 46, 47, 61, 80, 99–103]. For example, Yang *et al.* [101] proposed

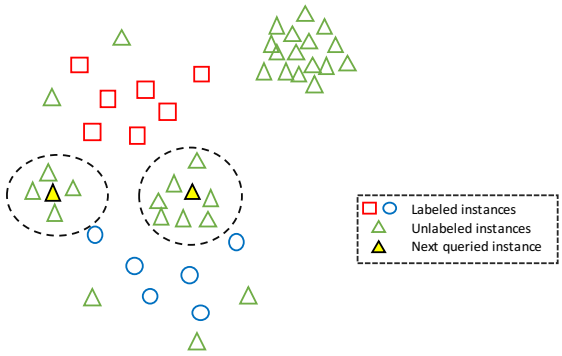


Figure 1.3: Clustering-based batch mode active learning with a batch size $k = 2$.

a multi-class batch mode active learning by combining uncertainty and diversity with a trade-off parameter. Similarly, Chakraborty *et al.* [44] proposed to select a batch of samples which are highly uncertain and have minimal redundancy among each other. The key differences among these approaches are the criteria used to evaluate the exploitation and exploration. They share a common weakness: their performances are very sensitive to the choice of the trade-off parameter. How to set this parameter is a challenge referred to as the exploration-exploitation dilemma [104–106].

Combinatorial optimization methods: The third class is composed of various approaches which deal with batch selection using some, more or less, sophisticated optimization techniques [13, 107–110]. For example, Hoi *et al.* [13] extended the work of Zhang and Oles [111] to the batch mode setting and solved the combinatorial optimization problem by using a greedy algorithm. Guo [108] put forward an NP-hard combinatorial optimization problem to query a batch of examples. Chen and Krause [109] viewed the batch selection as an adaptive submodular problem and proposed a greedy solution. The drawback of these algorithms is that they usually rely on a relaxation of the actual problem and will typically converge only to a local optimum.

Some of the aforementioned works are specifically designed for one particular classifier. The methods in [11, 107, 112] use logistic regression for instance, while the approaches in [61, 80] employ SVMs. Since deep convolutional neural networks (CNN) [113] have attracted much attention in recent years, many efforts have been devoted to combining active learning and CNNs in the batch mode setting. The reason of only batch mode setting being considered is that training the CNNs is usually time-consuming and batch selection can save training cost. Stanitsas *et al.* [114] investigated several active learning strategies for actively selecting samples to train CNN models. Wang *et al.* [115] proposed to use a self-labeling strategy which presents pseudo labels to high confidence samples and used pseudo-annotated samples to jointly fine-tune the CNN models. Finally, they applied uncertainty sampling techniques to choose the top k instances. Sener and Savarese [116] formulated active learning as a core-set selection problem based on the extracted features by using CNN models. Yang *et al.* [117] first selected top m ($m > k$) instances based on the uncertainty scores and then chose the most representative in-

stances by solving an approximated maximum set cover problem.

Several approaches have been proposed to integrate deep transfer learning with active learning [118–120]. For example, two active selection criteria, uncertainty and diversity, are combined to select samples for fine-tuning deep neural networks [119]. Huang *et al.* [120] proposed a new active learning criterion called distinctiveness and further combined distinctiveness and uncertainty for sample selection. Also meta-learning is combined with active learning by learning an active learner [121–123]. For example, Contardo *et al.* [121] proposed to learn the label acquisition strategy using bi-directional recurrent networks and selected all the required samples in a single step. Bachman *et al.* [122] learned effective active learning algorithms in an end-to-end fashion for pool-based setting while Woodward and Finn [123] focused on the setting where a stream of samples come one after the other.

1.3. CONTRIBUTIONS

This thesis focuses on pool-based active learning, especially for classification tasks. Our main contributions are providing a benchmark and comparison of current active learning algorithms, proposing new myopic and (single) batch mode active learning strategies, and investigating how to efficiently construct an initial labeled subset. In the remainder of this section, I provide a more detailed overview of the individual chapters.

In Chapter 2, we provide a comparison of state-of-the-art active learning algorithms on the basis of a logistic regression classifier and explicate the links and relationships between methods. Specifically, a preference map is proposed to reveal characteristic similarities and differences of compared methods. A fair and comprehensive comparison of the empirical performance of these active learning methods is conducted on three synthetic datasets and 44 real-world datasets. We find that uncertainty sampling, one of the earliest and simplest suggested active learning methods, performs exceptionally well in comparison with other supposedly more sophisticated algorithms.

Motivated by the fact that many active learning algorithms fail to outperform uncertainty sampling, we focus on investigating new myopic active learning strategies which can perform better than the current state of the art in Chapters 3 and 4. These two chapters concentrate on retraining-based active learning methods which estimate the usefulness of an instance by adding it to the training set and examining its influence to the current model. Since the true label of the selected instance is unknown before querying a human expert, existing retraining-based methods resort to calculating either the average-case [40] or worst-case criteria [80] with respect to the unknown label. Chapter 3 proposes a new technique which uses the uncertainty information (measured by the estimated posterior probabilities) to address the problem of not knowing the true labels. In particular, this technique estimates the usefulness of unlabeled instances by incorporating uncertainty sampling with retraining-based algorithms. We employ this new technique on two state-of-the-art retraining-based models and verify its effectiveness.

Chapter 4 proposes a new myopic active learning method which measures the usefulness of unlabeled data according to the predictive variance. The idea is that an instance has a large uncertainty if its posterior probability is very susceptible to the variations of input training data and possible labelings. More specifically, we keep track of the estimated probability of each unlabeled instance during the retraining procedure.

Two types of variance are exploited to measure the informativeness and representativeness of unlabeled instances. The proposed method clearly improves upon uncertainty sampling and other state-of-the-art algorithms in both binary and multi-class tasks.

Different from previous chapters which focus on myopic active learning, Chapter 5 turns to batch mode active learning. We consider the scenario in which human annotators are not readily available during the active selection procedure. Therefore, a single shot setting is studied: all the required instances should be selected in one pass. The challenge is that no human annotation can be exploited during the selection process. We turn the single shot selection into a sequential selection by substituting the human annotator for multiple pseudo annotators. These pseudo annotators uniformly and randomly annotate queried samples, which urges standard active learners to explore. Furthermore, the exploratory behavior is promoted by selecting the most representative sample via minimizing nearest neighbor distance between unlabeled data and queried data.

Chapter 6 studies a practical issue for myopic and batch mode active learning: how to initialize active learning algorithms. That is finding a set of labeled samples which contains at least one instance from each class. The goal is to construct such an initial set with as few annotations as possible. Some previous methods which can be used for the initialization problem are revisited and compared with our proposed criterion, the Nearest Neighbor Criterion. Our method sequentially queries the most representative instance from unlabeled data such that the overall distance between queried data and unlabeled data achieves a minimum value. Our method can be seen as a deterministic greedy clustering method, which can find the representative samples in an efficient way. Experiments on various test sets show that the number of queried samples obtained by our method is much less than that of other compared algorithms.

2

A Benchmark and Comparison of Active Learning for Logistic Regression

Logistic regression is by far the most widely used classifier in real-world applications. In this chapter, we benchmark the state-of-the-art active learning methods for logistic regression and discuss and illustrate their underlying characteristics. Experiments are carried out on three synthetic datasets and 44 real-world datasets, providing insight into the behaviors of these active learning methods with respect to the area of the learning curve (which plots classification accuracy as a function of the number of queried examples) and their computational costs. Surprisingly, one of the earliest and simplest suggested active learning methods, i.e., uncertainty sampling, performs exceptionally well overall. Another remarkable finding is that random sampling, which is the rudimentary baseline to improve upon, is not overwhelmed by individual active learning techniques in many cases.

This chapter is published as:
Yazhou Yang, and Marco Loog."*A Benchmark and Comparison of Active Learning for Logistic Regression.*" Pattern Recognition 83C (2018) pp. 401-415.

2.1. INTRODUCTION

In practice, it is easy to acquire a large amount of data, yet difficult, time-consuming, and expensive to label data since human experts are usually involved [36]. For instance, collecting millions of images from Google is not that difficult, while categorizing these images may need a lot of manpower and other resources. Active learning addresses this challenge by selecting the most valuable subset from the whole data set for human annotation. Many research studies have demonstrated that active learning is effective in maintaining good performance while reducing the overall labeling effort over a diverse range of applications, such as text categorization [24, 26], medical image classification [13, 14], remote sensing [19, 20, 22], image retrieval [15–17] and natural language processing [30].

To choose the most informative subset, it is of vital importance to choose an appropriate criterion which measures the usefulness of unlabeled instances. Most commonly used criteria in active learning include query-by-committee [62], uncertainty sampling [24], expected error minimization [40, 75, 124], and variance reduction [52, 76, 111], variance maximization [125], maximum model change [45, 77, 79, 126] and the "min-max" view active learning [80, 82]. They are derived from diverse heuristics and classifier dependent. Some of them are specifically designed for one particular classifier, *e.g.* the simple margin criterion for support vector machines [24], while others can be adapted to different types of classifiers, *e.g.* expected error reduction for logistic regression and naive Bayes [40].

In this work, we benchmark the state-of-the-art active learning algorithms built on logistic regression. Logistic regression is chosen because it is the most widely applied classifier in general and especially outside of machine learning in the applied sciences[1]. In addition, it is also used by most active learners (see, for instance, [13, 43, 75, 76, 107, 112, 127–130]). In part, the latter is because logistic regression readily provides an estimate of the posterior class probability, which is often exploited in active learning. In the binary classification setting, logistic regression models a posterior probability $P(y_i|x_i) = 1/(1+\exp^{-y_i w^T x_i})$, where $x_i \in \mathbb{R}^d$ is a training feature vector labeled with $y_i \in \{+1,-1\}$ and w is the d-dimensional parameter vector that is determined at training time. During training, we minimize the log-likelihood of the training data $\mathscr{L}$ to learn the model parameter w as follows:

$$\min_w \frac{\lambda}{2}\|w\|^2 + \sum_{x_i \in \mathscr{L}} \log(1+\exp^{-y_i w^T x_i}) \tag{2.1}$$

where $\|w\|^2$ is a regularization term for which λ controls its influence.

All in all, we study six different categories of active learning algorithms in which nine active learners are compared in an extensive benchmark study. Our work differs from two relevant earlier surveys on active learning [36, 49] in two important respects: (1) our work constructs extensive experiments to investigate the empirical behaviors of these active learning algorithms while these two surveys do not compare the performance of

[1] An advanced search on www.nature.com on October 1, 2017, gives us, for example, 1,126 hits for "support vector machine", 6,182 for "nearest neighbor" (containing more hits than just to the classifier), 1,231 for "LDA", and 14,715 for "logistic regression". Other classifiers are retrieved even less often.

different methods; (2) our paper presents a detailed summary of the active learning algorithms on the basis of logistic regression classifier because of its popularity while these two surveys offer an overview of existing active learning algorithms without specifying a type of classifiers. We believe that an empirical comparison can lead to a better understanding of the characteristics of active learning algorithms and provide guidance to the practitioner to choose a proper active learning algorithm. We should also mention the work by Schein and Ungar [76] here, that already provided an evaluation of active learning methods using logistic regression. In this chapter, however, we compare some new methods, which appeared only recently [11, 45, 82], and we generally provide a fair and comprehensive comparison with much more extensively conducted experiments. We also investigate how active learning algorithms generally perform in comparison to random sampling, and point out the underlying relationships among the compared methods. The computational cost of each method is also evaluated.

In this chapter, we focus on the pool-based setting, where few labeled samples and a large pool of unlabeled samples are available [36]. We consider the myopic active learning which assumes that a single unlabeled instance is queried at a time. Batch mode active learning, which selects a batch of examples simultaneously, is not considered in this work and we refer to [13, 107, 131–134] for further background of typical approaches.

The main contributions of this work can be summarized as follows:

- A review of the state-of-the-art active learning algorithms built on logistic regression is presented, in which links and relationships between methods are explicated;
- A preference map is proposed to reveal characteristic similarities and differences of the selection locations in 2D problems;
- Extensive experiments on 44 real-world datasets and three artificial sets are carried out;
- Insight is provided for the behaviors of classification performance and computational cost.

2.1.1. OUTLINE

The remainder of this chapter is organized as follows. Section 2.2 describes the general procedure of active learning and reviews the various approaches to active learning built on logistic regression. At the same time it sketches the relationships among different methods. Extensive experimental results on synthetic and real-world datasets are given in Section 2.3. The experimental setup is described and the outcomes are reported. More importantly, it provides an extensive discussion of the findings and aims to critically evaluate these compared methods. Section 2.4 concludes our work.

2.2. ACTIVE LEARNING STRATEGIES AND METHODS

For myopic active learning in the pool-based scenario, we assume that a small set of labeled instances with a large pool of unlabeled samples are available. Let $\mathscr{L} = \{(x_i, y_i)\}_{i=1}^{l}$ represent the training data set that consists of l labeled instances and let $\mathscr{U}$ be the pool of unlabeled instances $\{x_i\}_{i=l+1}^{n}$. Each $x_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in C$ is the class label of x_i. In this work we restrict ourselves to binary classification,

which does not pose any essential limitation. For this reason, C is simply taken to be the set $\{+1,-1\}$. The active learner will select an instance x^* from the unlabeled pool based on its measure of utility, obtain the corresponding label y^* by manual annotation and extend the training set with the new labeled sample $\mathscr{L} = \mathscr{L} \cup (x^*, y^*)$. The whole procedure is repeated until some stopping criteria are satisfied.

The remaining part of this section presents six different categories of active learning algorithms built on logistic regression, i.e., uncertainty sampling, error reduction, variance reduction, minimum loss increase, maximum model change and an adaptive approach, one per subsection. As also shown in Fig 2.1, nine different active learners which relate to the above six categories are used in our benchmark and comparison.

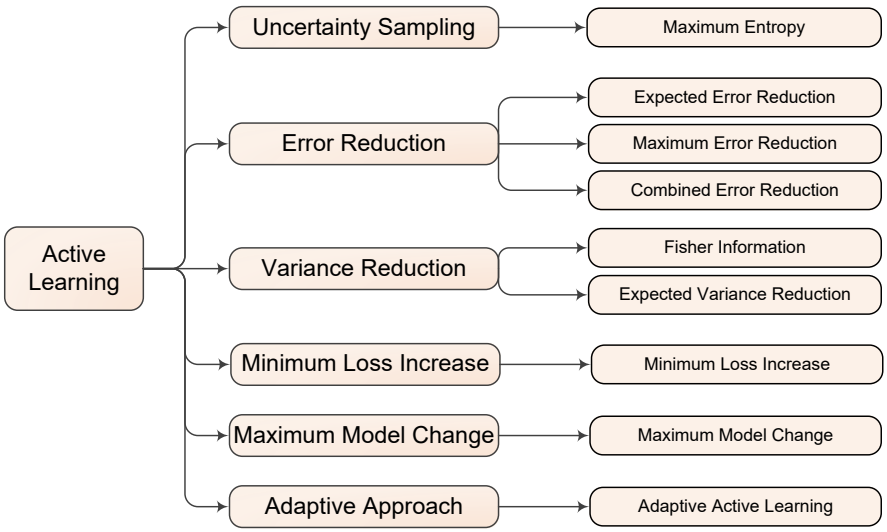


Figure 2.1: Nine active learners from six different categories are used in our comparison.

2.2.1. UNCERTAINTY SAMPLING

Uncertainty sampling, which selects the instances for which the current classifier is least certain, is a widely used active learning method [36, 73]. Querying these least certain instances can help the model refine the decision boundary. Intuitively, the distances between unlabeled instances and the decision boundary can be measures of the uncertainty. Tong and Koller [24] proposed a simple margin approach which queries the instance closest to the decision boundary.

Entropy is a different and more widely used general measure of uncertainty [135]. Entropy-based approaches query the instances with *maximum entropy*:

$$x^* = \arg\max_{x \in \mathscr{U}} -\sum_{y \in C} P_{\mathscr{L}}(y|x) \log P_{\mathscr{L}}(y|x) \tag{2.2}$$

where $P_{\mathscr{L}}(y|x)$ is the conditional probability of y given x according to a logistic classifier trained on $\mathscr{L}$. This method is called ENTROPY for short. It calculates the entropy of

each $x \in \mathscr{U}$ and selects the instance x^* which has maximum entropy. It can be used with any classifier that produces probabilistic outputs. For binary classification, ENTROPY is equivalent to the simple margin approach [24].

One of the main risks of such uncertainty sampling based approaches lies in the fact that, due to a lack of exploration, they can get stuck at suboptimal solutions, continuously selecting instances which do not improve the current classifier at all [82].

2.2.2. ERROR REDUCTION

Error reduction approaches are another type of popular active learning methods [40, 75, 107, 124]. These approaches attempt to measure how much the generalization error is likely to be reduced when adding one new instance into the labeled dataset. Though one does not have direct access to the future test data, Roy and Mccallum [40] proposed to estimate the expected error rate over the unlabeled examples under the assumption that the unlabeled data is representative of the test data. In other words, the unlabeled pool can be viewed as a validation set. Roy and Mccallum proposed to estimate the expected error using expected log-loss or 0-1 loss. For the former, which we consider in our work, the following objective is considered:

$$x^* = \arg\min_{x \in \mathscr{U}} \sum_{y \in C} P_{\mathscr{L}}(y|x) \left(- \sum_{x_i \in \mathscr{U}} \sum_{y_i \in C} P_{\mathscr{L}^+}(y_i|x_i) \log P_{\mathscr{L}^+}(y_i|x_i) \right) \tag{2.3}$$

Here, $\mathscr{L}^+ = \mathscr{L} \cup (x, y)$ indicates that the selected instance x is labeled y and added to the labeled dataset $\mathscr{L}$. We refer to this method as Expected Error Reduction (EER) in this chapter. The first term $P_{\mathscr{L}}(y|x)$ is the posterior probability of y given x trained on the labeled dataset $\mathscr{L}$.

However, since using the labeled data $\mathscr{L}$, which is typically of small size, can result in a bad classifier, $P_{\mathscr{L}}(y|x)$ may not be estimated very adequately [75]. To avoid problems with such misspecifications, Guo and Greiner [75] proposed an optimistic or, equivalently, *maximum* error reduction approach (called MAXER in this chapter), which estimates the best-case error reduction, without considering $P_{\mathscr{L}}(y|x)$. MAXER considers the following objective instead:

$$x^* = \arg\min_{x \in \mathscr{U}} \min_{y \in C} \left(- \sum_{x_i \in \mathscr{U}} \sum_{y_i \in C} P_{\mathscr{L}^+}(y_i|x_i) \log P_{\mathscr{L}^+}(y_i|x_i) \right) \tag{2.4}$$

Note that the error reduction approaches above only take the unlabeled data into consideration when estimating the future error. To obtain better generalization performance, it has been suggested to compute the loss both over the training set $\mathscr{L}$ and over the unlabeled set $\mathscr{U}$. This idea was proposed in [136] for semi-supervised learning, while Guo and Schuurmans [107] extended it to the batch mode active learning. Focusing on the myopic setting, one can adopt the related criterion as follows:

$$x^* = \arg\min_{x \in \mathscr{U}} \min_{y \in C} \left(- \sum_{x_j \in \mathscr{L}^+} \log P_{\mathscr{L}^+}(y_j|x_j) - \alpha \sum_{x_i \in \mathscr{U} \setminus x} \sum_{y_i \in C} P_{\mathscr{L}^+}(y_i|x_i) \log P_{\mathscr{L}^+}(y_i|x_i) \right) \tag{2.5}$$

where α is a trade-off parameter used to adjust the importance of loss over labeled and unlabeled data. We name this combined approach CEER in this chapter.

One general, potential disadvantage of error reduction approaches is the high computational cost [36]. Each time a new queried instance x with its label y is added to the training dataset, we need to retrain the classifier to get the new posterior probabilities $P_{\mathscr{L}^+}(y_i|x_i)$. This retraining step may amount to great computational efforts.

2.2.3. VARIANCE REDUCTION

Optimal experimental design, which attempts to minimize particular statistical criteria with the aim of saving in sampling cost, is an approach that has been classically used in the design of linear regression experiments [52, 137, 138]. A-optimality, which is one of the classic, commonly used measures, is the trace of the inverse of the information matrix [137]. Minimizing A-optimality can also be seen as reducing the average variance of the estimates of model parameters and therefore is wildly practised in active learning [13, 76, 111].

In the binary classification setting, regarding regularized logistic regression, the Fisher information matrix over the unlabeled pool $\mathscr{U}$ is defined as $\mathscr{I}_{\mathscr{U}}(w) = \frac{1}{|\mathscr{U}|}\sum_{i\in\mathscr{U}}\sigma_i(1-\sigma_i)x_i x_i^T + \lambda I_d$ where $\sigma_i = \sigma(w^T x_i) = 1/(1+\exp(-w^T x_i))$ is the posterior probability of $P(y=1|x_i)$, and I_d is the identity matrix of size $d\times d$. Zhang and Oles [111] utilized A-optimal design to minimize the Fisher information ratio between $\mathscr{I}_{\mathscr{U}}(\hat{w})$ and $\mathscr{I}_x(\hat{w})$:

$$x^* = \arg\min_{x\in\mathscr{U}} \mathrm{tr}(\mathscr{I}_x(\hat{w})^{-1}\mathscr{I}_{\mathscr{U}}(\hat{w})) \tag{2.6}$$

where $\mathscr{I}_x(\hat{w}) = \sigma_i(1-\sigma_i)x_i x_i^T + \lambda I_d$ and $\hat{w}$ is the maximum likelihood estimator. The entity $\mathscr{I}_{\mathscr{U}}(\hat{w})$ can be interpreted as the variance of model output with respect to unlabeled data $\mathscr{U}$, and $\mathscr{I}_x(\hat{w})^{-1}\mathscr{I}_{\mathscr{U}}(\hat{w})$ can be viewed as the future output variance once x has been labeled. The criterion suggested selects unlabeled examples that minimize the Fisher information ratio or, equivalently, reduce the future variance. We call this approach Fisher information variance reduction (FIVR) in this chapter. Hoi *et al.* [13] exploited the same idea as in [111] and extended it to the batch mode setting. When the batch size is set to one, Hoi's method is identical to FIVR apart from some approximations introduced for dealing with the batch setting.

Schein and Ungar [76] proposed a similar A-optimal active learning method based on logistic regression. In doing so, one can define the Fisher information matrix over the training data $\mathscr{L}$ as $F = \frac{1}{l}\sum_{i\in\mathscr{L}}\sigma_i(1-\sigma_i)x_i x_i^T + \lambda I_d$. Schein proposed to minimize the variance of the estimated distribution of the estimator $\sigma(\hat{w}^T x_i)$ as follows:

$$\mathrm{Var}(\sigma(\hat{w}^T x_i)) \simeq c_i^T F^{-1} c_i$$

where $c_i = \sigma_i(1-\sigma_i)x_i$ is the gradient vector of σ_i. The variance over all the unlabeled instances can be formulated as follows:

$$g(\mathscr{L},\mathscr{U}) = \sum_{x_i\in\mathscr{U}}\sum_{y\in\{+1,-1\}}\mathrm{Var}(\sigma(y\hat{w}^T x_i)) \simeq 2\sum_{x_i\in\mathscr{U}} tr\{c_i^T F^{-1} c_i\}$$

The benefit of a newly selected instance, can then be measured in terms of the ex-

pected variance reduction:

$$x^* = \arg\min_{x \in \mathcal{U}} \sum_{y \in C} P_{\mathcal{L}}(y|x) g(\mathcal{L} \cup (x,y), \mathcal{U}) \tag{2.7}$$

We refer to this method as Expected Variance Reduction (EVR) in this chapter. EVR represents the potential variance changes weighted by current estimated model $P_{\mathcal{L}}(y|x)$. EVR can also be extended to log-loss based EVR [76], but we do not consider this algorithm any further since we observed that it generally performs poorer than EVR in our experiments.

EVR is similar to EER in some respects. First, see Equation 2.3 and 2.7, we can find that both EER and EVR measure the utility of an unlabeled instance x by repeatedly labeling it y (i.e. $y \in \{+1,-1\}$) and retraining the model on $\mathcal{L} \cup (x,y)$. Second, both of them calculate the expectation value, e.g. EER evaluates the expected future error while EVR computes the expected future variance.

EVR is also computationally expensive since it goes over all the pool and re-estimates $\hat{w}$ and Fisher information matrix F each time. The need to calculate the inverse of matrix typically makes it even slower than expected error reduction approaches.

2.2.4. MINIMUM LOSS INCREASE

The next heuristic we consider is minimum loss increase (MLI), which directly bases its criterion on already labeled samples. Related to this class, Hoi *et al.* [80] originally proposed a min-max view of active learning that minimizes the gain of the objective function. We here look at the work of Hoi *et al.* [80] in a more general formulation and demonstrate its relationship with the expected error reduction framework.

Let us consider an unconstrained optimization problem using an L2-loss regularized linear classifier and a loss function $V(w; x_i, y_i)$:

$$\min_{w} \qquad g(w) = \frac{\lambda}{2}||w||^2 + \sum_{i=1}^{l} V(w; x_i, y_i) \tag{2.8}$$

where $y_i \in \{+1,-1\}$. Many loss functions can be adopted for linear classification. For example, hinge loss, $V(w; x_i, y_i) = \max(0, 1 - y_i w^T x_i)$, results in linear SVM and squared loss, $V(w; x_i, y_i) = (y_i - w^T x_i)^2$, leads to ridge regression. We will consider the logistic loss in the experimental section: $V(w; x_i, y_i) = \log(1 + \exp^{-y_i w^T x_i})$, which results in L2-regularized logistic regression.

Now, to identify the most valuable instances for labeling, we could select the example that, once labeled, results in the minimum gain in terms of the score of objective function. That is, we consider

$$x^* = \arg\min_{x \in \mathcal{U}} \max_{y \in C} g_{\mathcal{L}^+}(w) - g_{\mathcal{L}}(w) \tag{2.9}$$

where $\mathcal{L}^+ = \mathcal{L} \cup (x,y)$ and $g_{\mathcal{L}}(w)$ denotes the value of objective function over the training data $\mathcal{L}$. Since $g_{\mathcal{L}}(w)$ is independent of the next queried instance, we can rewrite Equation 2.9 as follows:

$$x^* = \arg\min_{x \in \mathcal{U}} \max_{y \in C} \min_{w} \frac{\lambda}{2}||w||^2 + \sum_{x_i \in \mathcal{L}^+} V(w; x_i, y_i) \tag{2.10}$$

This method can be interpreted as directly minimizing the worst-case value of the objective function when labeling a new instance. Considering kernel versions instead of linear classifiers in the above, they would entail the earlier mentioned min-max active learning methods [80, 82], which use the hinge loss and square loss, respectively. Hoi *et al.* [80] originally presented the min-max view method and extended it to the batch mode active learning. Huang *et al.* [82] extended the min-max view to consider all the unlabeled data and proposed an active learning method which QUeries Informative and Representative Examples (QUIRE for short) as follows:

$$x^* = \arg\min_{x\in\mathscr{U}} \min_{y_u\in C^{\{n_u-1\}}} \max_{y\in C} \min_{w} \frac{\lambda}{2}||w||^2 + \sum_{x_i\in\mathscr{L}\cup\mathscr{U}} V(w;x_i,y_i) \tag{2.11}$$

where y_u indicates the labels of remaining unlabeled pool $\mathscr{U}\setminus x$ and n_u is the number of samples of U. We need to point out, however, that the unlabeled part $\mathscr{U}\setminus x$ is of no use since QUIRE relaxed the constraints of y_u. Optimizing this unconstrained y_u can guarantee that the remaining unlabeled data $\mathscr{U}\setminus x$ is useless, which can also be observed from Equation (9) and (10) in the original work [82]. Therefore, QUIRE also fits this general framework.

As we consider the logistic loss for the above framework, MLI will refer to this particular choice. The particular objective function we consider is as follows:

$$\begin{aligned} x^* &= \arg\min_{x\in\mathscr{U}} \max_{y\in C} \min_{w} \frac{\lambda}{2}||w||^2 + \sum_{x_i\in\mathscr{L}^+} V(w;x_i,y_i) \\ &= \arg\min_{x\in\mathscr{U}} \max_{y\in C} \min_{w} \frac{\lambda}{2}||w||^2 + \sum_{x_i\in\mathscr{L}^+} \log(1+\exp^{-y_i w^T x_i}) \\ &= \arg\min_{x\in\mathscr{U}} \max_{y\in C} \frac{\lambda}{2}||\hat{w}||^2 + \sum_{x_i\in\mathscr{L}^+} -\log P_{\mathscr{L}^+}(y_i|x_i) \end{aligned} \tag{2.12}$$

where $\hat{w}$ is the estimated parameter of the L2-regularized logistic regression model trained on the labeled data $\mathscr{L}^+ = \mathscr{L}\cup(x,y)$ and $P_{\mathscr{L}^+}(y_i|x_i) = 1/(1+\exp^{-y_i\hat{w}^T x_i})$. Comparing Eqs. (2.5) and (2.12), we find that MLI differs from CEER in two respects: (1) MLI adopts the min-max criterion while CEER considers the best optimistic scenario (i.e. the smallest loss); (2) MLI only measures the log-likelihood on labeled data while CEER also takes the unlabeled data into account.

2.2.5. MAXIMUM MODEL CHANGE

Maximum mode change (MMC) is another strategy for active learning [45, 77–79, 126]. These approaches query the sample which can lead to a great change of the current model once labeled. The differences among these approaches lies in the criteria to measure the model change. Settles *et al.* [77] proposed to measure the expected gradient length of the objective function. Freytag *et al.* [79] estimated the change of model outputs instead of model parameters.

Cai *et al.* [45] proposed to use the gradient of the loss function to approximate the model change and derived algorithms for both SVM and logistic regression classifier. We briefly review this method built on logistic regression [45]. Assumed that the loss of

logistic regression after adding a new sample (x, y) is

$$L(w) = -\sum_{i \in \mathscr{L}^+} \log(1 + \exp^{-y_i w^T x_i})$$

where $\mathscr{L}^+ = \mathscr{L} \cup (x, y)$ and w is the parameter of logistic regression model. MMC approximates the model change as follows:

$$\frac{\partial L(w)}{\partial w} \approx \frac{\partial \log(1 + \exp^{-yw^T x})}{\partial w} = \frac{yx}{1 + \exp^{-yw^T x}}$$

Since the label y is unknown, MMC calculates the expected model change

$$\mathscr{M}(x) = \mathbb{E}_y \left\| \frac{yx}{1 + \exp^{-yw^T x}} \right\| = \frac{2\|x\|}{(1 + \exp^{-w^T x})(1 + \exp^{w^T x})} \tag{2.13}$$

Finally, MMC selects the sample x^* that leads to the largest mode change as follows:

$$x^* = \arg\max_{x \in \mathscr{U}} \mathscr{M}(x) \tag{2.14}$$

Note that $\frac{1}{(1+\exp^{-w^T x})(1+\exp^{w^T x})}$ corresponds to $P(+1|x) \times P(-1|x)$. This value will be maximum when $P(+1|x) = 0.5$, which means that MMC prefers the sample with high uncertainty. In addition, MMC is also likely to query the instance with large norm. Therefore, MMC trades off the uncertainty and the norm of a sample.

2.2.6. ADAPTIVE ACTIVE LEARNING

Li and Guo [11] proposed an active learning approach which combines uncertainty sampling and information density measure in an adaptive way. We call this method Adaptive Active Learning (AAL). We should consider the instances which are located in a dense region for two reasons. One is that they are less likely to be the outliers. And secondly, they can represent the underlying distribution. By combining the uncertainty and information density measure, their proposed method can balance the informativeness and representativeness. There are some active learning methods that share a similar idea [25, 61, 84, 101].

First, AAL trains a logistic regression classifier and uses the entropy as a measure of uncertainty, which is equivalent to the ENTROPY approach in Subsection 2.2.1. Then, AAL measures the information density by employing a Gaussian Process framework to calculate the mutual information between the candidate instance and the unlabeled pool. Finally, it combines the two criteria using a trade-off parameter β ($0 \le \beta \le 1$):

$$h_\beta(x_i) = u(x_i)^\beta \times d(x_i)^{1-\beta} \tag{2.15}$$

where $u(x_i)$ and $d(x_i)$ are the uncertainty and density values of $x_i \in \mathscr{U}$, respectively.

It is difficult, however, to set a proper weighting parameter β. Instead of using a predefined value of β, Li and Guo [11] proposed to adaptively choose the β value from a given set $[0.1, 0.2, \ldots, 0.9, 1]$. Each different β leads to picking a candidate instance from unlabeled samples. Among these candidates, AAL chooses the sample which has minimal expected classification error according to expected error reduction method [40]. In other words, AAL adaptively changes the β value to form a candidate set, from which the most informative sample is selected by using EER.

2.3. EXPERIMENTS

The experimental setup is first described, followed by an analysis of the results on synthetic datasets and real-world datasets, respectively. Finally, we investigate the computational costs of different active learning algorithms.

2.3.1. EXPERIMENTAL SETTING

We present the necessary information of three synthetic datasets and 44 real-world datasets that we used in the following subsections, along with a description of the evaluation design.

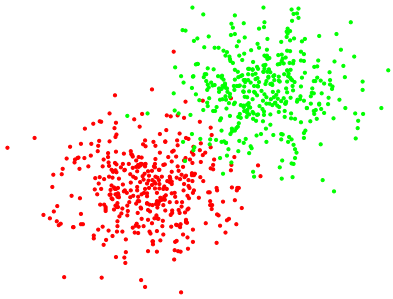
SYNTHETIC DATA SETS

Three binary synthetic datasets are constructed to intuitively demonstrate the different behaviors of active learning algorithms. The first dataset *Synth1* is a standard *2D* binary problem which is shown in Fig 2.2a. Positive and negative classes are generated according to two multivariate normal distributions centered at $[1,1]^T$ and $[-1,-1]^T$, respectively. We want to explore which active learning method works well on this unambiguously specified problem. The second dataset, *Synth2*, displayed in Fig 2.2c, is generated according to the description in [82]. We can observe that Synth2 has a clear cluster structure. On this kind of data, uncertainty sampling has substantial problems since it only considers the most *uncertain* instance which comes closest to the decision boundary. Initially, the decision boundary estimated from the limited number of labeled data may be far away from the actual boundary and therefore uncertainty sampling may select less informative instances due to a poorly estimated posterior probability. This is exactly what this dataset was designed for and set out to illustrate. This dataset prefers some kind of active learning methods which can consider the so-called representativeness along with the informativeness at the same time [82]. Representative instances are those that drive exploration, and not exploitation. The latter is what uncertainty sampling typically aims for. The third synthetic dataset, named *Synth3*, is also a 2D classification problem which is shown in Fig 2.2e. Synth3 is constructed to have a shape which looks like a tilted ⊔. Each part is generated from two multivariate normal distributions with small overlap. Compared with Synth1, Synth3 is a more challenging dataset since it has relatively complex structure and may mislead some active learning methods. We are curious whether active learning can outperform random sampling on this kind of data. We investigate how active learning approaches work in the above three synthetic datasets and whether they perform better than random sampling.

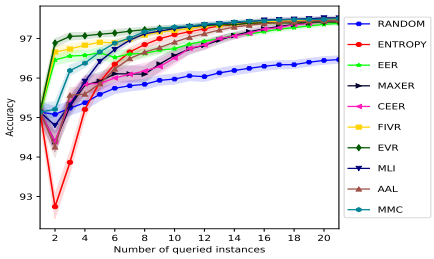
REAL-WORLD DATA SETS

As real-world benchmarks, we use various UCI datasets [139], the MNIST handwritten digit dataset [140], the 20 Newsgroups dataset [141] and the 80 subsets of the ImageNet database [6]. Table 2.1 lists the preprocessed datasets used in our study together with some basic information. All the datasets are pre-processed to become binary classification problems.

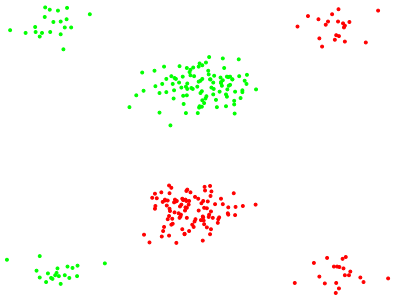
There are a total of 44 benchmark datasets used in this comparison, including the ImageNet dataset on which extensive experiments on 80 binary subsets are conducted.



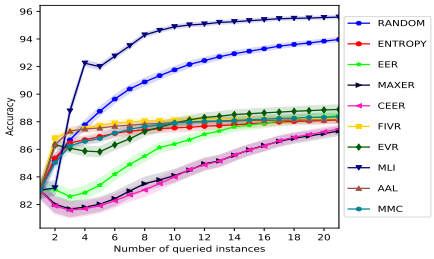
(a) Synth1



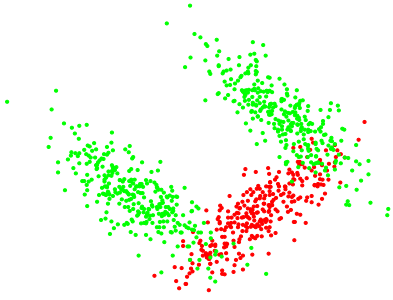
(b) Accuracy on Synth1



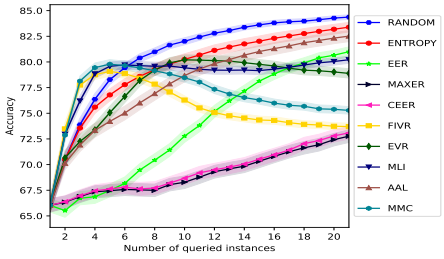
(c) Synth2



(d) Accuracy on Synth2



(e) Synth3



(f) Accuracy on Synth3

Figure 2.2: Distribution for three synthetic datasets and the results on these same sets in terms of classification accuracy with 90% confidence interval. Red and green points represent the two different classes. (a) shows the distribution of the Synth1 dataset; (b) presents the average accuracy of each active learning method on the test set for Synth1. (c)-(d) and (e)-(f) represent the same results for Synth2 and Synth3, respectively.

Most datasets are pre-processed to have zero mean and unity standard deviation according to [142]. Some datasets are linearly scaled to $[-1,1]$ or $[0,1]$ according to [143] [2]. These datasets have various sample sizes and diverse feature dimensionalities. Some of them can be quite easily handled while others are quite difficult classification problems. The Letter Recognition Data Set from UCI, which consists of 20,000 examples of 26 uppercase letters in various fonts and distortions, is also used as a test bed in [144]. As in this last work, 16 attributes are extracted from the letters as the feature and we consider the following six classification tasks between pairs of letters: D vs. P, E vs. F, I vs. J , M vs. N, V vs. Y, and U vs. V. These pairs of letters are selected since they have a somewhat similar appearance and distinguishing them is challenging.

The MNIST [3] contains 60,000 training examples and 10,000 test examples which have been pre-processed to the same size of 28×28 pixels. The pairs 3 vs. 5, 5 vs. 8, and 7 vs. 9 constitute three difficult classification tasks and are used as the binary sets in our benchmark. For each of the three pairs, we randomly subsample 1500 instances from the original dataset for computational reasons. Each pixel value is extracted as a feature, resulting in a 784-dimensional feature.

The 20 Newsgroups is a common benchmark used for document classification [4]. We use one version of this dataset which consists of 18,846 instances of 20 different news topics. Similar to the work of [145], our work also evaluates three binary tasks from this dataset: *sport.baseball* vs. *sport.hockey*, *pc.hardware* vs. *mac.hardware*, and *talk.religion.misc* vs. *alt.atheism*. All the documents have been pre-processed into 26,241 dimensional tf.idf vectors to which we initially apply PCA to reduce the dimensionality to 500 for computational reasons.

In addition, we also compare these active learning algorithms on a total of 80 binary subsets taken from the large visual ImageNet database [6]. First, following the work of [45], we take 8 different subsets of ImageNet: five categories of cats (i.e. Egyptian, Persian, Siamese, Tabby and Tiger) and elephant, rabbit and panda. Subsequently, we construct eight binary-class classification problems by considering cat vs. elephant, cat vs. rabbit, cat vs. panda and each category of cat vs. the four remaining cats. Moreover, we also randomly chose 72 paired classes to generate 72 binary data sets from the ImageNet database provided by Tommasi and Tuytelaars [146]. SIFT features are first extracted and then encoded into 1000-bin histograms. Detailed information of the 80 subsets of the ImageNet dataset is included in Table 2.4.

EVALUATION DESIGN

In the evaluation, each dataset is randomly divided into training and test data sets of equal size. Following some previous work [23, 24, 108, 112, 145, 147], we consider a difficult case of active learning, where only two labeled instances are provided as the initial labeled set, one from each class. We repeat each experiment 20 times on each real-world dataset. As for the synthetic datasets, we repeat the experiments 1000 times and every time we randomly regenerate the whole dataset. The average performance of each active learning method on each dataset is reported. In all the experiments, regularized logistic

[2] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[3] http://yann.lecun.com/exdb/mnist/
[4] http://qwone.com/ jason/20Newsgroups/

regression included in LIBLINEAR package [148] is used as the classifier. We set the regularization parameter λ to 0.01. The trade-off parameter present in the active learners considered, α for CEER, is set to 1.

For performance comparison, classification accuracy (or equivalently, the error rate) is the defacto standard evaluation criterion: the higher the accuracy, the better the algorithm. In active learning, however, performance varies depending on the number of labeled samples that one is allowed to take and we cannot settle on a single number of added labeled samples. For this reason, we use the area under the learning curve (ALC) [149] as the evaluation criterion. The larger this value, the better the performance. The optimal score is 1.

Table 2.1: Data sets information: It shows the number of instances (# INS) and the feature dimensionality (# FEA)

Data set	# Ins	# Fea	Data set	# Ins	# Fea
ac-inflam	120	6	acute	120	6
australian	690	14	blood	748	4
breast	683	10	credit	690	15
cylinder	512	35	diabetes	768	8
fertility	100	9	german	1000	24
haberman	306	3	heart	270	13
hepatitis	155	19	hill	606	100
ionosphere	351	34	liver	345	6
mushrooms	1000	112	mammographic	961	5
musk1	476	166	ooctris2f	912	25
ozone	1000	72	parkinsons	195	22
pima	768	8	planning	182	12
sonar	208	60	splice	1000	60
tictactoe	958	9	vc2	310	6
vehicle	435	18	wisc	699	9
wdbc	569	31	letterDP	1608	16
letterEF	1543	16	letterIJ	1502	16
letterMN	1575	16	letterVY	1577	16
letterUV	1550	16	3 vs 5	1500	784
5 vs 8	1500	784	7 vs 9	1500	784
baseball vs hockey	1993	500	pc vs mac	1945	500
misc vs atheism	1427	500	subsets of ImageNet	180,821	1000

2.3.2. ANALYSIS ON SYNTHETIC DATASETS

In Fig 2.2, we display the distributions of the three synthetic datasets, along with the performance of each active learning method in terms of the classification accuracy on the test set. We also present the 90% confidence interval around each learning curve. To start with, note that no single method outperforms all the other methods on all the datasets.

PREFERENCE MAP

To generally show a difference in characteristic of the various active learning methods, we introduce a visualization technique, called Preference Map, for our synthetic datasets (see Fig 2.3 and 2.4).

The preference map is generated by keeping track of the locations of the queried instances selected by each active learning algorithm. Presenting kernel density plots of all these locations and displaying them using pseudo-colors gives an impression where in feature space the active learners request their data from. The highest density regions are marked in red while the lowest density regions are indicated in blue. The preference map of the instance first queried is shown in Fig 2.3a. More specifically, for our 2D synthetic datasets, we record the location of the first queried sample selected by each active learner during 1000 repetitions of the experiment and generate the density plots.

We also plot the preference maps corresponding to the complete learning, where we exponentially weigh down later observations based on the intuition that the examples selected early on in the process are more valuable than the examples selected in the later rounds. The specific weight function we employ is $\exp(-r/R)$, where r and R are the current round and the total rounds, respectively. In other words, we make a record of the locations of all queried samples during the whole active learning process, followed by producing weighted preference maps. The corresponding preference maps are in Fig 2.3b.

RESULT ON SYNTH1

Synth1 is a simple classification problem and some algorithms perform well in the beginning stage, such as the variance reduction approaches FIVR and EVR. On the other hand, ENTROPY achieves rather poor performance at the beginning and is the worst approach at the first selected point in Fig 2.2b. To understand this specific aspect of how uncertainty sampling behaves, we refer to the preference map in Fig 2.3a. We can see that random sampling prefers the region where the mean vector of each class is. Clearly, the preference map for random sampling should ultimately reproduce the original underlying distribution, which is a mixture of two normal distributions in the setting we consider. Uncertainty sampling clearly prefers to query the points in the middle of two clusters since it focuses on the instances near the estimated decision boundary. Even though these samples may be close to the true decision boundary, they may not be a good choice, as they lead to instable estimates. This is what we see in the results, where ENTROPY performs rather poor in the beginning stage. CEER and MAXER show similar behaviors in the preference map and also seem to give relatively worse performance at the start of the active learning cycle. Their maps, however, seem a bit more rectangular, which may lead to slightly improved stability and therefore better performance as compared to ENTROPY. Variance reduction methods like EVR and FIVR also sample parallel to the decision boundary, but more through the respective class centers, which indeed leads to more stable and therefore better performing estimates. MLI, on the other hand, seems to sample perpendicular to the decision boundary, away from the regions with high density. This may be because MLI, which is similar to QUIER [82], tends to balance the informativeness and representativeness. When only two initial points are available, MLI prefers to select the instance far away from already labeled ones. AAL queries the

first instance from a broad region since it is able to explore a large region by adaptively changing the trade-off parameter. MMC performs similarly to MLI. The reason may be that MMC balances uncertainty and the norm of unlabled instance. MMC prefers the samples with large norm and high uncertainty.

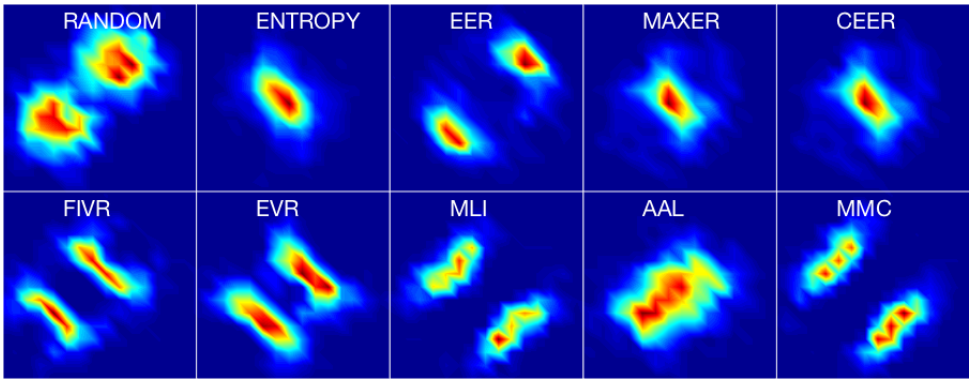
Turning our attention to the overall weighted preference maps Fig 2.3b, we see a dramatic change in behavior for at least six strategies. FIVR and especially EVR change their sampling from parallel to more perpendicular to the decision boundary. The changes we see for EER and MLI may be interpreted as changes from the more explorative initial phase to a more exploitative later stage, where a sampling around the decision boundary is performed to refine it. That active learning should actually deal with the exploration-exploitation tradeoff is at the basis of MLI. AAL also changes from the initial exploration to the subsequent exploitation. MMC seems to attach more importance to the uncertainty than the norm of sample. In addition, we observe that some active learning approaches, of which overall preference maps are similar to each other, performs similarly to each other in the later stage (e.g. after 8 samples are queried). For example, FIVR and MMC have similar maps while their performances are almost identical when 8 instances are labeled.

RESULT ON SYNTH2

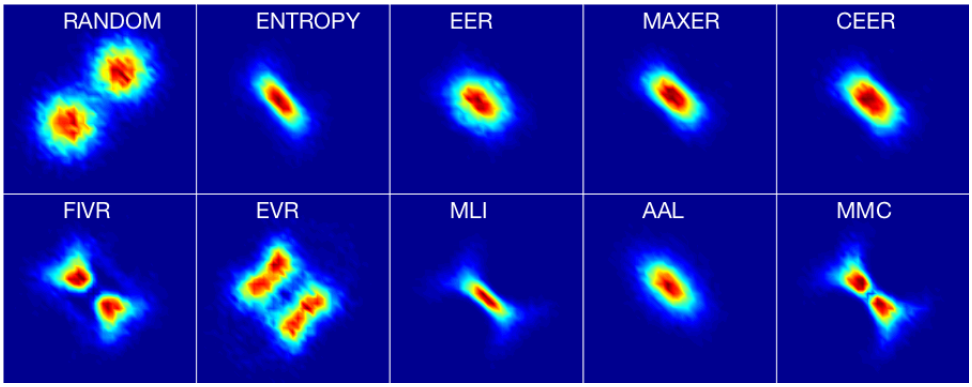
Fig 2.2d shows that on the second synthetic problem, random sampling far surpasses all the active learning methods except for MLI, which is the best performing strategy. We use the overall preference map to explain this result. Fig 2.4 displays preference maps corresponding to the whole learning curve on Synth2 dataset.

We can see that the preference map of random sampling reproduces the underlying distribution. The preference maps of the remaining active learning methods except MLI, which are almost identical to each other, only highlight the two large clusters in the middle. This indicates that most of the queried samples are from the two middle clusters. This happens because that these active learning methods are misled by an incorrect model estimated with limited initial training samples. For instance, assume that we have two initial labeled points separately located in the two middle clusters. This initial training data will lead to a completely wrong estimation of the decision boundary. Then, these active learners will keep selecting the points which come close to the wrong decision boundary. However, these selected points cannot provide much more information about the true underlying distribution. Finally, they miss a chance of selecting the samples from other small clusters to discover the underlying distribution. This shows a common situation where some active learning methods get stuck in keeping querying useless instances due to inaccurate estimation of model parameter. Random sampling does not suffer from this because it acts purely random in selecting new instances. This is why random sampling surpasses these active learning methods.

MLI can perform even better in this situation. MLI can select the samples in the upper left corner and lower right corner on Synth2 dataset since it also considers the so-called representativeness of each instance, such as whether the instances are inside of some clusters [82]. This leads to the exploratory behavior of MLI. As shown in Fig 2.4, MLI is more likely to query the instances along four clusters on the border line while some methods like uncertainty sampling and error reduction approaches favor the in-



(a)



(b)

Figure 2.3: (a) Preference maps of first queried example selected by each active learning method on the Synth1 dataset; (b) Weighted preference maps over the whole learning process on the same problem.

stances in the two middle clusters. This is the reason that MLI can significantly outperform random sampling and other active learning methods on this artificial set.

RESULT ON SYNTH3

From Fig 2.2f, we can observe some negative results that random sampling outperforms all the other active learning methods after 6 instances are selected. The possible reason is that random sampling can explore the whole structure of this dataset while other methods just pay attention to some local parts without exploring the whole dataset. And another reason may be that it is difficult to achieve good classification result on this dataset due to its complex structure. On this kind of hard datasets, active learning methods can easily get stuck in local structure while ignoring the global view of the problem. Due to space limitations, the preference maps of Synth3 are omitted.

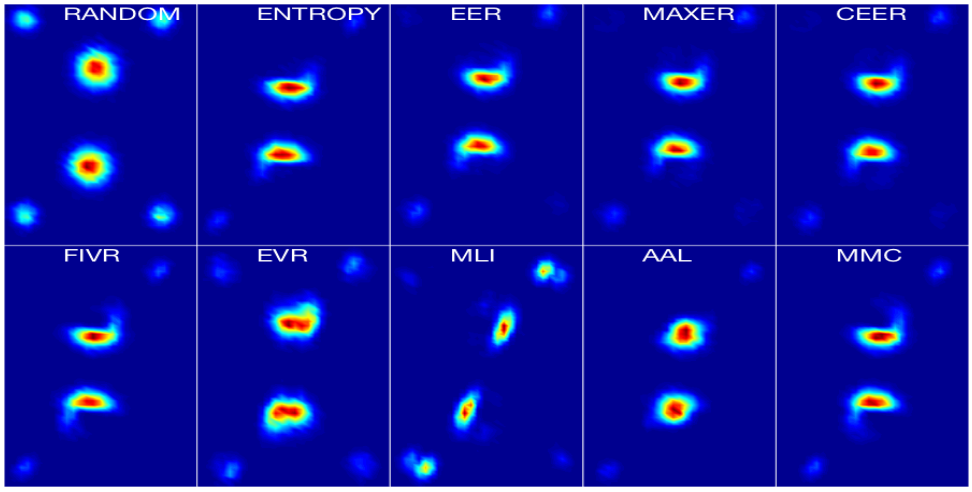


Figure 2.4: Weighted preference maps for the overall learning procedure by each of the active learning methods on the Synth2 dataset.

2.3.3. ANALYSIS ON REAL-WORLD DATASETS

Table 2.2 presents the results for applying each active learning method on the real-world datasets. We adopt the paired t-test at a 95% significance level on all the experiments to test which method does not significantly differ from the best method. The best performance is highlighted in bold face and surrounded with a box, together with the competitors that perform at a comparable level. The average ALC ("Mean" in Table 2.2) of each method is also reported. "Average Ranking" shows the average ranking of compared methods. "Win counts" shows the total number of datasets on which one method achieves the best performances. "win/tie/loss" demonstrates the win, tie, and loss counts of one method versus random sampling over all of the datasets.

As shown in Table 2.2, no single algorithm outperforms all others on all the datasets. Still ENTROPY and EER seem to markedly outperform other active learning methods. It may be surprising that uncertainty sampling can compete with relatively sophisticated active learning algorithms as it is a rather simplistic approach. In fact, uncertainty sampling gets the highest ALC score and performs best in terms of win/tie/loss counts versus random sampling. It also obtains the best average ranking. MLI and MMC behave the second best among the remaining methods in terms of win counts while their average ALC and average ranking are outperformed by uncertainty sampling.

Considering the error reduction approaches, it is clear that EER outperforms MAXER and CEER. The overall performances of MAXER and CEER remain close to that of random sampling. MAXER merely surpasses random sampling on 20 of the 44 datasets. This seems to demonstrate that the best-case criterion is not an appropriate choice for active learning, at least for error reduction approaches. The possible cause may be that such optimistic measure simply puts too much trust in a typically badly estimated model. As a result, initial errors may get reinforced rather than mitigated by correctly chosen additional samples. This is comparable to some of the issues that arise in self-learning

Table 2.2: Performance comparison on the Area under the Learning Curve (Accuracy). The higher the score, the better the performance. For each data set, the best performances and its comparable competitors are highlighted in bold face and surrounded with a box. Average performance of all the active learning methods are also reported as "Mean". "Average Ranking" shows the average ranking of compared methods. "Win counts" shows the total number of datasets on which one method achieves the best performances. "win/tie/loss" demonstrates the win/tie/loss counts of one method versus random sampling on all the datasets based on paired t-test at 95 percent significance level.

Dataset	Random	ENTROPY	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
hill	0.581	0.593	0.616	0.606	0.575	0.549	0.590	**0.626**	0.544	0.570
planning	0.586	0.584	0.580	0.568	0.574	0.574	0.587	**0.614**	0.575	0.581
cylinder	0.586	0.593	0.610	0.575	0.576	**0.630**	0.591	0.608	0.590	0.607
liver	0.627	0.612	0.635	0.623	0.621	**0.645**	0.632	0.615	0.616	0.631
splice	0.659	**0.685**	0.679	0.666	0.663	0.676	0.664	0.650	0.672	0.650
german	0.664	0.694	0.673	0.652	0.654	0.687	0.678	0.691	0.679	**0.707**
ooctris2f	0.679	0.665	0.678	0.648	0.652	0.651	0.680	**0.686**	0.669	0.646
musk1	0.682	**0.702**	**0.699**	0.672	0.668	0.679	0.689	**0.702**	0.684	0.675
fertility	0.693	0.701	0.706	0.679	0.674	0.716	0.686	**0.727**	0.699	0.705
haberman	**0.711**	**0.712**	**0.712**	0.704	0.704	0.692	0.708	0.694	**0.710**	0.691
sonar	0.713	0.711	0.715	0.713	0.715	0.720	0.720	0.708	**0.723**	0.718
pima	0.716	0.717	0.706	0.710	0.707	**0.727**	0.708	0.711	0.709	0.700
pcmac	0.717	0.727	0.715	0.698	0.696	0.717	0.711	0.747	0.713	**0.776**
diabetes	0.719	0.721	0.723	0.726	0.724	0.707	0.725	0.726	0.709	**0.736**
religionatheism	0.720	**0.740**	0.710	0.687	0.687	0.712	0.704	0.691	0.702	0.720
hepatitis	0.731	**0.753**	0.753	0.744	0.741	0.745	**0.754**	0.730	**0.757**	0.708
blood	**0.743**	0.718	0.740	0.732	0.736	0.723	0.732	0.730	0.728	0.728
heart	0.774	0.793	0.791	0.788	0.788	**0.799**	0.781	0.797	0.784	0.787
ImageNet	0.778	**0.783**	0.763	0.761	0.760	0.775	0.765	0.762	0.761	0.774
ionosphere	0.779	0.782	**0.818**	0.806	0.801	0.790	0.812	0.674	0.812	0.768
credit	0.779	**0.822**	0.793	0.795	0.804	**0.819**	0.791	0.797	0.758	0.780
mammographic	0.780	0.779	0.774	0.781	0.784	**0.795**	0.777	0.766	0.775	0.775
basehockey	0.793	0.822	0.784	0.772	0.770	0.820	0.783	0.817	0.768	**0.857**
vc2	0.807	0.814	0.815	0.802	0.803	0.816	0.822	**0.825**	0.796	0.823
parkinsons	0.811	0.823	0.824	0.824	0.828	0.825	0.825	0.830	0.803	**0.838**
australian	0.823	**0.844**	0.832	0.839	0.838	0.817	0.831	0.842	0.829	0.828
letterIJ	0.853	0.871	0.879	0.807	0.806	0.841	0.869	0.865	0.865	**0.889**
letterVY	0.855	0.880	0.878	0.814	0.814	0.753	**0.886**	0.861	0.876	0.830
3vs5	0.856	0.884	**0.903**	0.890	0.889	0.869	**0.903**	0.859	0.894	0.884
vehicle	0.859	**0.884**	0.878	0.851	0.855	0.830	**0.884**	**0.883**	0.837	0.847
5vs8	0.864	0.891	**0.907**	0.896	0.895	0.875	**0.909**	0.850	0.899	0.891
7vs9	0.876	0.904	0.914	0.905	0.906	0.909	**0.917**	0.841	0.912	0.904
ozone	0.882	0.884	0.860	0.862	0.861	0.843	**0.901**	0.892	0.863	0.868
tictactoe	0.894	0.902	**0.912**	0.673	0.684	0.903	0.898	0.853	0.902	0.843
letterMN	0.916	**0.939**	**0.944**	0.910	0.910	0.932	0.941	0.927	0.930	0.935
mushrooms	0.931	**0.971**	0.969	0.967	0.967	**0.972**	0.960	**0.971**	0.968	**0.971**
letterEF	0.933	**0.959**	0.954	0.949	0.950	0.954	0.956	0.956	0.952	**0.957**
wdbc	0.938	0.955	0.953	0.955	0.954	0.943	0.951	**0.958**	0.948	0.954
letterDP	0.939	0.964	0.963	0.950	0.950	0.954	0.961	**0.967**	0.956	**0.967**
letterUV	0.945	0.970	**0.972**	0.955	0.955	0.963	0.966	**0.974**	0.963	**0.975**
wisc	0.949	0.956	0.951	**0.958**	0.957	0.954	0.953	0.956	0.956	0.956
breast	0.950	0.958	0.956	0.956	0.957	0.960	0.955	**0.962**	0.947	**0.962**
ac-inflam	0.955	**0.985**	0.981	0.962	0.965	0.982	0.967	0.980	**0.983**	**0.983**
acute	0.977	**0.991**	0.971	0.958	0.965	**0.991**	0.954	**0.992**	0.986	**0.991**
Mean	0.796	**0.810**	0.809	0.791	0.790	0.801	0.806	0.803	0.800	0.804
Average Ranking	6.86	**3.89**	4.36	6.89	6.89	5.36	4.84	4.70	6.11	5.09
Win counts	2	**14**	8	1	0	8	7	13	4	13
win/tie/loss	-	**33/7/4**	32/3/9	20/5/19	21/3/20	29/3/12	32/3/9	30/2/12	25/2/17	25/5/14

and EM-based approaches to semi-supervised learning [150, 151]. Guo and Greiner [75] proposed the on-line adjustment for MAXER, which switches to another active learning method when MAXER supposedly guesses wrong about the true label of latest queried instance. We do not adopt this adjustment since it can be used for any active learning algorithms and we only focus on the performance of original, pure active learning methods. CEER obtains performance comparable to that of MAXER, and it shares the same problem with MAXER since it also uses the optimistic strategy [107]. One possible reason why CEER underperforms is that the trade-off parameter α is not well determined.

As for the variance reduction approaches, EVR slightly outperforms FIVR in terms of average scores and win/tie/loss counts. While FIVR achieves better performance than MAXER and CEER, it is still exceeded by random sampling on 12 datasets. EVR behaves comparably to EER. Three remaining methods, MLI, AAL and MMC, have similar performances on average ALC score. ENTROPY, MLI and MMC perform the best in terms of win counts. However, MMC is surpassed by random sampling on 14 datasets. AAL performs worse than random sampling on 17 datasets.

As random sampling is the technique to beat, it is important to see how the active learners perform in comparison to random sampling over the 44 datasets. Therefore, we consider the ratio $\frac{V_{active}}{V_{random}}$ where V_{active} and V_{random} are the ALC scores of active learning and random sampling, respectively. This gives us an indication of the relative improvement (or deterioration) the active learning schemes provide. We compute the ratios over all the datasets and visualize the outcomes with a box plot in Fig 2.5. The 25th, 50th and 75th percentiles are shown and the green crosses indict the average values of the ratios. We can observe that ENTROPY and EER may deliver satisfactory performances, while MAXER and CEER behave rather poorly. MLI achieves the highest ratio on one dataset, which means that MLI can improve most upon random sampling in some instances. Possibly more importantly, however, ENTROPY and EER may be considered safer: they may not reach the relative improvements that MLI does, but at least they also do not show dramatic decreases in performance. Even though random sampling strategy is expected to be less efficient than actual active learning algorithms, at times, it can perform so well in comparison to the latter. Similar observations have been made before in [152] that random sampling is a runner-up in the active learning challenge.

Table 2.2 is divided into three different sections according to the ALC value achieved by random sampling. The first group, in which ALC scores range from 0.5 to 0.75, represents the datasets on which reaching good performance seems difficult. The second group, ranging from 0.75 to 0.90, corresponds to the datasets which have medium levels of difficulty for classification. The last group consists of the remaining datasets, which seem fairly easy to solve by a linear logistic classifier. We can see that random sampling surpasses all the other methods on the blood dataset. On the medium and easy datasets, random sampling does not achieve the best performances, which may indicate that we need only consider random sampling on relatively hard tasks. For the difficult classification datasets in the first group, ENTROPY, FIVR and MLI achieve comparable performances. FIVR performs best on the datasets in the first group while it performs poorly on the easy and medium datasets. In the second group, EVR obtains the best performance, while it underperforms in the last group. For the easy datasets, MMC, MLI and ENTROPY are slightly better than the other methods. ENTROPY also performs well

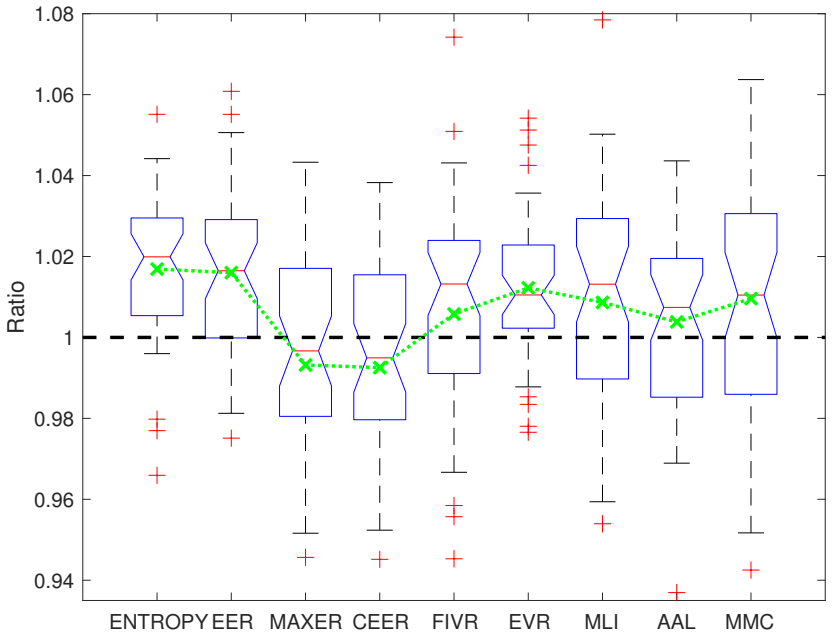


Figure 2.5: A box plot of the ratios of the ALC scores of active learning to that of random sampling over all the datasets. The green crosses represent the average values of the ratios. The black dashed line is at one, at the performance of random sampling.

on medium and hard datasets. The experiments demonstrate that uncertainty sampling is a robust active learning algorithm, regardless of the difficulty-level of the tasks.

Table 2.3 shows the average performance of all the methods over 80 subsets of the ImageNet database. A detailed description of the performance on each subset is shown in Table 2.4. We can observe that, also in this setting, ENTROPY performs the best in terms of the four measures, i.e. average ALC, average ranking, win counts, and win/tie/loss. Interestingly, all other methods are outperformed by random sampling in terms of average ALC and average ranking. In conclusion, all but the simplest approach overall fail to outperform random sampling on this particular ImageNet database. This seems to indicate that more attention may have to be devoted to seeking safe, yet effective active learning algorithms [153].

Table 2.4: Performance comparison on the Area under the Learning Curve (Accuracy). The higher the score, the better the performance. For each data set, the best performances and its comparable competitors are highlighted in bold face and surrounded with a box. Average performance of all the active learning methods are also reported as "Mean". "Average Ranking" shows the average ranking of compared methods. "Win counts" shows the total number of datasets on which one method achieves the best performances. "win/tie/loss" demonstrates the win/tie/loss counts of one method versus random sampling on all the datasets based on paired t-test at 95 percent significance level.

Dataset	Random	ENTROPY	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
Egyptian	0.554	0.550	**0.559**	0.546	0.545	0.546	0.555	0.544	0.553	0.533

Continued on next page

Table 2.4 – *Continued from previous page*

Dataset	Random	ENTROPY	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
Tabby	0.565	0.569	0.569	0.568	0.570	0.564	**0.586**	0.559	0.563	0.570
Siamese	0.616	0.622	0.621	0.615	0.618	**0.630**	**0.632**	0.611	0.628	0.614
Persian	0.621	0.629	0.640	0.628	0.633	**0.661**	0.650	0.606	0.638	0.619
umbrella vs. ball	0.622	0.614	0.604	0.572	0.577	**0.626**	0.592	0.604	0.554	0.599
computermouse vs. helmet	**0.638**	0.624	0.617	0.599	0.601	0.620	0.617	0.624	0.592	0.630
scissors vs. cellphone	0.638	**0.641**	0.624	0.616	0.623	0.633	0.615	0.619	0.636	0.632
bottle vs. cellphone	0.638	**0.645**	0.616	0.617	0.616	0.624	0.631	0.629	0.624	0.631
ewer vs. knife	**0.644**	0.629	0.610	0.586	0.586	0.588	0.631	0.637	0.592	0.630
spoon vs. telephone	**0.649**	0.641	0.632	0.623	0.623	0.635	0.639	0.642	0.618	0.626
catrabbit	0.652	0.663	0.667	0.652	0.659	0.648	0.659	0.657	**0.669**	0.637
Tiger	0.655	0.660	0.667	0.658	0.670	0.645	**0.672**	0.629	0.658	0.637
bottle vs. spoon	0.666	0.680	0.648	0.667	0.657	0.635	0.649	0.639	**0.687**	0.635
calculator vs. cellphone	0.672	**0.674**	0.649	0.652	0.634	0.650	0.656	0.659	0.639	0.644
teapot vs. lightbulb	**0.678**	0.676	0.667	0.620	0.646	0.668	0.658	0.659	0.640	**0.678**
spoon vs. cartire	**0.679**	0.675	0.658	0.647	0.659	0.648	0.655	0.672	0.650	0.671
flag vs. tower	**0.682**	0.679	0.667	0.664	0.670	0.678	0.673	0.661	0.679	0.667
rifle vs. eyeglasses	0.701	**0.713**	0.692	0.684	0.683	0.686	0.667	0.672	0.693	0.684
truck vs. boat	0.714	0.728	0.695	0.681	0.687	**0.736**	0.704	0.688	0.662	0.714
table vs. cellphone	**0.718**	0.713	0.689	0.673	0.670	0.674	0.682	0.698	0.651	0.680
motorcycle vs. baseballbat	**0.719**	0.717	0.681	0.658	0.658	0.664	0.700	0.674	0.666	0.678
skunk vs. umbrella	0.721	**0.724**	0.699	0.699	0.683	0.720	0.698	0.695	0.702	0.701
catpanda	0.725	**0.742**	0.726	0.731	0.728	0.691	0.736	0.703	**0.743**	0.681
apple vs. cup	0.725	0.728	0.723	0.715	0.707	**0.735**	0.708	0.714	0.718	0.722
sheep vs. skunk	0.726	0.725	0.719	0.718	0.716	0.724	0.716	0.708	**0.735**	0.705
motorcycle vs. bridge	0.732	**0.737**	0.661	0.637	0.634	0.727	0.699	0.697	0.603	0.734
bike vs. spoon	0.754	**0.769**	0.733	0.751	0.750	0.744	0.750	0.736	0.765	0.753
bathtub vs. basketball_hoop	**0.756**	**0.756**	0.736	0.734	0.733	0.749	0.734	0.734	**0.756**	0.741
washingmachine vs. cup	0.759	**0.772**	0.736	0.729	0.727	0.754	0.736	0.746	0.754	0.748
piano vs. scissors	**0.763**	0.755	0.721	0.728	0.728	0.713	0.725	0.743	0.731	0.756
lama vs. kangaroo	0.766	0.771	0.779	0.780	0.788	0.790	0.782	0.764	0.766	**0.793**
catelepant	0.768	0.795	0.781	0.788	0.791	0.765	0.789	0.762	**0.797**	0.763
horse vs. windmill	0.770	0.771	0.728	0.741	0.722	**0.795**	0.754	0.750	0.737	0.769
washingmachine vs. umbrella	0.771	**0.785**	0.770	0.767	0.767	0.771	0.762	0.762	0.776	0.780
bear vs. flower	0.772	**0.775**	0.737	0.708	0.700	0.770	0.748	0.765	0.734	0.726
piano vs. ball	0.772	**0.781**	0.755	0.773	0.760	0.779	0.753	0.754	0.759	0.779
scorpion vs. scissors	0.773	0.772	0.763	0.759	0.759	0.739	0.759	0.750	**0.776**	0.749
tomato vs. umbrella	0.774	**0.785**	0.765	0.765	0.764	0.778	0.762	0.752	0.769	0.751
buildings vs. lamp	0.776	**0.783**	0.757	0.761	0.759	0.775	0.757	0.753	0.769	0.762
billiards vs. umbrella	0.779	**0.788**	0.742	0.728	0.726	0.759	0.746	0.762	0.713	0.770
binder vs. grand_piano	0.790	0.785	0.774	0.779	0.782	0.805	0.775	0.772	0.782	**0.819**
chandelier vs. baseballglove	0.791	0.797	0.768	0.755	0.762	0.798	0.791	0.779	0.740	**0.806**
knob vs. giraffe	0.791	**0.805**	0.765	0.755	0.758	0.789	0.774	0.775	0.770	0.798
palmtree vs. rifle	0.799	0.808	0.776	0.792	0.786	0.798	0.767	0.777	**0.811**	0.768
bird vs. tombstone	0.805	0.804	0.807	0.798	0.784	0.804	0.778	0.807	0.758	**0.813**
motorcycle vs. buildings	0.808	**0.820**	0.777	0.758	0.762	0.791	0.794	0.780	0.787	0.815
flower vs. ewer	0.815	0.812	0.805	0.795	0.793	**0.834**	0.806	0.804	0.797	0.816
dog vs. headphone	0.817	**0.825**	0.795	0.793	0.786	0.816	0.810	0.813	0.803	0.823

Continued on next page

Table 2.4 – *Continued from previous page*

Dataset	Random	ENTROPY	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
horse vs. shoe	0.819	0.823	0.805	0.791	0.795	**0.830**	0.804	0.817	0.815	0.826
umbrella vs. gorilla	0.819	**0.832**	0.812	0.828	0.827	0.812	0.809	0.807	**0.833**	0.816
chandelier vs. mushroom	0.820	**0.829**	0.808	0.810	0.800	0.803	0.803	0.794	0.810	0.810
bottle vs. boat	0.829	**0.836**	0.812	0.820	0.818	**0.836**	0.813	0.819	0.828	0.821
lighthouse vs. bike	0.830	**0.845**	0.817	0.808	0.799	0.842	0.811	0.809	0.805	0.827
bridge vs. skunk	0.831	**0.842**	0.784	0.790	0.780	0.840	0.792	0.806	0.799	0.824
lama vs. can_soda	0.833	**0.841**	0.813	0.791	0.812	**0.840**	0.811	0.820	0.751	0.836
lightbulb vs. giraffe	0.843	0.850	0.831	0.824	0.826	**0.865**	0.841	0.827	0.826	0.863
cerealbox vs. skunk	0.846	0.859	0.845	0.858	0.855	**0.865**	0.844	0.830	0.853	0.854
dog vs. stapler	0.847	0.848	0.827	0.833	0.828	0.841	0.831	0.841	0.833	**0.850**
traffic_light vs. chimp	0.848	0.856	0.819	0.835	0.827	**0.858**	0.822	0.835	0.801	0.850
bridge vs. washingmachine	0.849	**0.864**	0.839	0.844	0.850	0.857	0.838	0.834	0.837	0.839
octopus vs. scissors	**0.851**	**0.850**	0.820	0.821	0.821	0.800	0.828	0.832	0.824	0.843
bear vs. fireextinguisher	0.853	0.861	0.837	0.823	0.820	0.856	0.842	0.845	0.834	**0.864**
cannon vs. tombstone	0.857	0.868	0.857	0.864	0.863	0.852	0.852	0.854	0.863	**0.871**
aeroplane vs. sheep	0.857	0.871	0.861	0.857	0.866	0.878	0.861	0.850	0.862	**0.881**
frying_pan vs. bear	0.864	0.865	0.828	0.816	0.812	**0.880**	0.831	0.845	0.801	0.875
chandelier vs. kangaroo	0.865	0.870	0.831	0.814	0.808	**0.878**	0.827	0.841	0.787	0.872
spoon vs. tombstone	0.867	0.874	0.857	0.870	**0.877**	0.845	0.852	0.850	0.853	0.872
cerealbox vs. monitor	0.871	0.876	0.865	0.866	0.866	**0.882**	0.848	0.845	0.848	0.881
butterfly vs. cerealbox	0.873	**0.883**	0.873	0.873	0.878	0.870	0.868	0.845	0.873	0.879
headphone vs. people	0.875	**0.891**	0.862	0.868	0.877	0.874	0.877	0.860	0.880	**0.891**
tenniscourt vs. ladder	0.879	0.886	0.886	**0.892**	0.891	0.890	0.885	0.878	0.889	0.877
people vs. computermouse	0.895	0.906	0.890	0.908	0.909	0.911	0.889	0.894	0.904	**0.914**
firetruck vs. bathtub	0.899	0.903	0.879	0.904	0.902	0.910	0.897	0.892	0.897	**0.915**
keyboard vs. bonsai	0.900	0.913	0.906	**0.921**	0.918	0.906	0.886	0.879	0.914	0.913
keyboard vs. bonsai	0.900	0.913	0.906	**0.921**	0.918	0.906	0.886	0.879	0.914	0.913
skyscraper vs. bonsai	0.901	0.903	0.867	0.870	0.869	0.907	0.884	0.882	0.896	**0.909**
teapot vs. tree	0.911	**0.918**	**0.920**	0.914	0.911	**0.920**	0.895	0.895	0.894	0.914
frying_pan vs. microwave	0.919	0.907	0.868	0.906	0.898	0.911	0.881	0.895	0.892	**0.921**
flashlight vs. tombstone	0.922	0.922	0.922	0.925	0.926	0.930	0.919	0.909	0.916	**0.939**
tree vs. dinosaur	0.936	0.941	0.939	0.944	**0.946**	0.943	0.938	0.938	0.942	**0.946**
Mean	0.778	**0.783**	0.763	0.761	0.760	0.775	0.765	0.762	0.761	0.774
Average Ranking	3.96	**2.58**	6.54	6.66	6.78	4.40	6.53	7.22	6.03	4.31
Win counts	11	**30**	2	3	2	17	3	0	9	16
win/tie/loss	-	**58/7/15**	15/7/58	17/8/55	21/4/55	35/8/37	13/3/64	3/4/73	22/6/52	31/10/39

2.3.4. COMPUTATIONAL COST ANALYSIS

Computational cost can also be a critical issue when employing active learning methods. Table 2.5 assesses the average computational cost of selecting 40 unlabeled samples of each of the methods. All the experiments are constructed with MATLAB 9.1 on an Intel(R) Core 2.8GHz i7-4980HQ CPU PC with 16 GB memory. We test the computational cost on 8 datasets that vary in the numbers of instances and the feature dimensionalities. Clearly, random sampling, ENTROPY and MMC are the most efficient methods due

Table 2.3: Average performance of the Area under the Learning Curve on 80 subsets of the ImageNet database. The best performances are highlighted in bold face and surrounded with a box. "Mean" reports the average performance of the Area under the Learning Curve. "Average Ranking" shows the average ranking of compared methods. "Win counts" shows the total number of datasets on which one method achieves the best performances. "win/tie/loss" demonstrates the win/tie/loss counts of one method versus random sampling on all the datasets based on paired t-test at 95 percent significance level.

Dataset	Random	ENTROPY	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
Mean	0.778	**0.783**	0.763	0.761	0.760	0.775	0.765	0.762	0.761	0.774
Average Ranking	3.96	**2.58**	6.54	6.66	6.78	4.40	6.53	7.22	6.03	4.31
Win counts	11	**30**	2	3	2	17	3	0	9	16
win/tie/loss	-	**58/7/15**	15/7/58	17/8/55	21/4/55	35/8/37	13/3/64	3/4/73	22/6/52	31/10/39

to their simplicity. MLI also has a low computational burden compared with other algorithms. Error reduction and variance reduction have, on the other hand, a significantly higher computational cost than other methods. Both of them are especially significantly less efficient for handling high dimensionality datasets like 3vs5 and basehockey. The reason may be that both need to retrain the logistic regressor in every selection step over all the unlabeled instances and all possible labels, which is relatively time consuming especially in higher dimensions. We also note that EVR has the highest computational cost. This is because EVR has to repeatedly calculate the inverse of matrix, which is extremely computationally expensive. AAL has the second-highest computational cost since it also needs to compute the inverse matrix multiple times.

Table 2.5: Computational cost comparison of querying 40 unlabeled instances for each active learning method (in seconds)

Dataset (#Ins #Fea)	Random	Entropy	EER	MAXER	CEER	FIVR	EVR	MLI	AAL	MMC
acute (120,6)	0.006	0.015	0.520	0.502	0.824	0.085	0.719	0.530	0.171	0.017
australian (690,14)	0.013	0.030	6.794	6.619	8.618	0.830	11.287	4.794	11.848	0.044
musk1 (476, 166)	0.005	0.054	16.855	16.113	18.077	8.725	39.832	10.510	14.159	0.059
hill (600, 100)	0.006	0.045	16.842	16.534	19.355	5.011	34.041	8.121	16.025	0.051
mushrooms (1000, 112)	0.007	0.029	15.179	15.313	17.706	7.787	89.594	9.686	46.020	0.110
letterEF (1543, 16)	0.006	0.031	25.400	25.382	28.850	1.129	44.214	11.635	203.500	0.082
3vs5 (1500, 784)	0.006	0.186	219.578	219.838	230.834	609.544	1806.288	55.369	216.266	0.247
basehockey (1993, 500)	0.007	1.132	1133.409	1122.366	1139.517	289.424	2060.489	165.871	2251.607	0.719
Mean	0.007	0.190	179.322	177.834	182.973	115.317	510.808	33.315	344.949	0.166

2.4. DISCUSSION AND CONCLUSION

This survey focuses on logistic regression because it is broadly applied and because of the fact that many active learning methods can be used in combination with this particular classifier. It should be clear, however, that some categories of active learning discussed in this work can also be used in combination with other types of classifiers. For instance, uncertainty sampling and error reduction approaches can be readily employed in combination with other probabilistic classifiers that can provide a posterior probability per sample, e.g. like naive Bayes [40]. Especially in the two-class case, uncertainty sampling can already be applied as soon as one has a notation of distance to the deci-

sion boundary. A technique like minimum loss increase has also been studied in relation with SVMs [80] and ridge regression [82]. Maximum model change can also be used in combination with SVMs [45].

On the other hand, there are also some active learning algorithms which are not easily combined with logistic regression. Examples are particular graph-based methods [53, 54] and methods that rely on model change with a closed-form estimate [78] as these methods are specifically derived on the basis of Gaussian Processes. Other approaches rely on the notion of a version space or a margin [24, 154] and therefore can also not be combined readily with logistic regression.

More recently, quite some effort has gone into the study of scenarios that deviate to a smaller or larger extent from the standard myopic active learning setting that we focus on. The main research directions that we identified are multi-label active learning where every instance may have multiple labels simultaneously [155, 156], multi-task active learning in which various tasks are meant to be learned jointly [157, 158], multiple instance active learning where human experts annotate an entire set that contains some samples instead of individual instances [17, 159], cost-sensitive active learning where different samples have varying labeling costs [12, 160], and active transfer learning which combines transfer learning and active learning [118, 161].

Finally, there are of course approaches in which deep learning and active learning come together. An original application is [162] which proposed to use a generative adversarial network (GAN) to synthesize training instances for labeling instead of using real, observed samples. Another contribution, offering an original way of active labeling is [163]. In that work, a GAN is used to generate new images along a 1-dimensional query line and a human expert is asked, rather than to provide a label, to provide the point where the images change class.

In this chapter, we compared current state-of-the-art active learning methods for logistic regression and pointed out their main similarities and dissimilarities. The experiments on the synthesis datasets and the large number of real-world datasets show some of the chief underlying characteristics of each of the active learning methods. On average, we would deem ENTROPY the most promising method. Though ENTROPY is a rather simplistic criterion and quite short-sighted when picking instances, it does outperform the min-max view approach, variance reduction methods and maximum model change algorithm in our experiments. Uncertainty sampling was first proposed in 1994, which may indicate that, in some sense, little progress has been made since then. MLI demonstrates its advantage in querying the representative instances on the synthesis data Synth2. A possible downside of expected error reduction approaches is the high computational cost it incurs. Variance reduction approaches and MLI suffer the same problem. How to speed up these methods is definitely a worthwhile problem for further research.

Overall, on the positive side, we can conclude that active learning can indeed provide improved performance over random sampling, most certainly if we consider the whole ensemble of active learners in this work. This, however, also seems to be a negative aspect. On its own, none of these methods can prevent becoming worse than random sampling. While this seems impossible anyway for every single instantiation of a problem, our results indicate that it does not even hold in the average. That is, for every active

learner there are (real-world) datasets on which the active learner performs significantly worse than random sampling, even when averaged over multiple runs. Finding active learning methods that are, in some sense, safe and yet give significant performance gains at times, still seems to be the challenge ahead (cf. [93, 153, 164–166]).

3

ACTIVE LEARNING USING UNCERTAINTY INFORMATION

Many active learning methods belong to the retraining-based approaches, which select one unlabeled instance, add it to the training set with its possible labels, retrain the classification model, and evaluate the criteria that we base our selection on. However, since the true label of the selected instance is unknown, these methods resort to calculating the average-case or worst-case performance with respect to the unknown label. In this chapter, we propose a different method to solve this problem. In particular, our method aims to make use of the uncertainty information to enhance the performance of retraining-based models. We apply our method to two state-of-the-art algorithms and carry out extensive experiments on a wide variety of real-world datasets. The results clearly demonstrate the effectiveness of the proposed method and indicate it can reduce human labeling efforts in many real-life applications.

This chapter is published as:
Yazhou Yang, and Marco Loog. "*Active Learning using Uncertainty Information.*" In *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 2646-2651, 2016.

3.1. INTRODUCTION

Over the past decade, a primary foundation of much progress in machine learning is the rapid growth of the number and size of data sets available, such as ImageNet [6] containing over 14 million labeled images for object recognition. In a practical scenario, we frequently encounter the situation where few labeled instances along with abundant unlabeled samples are available. Labeling a large amount of data is, however, very difficult due to the huge amount of time required or expensive because of the need of human experts [36]. Thus, it is very attractive to propose a proper labeling scheme to reduce the number of labels required in order to train a classifier.

Active learning has been put forward to overcome the above labeling problem. The main assumption behind active learning is that if an active learner can freely select any samples it wants, it can outperform random sampling with less labeling [36]. Thus, the main task of active learning is querying as little data as possible to minimize the annotation cost while maximizing the learning performance. Active learning tries to achieve this by selecting the most valuable samples. However, it is difficult to define or measure the value of one instance to the learning problem. We can view it as the amount of information carried which potentially promotes the learning performance, once its true label is known [100]. As a result of the fact that we do not have an exact measure of the value, there are a great number of selection criteria proposed from different perspectives on how to estimate the usefulness of each sample.

Most commonly used criteria in active learning include query-by-committee [62], uncertainty sampling [24, 167, 168], expected error reduction [40, 75, 107, 124], expected model change [77, 79, 169, 170], variance reduction [13, 52, 76, 111] and "Min-max" view active learning [80, 171]. Query-by-committee put forward multiple models as the committees and selected the samples which receive highest level of disagreement from the committees [62]. Uncertainty sampling approach preferred the instances with maximum uncertainty. Based on the measurement of uncertainty, uncertainty sampling can be roughly divided two categories: maximum entropy of the estimated label [167] and minimum distance from the decision boundary [24, 168]. For example, Tong and Koller [24] proposed to query the instance which is closed to the current learning boundary using the classifier of support vector machines. Campbell *et al.* [168] shared the same idea with Tong and Koller [24].

Roy and McCallum [40] proposed the expected error reduction (EER), which is a popular active learning method. EER aimed to reduce the generalization error when labeling a new instance. Since we do not have access to the test data, Roy and McCallum suggested to compute the "future error" on the unlabeled pool under the assumption that the unlabeled data set is representative of the test distribution. In other words, the unlabeled pool can be viewed as a validation set. Also, we have no knowledge about the true labels of unlabeled samples. EER estimated the average-case criterion of potential loss instead. Expected model change followed the idea of EER, but turned to select the instance which leads to maximum change of the current model. The variance reduction methods tried to minimize the output variances [36]. Schein and Ungar [76] extended this approach to expected variance reduction method on logistic regression by following the idea of EER. "Min-max" view active learning was originally proposed by Hoi *et al.* [80], where "Min-max" indicates the worst-case criterion is adopted. The key idea

behind is to select the sample which minimizes the gain of objective function no matter what its assigned label is. Huang *et al.* [171] extended this framework by taking into account all the unlabeled data when calculating the objective function.

Current active learning methods can be split in two classes: retraining-based and retraining-free active learning. Retraining-based active learning represents methods which measure the information of unlabeled sample by labeling it (any possible label) and adding it to the training set to retrain the classification model. Then, some appropriate criteria can be evaluated and used for the sample selection. The second class, retraining-free active learning, contains the remaining methods which not need repeatedly train the model for each unlabeled instance during one single selection. For example, uncertainty sampling and query-by-committee belong to this category.

However, since the true label of the selected unlabeled instance is unknown, these methods resort to calculating the average-case or worst-case criteria with respect to the unknown label. In this chapter, we propose a different criterion for retraining-based methods. We incorporate the uncertainty information (measured by the posterior probabilities within the min-max framework) for the selection. The proposed criterion can be seen as a trade-off of the exploration and the exploitation. The uncertainty information plays the role of the exploitation while the retraining-based models act as the exploration part. We concentrate on the pool-based active learning setting which assumes a large pool of unlabeled data along with a small set of labeled data already available [36]. We consider the myopic active learning which sequentially and iteratively selects unlabeled instance.

3.1.1. OUTLINE

The rest of this chapter is organized as follows. Section 3.2 firstly reviews the framework of retraining-based active learning. Then two state-of-the-art methods under the retraining framework are briefly described. Section 3.3 demonstrates the primary motivation of the proposed method and derives a general algorithm for retraining-based active learning in detail. It also illustrate how to extend the proposed criterion to current methods. Experimental design and results are reported in 3.4 ; Section 3.5 concludes this work followed by some future issues.

3.2. RETRAINING-BASED ACTIVE LEARNING

In this section, we summarize a general framework of retraining-based active learning. Then we demonstrate two examples under this framework: Expected error reduction and Minimum Loss Increase.

3.2.1. RETRAINING-BASED ACTIVE LEARNING

Firstly, let us introduce some preliminaries and notation. Let $\mathscr{L} = \{(x_i, y_i)\}_{i=1}^{m}$ represent the training data set that consists of m labeled instances and $\mathscr{U}$ be the pool of unlabeled instances $\{x_i\}_{i=m+1}^{n}$. Each $x_i \in \mathbb{R}^d$ is a d dimensional feature vector, and $y_i \in C = \{+1, -1\}$ is the class label of x_i. In this chapter, let us focus on binary classification problem firstly, and it is easy to extend this work to multi-class problem by extending C to multi-labels set. We denote $P_{\mathscr{L}}(y|x)$ be the conditional probability of y

Algorithm 1 General Retraining-based Active Learning Procedure

Input: Labeled data $\mathscr{L}$, unlabeled data $\mathscr{U}$
repeat
  Train the classifier on $\mathscr{L}$ and calculate $P_{\mathscr{L}}(y_i|x_i)$ for each $x_i \in \mathscr{U}$, each $y_i \in C$;
  for each $x_i \in \mathscr{U}$ **do**
    for each $y_i \in C$ **do**
      Re-train the model on $\mathscr{L} \cup \{x_i, y_i\}$;
      Calculate some criterion $V(x_i, y_i)$, (*e.g.*, error or variance);
    end for
  end for
  Compute some kind of performance based on $P_{\mathscr{L}}(y_i|x_i)$ and $V(x_i, y_i)$;
  Query the instance x^* which leads to the best performance and label it y^*, update $\mathscr{L} \leftarrow \mathscr{L} \cup \{x^*, y^*\}, \mathscr{U} \leftarrow \mathscr{U} \backslash \{x^*\}$;
until Stopping criterion is satisfied

given x according to a classifier trained on $\mathscr{L}$.

For the retraining-based active learning, its framework can be summarized in Algorithm 1, where $V(x_i, y_i)$ represents any selection criterion associated with (x_i, y_i). The main procedure contains the loops which checks all the points in unlabeled pool $\mathscr{U}$ over all the possible labels. For example, we firstly select one instance from the unlabeled pool and assign it any possible label. Then we update the labeled set (since we acquire a new labeled sample) and retrain the classifier we use. Based on the new trained classifier, we can measure some kind of selection criteria (*e.g.*, generalization error in EER [40]). However, since the true label information of last selected sample is unknown, we need calculate some kind of performance, *e.g.*, the average-case in [40, 76, 79], worst-case in [171], or even the best-case criteria in [75]. Finally, we will query the instance which leads to maximum or minimum value in terms of the criterion we are interested in.

EER is one example of retraining-based active learning, which uses the generalization error as $V(x_i, y_i)$. We get expected model change [77, 79, 169, 170] by adopting model change as the criterion. By adopting variance and logistic regression as the classifier, we get expected variance reduction [76]. Similarly, if we want to minimize the value of objective function after labeling a new instance and use the worst-case performance (corresponding to min-max framework), then we can get [80, 171]. Clearly, the retraining-based approaches may suffer from high computational cost due to the fact that they need go over all the unlabeled data and all the possible labels.

3.2.2. EXPECTED ERROR REDUCTION

Expected error reduction has demonstrated its effectiveness on text classification domain [40]. There are also some follow-up work of EER contributed by other researchers [75] [124] [107]. EER aims to select the sample which will reduce the future generalization error. Since we can not see the test data, the unlabeled pool can be used as the validation set to predict the future test error. We encounter a new problem since we do not know the true labels of the pool. Roy and McCallum [40] suggested, in practice,

we can approximately estimate the error using the expected log-loss or 0/1 loss over the pool. For example, if we adopt the log loss, EER can be written as follows:

$$\arg\min_{x\in\mathscr{U}} \sum_{y\in C} P_{\mathscr{L}}(y|x)\left(-\sum_{x_i\in\mathscr{U}}\sum_{y_i\in C} P_{\mathscr{L}^+}(y_i|x_i)\log P_{\mathscr{L}^+}(y_i|x_i)\right)$$

where $\mathscr{L}^+ = \mathscr{L}\cup(x,y)$ means that the selected instance x is labeled y and added to $\mathscr{L}$. Note that the first term $P_{\mathscr{L}}(y|x)$ contains the pre-trained label information. The second term is the sum of potential entropy over the unlabeled data set $\mathscr{U}$.

3.2.3. MINIMUM LOSS INCREASE

We can find that EER attempts to reduce the future generalization error, however, it is not easy due to the missing of test data and true label information of unlabeled data. There are some researchers which try to solve this problem from a different perspective. Hoi *et at.* [80] presented a so called "min-max" view active learning. It prefers the instance which results in a small value of an objective function in spite of its assigned label. This is because the smaller the value of an objective function, the better the learning model, at least in high probability. Assume $G_{\mathscr{L}}$ is the value of an objective function on current labeled data $\mathscr{L}$. When we label a new instance and update the training data $\mathscr{L}^+ = \mathscr{L}\cup\{x_i,y_i\}$, we get a new value of objective function $G_{\mathscr{L}^+}$. What we want is the minimum increase of objective function, *i.e.*, $G_{\mathscr{L}^+} - G_{\mathscr{L}}$, when adding one more labeled sample. However, because the second term $G_{\mathscr{L}}$ is independent of the next queried instance, so we can ignore it and focus on minimizing $G_{\mathscr{L}^+}$. Since we expect a minimum value of $G_{\mathscr{L}^+}$ regardless of the assigned label of x_i, we adopt the worst-case performance as follows, instead of the average-case version.

$$\arg\min_{x_i\in\mathscr{U}} \max_{y_i\in C} G_{\mathscr{L}^+}$$

Note that we can view $G_{\mathscr{L}^+}$ as one choice of $V(x_i,y_i)$ mentioned in Algorithm 1.

Let us consider an unconstrained optimization problem using L_2-loss regularized classifier with arbitrary loss $l(w;x_i,y_i)$: $g(w) = \frac{1}{2\lambda}||w||^2 + \sum_{x_i\in\mathscr{L}} l(w;x_i,y_i)$, where w is the parameter of learning classifier. If we adopt the Hinge loss $l(w;x_i,y_i) = \max(0,1-y_i w^T x_i)$, we can derive the same model with "min-max" view active learning described in [80], but without extend it to batch model setting. If we use square loss $l(w;x_i,y_i) = (y_i - w^T x_i)^2$, we can get the same model with [171]. Note that, as is stated in [172], though [171] includes all the unlabeled data when calculating the objective function, the unlabeled examples play no role since [171] relaxes the constraint of the labels of unlabeled pool in the end. This operation can guarantee *zero* contribution of unlabeled data to the objection function. Thus, [171] is also one special case using the square loss. Moreover, we can conclude that the main idea of min-max view active learning is to minimize the increase of the value of an objective function.

In our paper, we consider the logistic loss $l(w;x_i,y_i) = \log(1+\exp^{-y_i w^T x_i})$, which results in:

$$\arg\min_{x\in\mathscr{U}} \max_{y\in C} \frac{1}{2\lambda}||\hat{w}||^2 + \sum_{x_i\in\mathscr{L}^+} -\log P_{\mathscr{L}^+}(y_i|x_i) \tag{3.1}$$

where $\hat{w}$ is estimated parameter of L_2-regularized logistic regression model. Logistic regression is chosen as the base classifier since it is generally widely used in many fields and can output the conditional probability straightly, which can be used in active learning [172]. We call this method Minimum Loss Increase (MLI) in this chapter. EER tries to minimize the error on unlabeled data while MLI aims to minimize the loss on data already labeled.

3.3. A NEW RETRAINING-BASED ACTIVE LEARNER

In this section, we motivate our proposed method and, subsequently, describe a general adaptation for retaining-based active learning models.

3.3.1. MOTIVATION

Obviously, not knowing the true labels of the unlabeled data complicates calculating the final score of each instance in step 10 in Algorithm 1. One simple possibility is computing the average-case [40] or worst-case criteria [171], or even the best-case criterion [75]. These choices, however, may fail to take into account some potentially valuable information: Firstly, although the average-case criterion makes use of the label distribution information $P_{\mathscr{L}}(y_i|x_i)$ already known, the expectation calculation can hide or underestimate some outstanding samples due to the re-weighting by $P_{\mathscr{L}}(y_i|x_i)$. For example, the true label of instance x_i is +1 but the estimated $P_{\mathscr{L}}(+1|x_i) = 0.1$, and the $V(x_i,+1)$ has a maximum value compared with other instances. Then the average-case criterion of x_i, namely $\sum_{y_i} P_{\mathscr{L}}(y_i|x_i)V(x_i,y_i)$, is highly likely to be surpassed by other instances. Secondly, as to the worst-case criterion, it suffers from not taking advantage of label distribution information at all. Worst-case analysis is a safe analysis since it is never underestimated. However, making no use of the available label information $P_{\mathscr{L}}(y_i|x_i)$ can lose sight of some valuable information.

Thus, to overcome the shortcomings mentioned, a new criterion for retraining-based active learning is proposed. The main motivation is that we want to incorporate the uncertainty information (*e.g.*, known label distribution information) within min-max framework for retraining-based models. The proposed criterion is therefore as follows:

$$\min_{x_i \in \mathscr{U}} \max_{y_i \in C} P_{\mathscr{L}}(y_i|x_i)V(x_i,y_i) \tag{3.2}$$

where $P_{\mathscr{L}}(y_i|x_i)$ contains the pre-trained label information and $V(x_i,y_i)$ represents any criteria we are interested. Note that for some classifiers like logistic regression, we can use the estimated posterior probability as $P_{\mathscr{L}}(y_i|x_i)$. For classifiers which do not produce a probabilistic output, *e.g.*, SVMs, we can transform their output to some probability using Platt's [173] or Duin & Tax's method [174]. And for $V(x_i,y_i)$, various choices are possible, such as the test error on the unlabeled pool in EER, the output variance as in [76], or the value of an objective function [171].

The proposed method can be interpreted as follows: it utilizes the pre-trained label information, although this kind of information might be inaccurate due to limited labeled data we have, it still shows some underlying or potential useful clues which may promote active learning. Firstly, it improves upon the average-case criterion since it does not compute the expected value. The calculation of expectation tends to ruin the dis-

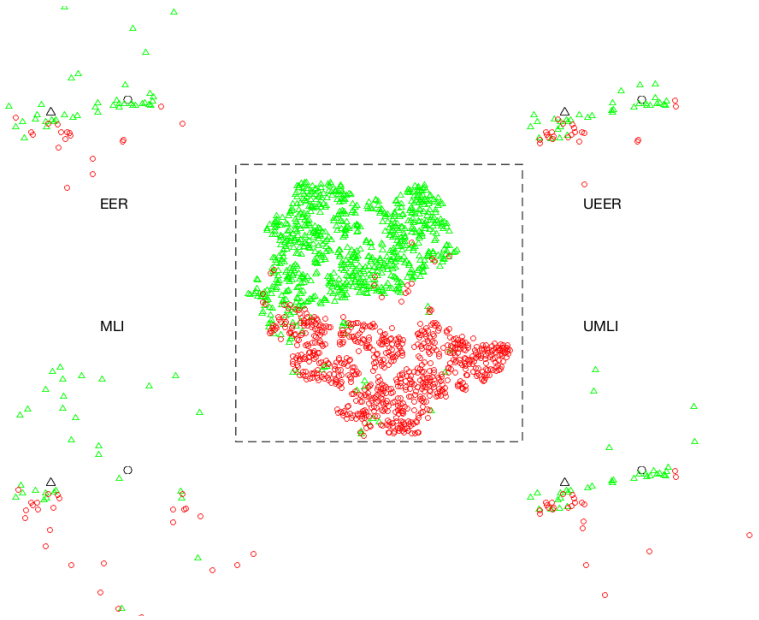


Figure 3.1: Illustration of the inherent characteristics of the proposed method. The middle is the distribution of a synthetic binary data set. Four corners represent the performance of four active learning methods, EER, UEER, MLI and UMLI, respectively. One black triangle and circle represent the initial labeled set.

criminative information contained in the data due to its averaging manner. Secondly, it outperforms the worst-case criterion because it takes advantage of the knowledge of the potential label distribution while worst-case analysis does not use this at all. Thus, it avoids the disadvantages of average-case and worst-case criteria. It can be seen as a trade-off between the average-case and the worst-case criteria. Lastly, it can be considered as incorporating uncertainty sampling (encoded by the posterior probabilities) for retraining-based model. If all $V(x_i, y_i)$ become one constant term like 1 or $P_{\mathscr{L}}(y_i|x_i)$ itself, then the proposed method will turn into exactly the uncertainty sampling. More specifically, $\min_{x_i \in \mathscr{U}} \max_{y_i \in C} P_{\mathscr{L}}(y_i|x_i)$ or $\min_{x_i \in \mathscr{U}} \max_{y_i \in C} [P_{\mathscr{L}}(y_i|x_i)]^2$ will act as totally same as uncertainty sampling since they will select the instance whose posterior probability comes closest to 0.5 on the binary problem. This shows that our proposed method actually fuses uncertainty sampling with retraining-based models.

3.3.2. TWO EXAMPLES OF THE PROPOSED METHOD

To provide valuable insights on the underlying characteristic of the proposed method, we apply it to two state-of-the-art retraining-based models EER and MLI. We also demonstrate its advantage on a synthetic data set in Figure 3.1.

Since our method tries to make use of the uncertainty information, the following adapted methods are termed uncertainty retraining-based active learners. It is easy to extend EER to uncertainty-based error reduction by adopting our method in Equation

3.2 as follows:

$$\arg\min_{x\in\mathscr{U}} \max_{y\in C} P_{\mathscr{L}}(y|x)\left(-\sum_{x_i\in\mathscr{U}}\sum_{y_i\in C} P_{\mathscr{L}^+}(y_i|x_i)\log P_{\mathscr{L}^+}(y_i|x_i)\right)$$

This method is called UEER for short. We can also apply our proposed criterion on MLI. The new approach is called UMLI in this chapter. Note that the regularization parameter $\frac{1}{2\lambda}$ in Equation 3.1 is usually quite small, so we ignore it in our adapted criterion:

$$\arg\min_{x\in\mathscr{U}} \max_{y\in C} P_{\mathscr{L}}(y|x)\sum_{x_i\in\mathscr{L}^+} -\log P_{\mathscr{L}^+}(y_i|x_i)$$

As is shown in Figure 3.1 , we construct a synthetic binary data set and two colours represent different classes. We demonstrate the performance of four retraining-based active learners EER, UEER, MLI and UMLI on four corners, respectively. One black triangle and circle in each corner represent two initial labeled points. When we compare UEER with EER, it is obvious that UEER selects a number of instances near the decision boundary while EER explores points in a wider range. This is because our method helps UEER make use of the uncertainty information and uncertainty information makes UEER focus on the region which is least certain about. Similar results can also be found between UMLI and MLI. MLI explores over the data space and queries the points around the border while UMLI balances the exploration and the exploitation. UMLI concentrates on the central part (exploitation) and also searches around the edge. Therefore, we can see that our method enhances retraining-based model by balancing the exploration and the exploitation.

3.4. EXPERIMENTS

In this section, we investigate the performance of our proposed methods to examine the effectiveness and robustness of our new criterion. The following experiments are limited to binary classification problems. Firstly, we show the experimental setting, then present the extensive experiment results, followed by further discussion and analysis.

3.4.1. EXPERIMENTAL SETTING

We compare the our proposed methods UEER and UMLI against their original version EER and MLI, respectively. Random sampling is also included in this comparison. In all the experiments, we use L_2-regularized logistic regression included in LIBLINEAR package [148] as default classifier with the same regularization parameter, $\lambda = 100$, for all methods.

The classification accuracy is used as the comparison criterion in our experiment. However, since active learning is a iteratively labeling procedure, we care about the performance during the whole learning process. Thus, it is not reasonable to merely compare the accuracy at some single points. Instead, we generate the learning curve of classification accuracy versus the number of labeled instances. Then, we calculate the area under the learning curve (ALC) as a measure of evaluation.

We test on totally 49 real-world data sets from various real-life applications, including many UCI data sets [139], MNIST handwritten digit dataset [140] and 20 Newsgroups

dataset [141]. There are 39 datasets from UCI benchmark datasets, such as breast, vehicle, heart and so on. These datasets are pre-processed according to [142]. For wine data set, we conduct class 2 against class 1 and 3 as binary problem. For glass data set, we also split it into two groups (class 1-3 vs. class 5-7) to build binary case. We randomly sub-sample 1000 instances from mushroom for computing efficiency. We select six pairs of letters from Letter Recognition Data Set [139], *i.e.,* D vs. P, E vs. F, I vs.J , M vs.N, V vs. Y and U vs. V since these pairs look similar to each other and distinguishing them is a little challenging. 3 vs. 5, 5 vs. 8 and 7 vs. 9 are three difficult pairs taken from MNIST data set [1] and used as the binary classification data set. We randomly sub-sample 1500 instances from the three data sets for computing efficiency. We also test the performance on 20 Newsgroups dataset which is a common benchmark used for text classification [2]. Following the work of [145], we also evaluate three binary tasks from 20 Newsgroups dataset: baseball vs. hockey, pc vs. mac, and religion.misc vs. alt.atheism. And the three pairs represent easy, moderate and difficult classification problems, respectively. We apply PCA to reduce the dimensionality of the above three datasets to 500 for computation efficiency. We also use the pre-processed data autos, motorcycles, baseball, hockey used in [52].

To objectively evaluate the performance, each data set is randomly divided into training and test data set of equal size. At the very beginning of active learning, we assume that only two instances randomly picked up from the training data are labeled, and one of them is from the positive class and the other is from the negative class. We run each active learning algorithm 20 times on each real-world dataset. The average performance of each active learning method is reported in the following section.

3.4.2. RESULTS

Table 3.2 shows the experimental results on 49 data sets. The datasets in Table 3.2 are sorted with respect to the performance of random sampling. We can find that the comparisons contain the datasets which vary from very difficult problems (*e.g.*, hill) to easy tasks (*e.g.*, acute). To clearly demonstrate the advantage of the proposed method, we do pairwise comparison between the original algorithm and its counterpart, *e.g.*, EER vs. UEER and MLI vs. UMLI, respectively. On each data set, a paired t-tests at 95% significance level is used to determine which method has the best performance or provides comparable outcome. These methods are highlighted in bold face. Over all the experiments, average performances are reported in Table 3.2. "Average Rank" shows the average rank of all the methods with regard to their performances on all the experiments. The lower the value of average rank, the better the method. The "win/tie/loss counts" represents times of our proposed methods versus its counterparts over all the 49 datasets.

As is shown in Table 3.2, our proposed methods UEER and UMLI evidently outperform their counterparts EER and MLI, respectively. UEER surpasses EER in terms of average accuracy, and improves its performance from 0.812 to 0.822. UEER also outperforms EER in terms of "average rank", which demonstrates the effectiveness of our method. Similar results can be found between UMLI and MLI. UMLI is superior to MLI on the overall performance. Moreover, it is interesting to observe that UEER attains the

[1] http://yann.lecun.com/exdb/mnist/

[2] http://qwone.com/ jason/20Newsgroups/

Table 3.1: Data sets information: It shows the number of instances (# INS) and the feature dimensionality (# FEA)

Data set (# Ins, # Fea)	Data set (# Ins, # Fea)	Data set (# Ins, # Fea)
ac-inflam (120, 6)	acute (120, 6)	australian (690, 14)
blood (748, 4)	breast (683, 10)	credit (690, 15)
cylinder (512, 35)	diabetes (768, 8)	fertility (100, 9)
german (1000, 24)	glass (214, 9)	haberman (306, 3)
heart (270, 13)	hepatitis (255, 19)	hill (606, 100)
ionosphere (351, 34)	liver (345, 6)	mushrooms (1000, 112)
mammographic (961, 5)	musk1 (476, 166)	ooctris2f (912, 25)
ozone (1000, 72)	parkinsons (195, 22)	pima (768, 8)
planning (182, 12)	sonar (208, 60)	splice (1000, 60)
tictactoe (958, 9)	vc2 (310, 6)	vehicle (435, 18)
wine (178, 13)	wisc (699, 9)	wdbc (569, 31)
d vs p (1608, 16)	e vs f (1543, 16)	i vs j (1502, 16)
m vs n (1575, 16)	v vs y (1577, 16)	u vs v (1550, 16)
3 vs 5 (1500, 784)	5 vs 8 (1500, 784)	7 vs 9 (1500, 784)
base-hockey (1993, 500)	pc-mac (1945, 500)	misc-atheism (1427, 500)
autos (3970, 8014)	motorcycles (3970, 8014)	baseball (3970, 8014)
hockey (3970, 8014)		

best overall performance among all the active learning methods. Over all the experimental data sets, the "win/tie/loss" counts of UEER versus EER is 29/7/13, meaning that UEER is the preferred active learner in over half the cases. With regard to UMLI and MLI, the "win/tie/loss" count is 27/11/11, which also shows the clear benefit of our scheme nonetheless. We also notice that even random sampling can surpass all the other methods, *e.g.*, on the blood data set, indicating that, generally, one might not want to use active learners in a blind way.

To investigate the robustness of our method, we also apply the worst-case criterion on EER and the average-case criterion on MLI, respectively. Due to the lack of space, we omit the results on each data set and only report the average performances. The average performance (ALC) of the worst-case on EER is 0.771 while that of the average-case on MLI is 0.710. To our surprise, they definitely show poorer performances in comparison with our method and even perform worse than random sampling. The possible reason may be that: EER computes the error on the unlabeled data and none of the true label are known, the average-case criterion is a safe choice for EER. Since MLI estimates the loss on the enlarged labeled set $\mathscr{L} \cup \{x_i, y_i\}$ and only the true label of x_i is unknown, the worst-case criterion is more appropriate for MLI than the average-case criterion. However, since the proposed method is a trade-off of the two criteria, it can adapt to both settings and show a robust performance for different retraining-based models.

3.5. CONCLUSIONS

In this chapter, we propose a new general method for retraining-based active learning. The proposed method can balance a trade-off of the average-case and worst-case criteria by incorporating uncertainty information (carried by the pre-trained posterior probabilities) within min-max framework. It drives current retraining-based models to pay more attention to the exploitation. We employ the new idea on two state-of-the-art methods to investigate its effectiveness. The synthetic data demonstrates that our method prefers to select the instances which are near the decision boundary in comparison with the original retraining-based approaches. Moreover, extensive experiments on 49 real-world datasets also prove that the proposed method is a promising approach for promoting retraining-based active learners.

Table 3.2: Performance Comparison on the areas under the learning curve (ALC)

Dataset	Random	EER	UEER	MLI	UMLI
hill	0.581	**0.616**	0.599	**0.626**	0.612
planning	0.586	**0.58**	0.578	**0.614**	0.586
cylinder	0.586	**0.61**	0.597	0.608	**0.617**
liver	0.627	**0.635**	0.626	**0.615**	0.607
splice	0.659	0.679	**0.682**	0.65	**0.666**
german	0.664	0.673	**0.679**	0.691	**0.703**
ooctris2f	0.679	**0.678**	0.673	**0.686**	0.663
musk1	0.682	0.699	**0.71**	**0.702**	0.688
fertility	0.693	0.706	**0.712**	**0.727**	0.711
haberman	0.711	0.712	**0.715**	0.694	**0.7**
sonar	0.713	**0.715**	0.707	0.708	**0.712**
pima	0.716	0.706	**0.714**	0.711	**0.722**
pcmac	0.717	**0.715**	**0.716**	0.747	**0.751**
diabetes	0.719	**0.723**	**0.723**	**0.726**	**0.728**
religionatheism	0.72	0.708	**0.718**	0.691	**0.739**
hepatitis	0.731	**0.753**	**0.75**	0.73	**0.738**
blood	0.743	**0.74**	0.718	0.73	**0.732**
baseball	0.753	0.765	**0.872**	0.832	**0.847**
motorcycles	0.763	0.78	**0.883**	0.854	**0.859**
autos	0.768	0.768	**0.872**	**0.838**	**0.835**
heart	0.774	0.791	**0.795**	**0.797**	**0.799**
hockey	0.775	0.787	**0.901**	0.875	**0.882**
ionosphere	0.779	**0.818**	0.806	0.674	**0.766**
credit	0.779	0.793	**0.814**	0.797	**0.809**
mammographic	0.78	0.774	**0.795**	0.766	**0.779**
basehockey	0.793	0.785	**0.801**	0.817	**0.847**
vc2	0.807	**0.815**	0.812	**0.825**	0.82
parkinsons	0.811	**0.824**	**0.821**	**0.83**	0.826
australian	0.823	0.832	**0.84**	**0.842**	0.83
letterIJ	0.853	**0.879**	0.853	0.865	**0.874**
letterVY	0.855	0.878	**0.884**	0.861	**0.867**
3vs5	0.856	**0.903**	0.897	0.859	**0.872**
vehicle	0.859	0.878	**0.888**	0.883	**0.89**
5vs8	0.864	**0.907**	0.901	0.85	**0.87**
7vs9	0.876	0.914	**0.921**	0.841	**0.874**
ozone	0.882	0.86	**0.899**	**0.892**	0.882
tictactoe	0.894	**0.912**	0.899	0.853	**0.88**
glass	0.904	**0.914**	**0.914**	**0.917**	0.912
wine	0.906	0.936	**0.943**	**0.94**	**0.939**
letterMN	0.916	**0.944**	**0.941**	0.927	**0.932**
mushrooms	0.931	0.969	**0.974**	**0.971**	**0.972**
letterEF	0.933	0.954	**0.961**	0.956	**0.957**
wdbc	0.938	0.953	**0.956**	**0.958**	**0.957**
letterDP	0.939	0.963	**0.969**	**0.967**	**0.966**
letterUV	0.945	0.972	**0.979**	**0.974**	**0.974**
wisc	0.949	0.951	**0.954**	0.956	**0.956**
breast	0.95	0.956	**0.959**	**0.962**	**0.962**
ac-inflam	0.955	**0.981**	**0.984**	**0.98**	**0.983**
acute	0.978	0.971	**0.984**	**0.992**	**0.992**
Mean	0.798	0.812	**0.822**	0.812	**0.818**
Average Rank	4.143	3.102	**2.388**	2.857	**2.510**
Win/tie/loss counts	-	29/7/13		27/11/11	

4

A Variance Maximization Criterion for Active Learning

Active learning aims to train a classifier as fast as possible with as few labels as possible. The core element in virtually any active learning strategy is the criterion that measures the usefulness of the unlabeled data based on which new points to be labeled are picked. We propose a novel approach which we refer to as maximizing variance for active learning or MVAL for short. MVAL measures the value of unlabeled instances by evaluating the rate of change of output variables caused by changes in the next sample to be queried and its potential labelling. In a sense, this criterion measures how unstable the classifier's output is for the unlabeled data points under perturbations of the training data. MVAL maintains, what we refer to as, retraining information matrices to keep track of these output scores and exploits two kinds of variance to measure the informativeness and representativeness, respectively. By fusing these variances, MVAL is able to select the instances which are both informative and representative. We employ our technique both in combination with logistic regression and support vector machines and demonstrate that MVAL achieves state-of-the-art performance in experiments on a large number of standard benchmark datasets.

This chapter is published as:
Yazhou Yang, and Marco Loog. "*A variance maximization criterion for active learning.*" Pattern Recognition 78 (2018): 358-370.

4.1. INTRODUCTION

In many real-world applications of classification problems, we face the problem that obtaining labels is more difficult than collecting input data: we can easily acquire a large amount of such input data, but labeling these instances is quite burdensome, time-consuming, or expensive [36]. For a large part, this is because of the heavy involvement of human supervision during the labeling process. For example, a hospital produces large amounts of digital images every day, but when categorizing these medical images one often needs to rely on medical doctors with a particular, and therefore expensive, expertise. Hence, it is essential to reduce the need for human annotation, bringing down cost by labeling fewer yet more informative samples. The problem studied in active learning is how to select the most valuable subset and how to measure the value of individual instances or collections of these.

In this work, we focus on, what we refer to as, retraining-based active learning in which one measures the usefulness of particular instances based on all the possible models that are obtained by adding the instances to the labeled dataset and retraining the classifier with the different labels possible [24, 40, 76]. This means that with n unlabeled points and k different classes to choose from, we train nk different classifiers. The key idea behind this is that the value of an unlabeled instance can be estimated by the change it brings to the model when it is queried and used to retrain the model.

Here we propose a new retraining-based active learning method: maximizing variance for active learning (MVAL). Our method selects the instances with maximum retraining variance. This variance stems from the variation presented in the next sample to query and the possible labels those samples can have. The idea is that if the output of an instance changes dramatically, it means that this instance is very susceptible to the variations of input training data. On the other hand, if an instance's output does not vary much, this indicates that the current classifier is very certain about it. A sample with the largest changes in output value is most uncertain and this rate of change can be naturally measured by the variance. Thus, the larger the variance of the output of an unlabeled instance, the higher the uncertainty it has. We propose to keep track of the estimated probability (or decision output) of each unlabeled instance during the retraining procedure. The recorded information is utilized to produce so-called retraining information matrices (RIMs), which are used to calculate the variances for all unlabeled samples. More specifically, two different kinds of variance are computed to measure the informativeness and representativeness. By selecting the instances with maximum variance, MVAL is able to query instances that are both informative and representative. Furthermore, MVAL can be incorporated with both probabilistic and non-probabilistic classifiers, such as logistic regression, Naive Bayes, support vector machines and least squares classifier. In this chapter, we construct the experiments of MVAL with logistic regression and support vector machines.

The remainder is organized as follows. Section 4.2 reviews related work, focussing on retraining-based active learning algorithms. The proposed method is presented in detail in Section 4.3, followed by an extension of the proposed method to multiclass classification problems in Section 4.4. Section 4.5 and 4.6 report the experimental results on binary and multi-class classification problems, respectively. Finally, we conclude this chapter in Section 4.7.

4.2. RELATED WORK

In the past decades, various active learning algorithms, based on many different selection criteria, have been proposed. These approaches rely on different heuristics. We can roughly divide these heuristics into two categories: informativeness and representativeness. Informativeness estimates the ability of an instance in decreasing the uncertainty of a statistical model, while representativeness indicates whether a sample is representative of the underlying distribution [36]. For example, query-by-committee [62], uncertainty sampling [24, 73, 175], error reduction [40, 75], model change [77, 79, 169, 170], expected variance reduction [76] belong to the informativeness category, but each of them has its own criterion of informativeness. Clustering-based approaches [14, 176, 177] and variance minimization methods [52–54, 138] are included in the representativeness group. There are also methods that try to combine the two criteria, such as min-max view active learning [82], density or diversity weighted methods [16, 25, 61, 84, 101] and multi-criteria fusion [21, 46, 178, 179].

The framework of retraining-based active learning, which our method is also an instantiation of, was first proposed by Roy and Mccallum [40] to perform so-called expected error reduction (EER for short). Tong and Koller [24] used a retraining approach in combination with SVMs to find instances that, after labeling, approximately halve the version space. A series of active learning methods which propose a scheme similar to EER, but with somewhat different motivations, were put forward in [75, 76, 79, 180]. All in all, retraining-based active learners can be roughly divided into four categories: error reduction [40, 75], variance reduction [76], model change [79, 84, 126, 169], and min-max view active learning [80, 82]. The principal difference among the above methods lies in how they measure the usefulness of unlabeled samples after retraining the model. For example, error reduce methods like EER [40] attempt to estimate the future generalization error as an indicator of the value of an instance while variance reduction approach [76] turns to use the model variance as a measure of the informativeness. Similarly, model change algorithms seek various ways of defining such change, *e.g.* as gradient length [84], and choose the instance which leads to maximum change. The min-max view active learning directly measures the value of objective function during retraining procedure and selects the instance with minimum score in the worst case scenario. Recently, Yang and Loog [81] proposed to improve the retraining-based algorithms by integrating the uncertainty information in the selection criterion.

We finally note that there exist close relationships between the proposed method and various active learning techniques, such as query-by-committee (QBC) [62], and variance minimization [52–54]. Their connections will be particularly explained in Subsection 4.3.5.

4.3. MAXIMIZING VARIANCE FOR ACTIVE LEARNING

We give a detailed description of the proposed method. We provide the full algorithm and introduce what is at the core of our method: so-called retraining information matrices (RIMs). Based on these RIMs, we introduce the two main types of variance and describe how these are fused into a single criterion for instance selection. In all of this, we focus on probabilistic classifiers. In Subsection 4.3.4, we show one way to adapt our

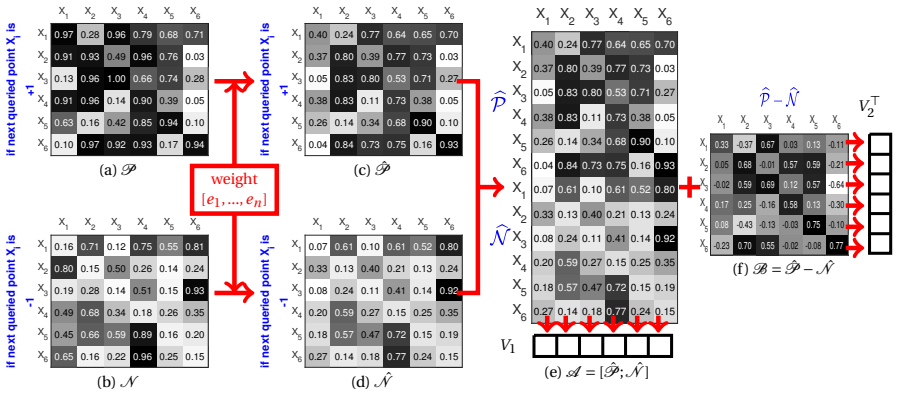


Figure 4.1: An overview of the proposed method MVAL. (a) retraining information matrix $\mathscr{P}$ represents that each of the next queried instance $x_i \in \mathscr{U}$ is labeled +1; (b) $\mathscr{N}$ means that each of the next queried instance $x_i \in \mathscr{U}$ is labeled −1; (c) and (d) are the weighted retraining information matrices of $\mathscr{P}$ and $\mathscr{N}$, respectively, where $[e_1, ..., e_n]$ are the defined weights; (e) and (f) correspond to two matrices $\mathscr{A}$ and $\mathscr{B}$, which are the combinations of $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$. V_1 is the variance of each column in $\mathscr{A}$ while V_2 corresponds to the variance of each row in $\mathscr{B}$. $V_2^\top$ is the transpose of V_2. MVAL fuses V_1 and V_2 to evaluate the usefulness of unlabeled data.

method to a non-probabilistic classifier that does not directly provide a posterior probability estimate. We particularly focus on the SVM, which is the classifiers we are going to experiment with next to logistic regression. In Subsection 4.3.5, we analysis the connections of the proposed method and several existing active learning approaches. First however, we spend a few words on the specific active learning setting we consider.

4.3.1. SPECIFIC SETTING

We study pool-based active learning in which the selection of individual instances to be labeled is sequential and myopic. This means that we assume we already have a large pool of unlabeled data with a small number of labeled data, and a single sample is selected for labeling at a time [36]. We start with the binary classification problem, then present how to extend the proposed method to multiclass tasks in the following section, Section 4.4. We take $\mathscr{U}$ to be the pool of n unlabeled instances $\{x_i\}_{i=1}^n$ and $\mathscr{L}$ to be the already labeled training set, where $y_i = \{+1, -1\}$ is the class label of x_i. $P_{\mathscr{L}}(y|x)$ represents the conditional probability of y given x on the basis of a classifier trained on $\mathscr{L}$.

4.3.2. RETRAINING INFORMATION MATRICES

Figure 4.1 gives a pictorial overview of the proposed method. The proposed method can be used with different types of classifiers. In addition, Algorithm 2 summarizes the overall training procedure of MVAL for probabilistic classifiers. The proposed method generates two matrices $\mathscr{P}$, $\mathscr{N}$, with the purpose of recording the probability of all unlabeled instances after each retraining procedure. We first assume that the next queried instance is labeled as +1, we then extend the current labeled set $\mathscr{L}^+ = \mathscr{L} \cup \{x_i, +1\}$, retrain the classifier on $\mathscr{L}^+$, and calculate the conditional probability $P_{\mathscr{L}^+}(+1|x_j)$ for all

Algorithm 2 Maximizing Variance for Active Learning

1: **Input:** Labeled data $\mathscr{L}$, unlabeled data $\mathscr{U}$
2: **repeat**
3: Train on $\mathscr{L}$ and calculate entropy e_j for all $x_j \in \mathscr{U}$;
4: For each $x_i \in \mathscr{U}$, retrain on $\mathscr{L}^+ = \mathscr{L} \cup \{x_i, +1\}$, let $\mathscr{P}_{i,j} = P_{\mathscr{L}^+}(+1|x_j)$, $x_j \in \mathscr{U}$;
5: For each $x_i \in \mathscr{U}$, retrain on $\mathscr{L}^+ = \mathscr{L} \cup \{x_i, -1\}$, let $\mathscr{N}_{i,j} = P_{\mathscr{L}^+}(+1|x_j)$, $x_j \in \mathscr{U}$;
6: Obtain weighted $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$ and compute the variance using Eq. 4.3;
7: Query the instance x^* with maximum variance and label it y^*, update $\mathscr{L} \leftarrow \mathscr{L} \cup \{x^*, y^*\}, \mathscr{U} \leftarrow \mathscr{U} \backslash \{x^*\}$;
8: **until** Stopping criterion is satisfied

$x_j \in \mathscr{U}$. Each $x_i \in \mathscr{U}$ is used to retrain the model, resulting in a matrix $\mathscr{P}$ of size $n \times n$, where each element (i, j) in $\mathscr{P}$ is assigned $P_{\mathscr{L}^+}(+1|x_j)$. For example, assuming that $\mathscr{U}$ consists of six unlabeled samples $x_i, i = 1,2,\ldots,6$, we could get the matrix $\mathscr{P}$ in Figure 4.1a. Equivalently, if we categorize all of the next queried instances as -1, we retrain the model with $\mathscr{L}^+ = \mathscr{L} \cup \{x_i, -1\}$ for all $x_i \in \mathscr{U}$, we can construct a matrix $\mathscr{N}$ that contains the elements $\mathscr{N}_{i,j} = P_{\mathscr{L}^+}(+1|x_j)$, of which an example is shown in Figure 4.1b.

The matrices $\mathscr{P}$ and $\mathscr{N}$ are the RIMs that collect and preserve the output information over the unlabeled pool during the retraining process. We note here already that since we assign all the elements in the RIMs the value of $P_{\mathscr{L}^+}(+1|x_j)$, for the variance computation it will make no difference if we change the value to $P_{\mathscr{L}^+}(-1|x_j)$ since we are dealing with binary classification problem and $P_{\mathscr{L}^+}(-1|x_j) = 1 - P_{\mathscr{L}^+}(-1|x_j)$.

We subsequently introduce a entropy weighted version of these RIMs, similar to the correction strategy that was presented in [81], which reflects the ideas behind uncertainty sampling [73], in which the selection mechanism is purely based on the current classifier trained on the original $\mathscr{L}$ (rather than retrained on $\mathscr{L}^+$). With this weighting we aim to trade off uncertainty due to instability of an instance and uncertainty due to closeness to the decision boundary. Specifically, we firstly compute the pre-retraining entropy $e_j = -\sum_{y_j \in \pm 1} P_{\mathscr{L}}(y_j|x_j) \log(P_{\mathscr{L}}(y_j|x_j))$, $j = 1,\ldots,n$ and subsequently obtain two weighted matrices $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$, where $\hat{\mathscr{P}}_{i,j} = e_j \times \mathscr{P}_{i,j}$ and $\hat{\mathscr{N}}_{i,j} = e_j \times \mathscr{N}_{i,j}$.

4.3.3. VARIANCE COMPUTATIONS

The two information matrices we compute do not lead directly to a selection criterion that we can determine for each instance. Here it is where we consider particular variances derived from these RIMs. As shown in Figure 4.1, we firstly construct two different matrices by combining $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$. The first one concatenates $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$ column-wise, resulting in a new matrix $\mathscr{A} = [\hat{\mathscr{P}}; \hat{\mathscr{N}}]$ of size $2n \times n$ in Figure 4.1e. We obtain second matrix $\mathscr{B} = \hat{\mathscr{P}} - \hat{\mathscr{N}}$ of size $n \times n$ by subtracting $\hat{\mathscr{N}}$ from $\hat{\mathscr{P}}$, as illustrated in Figure 4.1f.

For matrix $\mathscr{A}$, the column-wise variance is derived to form a vector denoted as V_1, in which the j-th element corresponding to the variance in the j-th column is calculated by

$$V_{1,j} = \frac{1}{2n-1} \sum_{i=1}^{2n} (\mathscr{A}_{i,j} - \frac{1}{n} \sum_{i=1}^{2n} \mathscr{A}_{i,j})^2, \quad j = 1,\ldots,n \tag{4.1}$$

where $\mathscr{A}_{i,j}$ represents the value of element (i,j) in $\mathscr{A}$. In contrast, we compute the row-wise variance for matrix $\mathscr{B}$, which is stored in the vector V_2, i.e., the variance in the i-th row is calculated by

$$V_{2,i} = \frac{1}{n-1}\sum_{j=1}^{n}(\mathscr{B}_{i,j} - \frac{1}{n}\sum_{j=1}^{n}\mathscr{B}_{i,j})^2, \quad i = 1,\ldots,n \tag{4.2}$$

Herein $\mathscr{B}_{i,j}$ is the value of element (i,j) of $\mathscr{B}$.

The reasons for creating the matrices $\mathscr{A}$ and $\mathscr{B}$ and the way of calculating their variances V_1 and V_2 are the following. Firstly, the variance of each column of $\mathscr{A}$ is important since it captures the variations of unlabeled samples when we query a different sample in the next selection or label it a different category. Each column of $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$ represents the scenario that we choose different instances as the next candidate. Concatenating $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$ column-wise like $\mathscr{A}$ indicates that we attach totally contradictory label to the next queried sample. Therefore, V_1, which represents the instability or uncertainty when the next queried sample or its corresponding label changes, is a measure of the informativeness. Secondly, the element (i,j) in $\mathscr{B}$ represents the difference of $P_{\mathscr{L}}(+1|x_j)$ caused by assigning x_i a totally different label. If x_i is representative of x_j, *e.g.*, x_i and x_j come close to each other or belong to the same cluster, element (i,j) in $\hat{\mathscr{P}}$ should vary markedly from (i,j) in $\hat{\mathscr{N}}$ since x_i is labeled differently and the element (i,j) of $\mathscr{B}$ should significantly differ from zero. Hence, the variance of the row of $\mathscr{B}$ indicates the impact of an instance over other unlabeled data when its annotated label varies. V_2 can be seen as a measure of the representativeness. Finally, since the variances are calculated over weighted $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$, both V_1 and V_2 essentially take advantage of the uncertainty information provided by the entropy.

Now we need to fuse V_1 and V_2 to sort the unlabeled data. In this work, we use a simple approach: element-wise multiplication $V_1 \cdot V_2$. We propose the maximizer of this product as our new selection criterion for active learning:

$$\arg\max_{x\in\mathscr{U}} V_1 \cdot V_2 \tag{4.3}$$

Since V_1 and V_2 can measure the informativeness and representativeness, respectively, MVAL is able to select the samples which are both informative and representative.

4.3.4. ADAPTATION TO SVM

For classifiers which do not produce a probabilistic output, we can adapt the proposed method by using their decision values. The particular example we focus on, which will also be used in our experiments, is the SVM. Directly using the decision value $f(x_j)$ as the element of the RIMs leads the variance estimates to be overly sensitive to decision values which may be extremely large or small and empirical experiments indeed show poor results for the above choice. Therefore, similar to the scaling in [173], we are going to transform the decision values into a type of probabilistic outputs. We do not directly rely on Platt scaling, however, because the limited amount of labeled training data, especially in the beginning of active learning, fails to produce stable estimates for these probabilities. Instead, we take a fixed sigmoidal transfer function $(1+\exp(-f(x)))^{-1}$ to transform decision values into probabilities. This sigmoidal transfer corresponds to the

probabilistic output one would obtain if instead of the hinge loss, one would plugs in a logistic loss function that respects the same margin as the original hinge loss.

In order to obtain weighted RIM, $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$, we also need estimate the weight. Instead of firstly transforming the probability and then computing the entropy, we adopt the $e_j = \exp(-|f(x_j)|)$ as the weight, which means that the instance that is nearest to the decision margin receives the highest weight. The proposed method can be easily adapted to other classifiers. In the experimental section, we validate the performance of the proposed method with logistic regression and SVM, respectively.

4.3.5. COMPARISONS AND CONNECTIONS

The proposed method mainly preserves the relevant information during the retraining procedure and creates RIMs to capture the variance of unlabeled samples. Indeed, there exist several connections to other active learning approaches, such as QBC [23, 62, 63], bootstrap-local variance method (BSLV) [71], and variance-minimization approaches [52–54, 76].

MVAL shares a similar idea with QBC but performs slightly different. QBC approaches first constitute a committee of models and then measures the disagreement among the different committee members. Similarly, MVAL can be seen as a specific version of QBC since it also makes use of a number of committee (such as the model re-trained on $\mathscr{L}^+ = \mathscr{L} \cup \{x_i, \pm 1\}$ and estimates the variance as the disagreement. The slight differences lies in: (1) typical QBC algorithms use Gibbs algorithm [62] or re-sampling method such as boosting and bagging [63] to generate the a committee, while MVAL directly utilizes the current training data and one more unlabeled sample with its potential labels. The presence of additional unlabeled samples make the committee more flexible, which can increase the levels of disagreements among committees; (2) QBC normally employs vote entropy or KL divergence [23] to measure the disagreement, whereas MVAL designs two particular variances based on RIMs as the disagreement. And these variances correspond to the informativeness and representativeness, respectively. Therefore, one advantage of MVAL over QBC is that QBC is not able to estimate the representativeness of samples.

MVAL is also different from the BSLV [71], which bootstraps from the already labeled data and calculates the variance of each unlabeled instance. Several differences exist: (1) BSLV uses bootstrap sampling to generate various models; (2) BSLV only calculate a kind of variance which is slightly similar to the V_1 in MVAL; (3) BSLV is not a deterministic selection algorithm since it normalizes the variance as a randomly selection distribution.

There is a major difference between MVAL and several variance-minimization methods such as transductive experimental design (TED) [52], variance reduction [76, 111] and graph-based variance minimization [53, 54]. The sharpest distinction is that MVAL prefers the instance whose individual variance is the largest while these variance-minimization algorithms favour the sample which leads to a minimum variance of a statistical model. For example, experimental design approaches aim to minimize the output variance of some specific statistical models to sequentially reduce the future generalization error. Graph-based methods in [53, 54] focus on the tasks where the graph structure is available without the feature representation. Based on the Gaussian random field classifier, it selects the nodes which minimizes expected prediction variance once labeled. Ex-

pected variance reduction (EVR) [76], which also belongs to the retraining-based active learning, obtains an approximation of the model variance during the retraining process. Unlike experimental design, EVR and graph-based algorithms, MVAL directly estimates the variance of each unlabeled sample introduced by retraining with different training data instead of calculating the model variance. Another dissimilarity is that TED [52] and two graph-based algorithms [53, 54] do not make use of the label information of the queried samples. This means that these methods can not benefit from the feedback information which comes from the human annotator. On the contrary, our method utilizes the label information to update the model in each iteration. As shown in [181], the label information can provide useful hints for active learning. Therefore, these methods in [52–54] is less competitive than the proposed method. We will verify this through empirical experiments in Subsection 4.5.3 (See Table 4.3).

4

4.4. MVAL FOR MULTI-CLASS CLASSIFICATION

In this section, we extend MVAL to multi-class classification problems. A simple approach to addressing this issue is to reduce a multi-class task as multiple binary subtasks using one-vs-all strategy. As Yang *et al.* [101] addressed, however, this may lead to a degradation of the performance of active learners since it is difficult to fuse the results across multiple binary classifiers. We present an alternative approach, which also follows the retraining procedures and keeps record of relevant information. When it comes to the multiclass case, the main challenges are how to generate the RIMs and how to construct the variances.

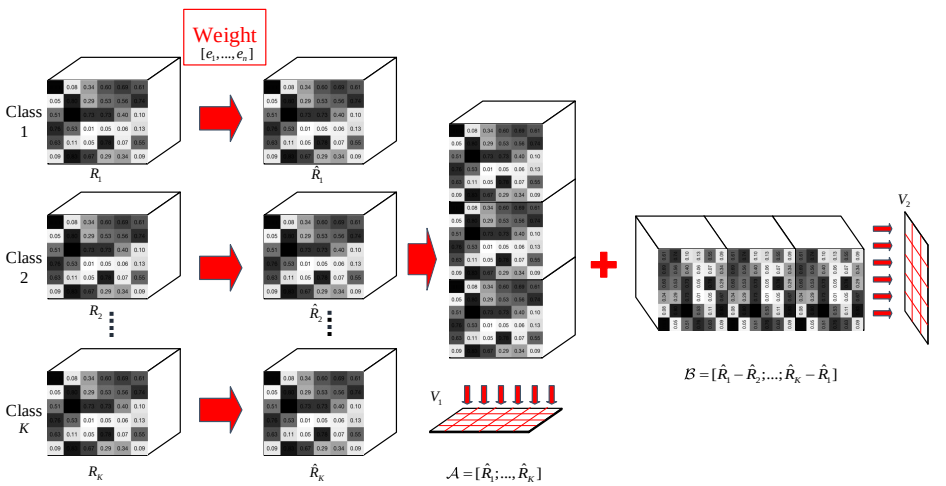


Figure 4.2: An illustration of MVAL for multi-calss classification. R_k and $\hat{R}_k$ are the original and weighted 3D retraining information matrices, respectively, where $k = 1,2,\ldots,K$ and $[e_1,\ldots,e_n]$ are the predefined weights of unlabeled samples. $\mathcal{A}$ and $\mathcal{B}$ are two combinations of $\hat{R}_k$ on which two kinds of variance V_1 and V_2 are computed. MVAL fuses V_1 and V_2 to evaluate the usefulness of unlabeled data.

For binary problem, RIMs are 2D matrices since each element of RIM is a single value $P_{\mathscr{L}^+}(+1|x_j)$. Nevertheless, for a multiclass task of K classes $\{1,2,\ldots,K\}$, we need record

all the posterior probabilities $P_{\mathscr{L}^+}(l|x_j)$, $l = 1,2,\ldots,K$ when the model is retrained. The advantage is that no information is discarded during the retraining process. Thus, we can constitute K different RIMs of size $n \times n \times K$, where n is the number of unlabeled samples and the third references dimension corresponds to the posterior probabilities. These RIMs, which are 3D matrices, are denoted as R_k, $k = 1,2,\ldots,K$. The whole procedure is shown in Figure 4.2. First, the model is retrained by adding each unlabeled instance with pseudo label k, resulting in a RIM R_k. The element (i,j,l) of R_k is the posterior probability $P_{\mathscr{L}\cup\{x_i,k\}}(l|x_j)$, $k,l = 1,2,\ldots,K$, $i,j = 1,2,\ldots,n$. Next, similar to the weighting scheme used in binary case, each R_k is transformed to weighted $\hat{R}_k$. The weighting function we use here is the margin sampling [36], which is equivalent to entropy-based uncertainty sampling in binary case but demonstrates much better performance than entropy-based uncertainty sampling on multi-class tasks [9]. Note that the original criterion of margin sampling is finding the minima of $\min_{x_j}(P_{\mathscr{L}}(\hat{y}_1|x_j) - P_{\mathscr{L}}(\hat{y}_2|x_j))$, where $\hat{y}_1$ and $\hat{y}_2$ are two class labels which rank first and second, respectively based on the model trained on current labeled data $\mathscr{L}$. To be consistent to our objective of querying the samples with maximum variance, we use $exp(-(P_{\mathscr{L}}(\hat{y}_1|x_j) - P_{\mathscr{L}}(\hat{y}_2|x_j)))$ as the weight e_j of sample x_j. More precisely, weighted $\hat{R}_k$ are obtained as $\hat{R}_k(i,j,l) = e_j \times R_k(i,j,l)$.

Finally, we estimate the variance of each unlabeled sample on the basis of these 3D RIMs. As shown in Fig 4.2, two new matrices are constructed as follows: (1) all the weighted $\hat{R}_k$ are concatenated column-wisely to form a matrix $\mathscr{A} = [\hat{R}_1;\hat{R}_2;\ldots;\hat{R}_K]$ of size $nK \times n \times K$; (2) in the binary case, we measure the difference between two RIMs $\hat{\mathscr{P}}$ and $\hat{\mathscr{N}}$ to estimate the representativeness. However, we own K different RIMs instead of two RIMs in the multi-class case. Here we propose to evaluate the differences between all adjacent pairs $\hat{R}_k$ and $\hat{R}_{k+1}$ and concatenate these results row-wisely, resulting in a new matrix $\mathscr{B} = [\hat{R}_1 - \hat{R}_2, \hat{R}_2 - \hat{R}_3, \ldots, \hat{R}_K - \hat{R}_1]$ of size $n \times nK \times K$. An alternative approach, which considers all the paired difference among $\hat{R}_i$ and $\hat{R}_j$, $i,j = 1,2,\ldots,K$, has a heavy computational cost, especially when K is large. Therefore, we only consider the difference of adjacent $\hat{R}_k$ and $\hat{R}_{k+1}$.

Even though a different ordering of the classes will, in principle, lead to a potentially different outcome, preliminary experimental results show that the ordering has a negligible effect on the overall performance of the proposed method.

Similarly to the binary case, the column-wise variance of $\mathscr{A}$ and the row-wise variance of $\mathscr{B}$ are calculated. Note that the $\mathscr{A}$ and $\mathscr{B}$ are 3D matrices, which means that the variances of $\mathscr{A}$ and $\mathscr{B}$ are still 2D matrices. The idea used here is that we first calculate the column-wise variance of $\mathscr{A}$ according to the first dimension and then measure its mean over the third dimension as V_1. On the other hand, V_2 are firstly computed on the second dimension and then averaged on the third dimension. In the end, the final selection criteria of multiclass MVAL is the same with Equation 4.3: the element-wise multiplication of V_1 and V_2. Accordingly, V_1 and V_2 indicate the informativeness and representativeness, respectively.

4.5. EXPERIMENTS WITH BINARY CLASSIFICATION

We empirically compare our proposed method with state-of-the-art active learning algorithms. Extensive results on 45 binary benchmark datasets demonstrate the effective-

ness and robustness of our method. We start by a brief description of the various test datasets. Subsequently, we examine how the proposed method works in comparison with other active learning methods using logistic regression and SVM, respectively.

4.5.1. DATASETS

To evaluate the performance of different active learning algorithms, 45 benchmark datasets are used as the test bed. Some basic information about the datasets after pre-processing is shown in Table 4.1. Many of these datasets are commonly used in other active learning experiments, such as the 20 Newsgroups dataset in [52, 145] and the Letter Recognition dataset in [82]. A large number of datasets originally comes from the UCI Machine Learning Repository [139][1], such as australian, mammographic, vehicle, wdbc and so on. Here, however, we use the preprocessed version such as presented in [142]. Datasets containing more than two classes are converted to binary datasets. Specifically, six pairs of letters from Letter Recognition dataset, *i.e.*, DvsP, EvsF, IvsJ, MvsN, VvsY and UvsV, are constructed as the binary datasets. Seven binary datasets are taken from the 20 Newsgroups dataset [141], which is a commonly used collection for text classification[2]. The first three datasets, baseball vs. hockey, pc vs. mac, and misc vs. atheism, are also used for comparison in [145]. The remaining four datasets, autos, motorcycles, baseball, hockey, are pre-processed according to [52][3]. Since a one-against-all scheme is used to create the above four binary datasets, it represents a case of fairly imbalanced binary classification problems. The MNIST database [140][4] is a commonly used handwritten digit dataset and we construct three paired datasets based on it, *i.e.*, 3vs5, 5vs8 and 7vs9, to test the performance of the different active learners.

For computational efficiency, we apply random sub-sampling and principal component analysis (PCA) on some datasets to reduce both the number of data points and the size of feature dimensionality.

4.5.2. DATA SPLIT AND INITIAL LABELED SET

We randomly split each dataset into training and test set of equal size. We consider a difficult setting for active learning and start out with only two labeled instances at the very beginning. We randomly labeled one example of the positive class and one example of the negative class from the training set. For each active learning algorithm, the experiment is repeated 10 times on each real-world dataset, followed by a report of the average performance. Active learning is terminated when 100 samples are labeled on all of the datasets, except on those datasets that have too few instances to leave a properly sized test set.

4.5.3. RESULTS USING LOGISTIC REGRESSION

Many active learning algorithms are derived using particular classifiers. For example, the simple margin [24] approach was created based on SVM, while QUIRE [82] was developed using ridge regression. In evaluating our active learning method, we benchmark

[1] http://archive.ics.uci.edu/ml/
[2] http://qwone.com/ jason/20Newsgroups/
[3] http://www.dbs.ifi.lmu.de/ yu_k/ted/
[4] http://yann.lecun.com/exdb/mnist/

Table 4.1: Datasets information after pre-processing: the number of instances (# Ins) and the feature dimensionality (# Fea)

Dataset (# Ins, # Fea)	Dataset (# Ins, # Fea)	Dataset (# Ins, # Fea)
fertility (100, 9)	wdbc (569, 31)	3vs5 (1500, 784)
ac-inflam (120, 6)	hill (606, 100)	5vs8 (1500, 784)
acute (120, 6)	breast (683, 10)	7vs9 (1500, 784)
wine (178, 13)	australian (690, 14)	IvsJ (1502, 16)
parkinsons (195, 22)	wisc (699, 9)	EvsF (1543, 16)
sonar (208, 60)	blood (748, 4)	UvsV (1550, 16)
glass (214, 9)	diabetes (768, 8)	MvsN (1575, 16)
hepatitis (155, 19)	pima (768, 8)	VvsY (1577, 16)
heart (270, 13)	ooctris2f (912, 25)	DvsP (1608, 16)
vc2 (310, 6)	tictactoe (958, 9)	pc-mac (1945, 500)
liver (345, 6)	mammographic (961, 5)	base-hockey (1993, 500)
ionosphere (351, 34)	mushrooms (1000, 112)	autos (3970, 8014)
vehicle (435, 18)	ozone (1000, 72)	motorcycles (3970, 8014)
musk1 (476, 166)	splice (1000, 60)	baseball (3970, 8014)
cylinder (512, 35)	misc-atheism (1427, 500)	hockey (3970, 8014)

against methods that either have been designed for the same classifiers or can be easily adapted to the same classifiers. In this subsection, we firstly conduct experiments using active learning algorithms whose base classifier is logistic regression. Subsection 4.5.4 then presents experiments with active learning methods that rely on SVMs.

The following state-of-the-art active learning algorithms based on logistic regression are considered in addition to the standard baseline, *i.e.*, random sampling (RS for short).

- BSLV: Bootstrap-LV algorithm, which bootstraps from the labeled data and estimates the variance as the randomly sampling probability distribution [71];
- US: an uncertainty sampling approach, which queries the example with highest entropy [36, 73];
- EER: Expected Error Reduction, which selects the sample with minimum future generalization error [40];
- UEER: Uncertainty based EER, an improved version of EER using the uncertainty information [81];
- MLI: Minimum Loss Increase, which switches from the square loss of QUIRE [82] to the logistic loss.
- BMDR: Batch mode active learning, which queries discriminative and representative samples. The batch size is set as 1 in this comparison [179].

Table 4.2: Performance comparisons of active learning algorithms in terms of the areas under the learning curve (ALC), based on logistic regression. "Average ALC" reports the average ALC scores over all the datasets. "Average Ranking" shows the average ranking within the compared methods. "Win Times" is the number of datasets on which an algorithm achieving the best or comparable performance. "W/T/L MVAL VS" represents the win/tie/loss counts of MVAL versus the other algorithms over all the datasets. Similarly, "W/T/L VS RS" shows the win/tie/loss counts of other methods versus random sampling.

	RS	BSLV	US	EER	UEER	MLI	BMDR	MVAL
hill	0.583	0.599	0.591	**0.619**	0.592	0.616	**0.622**	**0.621**
cylinder	0.596	0.585	0.597	**0.617**	0.601	**0.616**	0.587	0.602
liver	**0.628**	0.612	0.581	**0.629**	0.606	0.600	0.621	**0.631**
splice	0.651	0.663	**0.672**	**0.672**	0.668	0.644	0.646	**0.671**
ooctris2f	0.686	0.681	0.671	0.688	0.679	0.685	0.666	**0.699**
musk1	0.688	0.690	0.699	0.704	**0.714**	0.703	0.704	**0.713**
sonar	0.688	0.701	0.698	**0.707**	0.691	0.690	0.696	**0.704**
pcmac	0.688	0.710	0.693	0.686	0.677	0.719	**0.722**	**0.724**
religionatheism	0.689	0.701	0.709	0.679	0.686	0.643	**0.714**	0.698
pima	0.704	0.717	0.714	0.700	0.708	0.699	0.685	**0.725**
fertility	0.707	**0.729**	0.705	0.720	0.719	**0.728**	0.665	**0.728**
diabetes	0.708	0.722	0.718	0.727	0.726	0.728	0.711	**0.731**
blood	0.727	**0.738**	0.721	**0.740**	0.705	0.734	0.661	**0.739**
hepatitis	0.727	0.732	**0.773**	**0.768**	0.758	0.742	0.760	**0.767**
heart	0.758	0.789	0.783	0.782	0.782	**0.796**	0.786	0.791
baseball	0.759	0.793	0.850	0.765	**0.871**	0.836	0.781	0.857
autos	0.760	0.793	0.845	0.769	**0.871**	0.839	0.779	**0.866**
motorcycles	0.765	0.798	0.858	0.777	0.883	0.853	0.796	**0.888**
basehockey	0.766	0.784	0.780	0.736	0.749	0.770	0.816	**0.822**
hockey	0.783	0.823	0.886	0.786	0.899	0.872	0.811	**0.911**
mammographic	0.783	0.791	0.770	0.772	**0.801**	0.776	**0.796**	0.793
australian	0.785	0.832	0.839	0.818	0.832	0.839	0.837	**0.848**
ionosphere	0.797	0.801	0.769	**0.823**	0.800	0.666	0.790	**0.822**
parkinsons	0.811	0.819	0.824	0.816	0.818	**0.828**	0.816	0.825
vc2	0.812	0.821	0.813	0.811	0.813	**0.826**	0.794	0.814
letterIJ	0.849	0.859	0.861	0.874	0.824	0.851	0.878	**0.891**
5vs8	0.855	0.877	0.894	0.907	0.906	0.846	0.898	**0.914**
7vs9	0.856	0.891	0.901	**0.918**	**0.916**	0.849	0.909	**0.919**
vehicle	0.858	0.870	0.881	0.871	0.886	0.886	0.877	**0.900**
letterVY	0.864	0.856	0.881	0.880	0.881	0.860	0.880	**0.893**
3vs5	0.866	0.871	0.886	**0.906**	0.898	0.860	0.883	0.902
ozone	0.877	0.875	0.883	0.853	**0.889**	0.872	0.877	**0.887**
tictactoe	0.896	0.893	0.898	**0.907**	**0.905**	0.849	0.875	**0.905**
wine	0.899	0.925	0.923	0.938	0.942	0.936	0.934	**0.948**
glass	0.899	0.908	0.904	0.907	0.912	**0.915**	0.908	**0.913**
letterMN	0.911	0.925	**0.941**	0.941	**0.945**	0.928	0.934	**0.944**
wdbc	0.916	0.952	0.952	0.951	0.953	**0.956**	0.942	0.954
mushrooms	0.931	0.953	0.973	0.969	0.972	0.971	0.957	**0.976**
letterEF	0.934	0.948	0.958	0.954	0.960	0.956	0.953	**0.962**
letterDP	0.938	0.949	0.961	0.962	0.967	0.966	0.952	**0.970**
breast	0.943	0.960	0.960	0.957	0.962	**0.964**	0.950	0.961
ac-inflam	0.947	0.972	**0.984**	0.979	**0.983**	**0.979**	0.966	**0.981**
wisc	0.949	0.951	0.954	0.952	0.956	**0.956**	0.947	0.956
letterUV	0.949	0.962	0.969	0.972	**0.977**	**0.975**	0.964	**0.978**
acute	0.967	0.975	**0.991**	0.955	0.978	**0.993**	0.984	**0.991**
Average ALC	0.803	0.818	0.825	0.819	0.828	0.818	0.816	**0.839**
Average Ranking	6.84	5.11	4.31	4.49	3.71	4.42	5.29	**1.82**
Win Times	1	2	5	11	10	12	4	**34**
W/T/L MVAL VS	44/1/0	37/6/2	36/8/1	31/12/2	29/14/2	32/6/7	41/3/1	-
W/T/L VS RS	-	37/4/4	36/4/5	35/3/7	36/2/7	30/4/11	33/2/10	**44/1/0**

We use the L_2 regularized logistic regression method implemented in the LIBLINEAR package [148] as the classification model for all the algorithms that we compare. Default parameters are used and the penalty parameter C is set to 100 in all the experiments. For BMDR, a trade-off parameter β is used to balance the informativeness and representativeness. We carefully tuned this parameter and set $\beta = 1$ which shows the best average performance over all the datasets. We consider learning curves, which plot the classifier accuracy on test data as a function of the number of labeled training examples. The area under the learning curve (ALC) is then used as the performance measure [149].

The performance of seven active learning approaches based on logistic regression on our 45 datasets are presented in Table 4.2. A paired t-test at a 95% significance level is adopted to evaluate whether two methods are significantly different from each other. For each dataset, the active learning methods which perform the best or are able to compete with the best one are highlighted in bold face and coloured. Some criteria, like average ALC and average ranking, are also reported in Table 4.2. The win/tie/loss counts are also provided based on paired t-test at 95% significance level. All the datasets are sorted in ascending order based on the average ALC scores of random sampling, or in other words, they are sorted from difficult to easy classification tasks from the perspective of logistic regression.

We see that the proposed method achieves the best performance in terms of average ALC and average ranking. MVAL obtains the highest average ALC score 0.839 while the second best one is 0.828 achieved by UEER. The average ranking of MVAL is smaller than 2 and, in most cases, MVAL ranks in the first or second position. There are 34 datasets on which MVAL obtains the best performance or one not significantly different from the best scoring other method. The second best one is MLI on 12 datasets. Generally, MVAL demonstrates highly competitive performance in comparison with other methods over all the datasets, *e.g.* the win/tie/loss counts of MVAL versus the second best one UEER is 29/14/2. And this value of MVAL versus US is 36/8/1. This confirms the effectiveness of the proposed method. We also observe that though US is a quite simplistic approach, it still outperforms some sophisticated methods like BMDR and MLI with regards to average ALC. There are some datasets on which many active learning methods actually lose when compared to random sampling. For example, random sampling outperforms MLI on 11 datasets. It therefore is very interesting to note that MVAL never performs worse than random sampling over all 45 sets and only reaches a tie on 1 of the datasets.

Figure 4.3a presents the average accuracy of the first 30 labeled instances over all the 45 datasets for logistic-based active learning algorithms in. MVAL clearly outperforms other methods, while UEER is a good second, being slightly better than EER and US.

To further investigate the distinction between our variance-maximization method and variance-minimization methods, we also construct experiments to empirically compare their performance. Random sampling (RS) and two graph-based methods V-optimality (V-opt) [53] and Σ-optimality (Σ-opt) [54] are included in this comparison, followed by two experimental design algorithms, TED [52] and Logistic Bound [112]. As shown in Table 4.3, we only report the average performances of compared methods. A detailed description of the performances on each single dataset is included in Table 4.4. Note that we only show the results on 41 binary datasets since there are four relatively large datasets, i.e. autos, motorcycles, baseball and hockey, on which we can not manage to

conduct V-opt and Σ-opt due to high computational cost involved in computing the inverse matrix. Still, we find that the proposed method obtains the best average performance. Logistic Bound also has a very competitive performance and is far better than TED. This is because that (1) Logistic Bound can be seen as a weighted version of TED where the weights are closely related to the entropy of unlabeled samples and (2) Logistic Bound takes into account uncertainty information derived form label information while TED does not utilize this kind of label information. However, our method still outperforms Logistic Bound on 21 datasets and only fails on 7 datasets.

Table 4.3: Performance comparisons of the proposed method versus variance-minimization algorithms on 41 binary datasets.

	RS	V-opt [53]	Σ-opt [54]	TED [52]	Logistic Bound [112]	MVAL
Average ALC	0.807	0.806	0.815	0.810	0.827	**0.834**
Average Ranking	4.93	4.05	3.63	4.49	2.29	**1.61**
Win Times	1	6	6	3	18	**30**
W/T/L MVAL VS	40/1/0	33/7/1	34/3/4	34/6/1	21/13/7	-
W/T/L VS RS	-	27/3/11	28/6/7	20/8/13	35/2/4	**40/1/0**

Let us, for completeness, also report the overall performance of each component of MVAL based on the original RIMs and the weighted RIMs. The average ALC values over all the datasets are provided in Table 4.5. V_1, V_2, and $V_1 \cdot V_2$ represent the different types of variance as introduced in Subsection 4.3.3. The fusion of V_1 and V_2 significantly outperforms each single term on both original RIMs and weighted RIMs based on a paired t-test at a 95% significance level, which demonstrates the advantage of combining the informativeness introduced by V_1 and the representativeness carried by V_2. We observe that the same kind of variance on weighted RIMs markedly exceeds that on original RIMs. For example, a paired t-test shows that the performance of $V_1 \cdot V_2$ on $(\hat{\mathscr{P}}, \hat{\mathscr{N}})$ surpasses that on $(\mathscr{P}, \mathscr{N})$ at a 95% significance level. It is also the same situation for V_1 and V_2. This demonstrates that our proposed weighting scheme is able to enhance the performance of active learners.

4.5.4. RESULTS USING SVM

Support vector machines are a popular classification method used in active learning [24, 80, 154]. Here we compare our method with random sampling and several active learning approaches which are used in combination with SVM. These methods are named as follows:

- SIMPLE: simple margin, which selects the example closest to the decision boundary [24];
- CONF: confidence-based active learning, which estimates the uncertainty by its conditional error [182];
- I-ALSVM: inconsistency-based active learning, which considers two extreme hypotheses and selects instance with highest inconsistency value [183];

Table 4.4: Performance comparisons of the proposed method versus variance-minimization algorithms on 41 binary datasets.

	RS	V-opt	Σ-opt	TED	Logistic Bound	MVAL
hill	0.583	0.600	0.585	0.592	0.588	**0.621**
cylinder	0.596	0.583	**0.612**	0.598	0.603	0.602
liver	**0.628**	**0.630**	0.623	0.621	0.625	**0.631**
splice	0.651	0.612	0.667	0.661	**0.671**	**0.671**
ooctris2f	0.686	0.608	0.647	0.682	0.662	**0.699**
musk1	0.688	0.658	0.684	0.676	0.692	**0.713**
sonar	0.688	0.661	0.669	**0.707**	0.704	**0.704**
pcmac	0.688	**0.730**	0.687	0.679	0.712	0.724
religionatheism	0.689	0.691	**0.708**	0.625	0.638	0.698
fertility	0.694	0.698	0.695	0.671	0.693	**0.717**
pima	0.704	0.699	0.716	0.676	**0.729**	0.725
diabetes	0.708	0.715	0.715	0.702	**0.732**	0.731
blood	0.727	**0.736**	**0.736**	0.701	0.732	**0.739**
hepatitis	0.727	0.655	0.695	0.721	0.746	**0.767**
heart	0.758	0.777	0.770	0.776	0.787	**0.791**
basehockey	0.766	0.815	0.737	0.733	0.785	**0.822**
mammographic	0.783	0.778	0.785	0.767	0.780	**0.793**
australian	0.785	0.802	0.802	0.806	0.830	**0.848**
ionosphere	0.797	0.720	0.798	0.802	**0.820**	**0.822**
parkinsons	0.811	0.823	0.815	0.809	**0.827**	0.825
vc2	0.812	0.811	0.818	0.812	**0.822**	0.814
letterIJ	0.849	0.871	0.872	0.846	0.887	**0.891**
5vs8	0.855	0.892	0.907	0.905	**0.915**	**0.914**
7vs9	0.856	0.903	0.906	0.903	**0.919**	**0.919**
vehicle	0.858	0.885	0.875	0.877	0.896	**0.900**
letterVY	0.864	0.872	0.885	0.846	0.885	**0.893**
3vs5	0.866	0.880	0.896	0.887	**0.904**	0.902
ozone	0.877	0.635	0.827	0.859	**0.900**	0.887
tictactoe	0.896	0.865	**0.922**	0.899	0.889	0.905
wine	0.899	0.930	0.918	0.908	0.944	**0.948**
glass	0.899	**0.911**	0.909	0.905	0.911	**0.913**
letterMN	0.911	0.933	0.936	0.932	**0.945**	**0.944**
wdbc	0.916	0.938	0.933	0.932	**0.955**	**0.954**
mushrooms	0.931	0.965	0.966	0.961	0.975	**0.976**
letterEF	0.934	0.945	0.940	0.939	**0.962**	**0.962**
letterDP	0.938	0.954	0.953	0.957	0.968	**0.970**
breast	0.943	0.962	0.954	0.962	**0.964**	0.961
ac-inflam	0.947	**0.982**	**0.980**	0.974	**0.983**	**0.981**
wisc	0.949	0.953	0.948	**0.955**	**0.956**	**0.956**
letterUV	0.949	0.956	0.953	0.967	**0.976**	**0.978**
acute	0.967	**0.990**	**0.991**	**0.990**	**0.990**	**0.991**

Table 4.5: Average ALC of components of MVAL over all the datasets using logistic regression. V_1, V_2, and $V_1 \cdot V_2$ represent the different types of variance. $(\mathscr{P}, \mathscr{N})$ and $(\hat{\mathscr{P}}, \hat{\mathscr{N}})$ represent the original RIMs and the weighted RIMs, respectively.

RIMs \ variance	V_1	V_2	$V_1 \cdot V_2$
$(\mathscr{P}, \mathscr{N})$	0.827	0.786	0.833
$(\hat{\mathscr{P}}, \hat{\mathscr{N}})$	0.831	0.815	0.839

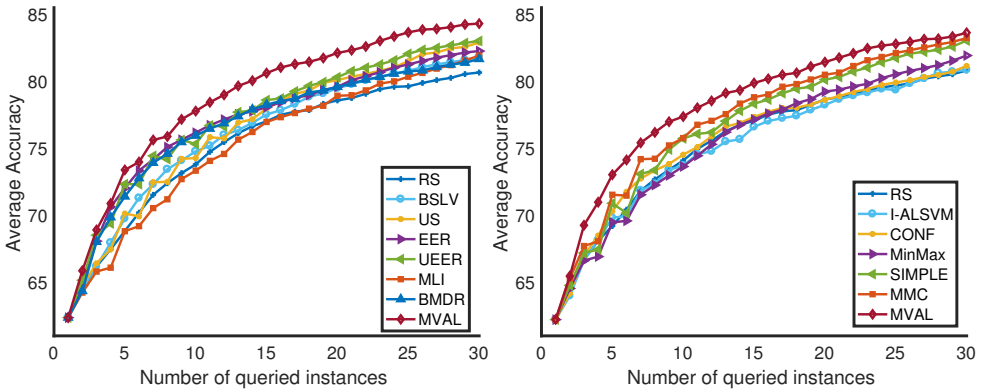


Figure 4.3: Average accuracy of the first 30 labeled examples over all the datasets. (a) shows the average performance of active learning methods based on logistic regression; (b) demonstrates the average result of active learning methods based on SVM.

- MMC: maximum model change, an adaptive version of simple margin. It selects the instance close to the decision boundary but also considers about its contribution to model change [169];
- MinMax: min-max view active learning, a new version of QUIRE [82], but uses the hinge loss instead;

For all the methods, we use linear SVM from the LIBSVM package [143] as classifier. The regularization parameter C is set to 10 in all the experiments. As in the previous subsection, we use the area under the learning curve (ALC) as the performance measure. Like for the hyper-parameters of the base classifiers, there typically are no additional labeled validation data available for tuning any hyper-parameters an active learning scheme might have. We empirically tuned these parameters over all the datasets to globally good working choices. For CONF, an uncertain threshold c and bin size $nBin$ are needed. The resulting parameters we found were $c = 0.5$ and $nBin = 4$. For MMC, a hyper-parameter γ is used to filter the instances within the margin. We validated this value from a candidate set and selected the one which presents the best overall performance. Finally, we set $\gamma = 0.01$ in our experiment. For I-ALSVM, we used the modified

version of I-ALSVM which outperforms original I-ALSVM by combining I-ALSVM and the simple margin method [183]. The modified version first selects a small candidate set based on original I-ALSVM, then chooses the instance which is closest to the decision boundary from the above subset. We tuned the size of this subset and finally this parameter was set 16 in our experiment. We note that the proposed method MVAL does not need to tune additional hyper-parameters.

As shown on the right side of Table 4.6, the proposed method also in this setting achieves the overall best performance. MVAL obtains 0.834 in terms of average ALC and performs best or at the best level on 31 datasets. The second best algorithm, MMC, only performs well on 16 datasets. Also here we used a paired t-test at 95% significance level to evaluate the scores of ALC over all 45 datasets and we can conclude that MVAL significantly outperforms other approaches. The win/tie/loss counts of MVAL versus other methods also demonstrate that MVAL compares favorably to all other methods. We also note that MMC and the simple margin outperform other active learning methods except MVAL. CONF and I-ALSVM perform slightly better than random sampling. The possible reason might be that their hyper-parameters need to be tuned very on each dataset. We plot the average performance of the first 30 annotated examples in Figure 4.3b. Also looking at the performance in this way, we see that MVAL performs better than other algorithms, especially in the early stage of active learning. MMC slightly outperforms SIMPLE and I-ALSVM performs similarly to random sampling.

4.6. EXPERIMENTS WITH MULTI-CLASS CLASSIFICATION

We present the experimental results on multi-class classification problems in this section. Since many of the compared active learning algorithms using SVM are only designed for binary case and it is not clear how to extend them to multi-class problems, we only compare the proposed method with active learning algorithms that are derived on the basis of logistic regression.

We use 12 UCI benchmark datasets and 8 real-word datasets as the test bed. For some relatively large datasets such as MNIST, scene13 [184], GTSRB [185] and CIFAR10 [186], we use randomly sub-sampling to reduce their sizes. The datasets information after sub-sampling and PCA is listed in Table 4.7. For the scene13 dataset, we use the GIST feature [187]; for the CIFAR10 dataset and GTSRB dataset, HOG feature [188] are extracted. With regards to the action recognition datasets, KTH [189] and UCFsports [190], we use the pre-extracted Action Bank features [191]. The Isolet is a letter speech recognition dataset [192]. TWSA03 is a player action recognition data set in tennis games taken from [193], of which HOG3D descriptors are extracted according to [194].

The experiments are repeated 10 times on each datasets and average performances are reported. As the initial training set, we randomly select one instance from each class. For the logistic regression classifier, the same setting is used as that in Section 4.5.3. Due to that BSLV, Logistic Bound, BMDR are specifically designed for binary tasks, they are omitted for comparison. The proposed method MVAL is compared with the remaining active learning algorithms.

As is shown in Table 4.8, MVAL consistently outperforms other active learning methods over 14 datasets, it achieves the best performance or behaves comparably to the best algorithms. Though it fails on 6 datasets such as CIFAR10, MNIST and dermatology, it

is never the worst one. This can demonstrate the advantages of MVAL, efficient and robust. We also observe that MLI totally fails on most of datasets and performs worse than random sampling. The probable reason may be that the min-max view used in [82] is not suitable for multi-class classification problems. The error reduction method EER achieves the second best scores while MVAL still outperforms it on 17 datasets based on paired t-test at a 95% significance level. Three variance-minimization approaches, V-opt, Σ-opt and TED, perform better than random sampling. However, they are still worse than the proposed method, e.g. the win/tie/loss of MVAL versus Σ-opt is 17/1/2.

4.7. DISCUSSION AND CONCLUSION

We proposed a novel active learning method called MVAL, which is based on the retraining-based active learning framework. MVAL builds weighted retraining information matrices (RIMs) to record the changes of the output of unlabeled data during the retraining process. Two types of variance based on these RIMs are calculated and fused to evaluate the combined informativeness and representativeness of unlabeled samples. MVAL then selects the instance with the largest combined variance. As an example, we demonstrated how to use MVAL both with logistic regression and support vector machines. Furthermore, an extension of MVAL to multi-class classification task is also presented in this chapter. Empirical results on both binary and multi-class datasets show excellent performance of our method in comparison with current state-of-the-art active learning methods.

We see two different extension of our approach as potentially interesting for future research. First of all, currently, MVAL is only feasible for myopic active learning setting. Like for many other active learning approaches, it may be interesting to investigate how to extend this idea to batch mode active learning, which queries a set of unlabeled examples simultaneously. Secondly, if there is one drawback our method has, it is the computational cost. It is not a problem that only our method has: MVAL actually has the same computational complexity as some of the state-of-the-art retraining-based methods that we compared to, namely EER [40], UEER [81], and MLI [82]. For some simple active learning methods, such as uncertainty sampling, a proper acceleration can be achieved by hyperplane hashing [195]. For our method and other retraining-based approaches, a feasible solution is to use parallel computing to improve the efficiency since retraining the classifier with different $x_i \in \mathcal{U}$ is independent of each other. Another direction to speed up these methods is using various heuristic approximations (e.g. a warm start in [75] and nearly zero assumption of the gradient of objective function in [84]) and subsampling strategies (e.g. selecting a subset of samples with maximum entropy [196]).

More important than the extension to the batch setting and the computational speed is that we at all have a criterion that can give us good active learning performance. With the current work, we have made an additional step in this direction, clearly improving upon current state of the art.

Table 4.6: Performance comparisons of active learning algorithms in terms of the areas under the learning curve (ALC) based on SVM. "Average ALC" reports the average ALC scores over all the datasets. "Average Ranking" shows the average ranking within the compared methods. "Win Times" is the number of datasets on which an algorithm achieving the best or comparable performance. "W/T/L MVAL VS" represents the win/tie/loss counts of MVAL versus the other algorithms over all the datasets. Similarly, "W/T/L VS RS" shows the win/tie/loss counts of other methods versus random sampling.

	RS	I-ALSVM	CONF	MinMax	SIMPLE	MMC	MVAL
hill	0.534	0.549	0.550	0.578	0.580	0.581	**0.587**
liver	0.599	**0.622**	0.613	0.611	**0.622**	0.612	0.609
cylinder	0.602	0.592	0.603	0.631	0.608	**0.637**	**0.641**
splice	0.658	0.667	0.658	0.630	**0.682**	0.670	0.670
religionatheism	0.673	**0.677**	0.673	0.641	0.673	0.662	0.647
ooctris2f	**0.682**	0.664	**0.681**	0.638	0.662	0.654	0.680
musk1	0.687	0.691	0.687	0.700	0.680	0.662	**0.703**
pcmac	0.690	0.679	0.690	0.674	0.680	**0.711**	0.675
pima	0.705	0.701	0.702	**0.727**	0.714	**0.726**	0.713
sonar	0.705	0.725	0.708	0.710	0.720	0.714	**0.731**
diabetes	0.715	0.687	0.717	0.726	0.720	0.722	**0.736**
fertility	0.729	**0.752**	0.733	0.738	0.757	**0.760**	**0.752**
basehockey	0.730	0.751	0.730	0.706	0.743	**0.765**	0.722
blood	0.732	0.725	0.732	**0.747**	0.726	0.735	0.740
hepatitis	0.750	0.759	0.755	0.760	0.776	**0.779**	**0.771**
heart	0.756	0.777	0.776	0.785	0.775	0.783	**0.790**
baseball	0.768	0.838	0.766	0.842	0.850	**0.867**	0.859
autos	0.771	0.836	0.775	0.844	0.857	0.852	**0.869**
motorcycles	0.776	0.849	0.777	0.861	0.862	0.870	**0.884**
mammographic	0.784	0.768	0.790	0.782	0.795	**0.803**	0.791
ionosphere	0.791	0.779	0.793	0.694	0.796	0.793	**0.811**
australian	0.793	0.801	0.819	**0.844**	0.835	0.832	0.838
hockey	0.798	0.878	0.797	0.888	0.880	**0.899**	**0.898**
vc2	0.803	0.779	0.814	0.822	0.793	0.811	**0.828**
parkinsons	0.824	0.832	0.829	0.835	**0.845**	**0.845**	0.835
letterIJ	0.847	0.787	0.863	0.867	0.868	0.879	**0.891**
vehicle	0.857	0.845	0.864	0.877	0.881	0.877	**0.887**
7vs9	0.858	0.883	0.869	0.850	0.901	0.907	**0.918**
5vs8	0.859	0.883	0.876	0.854	0.891	0.888	**0.910**
letterVY	0.860	0.778	0.867	0.856	0.868	0.876	**0.882**
3vs5	0.864	0.871	0.859	0.857	0.884	0.880	**0.895**
glass	0.897	0.903	0.895	**0.907**	0.902	0.903	**0.909**
wine	0.897	0.904	0.898	0.926	0.932	**0.930**	**0.939**
tictactoe	0.904	0.848	**0.908**	0.870	0.894	0.894	**0.912**
letterMN	0.912	0.872	0.912	0.927	0.934	0.934	**0.947**
wdbc	0.918	0.945	0.925	0.958	0.956	0.957	**0.961**
letterEF	0.926	0.921	0.927	0.956	0.956	**0.959**	**0.960**
ozone	0.928	0.942	0.928	0.930	0.937	0.934	**0.945**
mushrooms	0.931	0.968	0.930	0.964	0.970	0.970	**0.973**
letterDP	0.935	0.917	0.935	0.964	0.959	0.964	**0.967**
ac-inflam	0.942	0.943	0.942	**0.977**	**0.979**	**0.979**	**0.975**
wisc	0.944	0.936	0.940	**0.953**	0.951	**0.953**	0.950
breast	0.947	0.955	0.956	**0.964**	0.963	0.963	0.963
acute	0.949	0.955	0.949	**0.990**	**0.988**	**0.989**	**0.984**
letterUV	0.949	0.939	0.950	**0.976**	0.974	**0.977**	**0.979**
Average ALC	0.804	0.808	0.808	0.819	0.827	0.830	**0.834**
Average Ranking	5.64	5.11	5.22	4.04	3.16	2.69	**2.13**
Win Times	1	3	2	9	5	16	**31**
W/T/L MVAL VS	41/0/4	39/2/4	38/3/4	32/8/5	31/6/8	26/10/9	-
W/T/L VS RS	-	24/3/18	18/23/4	30/5/10	38/1/6	38/4/3	**41/0/4**

Table 4.7: Multi-class datasets information after pre-processing: the number of instances (# Ins), the feature dimensionality (# Fea) and class number (#C)

Dataset (#Ins, #Fea, #C)	Dataset (#Ins, #Fea, #C)	Dataset (#Ins, #Fea, #C)
car (900, 6, 4)	led_display (1000, 7, 10)	heart_cleveland(303, 13, 5)
contrac (1473, 9, 3)	pendigits (1000, 16, 10)	satimage (1000, 36, 6)
segment (1000, 19, 7)	stvehicle (846, 18, 4)	glass (214, 9, 6)
dermatology (366, 34, 6)	vowel (990, 10, 11)	USPS (1000, 60, 10)
MNIST(1000, 60, 10)	scene13 (1000, 90,13)	CIFAR10 (1000, 57,10)
KTH (599, 100, 6)	UCFsports(140, 100,10)	TWSA03 (1228,100, 3)
GTSRB (1000, 40, 20)	Isolet(1040, 40, 26)	

Table 4.8: Performance comparisons of active learning algorithms on 20 multiclass datasets. "Average ALC" reports the average ALC scores over all the datasets. "Average Ranking" shows the average ranking within the compared methods. "Win Times" is the number of datasets on which an algorithm achieving the best or comparable performance. "W/T/L MVAL VS" represents the win/tie/loss counts of MVAL versus the other algorithms over all the datasets. Similarly, "W/T/L VS RS" shows the win/tie/loss counts of other methods versus random sampling.

	RS	US	EER	UEER	MLI	V-opt	Σ-opt	TED	MVAL
CIFAR10	0.257	0.253	**0.270**	0.261	0.240	0.249	**0.269**	0.253	0.256
vowel	0.378	0.374	0.391	0.388	0.373	0.401	0.385	0.381	**0.413**
contrac	0.434	**0.444**	0.440	0.437	0.393	**0.443**	**0.441**	**0.446**	0.443
scene13	0.471	0.418	0.500	0.443	0.420	0.487	0.465	0.476	**0.504**
heart_cleveland	0.501	0.522	0.521	0.527	0.514	0.507	0.517	0.512	**0.531**
glass	0.521	**0.542**	0.535	0.526	0.520	0.491	**0.549**	0.473	0.539
GTSRB	0.628	0.621	0.669	0.644	0.674	0.643	0.664	0.677	**0.681**
MNIST	0.628	0.627	**0.709**	0.649	0.587	0.685	0.692	0.674	0.700
Isolet	0.629	0.631	0.645	0.631	0.637	0.592	0.651	0.654	**0.659**
led_display	0.633	**0.662**	0.653	0.659	0.542	0.640	0.641	0.632	**0.663**
stvehicle	0.652	0.664	0.668	0.675	0.631	0.643	0.659	0.662	**0.680**
car	0.694	0.730	0.725	**0.735**	0.627	0.727	0.729	0.710	**0.734**
pendigits	0.752	0.766	0.770	0.760	0.733	0.734	0.735	0.766	**0.786**
satimage	0.763	0.746	0.766	0.764	0.760	0.759	0.753	0.745	**0.793**
USPS	0.769	0.797	**0.816**	0.802	0.777	0.804	0.812	0.798	**0.817**
UCFsports	0.769	0.769	0.758	0.770	0.788	0.766	0.766	**0.797**	0.775
TWSA03	0.775	0.795	0.789	0.803	0.787	0.764	0.799	0.794	**0.811**
segment	0.809	0.810	0.828	0.850	0.794	0.842	0.846	0.820	**0.865**
KTH	0.918	0.951	0.936	0.948	0.932	0.927	0.927	0.941	**0.953**
dermatology	0.940	0.945	0.940	**0.952**	0.936	0.913	0.925	0.939	0.950
Average ALC	0.646	0.653	0.667	0.661	0.633	0.651	0.661	0.657	**0.678**
Average Ranking	6.9	5.15	3.85	3.85	7.25	6.3	4.85	5.2	**1.65**
Win Times	0	3	3	2	0	1	3	2	**14**
W/T/L MVAL VS	19/1/0	16/4/0	17/1/2	17/1/2	19/0/1	19/1/0	17/1/2	18/0/2	-
W/T/L VS RS	-	12/3/5	18/1/1	17/2/1	7/3/10	10/1/9	15/0/5	14/3/3	**19/1/0**

5

SINGLE SHOT ACTIVE LEARNING USING PSEUDO ANNOTATORS

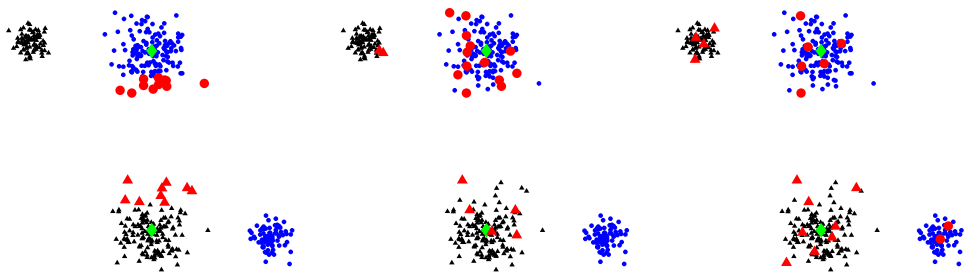
Standard myopic active learning assumes that human annotations are always obtainable whenever new samples are selected. This, however, is unrealistic in many real-world applications where human experts are not readily available at all times. In this chapter, we consider the single shot setting: all the required samples should be chosen in a single shot and no human annotation can be exploited during the selection process. We propose a new method, Active Learning through Random Labeling (ALRL), which substitutes single human annotator for multiple, what we will refer to as, pseudo annotators. These pseudo annotators always provide uniform and random labels whenever new unlabeled samples are queried. This random labeling enables standard active learning algorithms to also exhibit the exploratory behavior needed for single shot active learning. The exploratory behavior is further enhanced by selecting the most representative sample via minimizing nearest neighbor distance between unlabeled samples and queried samples. Experiments on real-world datasets demonstrate that the proposed method outperforms several state-of-the-art approaches.

This chapter is in preparation for publication.

5.1. INTRODUCTION

In many machine learning applications, the availability of a large amount of data offers opportunities to boost the prediction performance. Even if data is abundant, a major issue remaining is that labeling the data is usually time-consuming and expensive. For example, it is costly to hire many dermatologists to annotate the 129,450 clinical images of skin cancer used in [197]. Active learning, which iteratively selects the most informative samples and queries the labels from human experts, has demonstrated its ability to reduce the annotation cost and maintain good learning performance in various applications [36, 48].

The strength of active learning in reducing annotation cost stems from the fact that it can iteratively query its preferred unlabelled examples for labelling and simultaneously update its selection strategy according to the feedback from a human expert. Indeed, conventional active learning assumes a human in the loop such that it can iteratively learn from the received label information. This also implies that in the classical setting of active learning, human annotators should be always readily available for labeling whenever new unlabeled samples are queried. However, this assumption may not hold in some real-world applications since (1) human annotator is unlikely to be present at all time, e.g. human annotator may get tired or need a rest, (2) and active learning process has to be suspended until the annotator reappear.



(a) uncertainty sampling

(b) our method with a pseudo annotator

(c) our method with multiple pseudo annotators

Figure 5.1: An illustration of different active learning algorithms in the single shot setting. Blue and black points represent two different classes. Two green diamond points represent a set of labeled samples that is often of small size while the remaining points represent a large pool of unlabeled samples. These figures show the (red) queried points chosen by (a) standard uncertainty sampling (i.e. maximum entropy [36, 73]), (b) uncertainty sampling + random labeling (with a single pseudo annotator) and (c) uncertainty sampling + random labeling (with multiple pseudo annotators).

To mitigate the issue of human annotators being missing in the loop, we consider a single shot setting of pool-based active learning, where few labeled samples and a potentially large number of unlabeled instances are available and the active learner is asked to choose a query set $\mathcal{Q}$ in a single shot [121]. Simply using standard myopic active learning algorithms to select the top-ranked samples is not a good choice since it fails to consider the redundancy among these top instances [36]. Figure 5.1a shows an example of the failure of standard uncertainty sampling [36, 73]. We can observe that in the single shot scenario, uncertainty sampling chooses a subset of samples which overlap each other.

And it fails to explore the other two clusters.

In this chapter, we concentrate on adapting standard active learning algorithms, which need real label information for exploitation, to the single shot setting. We propose a new method, called Active Learning through Random Labeling (ALRL). Our method introduces multiple annotators, which we refer to as pseudo annotators, to take the place of the human expert used in active learning cycle. The pseudo annotators are independent from each other and present uniformly random labels whenever new unlabeled samples are queried. Even though the pseudo annotators do not add any information telling us anything about the true labels, regular active learning methods can still benefit from receiving such random labels. The improvement comes from the exploration ability provided by this random labeling mechanism. As we will see, the exploratory behavior is further enhanced by fine-tuning the results that come from multiple pseudo annotators by selecting the sample that well-represents the unlabeled data. The proposed method is a general approach, which can be incorporated with both simple active learners, e.g. uncertainty sampling [73] and sophisticated ones, e.g. variance maximization [125]. We show the efficiency of our method on real-world datasets, in comparison with state-of-the-art approaches.

5.2. RELATED WORK

A brief review of the work related to our single shot active learning is given, including myopic and batch mode active learning, optimal experimental design, data subset selection, and single shot selection.

Myopic vs. batch. Active learning can be roughly divided into two categories according to the number of queried samples at a time [172]. The first one is myopic active learning, where only a single instance is selected in each iteration. Many well-known algorithms, such as query-by-committee (QBC) [62], uncertainty sampling [24, 73], error reduction [40], maximum model change [79], variance reduction [76], and variance maximization [125] belong to this group.

The second one is batch mode active learning, where a batch of samples is selected simultaneously [13, 44, 107, 132]. Conventional batch mode active learning methods first select a fixed number of unlabeled instances and then ask for the real labels from human experts. Subsequently, they make use of the received real label information to update their selection criteria and continue choosing the next round of a group of unlabeled samples. When the batch size is very large, e.g. the number of required samples in total, batch mode setting is transformed into the considered single shot case. In other words, single shot active learning can be viewed as a particular case of batch mode active learning with the batch size being equal to the sampling budget. However, Brinker [61] found that it is preferable to set batch size as small as possible, with the possible reason that the selection criterion can be updated more frequently if batch size is small. This implies that directly using batch mode active learner for one single shot setting may lead to a decrease of the learning performance.

Optimal experimental design. There also exist some active learning methods which do not require true labels for samples selection at all, such as transductive experimental design (TED) [52] and graph-based variance minimization methods [53, 54]. These approaches usually attempt to minimize the expected variance of a statistical model, where

the label information is omitted in the calculation of such variance. Hence, they are well matched with this single shot setting since no human annotation is needed during the selection process. However, since they do not utilize label information, they mainly focus on selecting representative samples and do not make use of label information even when some labels of samples are available.

Data subset selection. Data subset selection is also closely related to our single shot setting since both of them aim to select an informative subset. The subset selection problem has been well studied in the literature. When the input samples are feature vectors, many works focus on finding representative samples by searching in low-dimensional subspace [198, 199] or using some clustering algorithms, e.g. k-means. Other efforts have been devoted to finding the representatives by using pairwise similarities between data points [55, 200]. For example, Elhamifar *et al.* [55] proposed a dissimilarity-based sparse subset selection (DS3) method to minimize the difference between source data and target data. However, some methods, e.g. DS3 and the work in [200], cannot exactly determine the number of selected representative points beforehand.

Single shot selection. Few efforts have been devoted to single shot active learning problem. Contardo *et al.* [121] combined meta-learning and active learning to learn an active learning strategy. After the selection strategy is learned, all the required samples are queried in a single shot. However, their method required additional supervised data to train their model, which is not realistic in many active learning applications since only little or even no labeled data is available. Our proposed method does not require extra supervision information.

5.3. ACTIVE LEARNING USING RANDOM LABELING

This section presents our novel approach for querying an informative subset for human annotation in a single shot in detail. The proposed method offers an alternative view of the subset selection problem, which adapts standard myopic active learners to the single shot setting through random labeling.

5.3.1. MOTIVATION

Many active learning algorithms, which make use of label information for samples selection, focus on exploitation by querying samples near the decision boundary to refine the classification model, e.g. uncertainty sampling. On the contrary, some other methods concentrate on the exploration by selecting the most representatives of the unlabeled instances. Most of these approaches, e.g. Transductive experimental design [52] and Hessian optimal design [138], do not use the class information of selected samples at all.

In the single shot setting, in the absence of experts annotations, regular exploitation-based active learning algorithms may fail because they cannot update their exploitation criterion for every sample selected. Conducting the exploitation on the basis of initial training data without further updating the selection criterion is likely to mislead the active learner to select uninformative and redundant samples. Take uncertainty sampling for example, it chooses the instances for which the current classifier is least cer-

tain. For probabilistic classifiers, a general uncertainty sampling strategy is selecting the instances with maximum entropy [36]: $x^* = \arg\max_x -\sum_i \hat{P}(y_i|x) \log \hat{P}(y_i|x)$, where $\hat{P}(y_i|x)$ is the estimated posterior probability based on current labeled data and y_i goes over all candidate labels. As is illustrated in Figure 5.1a, uncertainty sampling, which concentrates on exploitation, queries less informative samples in a single shot. And there are high redundancies among these queried samples. It shows that, in general, pure exploitation without subsequent updating can indeed be harmful in the single shot setting.

To overcome the disadvantage of pure exploitation, we enable standard myopic active learners to explore by using multiple of our so-called pseudo annotators. The value of employing such pseudo annotators will be explained in Subsection 5.3.2. First, however, we explain more precisely what a pseudo annotator does. A pseudo annotator does not know anything about the true labels and just randomly guesses a class category when annotating an unlabeled instance. For example, given that $C = \{c_1, c_2, \ldots, c_p\}$ is the set of possible labels, the pseudo annotator randomly and uniformly selects one label c_i from C for each queried unlabeled instance. This also implies that the randomly assigned labels of different unlabeled samples are totally independent from each other.

5.3.2. THE PROPOSED METHOD: RANDOM LABELING

We start with the basic setting of single shot active learning. We have relatively little labeled data $\mathscr{L} = \{(x_i, y_i)\}_{i=1}^{n_l}$, where $x_i \in \mathbb{R}^d$ is a feature vector and y_i is the label of x_i. In addition, a large pool of unlabeled examples $\mathscr{U}$ is also available. The task is to select a budget of N samples from the unlabeled pool $\mathscr{U}$ in a single shot. When N instances are determined, they are categorized by human annotators and added to the labeled data $\mathscr{L}$. The way to construct such an initial labeled set $\mathscr{L}$ is not widely studied in the literature. Like in [41, 101, 208], this work simply assumes that $\mathscr{L}$ is obtained by random sub-sampling, for instance, by randomly selecting a fixed number of instances per category from all the data.

Figure 5.2 shows the flow chart of our proposed method. $\mathscr{Q}$ denotes the already selected but still unlabeled data. Our method chooses N samples in a sequential way, which means only one sample is queried at a time.

To start with, we set $\mathscr{Q} = \emptyset$ and select the first instance according to the myopic active learner $\mathscr{A}$ trained on $\mathscr{L}$. From then on, the instances in $\mathscr{Q}$ are randomly labeled by the pseudo annotators and used to select the next sample. Trained on correctly labeled $\mathscr{L}$ and randomly labeled $\mathscr{Q}$, the active learner we use is able to explore larger regions than that only trained on $\mathscr{L}$. As shown in Figure 5.1b, uncertainty sampling is still used as the active learner but our method can make it select diverse samples instead of purely those samples lying close to initial decision boundary. This verifies that random labeling can indeed help exploration. Note that in each iteration, all the samples in $\mathscr{Q}$ are relabeled by the pseudo annotators, which means that the assigned labels may be different from that obtained in the last round.

A major difference between our random labeling and standard active learning algorithms is that method employs multiple pseudo annotators whereas classical active learning assumes that only a single annotator is available. The motivation of using multiple pseudo annotators is that a single random labeling strategy would make the out-

put of our algorithm too depend on the quality of randomly assigned labels. It could happen that our method unfortunately queries an uninformative sample because of the poor random labels. To tackle this problem, we simply decide to use m different pseudo annotators, i.e. $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_m\}$, and fine-tune the result by selecting the most representative instance obtained by using multiple pseudo annotators. Note that these pseudo annotators are independent from each other.

More specifically, we first use $\mathcal{O}_i$ to randomly label all the samples in $\mathcal{Q}$ and then apply an active learner $\mathcal{A}$ to choose one candidate s_i from unlabeled data $\mathcal{U}$ based on truly labeled $\mathcal{L}$ and randomly labeled $\mathcal{Q}$. We repeat this procedure m times with different pseudo annotators, result in obtaining m different candidates, i.e. $\{s_1, s_2, \ldots, s_m\}$, since each pseudo annotator is highly likely to present different random labels to $\mathcal{Q}$. Subsequently, we can evaluate the candidate samples and select the one which best represents the unlabeled samples.

Overall, as shown in Figure 5.2, the proposed random labeling mechanism uses a two-step strategy: it first employs multiple pseudo annotators to impel regular active learner to explore and choose an informative candidate set; then the most representative

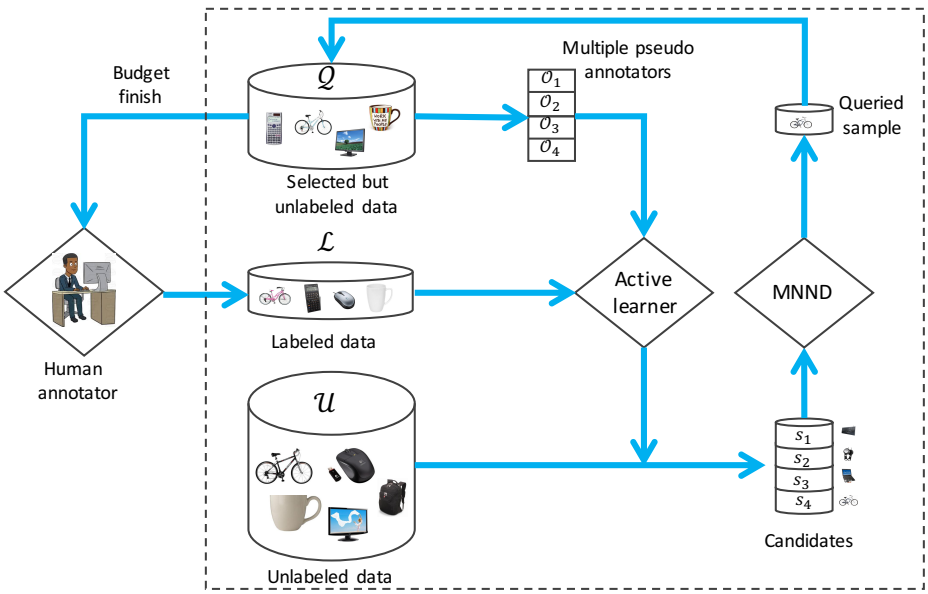


Figure 5.2: Schematic illustration of the proposed method. $\mathcal{L}$ and $\mathcal{U}$ denote the truly labeled data and unlabeled data, respectively. $\mathcal{Q}$ represents those already selected but still unlabeled data. $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_m\}$ ($m = 4$ in this figure) denotes a set of pseudo annotators where each pseudo annotator $\mathcal{O}_i$ always randomly and uniformly label these samples in $\mathcal{Q}$. $\mathcal{Q}$ starts with an empty set, and adds the first sample which is selected by an active learner trained on $\mathcal{L}$ only. $\{s_1, \ldots, s_m\}$ are the candidates selected by an active learner based on each pseudo annotator. MNND stands for minimizing nearest neighbor distance between unlabeled data and already queried data (see Equation 5.2). In the end, the most representative sample from $\{s_1, \ldots, s_m\}$ is chosen by MNND and is added to $\mathcal{Q}$. Only when budget is finished will the samples in $\mathcal{Q}$ be labeled by human annotator.

Algorithm 3 Active Learning with Random Labeling

Require: Labeled data $\mathscr{L}$, unlabeled data $\mathscr{U}$, subset $\mathscr{Q} = \emptyset$, Active Learner $\mathscr{A}$, pseudo annotators $\{\mathscr{O}_1, \mathscr{O}_2, \ldots, \mathscr{O}_m\}$

1: Train on $\mathscr{L}$ and select the first sample x^* from $\mathscr{U}$ according to active learner $\mathscr{A}$; update $\mathscr{Q} \leftarrow \mathscr{Q} \cup \{x^*\}, \mathscr{U} \leftarrow \mathscr{U} \backslash \{x^*\}$;
2: **repeat**
3: **for** $i = 1$ **to** m **do**
4: Samples in subset $\mathscr{Q}$ are randomly and uniformly labeled by the pseudo annotator $\mathscr{O}_i$;
5: Train on $\mathscr{L} \cup \mathscr{Q}$ and use active learner $\mathscr{A}$ to select the most informative sample denoted by s_i;
6: **end for**
7: Select the sample x^* from $\{s_1, s_2, \ldots, s_m\}$ by using MNND (see Equation 5.2);
8: Update $\mathscr{Q} \leftarrow \mathscr{Q} \cup \{x^*\}, \mathscr{U} \leftarrow \mathscr{U} \backslash \{x^*\}$;
9: **until** Budget is reached
10: Human expert annotators the instances in $\mathscr{Q}$ with true labels $Y_{\mathscr{Q}}$, update $\mathscr{L} \leftarrow \mathscr{L} \cup \{\mathscr{Q}, Y_{\mathscr{Q}}\}$;

sample is queried from the candidate set. The overall training procedure of our method is given in Algorithm 3.

In this work, we consider minimizing the overall nearest neighbor distance between unlabeled data and queried data to choose the most representative sample. We call this technique Minimizing Nearest Neighbor Distance (MNND for short). We will further explain why and how to implement the MNND in Subsection 5.3.3.

5.3.3. MINIMIZING NEAREST NEIGHBOR DISTANCE

Figure 5.3 illustrates the main idea behind minimizing nearest neighbor distance between unlabeled samples and queried samples. The yellow dots represent the unlabeled samples, e.g. samples in $\mathscr{U}$ whereas the red squares stand for these already chosen samples, e.g. samples in $\mathscr{L} \cup \mathscr{Q}$. The red dot indicates that this instance is chosen as the next queried data point, followed by a calculation of the minimum nearest neighbor distance. First, we define the overall nearest neighbor distance between unlabeled data $\mathscr{U}$ and the remaining already chosen data which includes both labeled data $\mathscr{L}$ and $\mathscr{Q}$ as follows:

$$Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q}) = \sum_{u \in \mathscr{U}} \min_{x \in \mathscr{L} \cup \mathscr{Q}} \|u - x\| \tag{5.1}$$

where $\|u - x\|$ denotes the Euclidean distance between unlabeled data point u and labeled (in our case possibly randomly labeled) data point x.

For example, in Figure 5.3, for all unlabeled samples, we first find their nearest neighbor from $\mathscr{L} \cup \mathscr{Q}$ and then sum over all the pair distances between unlabeled data points and their corresponding nearest neighbors. Finding such nearest neighbor can be interpreted in two ways: (1) it can be seen as classifying unlabeled samples using 1-nearest neighbor algorithm; (2) it can also viewed as clustering unlabeled instances according to these already chosen data points, where each instance in $\mathscr{L} \cup \mathscr{Q}$ is considered as the

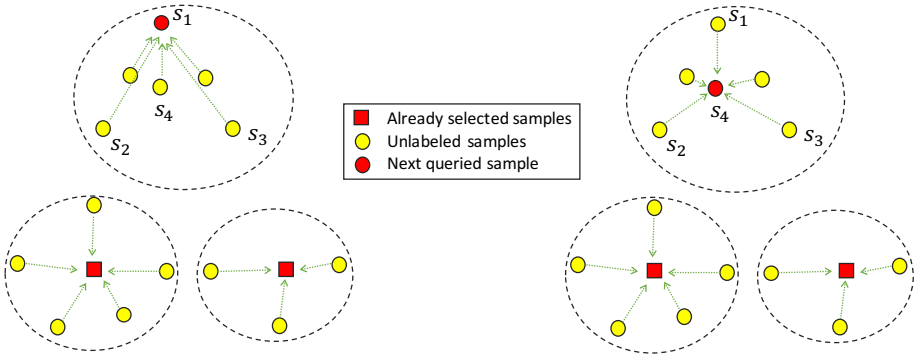


Figure 5.3: Illustration of minimizing nearest neighbor distance between unlabeled samples and queried samples. Arrows indicate the distance between unlabeled instances to their nearest already selected instances. The dashed ellipses indicates that these unlabeled samples inside share a common nearest neighbor. $\{s_1, s_2, s_3, s_4\}$ are the candidates chosen by using multiple pseudo annotators. (a) s_1 is assumed to be selected as the next queried sample; (b) s_4 is assumed to be selected as the next queried sample. Since choosing s_4 will lead to a smaller overall distance between unlabeled samples and queried samples, our algorithm chooses s_4 as the next queried instance.

cluster centroid. If $\mathscr{U}$ is not empty and $Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q})$ reaches its minimum value 0, it implies that all unlabeled data points can find some samples which are exactly the same as themselves. This also indicates that all unlabeled data can be perfectly classified by the 1-nearest neighbor algorithm. Therefore, $Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q})$ can be considered as a measure of how well the labeled data can represent the unlabeled data. The smaller the value of $Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q})$, the more representative of already queried samples.

Now let us return to how to select the most representative sample from these candidates obtained by employing multiple pseudo annotators. For example, assume that our method uses m pseudo annotators and obtains m candidates: $\mathscr{S} = \{s_1, \ldots, s_m\}$ with $s_i \in \mathscr{S}$. We prefer the sample s which can lead to a minimum nearest neighbor distance once chosen as the next queried sample. The intuition behind is that the smaller the value of $Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q} \cup s)$, the more representative the queried samples s is. Therefore, we consider selecting x^* to minimize the nearest neighbor distance between queried data and unlabeled data as follows:

$$x^* = \arg\min_{s \in \mathscr{S}} Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q} \cup s) \tag{5.2}$$

As we see, there are four candidates $\{s_1, s_2, s_3, s_4\}$ chosen by using four pseudo annotators in Figure 5.3. If s_4 is chosen as the next queried instance (shown in Figure 5.3(b)), it will result in a minimum nearest neighbor distance than that other samples are queried (e.g. as shown in Figure 5.3(a), selecting s_1 will lead to a larger value of $Dis(\mathscr{U}, \mathscr{L}, \mathscr{Q} \cup s)$). Finally, our algorithm chooses s_4 as the next queried instance. This also implies that our method prefers these samples which are located in the high density region.

To demonstrate the effectiveness of MNND, let us compare Figure 5.1b and Figure 5.1c. We observe that our method without using MNND fails to select the points in the

bottom right corner (in Figure 5.1b). However, when MNND is utilized, our method can select samples from all four clusters and most of the queried points are in high density regions. As it turns out, MNND is able to further enhance the exploratory behavior of standard active learning algorithms.

5.4. COMPARISONS AND CONNECTIONS

This section discusses the connections and differences between our method and other relevant approaches.

To start with, we first describe the active learning techniques which are combined with our random labeling mechanism. Our method is generally designed for active learning algorithms which make use of label information for selection. We employ our method in combination with two active learning methods. The first one is an uncertainty sampling strategy, the maximum entropy criterion (MaxE for short) [36]. Though MaxE is quite simple, it performs well in comparison with many myopic active learning algorithms [172]. The other one we consider is a recently proposed myopic active learning method called Maximizing Variance for Active Learning (MVAL for short). MVAL shares some similarities with the classical query-by-committee [62]. MVAL forms a committee that consists of models trained on currently labeled data and each unlabeled sample with all possible labels. It trains each committee member and records the posterior probabilities of unlabeled samples to form so-called retraining information matrices (RIMs). These RIMs are used to compute the disagreement among each committee member on all unlabeled samples. More specifically, MVAL estimates two kinds of variance to evaluate the informativeness and representativeness and fuses these variances as a measure of the disagreement. Finally, MVAL queries the sample that causes maximum disagreement among all committee members. The main differences between MaxE and MVAL are that MaxE only concentrates on exploitation while MVAL considers both exploitation and exploration.

Many efforts have been devoted to density or diversity based active learning [16, 25, 61, 84, 101]. The common idea behind these is that we should select samples which are representative of unlabeled data, e.g. both Settles and Craven [84] and Zhu *et al.* [25] used the similarity between unlabeled samples as a measure of density and combined the density measure with the uncertainty measure. However, these methods may get into trouble in the single shot setting. The reason is that the uncertainty measure is fixed during the selection process since no true labels can be obtained and these methods may fail to balance density and uncertainty. Similar to these density-based approaches, our method also prefers querying representative samples by minimizing the nearest neighbor distance. For example, in Figure 5.3, our algorithm prefers selecting these samples which are close to the cluster centers. In this sense, the proposed method provides an alternative view to select representative samples.

The proposed method also has some connections with k-means++ algorithm [201]. k-means++ uniformly and randomly chooses a data point as the first cluster center, and selects the next cluster center from remaining data points with probability proportional to their squared distance from the closest existing cluster center. After that, the chosen seeds are feed to start k-means clustering algorithm.

Four aspects distinguish our work from k-means++. To start with, the first point is

chosen by the active learner $\mathscr{A}$ trained on initially labeled data $\mathscr{L}$. Secondly, the subsequent point is not chosen from all the remaining unlabeled samples. Our method only chooses the next queried point from a candidate set S generated by multiple random annotators. Thirdly, our method determinately selects the sample which leads to a minimum neighbor distance once chosen while k-means++ randomly chooses the next sample with some kind of probability. Finally, our method does not use any subsequent clustering algorithms while k-means++ still needs use k-means clustering algorithm. In some sense, the proposed method can also be viewed as a sequentially adaptive clustering approach. The benefit of our method over k-means++ is that our method can make use of some existing supervised information, e.g. training active learner $\mathscr{A}$ on initially labeled data $\mathscr{L}$, while k-means++ is a pure unsupervised clustering approach.

The strength of the proposed method over random sampling is that our method considers both exploitation, i.e., by training on an initial labeled data set $\mathscr{L}$, and exploration, i.e., by training on the randomly-labeled data set $\mathscr{Q}$. More important is that we also use MNND to select representative instances. On the contrary, random sampling only does pure exploration and fails to consider the representativeness of unlabeled samples. One work [202] also proposed to use some randomly selected samples to explore. However, that technique is particularly designed for binary classification tasks and it is unclear how to extend it to multi-class classification.

5.5. EXPERIMENTS

We test the empirical performance of the proposed method and compare it against other subset selection approaches. Additional comparisons are also made between our method and conventional batch mode active learning algorithms. We first describe the used test datasets and experimental setup, followed by an analysis of the experimental results.

5.5.1. DATASETS

We use 10 benchmark datasets in our experiments, some of which are image classification tasks, such as two handwritten digit datasets, the MNIST [140] and the USPS dataset [203]. For the MNIST and the USPS dataset, the gray-scale pixel values are used as the features. We also use three pre-processed subsets in the *Office* dataset [204], including the Amazon, Webcam and Caltech datasets. These three sets contain 10 common classes which are from different sources, e.g. the Amazon dataset contains images downloaded from online merchants and the Webcam set uses low-resolution images by a web camera. SURF features are firstly extracted and then encoded into an 800-bin histograms. In addition, we also experiment on five standard datasets [205], including five categories of images taken from Caltech101 (C), ImageNet (I), LabelMe (L), SUN09 (S), VOC2007 (V). The selected five categories are: bird, car, chair, dog and person. Following [205], we use the pre-extracted DeCAF6 features. For computational efficiency, sub-sampling and principal component analysis (PCA) are applied on some of the larger datasets to reduce the sample size and feature dimensionality. The detailed information of these preprocessed test datasets after pre-processing is listed in Table 5.1.

Table 5.1: Characteristics of the preprocessed test datasets: the number of instances (#n), the feature dimensionality (#fea) and the number of class (#c). Refer to the text in the beginning of Subsection 5.5.1 to see what C, L, V, I, and S stand for.

Dataset	(#n, #fea, #c)	Dataset	(#n, #fea, #c)
C	(467, 4096, 5)	I	(500, 4096, 5)
L	(410, 4096, 5)	S	(350, 4096, 5)
V	(500, 4096, 5)	Amazon	(500, 50, 10)
Webcam	(295, 50, 10)	Caltech	(500, 50, 10)
MNIST	(1000, 60, 10)	USPS	(1000, 60, 10)

5.5.2. EXPERIMENTAL SETUP

In our experiments, each dataset is randomly and repeatedly divided into training and test sets of equal size. We randomly select one sample from each class as the initial labeled set. All the experiments are repeated 20 times and the average performances are reported. The number of queried samples varies from $\{20, 40, 60, 80, 100, 120\}$ in our experiment.

We compare the proposed method, Active Learning through Random Labeling (ALRL for short), with random sampling and the following algorithms:

- USDM: Uncertainty sampling with diversity maximization, which retains the uncertainty and maximizes the diversity simultaneously [101].
- BatchRank: It balances the informativeness and diversity and offers some relaxations to solve the optimization problem [44].
- k-means++: It applies k-means++ [201] algorithm and selects the sample nearest to the centroid of a cluster.
- TED: Transductive experimental design chooses examples to minimize the variance of ridge regression model [52].
- V-opt: It selects samples to minimize the V-optimality on Gaussian Random Fields (GRFs) [53].
- Σ-opt: Similar to V-opt, it minimizes the Σ -optimality on GRFs [54].
- DS3: It selects representative samples by minimizing the dissimilarity between selected data and the remaining data [55].

Among these compared methods, two methods, USDM and BatchRank, are the most recent state-of-the-art batch mode active learning algorithms. Since k-means++ is relatively sensitive to the initialization seeds, we repeat it 500 times with different random seeds and choose the one which performs the best based on the objective function of k-means++. For fairness, linear SVM with the same parameter setting is used to evaluate the performances of all compared algorithms. We use the LIBSVM package [143] and empirically set the regularization parameter $C = 10$. The number of pseudo annotators m is empirically set 10 in all experiments.

5.5.3. RESULTS

We first investigate whether the proposed random labeling mechanism can help improve the performance of standard active learning algorithms. Subsequently, we compare our method with other subset selection approaches.

THE EFFICIENCY OF RANDOM LABELING

We show the average performance over all test sets of our method with MaxE and MVAL in Figure 5.4. ALRL_MaxE denotes the combination of MaxE and our method with a default m=10 pseudo annotators. We also show the performance of ALRL_MaxE (m=1) in which a single random annotator is used such that MNND does not play a role. It is the same setting with ALRL_MVAL and ALRL_MVAL (m=1). MaxE_True and MVAL_True refer to the two active learners that are obtained of the true is obtained fro a human ob server in every iteration. This is, in a sense, the best one can do and therefore serves as a natural upper bound for the performance.

It is obvious that standard myopic active learning algorithms perform poorly in the single shot setting, e.g. MaxE and MVAL are significantly worse than random sampling. The reason is that these active learners are likely to select samples which extensively overlap each other. As shown in Fig 5.1a, standard uncertainty sampling keeps selecting instances which are close to initial decision boundary and have considerable overlapping within each other. And it fails to query new data points from the other two clusters. From another perspective, this also implies that the key advantage of active learning over random sampling is that active learner can iteratively learn from the labels obtained from human annotator. In the single shot scenario where standard myopic active learners cannot query human annotator for labels and cannot update their selection criteria, it does make sense that these approaches demonstrate poor performance.

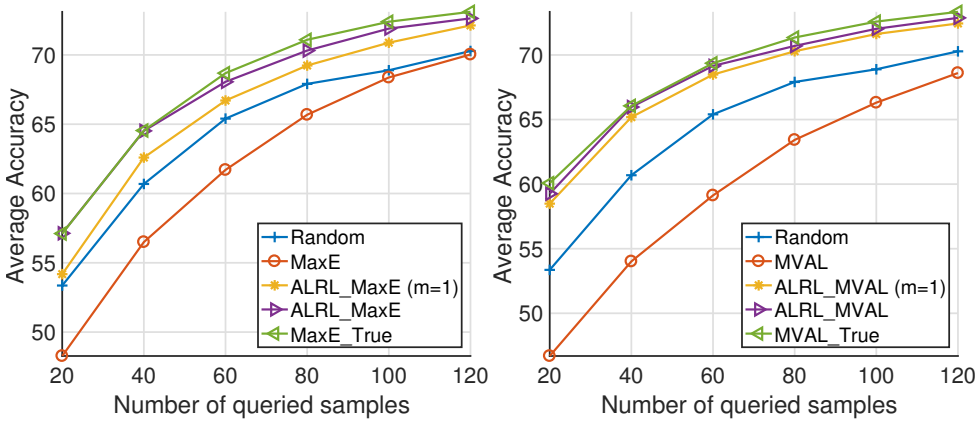
However, by adopting our proposed random labeling, both ALRL_MaxE (m=1) and ALRL_MVAL (m=1) outperform the original active learners (MaxE and MVAL) and random sampling. As shown in Fig 5.1b, our random labeling strategy impels uncertainty sampling to explore and select diverse data points without large overlapping. This demonstrates that in the single shot scenario, our random labeling mechanism can indeed boost the performance of regular active learners by promoting exploration.

We also investigate the benefit of our proposed MNND criterion. In Figure 5.4, ALRL_MaxE and ALRL_MVAL outperform their competitors, ALRL_MaxE (m=1) and ALRL_MVAL (m=1), respectively. This confirms the advantage of selecting representative samples by minimizing nearest neighbor distance. It also means that we can still expect better performance when multiple pseudo annotators are employed and the most representative sample are chosen by using MNND. We also observe that two active learners that received human feedback in every iteration, MaxE_True and MVAL_True, obtain the best performances. And our method comes very close to these two active learners. This demonstrates that even in the single shot setting, our proposed random labeling mechanism can produce promising results relatively comparable to that of active learner in the standard setting where human annotation is obtainable in each iteration.

THE INFLUENCE OF MYOPIC ACTIVE LEARNER

Figure 5.5 illustrates the influence of myopic active learner chosen in our method. ALRL_Random means that random sampling is used as the active learner $\mathscr{A}$ in Algorithm 3. In

addition, we also compare with Simple_MNND in which no pseudo annotators are used and the most representative sample is directly chosen from all the remaining unlabeled samples by using MNND. The difference between ALRL_Random and Simple_MNND is that ALRL_Random conducts MNND on a random subset while Simple_MNND implements MNND on all remaining unlabeled data. We can see that ALRL_Random and Simple_MNND perform similarly to each other. And both of them are surpassed by ALRL_MaxE and ALRL_MVAL. This implies that the exploitation produced by MaxE and MVAL increases the learning performance. Overall, ALRL_MVAL demonstrates the best performance. In the following experiments, ALRL refers to ALRL_MVAL, unless otherwise specified.



(a) MaxE (b) MVAL

Figure 5.4: Performance comparisons of random labeling in combination with two active learners (a) MaxE and (b) MVAL. ALRL_MaxE is the combination of MaxE and our method ALRL with a default m=10 pseudo annotators while ALRL_MaxE (m=1) means that we only use a single random annotator so that MNND is not utilized. It is the same setting with ALRL_MVAL and ALRL_MVAL (m=1). MaxE_True and MVAL_True refer to the two active learners that can obtain the true label from human expert when an unlabeled sample is selected.

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART ALGORITHMS

Figure 5.6 compares our method ALRL to several state-of-the-art algorithms over 10 test datasets. We can see that our method obtains the best performance on most datasets, such as MNIST, USPS, Caltech, Amazon, L and Webcam. ALRL also demonstrate excellent performance on the I and V datasets, ranking in the second place. It only fails to remain among the top two methods on the C and S datasets. TED also shows good results on several datasets, especially on the V and I, on which it achieve the highest accuracy. Most of the remaining methods can perform quite well only on one or two datasets, e.g. DS3 exceed other methods on the C dataset. However, these two batch mode active learning algorithms USDM and BatchRank perform poorly on most datasets, e.g. BatchRank obtains worse performance than random sampling on 7 datasets. Incidentally, this supports our claim that standard batch mode active learning algorithms are likely to under-perform in the single shot setting. We show the average performance

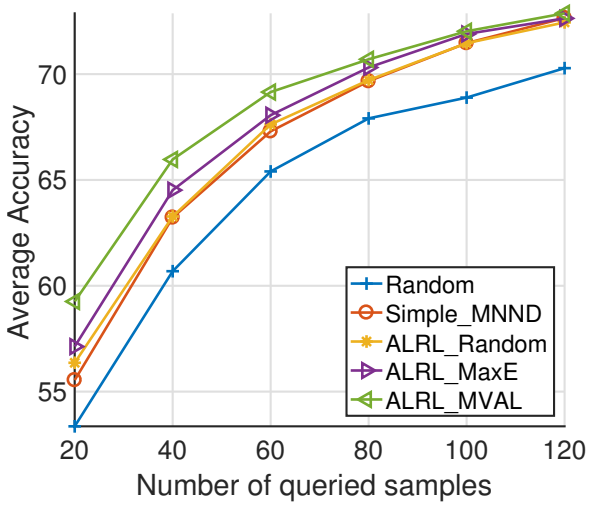


Figure 5.5: Performance comparisons of our method ALRL in combination with different active learners over all test sets. Simple_MNND means that we ignore the pseudo annotators and directly choose the most representative sample from all the remaining unlabeled samples by using MNND. ALRL_Random means that random sampling is used as the active learner $\mathscr{A}$ in Algorithm 3.

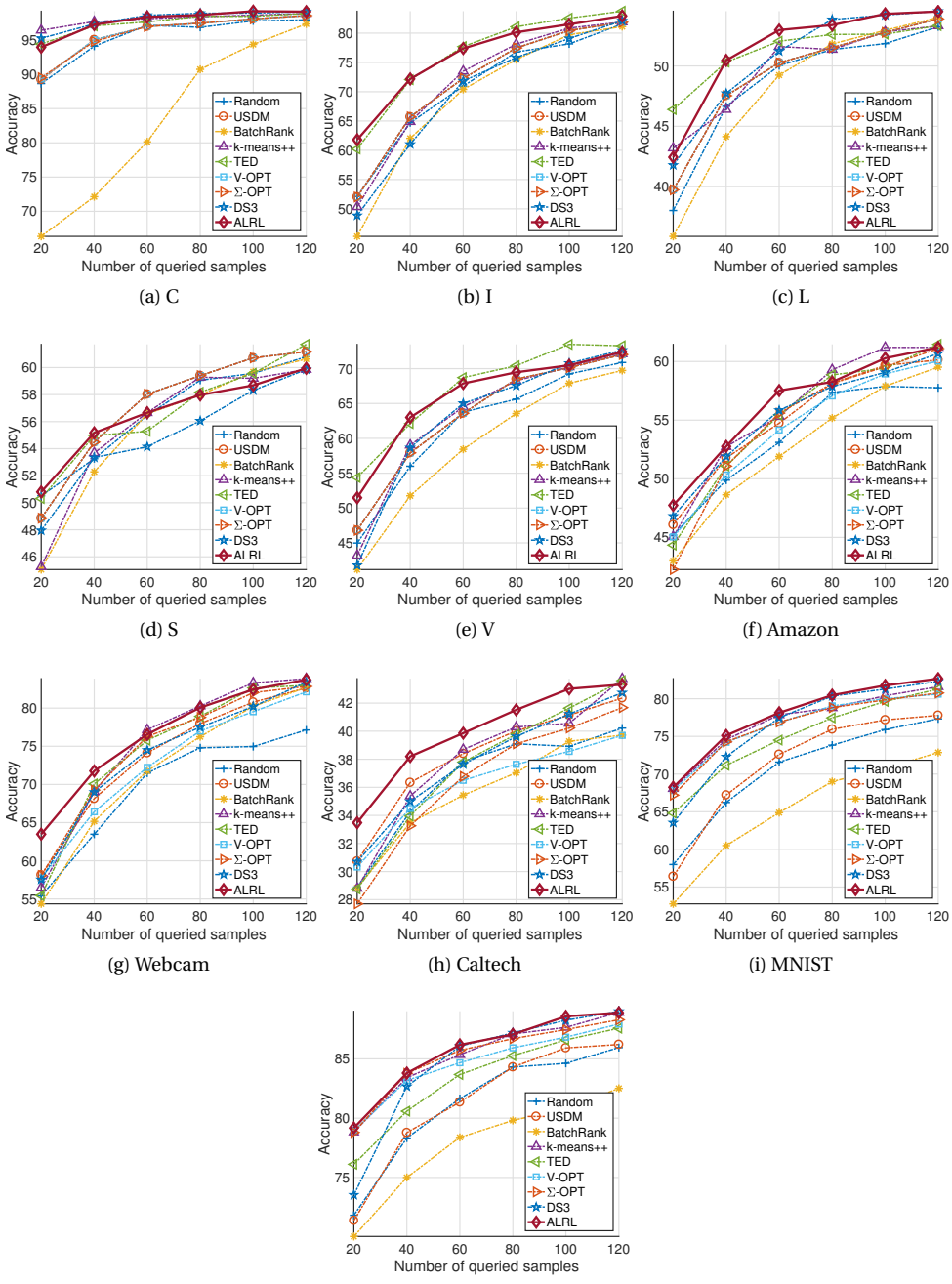
of the different algorithms in Figure 5.7. Our method achieves the best overall performance, with TED as a good second. k-means++ shows performance similar to TED, but underperforms when the budget is very small. Our method shows a clear advantage both with small and large budgets.

SENSITIVITY ANALYSIS

We also set up experiments to explore the influence of the number of pseudo annotators on the efficiency of our method. We use MaxE as the active learner and repeat the experiments by varying m from a set $\{1,4,8,10,12,16,20,24\}$. Figure 5.8 shows the average performance over 10 test datasets. Note that $m = 1$ means that no MNND is utilized since there is only a single candidate in $\mathscr{S}$. In the case of $m = 1$, a sharp decline in the performance of our proposed method is witnessed, which indicates that MNND can indeed enhance the performance by filtering out some poor random labelings. We can also observe that our method is not very sensitive to the number of pseudo annotators. For example, when m varies from 4 to 24, the overall average performance of the proposed method shows little change. This implies that our method is robust to the number of pseudo annotators.

5.6. DISCUSSION AND CONCLUSION

We tackle the problem of human experts being unavailable during the active data selection process by introducing multiple pseudo annotators. These pseudo annotators uniformly and randomly annotate queried samples, which provides standard active learning methods with the ability to explore. The exploratory behavior is further enhanced by selecting the most representative sample through minimizing nearest neighbor distance



(a) C (b) I (c) L

(d) S (e) V (f) Amazon

(g) Webcam (h) Caltech (i) MNIST

(j) USPS

Figure 5.6: Performance comparison of different methods over 10 test datasets. The x-axis is the number of queried samples while the y-axis is the classification accuracy.

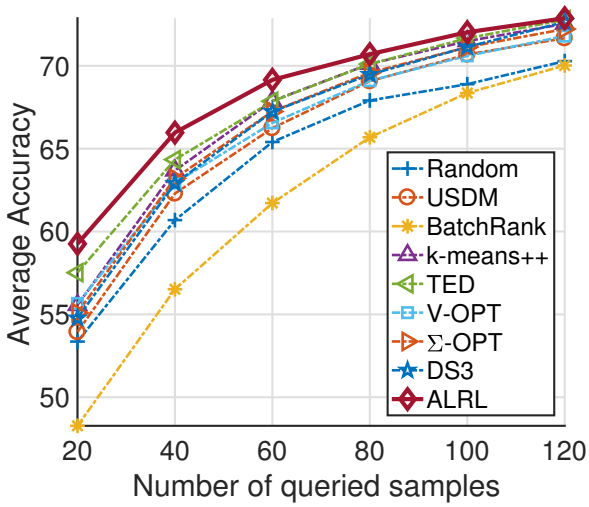


Figure 5.7: Overall average performance of compared methods on 10 test datasets.

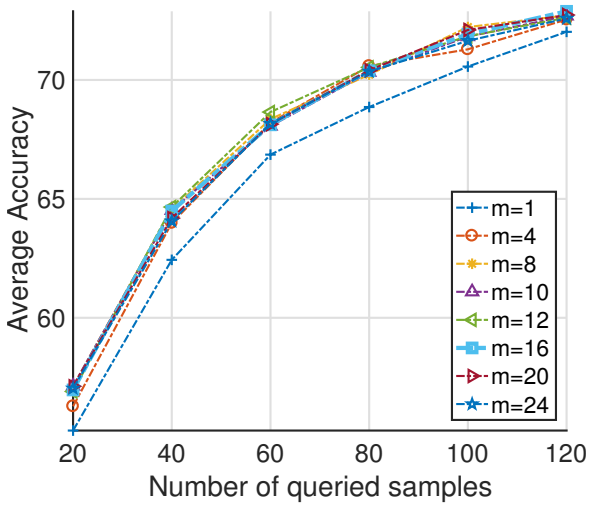


Figure 5.8: Sensitivity analysis: it shows average performance of our proposed method over 10 test datasets w.r.t. different number of pseudo annotators m.

between unlabeled data and queried data. Experiments on real-world datasets show that our method (ALRL) can indeed obtain close result to active learner that receives true label feedback from human annotator. Our method demonstrates very good performance when compared with state-of-the-art data selection methods.

We see several future research directions in which our method can be extended. First of all, currently, the proposed nearest neighbor distance $Dis(\mathcal{U},\mathcal{L},\mathcal{Q})$ treats all samples in $\mathcal{L},\mathcal{Q}$ equally, i.e., with equal weighting. An alternative is assigning a different weight to each instance, depending on whether its nearest neighbor is from $\mathcal{L}$ or from $\mathcal{Q}$. It requires more efforts to investigate this importance weighting strategy. Secondly, the proposed method assumes that there are a small number of labeled instances available. A challenge is how to adapt our method to the cold-start case where no labeled data is available.

6

ACTIVELY INITIALIZE ACTIVE LEARNING

Though many efforts have been devoted to designing new active learning algorithms, little attention has been paid to the initialization problem of active learning: how to find a set of labeled samples which contains at least one instance per category. We revisit previous methods that can be used for initialization and propose a new active initialization criterion, namely the Nearest Neighbor Criterion. Experiments on 16 benchmark datasets verify that the proposed method often finds an initialization set with fewer queried samples than other methods can.

This chapter is in preparation for publication.

6.1. INTRODUCTION

These days, we are witnessing a sharp increase of the amount of training data used in classification or regression tasks. Though the availability of large input data tends to boost the performance of machine learning models, it also leads to a big challenge: manually labeling these hundreds of thousands of samples is very time-consuming and expensive [36]. Active learning has been proposed to tackle this challenge by querying the most informative subsets from the whole data and maintaining good learning performance. Most active learning approaches, which we refer to as supervised, need an initial labeled subset to start the active learning cycle based on true labels obtained from a human annotator. The main reason for the need for labels is that the selection criteria typically depend, one way or the other, on a trained classifier. For example, the simple margin method [24] trains a classifier on the initial labeled data and then chooses the sample nearest to its decision boundary.

There also exist unsupervised active learning methods which do not use label information for sample selection. For instance, transductive experimental design (TED) [52] and graph-based variance minimization methods [53, 54] minimize the expected variance of a statistical model and omit the labels of all samples. Such unsupervised methods do not exploit the information made available through the labeling provided by the human annotator. Generally speaking, their performances can be improved by utilizing label information. For example, Zhen and Yeung [181] introduced a supervised version of TED, which adds a regularization term to incorporate label information. Gu *et al.* [112] proposed a weighted TED where the weight is estimated by using the class probability. Both of them show better performance than the original TED, indicating that supervised versions are to be preferred over unsupervised ones.

A crucial issue for supervised active learning, however, is that these methods assume to have a labeled dataset to start with. Though considerable efforts have been devoted to seeking new active learning strategies, little attention has been paid to the initialization of these supervised methods. As also suggested by [207], we can view active learning as a two-step process: find an initial labeled subset using some initialization strategy in the first step and employ any regular active learning algorithm preferred in the second. In this Chapter, we consider exactly that first initialization step: how to find an initial labeled subset containing at least one instance of each category to start the active learning process.

The simplest way to initialize active learning is to randomly select unlabeled instances until a subset containing at least one instance of each class has been obtained. This is also the most common strategy used in the active learning literature [24, 40, 41, 76, 208]. Oftentimes, it is simply assumed that there exists an initial set consisting of a fixed number of instances per category randomly chosen from the unlabeled samples [41, 101, 208]. This is not realistic in many real-world applications, since we cannot know the class labels in advance [207]. In this Chapter, we consider a more reasonable and realistic case where we start the initialization without any labeled samples and actively choose the initial instances to label. Given that the number of classes is known beforehand, the initialization phase will be stopped when the initial set contains at least one instance per class.

Section 6.3 covers various known approaches and techniques that can be adapted

to initialize active learning. That same section introduces and discusses a new strategy, which we call Nearest Neighbor Criterion (NNC for short). First, however, Section 6.2 presents background and related work for the initialization strategy. Following the section on different initialization strategies, Section 6.4 describes the experimental setup and analyses the results. This is followed by a conclusion of this work in Section 6.5.

6.2. RELATED WORK

To start with, we remark that this work focuses on pool-based active learning where a large pool of unlabeled samples are already available for querying [36]. This setting is the most widely studied one in active learning. Reference [36] provides an early overview of active learning that also covers many of its ins and outs. Some more recent, complementary surveys on active learning can be found in [154] and [51].

There already exist a couple of studies considering the Initialization of Active Learning (IAL for short), including [210–212]. These methods mainly use unsupervised clustering approaches (e.g. k-means) to find the representative samples in each dataset. Both of the works in [211, 212] show the superiority of k-means based initialization strategy. However, these methods use a fixed initialization size (e.g. 10% of all the data), which seems unwanted as it cannot guarantee that all classes have been identified. In the setting we consider, the initialization stage will be terminated once at least one instance from each class has been selected. Another downside of these k-means-based approaches is setting the parameter k as, clearly, we cannot guarantee that each instance selected from each cluster belongs to a different category.

As experimental design approaches are unsupervised active learning techniques, they can be directly used to initialize active learning. Examples are the D-optimality based transductive experimental design (TED) [52], V-optimality based graph variance minimization (V-opt) [53] and recently proposed Σ-optimality based graph variance minimization (Σ-opt) [54]. These methods select the representative samples which reduce the variance of a specific statistical model. The differences among them are the chosen optimality criteria measuring the overall variance and the actual model they rely on. Subsection 6.3.1 will provide some further specifics about these approaches and the clustering-based ones.

Further work similar to ours can be found in the area of rare category detection [214–217]. The aim in these settings is to use active learning to identify interesting and useful anomalies, which are assumed to be very rare and typically can be found in tiny classes. The selection procedure will also be terminated when at least one representative sample from each rare class has been found. For example, He and Carbonell [214] first performed density estimation and then selected the instance which leads to an expected maximum change in local density once queried. Fincham Haines and Xiang [215] proposed a criterion called pWrong, which selects the sample most likely to be wrong (in other words, belonging to an unseen category). Hospedales *et al.* [217] introduced a technique (Gen/Disc), which adaptively switches generative and discriminative classifiers in the learning progress to jointly discover new categories and maintain good learning performance. Hospedales *et al.* [216] proposed a criterion called Dirichlet Process Expected Accuracy (DPEA for short) to unify active learning and active class discovery.

The IAL differs from rare category detection in two respects. First, rare category de-

tection focuses on finding useful and useless anomalies from normal data points, while IAL does not make any assumption on whether or not anomalies exist. IAL only concentrates on seeking representative samples to start active learning. Second, rare category detection always assumes that the datasets are extremely imbalanced with large majority classes and relatively small rare classes. IAL does not make such imbalanced assumption.

6.3. ADAPTED AND NEW INITIALIZATION STRATEGIES

We briefly revisit various approaches from Section 6.2 that can be used to initialize active learning, but that need some minor adaptations. Subsequently, we present a new initialization criterion: Nearest Neighbor Criterion (NNC). But first some preliminaries.

In the setting we consider, a totally unlabeled dataset $\mathbf{P} = \{x_i\}_{i=1}^{n}$ is available, where $x_i \in \mathbb{R}^d$ is a feature vector. We denote with $\mathbf{I}$ the initial set and with $C(\mathbf{I})$ the number of classes in $\mathbf{I}$. $\mathbf{U}$ is the remaining unlabeled data such that $\mathbf{U} = \mathbf{P} \backslash \mathbf{I}$. Algorithm 4 presents a basic active initialization process in which the number of all classes we know beforehand is c. We sequentially choose an unlabeled sample and ask for its labels from a human annotator. When all classes have at least one instance, the initialization phase will be stopped, followed by a switching to the some preferred supervised active learning algorithm.

Algorithm 4 Actively Initialize Active Learning

Require: unlabeled data $\mathbf{U}$, number of classes c, initial set $\mathbf{I} = \emptyset$.

1: **while** $C(\mathbf{I}) < c$ **do**
2: Choose the sample x^* according to some initialization criterion.
3: Query it for its label y and update $\mathbf{I} = \mathbf{I} \cup \{x^*, y\}$, $\mathbf{U} \leftarrow \mathbf{U} \backslash \{x^*\}$;
4: **end while**
5: Start some preferred supervised active learning algorithm with initial labeled $\mathbf{I}$.

6.3.1. ADAPTED TECHNIQUES

We draw from what already has been covered in Section 6.2 and cover methods here that can be easily adapted to the task of initializing active learning.

Clustering-based Approaches We first consider using k-means to initialize the active learning algorithms due to its simplicity [211, 212]. Specifically, we use k-means++ which smartly chooses the initial seeds for k-means and performs well in practice [201]. There is no easy way to set the parameter k, but we suggest the following procedure. We perform k-means++ with $k = c$ in which c is the number of classes and query one instance nearest to the centroid from each cluster. If there are still some categories undiscovered, we continue the aforementioned procedure on the remaining unlabeled data until all categories own at least one data points. Algorithm 5 shows the overall initialization process. The main difference between Algorithm 5 and the works in [211, 212] is that the latter ones set k to be equivalent to a pre-defined budget and terminate the active annotation process when the budget is finished.

Algorithm 5 Initialization with k-means++

```
I = ∅;
while C(I) < c do
    Perform k-means++ with k = c on U and set i = 1;
    while C(I) < c & i ≤ k do
        Choose the sample x* closest to the centroid from ith cluster;
        Query its true label y and update I = I ∪ {x*, y}, U ← U\{x*}, i = i + 1;
    end while
end while
```

Optimal Design-based Approaches We show by example how to adapt, in a simple way, methods for experimental design for the initialization of active learning. The example criterion we use is TED [52]. In brief, TED minimizes the variance of a regularized least square model by solving the following optimization problem:

$$\min_{\mathbf{X}\subset\mathbf{P}} \quad \mathrm{Tr}(\mathbf{P}(\mathbf{X}^T\mathbf{X}+\lambda\mathbb{I})^{-1}\mathbf{P}^T), \tag{6.1}$$

where $\mathbf{X}$ represents a set of instances to be queried, λ is the regularization parameter, and $\mathbb{I}$ is the identity matrix. A sequential greedy solution to TED is to select instance x such that $\mathrm{Tr}(\mathbf{P}(\mathbf{X_{t+1}}^T\mathbf{X_{t+1}}+\lambda\mathbb{I})^{-1}\mathbf{P}^T)$ achieves its minimum value, where $\mathbf{X}_{t+1}=\mathbf{X}_t\cup x$ and t represents the t-th round selection. We query the instances one by one according to the TED criterion and terminate the initialization process when $C(\mathbf{I})$ is equal to c. Of course, other criteria used for optimal experimental design can be employed for initializing in exactly the same way.

6.3.2. MINIMUM NEAREST NEIGHBOR DISTANCE

As a last method, we present one that is new, simple, and fast. We will refer to it as the Nearest Neighbor Criterion (NNC). It sequentially queries the most representative instance from unlabeled data such that the overall distance between queried data and unlabeled data achieves its minimum value. The main motivation behind this is that we want to minimize the dissimilarity between labeled data and unlabeled data such that these labeled data can well-represent the remaining unlabeled data. We use the total nearest neighbor distance as a measure of the dissimilarity between queried data and unlabeled data, which is defined as follows:

$$NND(\mathbf{I},\mathbf{U}) = \sum_{u\in\mathbf{U}} \min_{x\in\mathbf{I}} \|u-x\|, \tag{6.2}$$

where $\|u-x\|$ denotes the Euclidean distance between an unlabeled instance u and a labeled instance x.

$NND(\mathbf{I},\mathbf{U})$ computes the sum of the Euclidean distance between each unlabeled data point $u\in\mathbf{U}$ and its corresponding nearest neighbor x chosen from queried data $\mathbf{I}$. $NND(\mathbf{I},\mathbf{U})$ obtains a relatively small value when each unlabeled sample is close to its nearest neighbor. In other words, if each unlabeled data point is very similar to its nearest neighbor, e.g. $\min_{x\in\mathbf{I}}\|u-x\|$ is small, then the overall neighbor distance between $\mathbf{U}$ and $\mathbf{I}$ is small too. This also implies that $NND(\mathbf{I},\mathbf{U})$ can be considered as a measure of how well the

queried data can represent the unlabeled data. The smaller the value of $NND(\mathbf{I},\mathbf{U})$, the more representative of queried data $\mathbf{I}$.

Therefore, to initialize active learning, we select an unlabeled sample s that leads to a minimum value of the dissimilarity between queried data and unlabeled data once labeled. In other words, we select an unlabel sample, denoted by s, as the next queried data point such that the nearest neighbor distance between queried data $\mathbf{I} \cup s$ and the remaining unlabeled data $\mathbf{U} \backslash s$ obtains its minimum value. Our nearest neighbor criterion (NNC) is defined as follows:

$$x^* = \arg\min_{s \in \mathbf{U}} NND(\mathbf{I} \cup s, \mathbf{U} \backslash s)\,. \tag{6.3}$$

Algorithm 6 provides the pseudo-code of NNC.

NNC can be seen as a sequential clustering algorithm. The clustering problem we consider is defined as follows:

$$\arg\min_{\mathbf{S} \subset \mathbf{P}} \sum_{u \in \mathbf{P} \backslash \mathbf{S}} \min_{x \in \mathbf{S}} \|u - x\| \tag{6.4}$$

where $\mathbf{S}$ is a set of samples we want to query. Each instance in $\mathbf{S}$ can be seen as an independent cluster seed. These seeds are used to cluster the remaining unlabeled data points based on the pairwise Euclidean distance. The number of seeds increase by 1 each time a new instance is chosen. The proposed NNC indeed provides a sequential greedy optimization approach to the problem in Equation 6.4. In the first iteration, since the initial set $\mathbf{I}$ is empty, NNC will choose the first point which is closest to the average mean of all data $\mathbf{P}$. NNC then chooses the subsequent sample by minimizing the nearest neighbor distance using Equation 6.3.

NNC has some further links with the earlier used k-means++ algorithm [201]. k-means++ first selects a random data point as the first cluster seed, and then selects the subsequent cluster seed with probability proportional to their squared distance from the closest existing cluster seeds. Two aspects distinguish NNC from k-means++. First, NNC queries the sample nearest to the mean of the data as the first point while k-means++ randomly chooses the first seed. Secondly, NNC selects the subsequent point by minimizing the nearest neighbor distance whereas k-means++ randomly chooses the next point based on some pre-defined probability.

Algorithm 6 Initialization with NNC

1: $\mathbf{I} = \emptyset$;
2: **while** $C(\mathbf{I}) < c$ **do**
3: **for** each unlabeled sample $x_i \in \mathbf{U}$ **do**
4: Assume that x_i is chosen as the next queried sample;
5: Compute the nearest neighbor distance $NND(\mathbf{I} \cup x_i, \mathbf{U} \backslash x_i)$;
6: **end for**
7: Choose the sample x^* with minimum nearest neighbor distance using Eq. 6.3.
8: Query its true label y and update $\mathbf{I} = \mathbf{I} \cup \{x^*, y\}$, $\mathbf{U} \leftarrow \mathbf{U} \backslash \{x^*\}$;
9: **end while**

6.4. EXPERIMENTS

We test the performance of the proposed method NNC and various methods described in the foregoing section. We first describe the experimental setup and, subsequently, present the way we compare the approaches.

6.4.1. EXPERIMENTAL SETUP

NNC is compared with the following algorithms: random sampling, k-means++ [201] (See Algorithm 5), TED [52], Σ-opt [54], pWrong [215], Gen/Disc [217], DPEA [216]. In addition, we also compare with a variant of uncertainty sampling, which first uses NNC to find instances which belong to two different classes (in order to train a classifier) and then use maximum entropy [36, 73] to adaptively choose the most uncertain sample. This method, what we refer to as MaxE, is considered because we want to know whether or not classical supervised active learning methods (e.g. uncertainty sampling) are suitable for the initialization task.

Among the methods compared, pWrong, Gen/Disc, and DPEA are three state of the art rare category detection algorithms. We do not show the performance of V-optimality based approach [53] since we observe that it, in general, performs worse than Σ-opt [54] in our experiments. Therefore, we only present the performance of Σ-opt in our comparison. Since k-means++ is also affected by the first randomly chosen seed, we try 100 different random seeds and choose the one which has lowest within-cluster sum of point-to-centroid distances.

Datasets. 16 multi-class datasets are used in our experiments. Eleven of them are taken from the UCI Machine Learning Repository [139], such as the pendigits, yeast, semeion, and vowel. The remaining datasets are vision datasets. For instance, the MNIST [140] and the USPS [203] are handwritten digit datasets, and CIFAR10 [186] and GTSRB [185] are image classification datasets. UCFsports [190] consists of 10 different categories of human actions collected from various sports videos, where the pre-extracted Action Bank features [191] are used in our experiment. For the MNIST and the USPS dataset, we extract the gray-scale pixel values as the features. The HOG feature are extracted [188] for the CIFAR10 and the GTSRB dataset. For computational efficiency, we use random sub-sampling and principal component analysis (PCA) on some datasets to reduce the sample size and feature dimensionality. Table 6.1 presents the properties of each dataset after pre-processing.

For each dataset, we randomly choose half of the data as the unlabeled data where the actively initialization is conducted. The remaining data is used as a separate test set to evaluate the classification accuracy of the chosen initial set. We repeat the random partition 100 times and average the results. We use the linear SVM from the LIBSVM package [143] as the classifier with a regularization parameter $C = 10$.

6.4.2. RESULTS AND ANALYSIS

Table 6.2 shows the performance of each method in terms of the means and standard deviations of the number of queried samples required to initialize active learning. The smaller the value, the better the performance. All the experiments are repeated 100 times and we use a paired t-test at a 95% significance level to check whether there is a statistical significant difference between two models. The method which obtains the best

Table 6.1: Multi-class datasets information after pre-processing: the number of instances (n), the feature dimensionality (d) and the number of class (c).

Dataset	n	d	c	Dataset	n	d	c
MNIST	1000	60	10	USPS	1000	60	10
CIFAR10	1000	57	10	GTSRB	1000	40	20
UCFsports	140	100	10	Isolet	1040	40	26
pendigits	1000	16	10	satimage	1000	36	6
segment	1000	19	7	vowel	990	10	11
dermatology	366	34	6	led_display	1000	7	10
yeast	1484	8	10	ecoli	336	7	8
lowres	531	50	9	semeion	1593	50	10

performance or performs comparably to the best model is highlighted in bold face and coloured. We also report the average score over all test sets and the average ranking of each algorithm (i.e. the "Mean" and "Average Ranking" in Table 6.2). "Wins" counts the number of datasets on which an algorithm behaves the best or acts comparably to the best. "win/tie/loss" reports the win/tie/loss counts of other methods versus random initialization also based on the paired t-test.

Table 6.2: Performance comparison: means and standard deviations of the number of queried samples. "Mean (± std)" reports means and standard deviations of the number of queried samples required to initialize active learning over all test sets. "Average Accuracy" reports the average classification accuracy of all methods over all the test sets. "Average Ranking" shows the average ranking of all methods over all the test sets. "Wins" counts the number of datasets on which an algorithm performs among the best. "win/tie/loss" shows the win/tie/loss counts of other methods versus random initialization.

	Random	k-means++	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	36.24 (±14.32)	52.44 (±35.03)	19.57 (±9.13)	**16.05 (±4.86)**	24.50 (±10.87)	56.57 (±31.43)	20.15 (±10.03)	30.52 (±11.58)	**15.38 (±3.78)**
MNIST	30.05 (±13.07)	47.60 (±22.44)	**21.15 (±6.82)**	23.60 (±6.71)	**21.06 (±5.72)**	47.12 (±26.56)	**21.37 (±10.34)**	23.34 (±7.65)	**22.24 (±4.91)**
CIFAR10	**28.06 (±10.16)**	32.05 (±11.00)	**28.61 (±11.59)**	**29.65 (±8.45)**	35.84 (±10.66)	50.84 (±23.64)	47.60 (±17.36)	**28.92 (±9.81)**	**28.85 (±11.10)**
UCFsports	47.65 (±32.31)	23.72 (±10.31)	17.86 (±5.31)	32.67 (±12.64)	32.86 (±12.77)	32.43 (±12.82)	32.86 (±12.77)	38.56 (±18.57)	**15.21 (±4.21)**
GTSRB	74.16 (±26.19)	70.15 (±25.32)	**57.88 (±15.61)**	**61.25 (±21.69)**	66.96 (±18.92)	75.38 (±22.17)	**62.53 (±18.45)**	**61.78 (±16.21)**	**60.25 (±21.22)**
Isolet	108.98 (±41.68)	102.81 (±37.88)	81.69 (±28.22)	81.30 (±24.84)	102.67 (±34.38)	90.54 (±26.90)	**74.32 (±22.33)**	93.70 (±29.71)	**71.57 (±23.65)**
pendigits	33.16 (±15.30)	47.67 (±27.89)	24.12 (±7.84)	30.02 (±12.37)	29.55 (±11.59)	20.80 (±5.98)	41.08 (±11.87)	28.36 (±15.23)	**18.45 (±4.89)**
satimage	21.70 (±14.52)	103.24 (±26.13)	23.15 (±13.19)	**9.34 (±1.97)**	19.56 (±9.46)	15.09 (±6.59)	**9.65 (±4.23)**	17.78 (±9.61)	**9.03 (±1.65)**
yeast	357.57 (±372.64)	87.20 (±44.23)	**66.09 (±40.30)**	**66.74 (±33.00)**	69.88 (±37.98)	71.94 (±46.74)	**58.95 (±27.32)**	106.44 (±116.38)	**62.08 (±33.95)**
segment	17.33 (±9.04)	20.95 (±8.03)	14.77 (±4.56)	**10.82 (±2.84)**	15.04 (±4.93)	26.79 (±10.07)	**10.93 (±3.42)**	18.26 (±7.84)	**10.47 (±2.72)**
vowel	37.31 (±13.55)	53.57 (±23.34)	79.84 (±60.82)	30.50 (±9.30)	34.36 (±9.39)	57.38 (±41.73)	28.60 (±10.70)	30.88 (±10.95)	**25.95 (±5.98)**
lowres	465.02 (±309.97)	108.75 (±82.48)	82.55 (±41.04)	**24.55 (±4.52)**	29.03 (±11.35)	73.48 (±18.22)	68.35 (±6.97)	78.14 (±31.87)	38.54 (±13.61)
dermatology	26.49 (±19.12)	12.10 (±7.03)	8.50 (±3.02)	25.45 (±10.44)	17.11 (±8.27)	11.09 (±4.54)	22.17 (±8.53)	10.96 (±4.74)	**6.46 (±1.35)**
led_display	32.41 (±15.60)	**20.47 (±8.25)**	67.20 (±30.40)	23.09 (±7.74)	24.51 (±7.79)	27.35 (±10.54)	**19.19 (±8.48)**	27.37 (±11.10)	**18.95 (±8.43)**
ecoli	241.71 (±159.46)	49.86 (±46.25)	87.89 (±40.23)	74.18 (±46.68)	47.16 (±28.47)	62.86 (±44.15)	**38.89 (±29.90)**	**32.68 (±21.49)**	45.93 (±35.13)
semeion	32.32 (±11.99)	23.58 (±12.58)	26.61 (±7.88)	75.73 (±19.84)	28.52 (±9.18)	23.69 (±8.44)	37.64 (±12.47)	20.58 (±6.11)	**18.57 (±5.13)**
Mean (± std)	99.39 (±134.96)	53.51 (±32.21)	44.22 (±29.01)	38.43 (±24.49)	37.41 (±23.60)	46.46 (±24.54)	37.14 (±20.37)	40.52 (±28.67)	29.25 (±20.43)
Average Accuracy (%)	64.14 (±16.82)	66.03 (±16.36)	63.06 (±15.98)	63.81 (±14.73)	62.46 (±15.58)	63.26 (±16.49)	65.48 (±16.52)	62.38 (±16.17)	65.23 (±15.99)
Average Ranking	7.38	6.56	4.50	4.38	5.06	6.12	4.31	5.00	1.69
Wins	1	1	4	7	1	0	8	3	14
win/tie/loss	–	7/2/7	12/2/2	12/3/1	11/4/1	10/1/5	13/0/3	14/2/0	15/1/0

First of all, we find that NNC obtains the best performance on most datasets except on the dermatology and ecoli dataset. It also outperforms other models in terms of Mean, Average Ranking and Wins. The average number of samples required for NNC is only around 30 while the second best needs, more or less, 37 data points. NNC achieves a higher average ranking 1.69 whereas the second best model DPEA gets a score of 4.31. NNC also shows a clear advantage over random initialization with a win/tie/loss count

Table 6.3: Performance comparison: the ratio of the number of necessary queries of different initialization models to that of random sampling over all the datasets. The medians and median absolute deviations of these ratios are presented. The smaller the ratio, the better the performance. "Mean (± std)" shows means and standard deviations of the medians of the ratios over all test sets. "Average Ranking" reports the average ranking of all methods over all the test sets. "Wins" counts the number of datasets on which an algorithm performs among the best. "win/tie/loss" reports the win/tie/loss counts of other methods versus random sampling.

	Random	k-means++	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	1.00	1.40 (±0.74)	0.53 (±0.17)	**0.45 (±0.13)**	0.66 (±0.19)	1.55 (±0.59)	0.52 (±0.16)	0.84 (±0.24)	**0.44 (±0.11)**
MNIST	1.00	1.55 (±0.62)	**0.75 (±0.26)**	0.85 (±0.23)	**0.77 (±0.21)**	1.51 (±0.58)	**0.68 (±0.22)**	0.81 (±0.29)	0.83 (±0.24)
CIFAR10	**1.00**	**1.21 (±0.39)**	**0.92 (±0.31)**	**1.06 (±0.35)**	1.33 (±0.39)	1.67 (±0.62)	1.77 (±0.56)	**1.00 (±0.33)**	**1.00 (±0.31)**
UCFsports	1.00	0.60 (±0.23)	0.43 (±0.19)	0.75 (±0.32)	0.76 (±0.31)	0.72 (±0.27)	0.76 (±0.31)	0.86 (±0.38)	**0.39 (±0.13)**
GTSRB	1.00	0.97 (±0.33)	**0.76 (±0.20)**	**0.81 (±0.21)**	0.92 (±0.27)	1.04 (±0.39)	**0.87 (±0.20)**	**0.84 (±0.23)**	**0.85 (±0.29)**
Isolet	1.00	0.90 (±0.32)	0.77 (±0.23)	0.73 (±0.20)	0.96 (±0.24)	0.85 (±0.26)	**0.70 (±0.21)**	0.83 (±0.22)	**0.69 (±0.19)**
pendigits	1.00	1.37 (±0.63)	0.79 (±0.31)	0.93 (±0.35)	0.93 (±0.37)	0.65 (±0.23)	1.33 (±0.41)	0.89 (±0.40)	**0.57 (±0.18)**
satimage	1.00	5.59 (±2.12)	1.10 (±0.47)	**0.52 (±0.21)**	0.92 (±0.39)	0.77 (±0.33)	**0.53 (±0.21)**	0.80 (±0.31)	**0.56 (±0.20)**
yeast	1.00	0.34 (±0.22)	**0.22 (±0.15)**	**0.26 (±0.17)**	**0.24 (±0.15)**	**0.25 (±0.13)**	**0.23 (±0.14)**	0.26 (±0.17)	**0.24 (±0.15)**
segment	1.00	1.30 (±0.46)	0.97 (±0.33)	**0.64 (±0.23)**	0.88 (±0.26)	1.57 (±0.66)	**0.64 (±0.23)**	1.13 (±0.49)	**0.63 (±0.19)**
vowel	1.00	1.45 (±0.54)	1.61 (±1.05)	0.83 (±0.26)	1.01 (±0.29)	1.23 (±0.51)	**0.70 (±0.25)**	0.86 (±0.31)	**0.67 (±0.21)**
lowres	1.00	0.20 (±0.14)	0.20 (±0.11)	**0.06 (±0.03)**	0.07 (±0.03)	0.19 (±0.08)	0.17 (±0.07)	0.18 (±0.09)	0.10 (±0.04)
dermatology	1.00	0.52 (±0.24)	0.38 (±0.18)	1.06 (±0.51)	0.74 (±0.41)	0.47 (±0.20)	0.94 (±0.38)	0.42 (±0.23)	**0.31 (±0.11)**
led_display	1.00	**0.67 (±0.21)**	2.15 (±0.95)	0.81 (±0.25)	0.84 (±0.29)	0.87 (±0.29)	**0.62 (±0.23)**	0.88 (±0.34)	**0.66 (±0.28)**
ecoli	1.00	**0.13 (±0.08)**	0.41 (±0.19)	0.30 (±0.19)	**0.20 (±0.10)**	**0.26 (±0.13)**	0.15 (±0.09)	0.15 (±0.08)	0.18 (±0.12)
semeion	1.00	0.67 (±0.27)	0.85 (±0.27)	2.41 (±0.92)	0.92 (±0.33)	0.74 (±0.22)	1.18 (±0.35)	0.63 (±0.20)	**0.57 (±0.19)**
Mean (± std)	1.00	1.18 (±1.26)	0.80 (±0.51)	0.78 (±0.52)	0.76 (±0.33)	0.90 (±0.49)	0.74 (±0.42)	0.71 (±0.30)	0.54 (±0.25)
Average Ranking	7.38	6.38	4.44	4.44	5.25	6.06	4.12	4.81	2.12
Wins	1	3	4	7	3	2	8	2	13
win/tie/loss	0/16/0	7/2/7	11/2/3	12/1/3	7/8/1	9/1/6	12/1/3	9/6/1	15/1/0

of 15/1/0, which means that it dose not perform worse than random sampling on the 16 test sets.

Secondly, among the remaining compared approaches, DPEA and Σ-opt also perform well on most datasets, achieving the best performance on 8 and 7 datasets, respectively. They require about 37 or 38 initial data points on average. Though pWrong obtains a reasonable performance in terms of Mean, i.e. the number is 37.41, it only performs among the best on one single dataset, the MNIST. MaxE also performs well compared to random sampling, i.e. obtaining a win/tie/loss count of 14/2/0. TED and Gen/Disc obtain a slightly worse performance than NNC, DPEA, Σ-opt. k-means++ demonstrates a very poor performance with respects to the average number of required samples. It needs around 54 instances on average to initialize active learning. This number far exceeds that of all other compared methods except random sampling. The main reason could be that it is difficult to set an appropriate k beforehand.

Furthermore, we can observe that random sampling are surpassed by all compared approaches in terms of Mean and Average Ranking. The average number of required samples for random sampling is around 99 while this value of the second worst model k-means++ is merely around 54. One exception is that on the CIFAR10 dataset, random sampling is among the best performing. We also find that some methods, e.g. DPEA, Gen/Disc, k-means++, fail to outperform random initialization on some particular datasets. Overall, however, random initialization is not a good candidate in most cases.

In addition, we also evaluate the relative improvement of the active initialization criteria over random sampling. We consider the ratio $\frac{n_{AL}}{n_R}$ where n_{AL} and n_R are the number

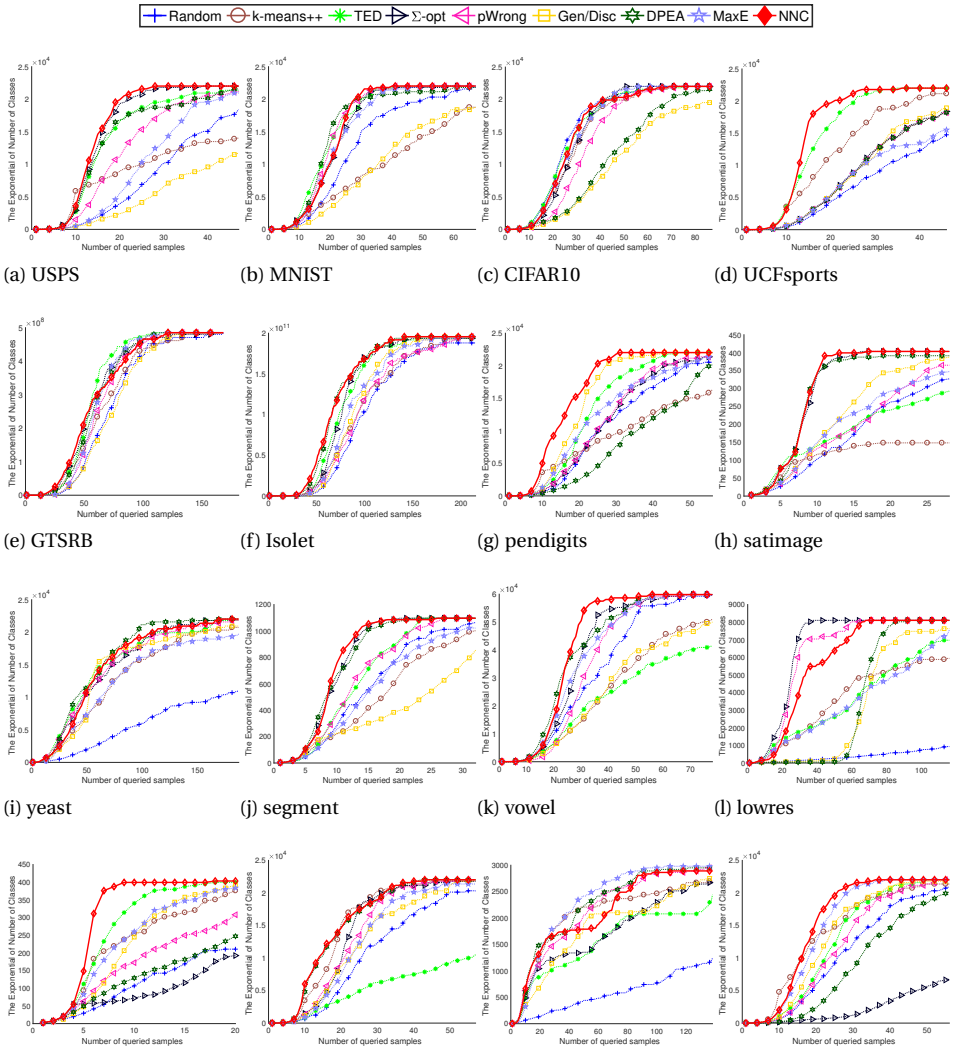
of necessary queries of active initialization models and random sampling, respectively. Table 6.3 reports the medians and median absolute deviations of these ratios over 100 trials. We use a Wilcoxon signed-rank test at a 95% significance level to check whether there is a statistical significant difference between two models. The method which obtains the best performance or performs comparably to the best model is highlighted in bold face and coloured. NNC clearly improves upon random sampling and other models in terms of mean, wins, and average ranking. It outperforms random sampling on 15 datasets except the CIFAR10 on which a tie is reached. The mean of the medians of the ratios of NNC to random sampling is 0.54, which means that NNC can save 46% annotation cost. DPEA, MaxE, pWrong, and Σ-opt also demonstrate good performances, reducing the cost around 24%. k-means++ shows a poorer performance than random sampling, obtaining an average ratio of 1.18 and a win/tie/loss count of 7/2/7.

Figure 6.1 shows the plots of the exponential of the number of discovered classes with respects to the number of queried samples. We use the exponential function for a better illustration of the behaviours of different methods in the latter part of the initialization stage. Obviously, the faster the increase, the better the method. We observe that the NNC clearly outperforms other approaches on most datasets, i.e. the USPS, UCFsports, Isolet, pendigits, satimage, segment, vowel, dermatology, and semeion. Random sampling performs among the best on the CIFAR10 dataset, and obtains poor performance on the remaining datasets, e.g. on the UCFsports, yeast, and ecoli. Σ-opt behaves the best on the lowres dataset while becoming the worst one on the semeion dataset. Gen/Disc performs the worst on several datasets, i.e. the USPS, CIFAR10, GTSRB, and segment. TED is the slowest one to discover classes on the vowel and led_display dataset. Another observation to make is that NNC is never the worst performing one over the 16 test sets.

When an initial set has been constructed, i.e. at least one instance has been queried from each class, we evaluate the classification accuracy on the test set to evaluate how informative the queried initial set is. Figure 6.2 illustrates the average classification accuracy of the initial set chosen by different initialization criteria *w.r.t* the average number of queried samples over 100 trials. The point in the upper left corner means that an algorithm has the overall best performance since it achieves the highest classification accuracy with the smallest number of queried samples. On the contrary, the point in the lower right corner indicates that this methods performs poorly even with a large number of initial instances.

We find that NNC obtains a relatively high accuracy with a reasonable number of queried instances on most datasets. In Figure 6.2a, NNC achieves the second highest accuracy with around 15 samples while k-means++ has the best accuracy with about 50 instances. NNC has a similar accuracy to k-means++ on the MNIST and segment dataset but it only needs about half of the samples required by k-means++. NNC also performs well in terms of the average accuracy on the CIFAR10, pendigits, segment, dermatology, led_display, and semeion.

Table 6.2 also reports the average classification accuracy of all methods over all the test sets. NNC obtains about 65% accuracy with less than 30 samples. DPEA has a similar performance to NNC in terms of the average accuracy, but it requires around 37 instances. k-means++ obtains the best accuracy 66% but with a cost of requiring 24 more samples than that of NNC. Overall, the new method shows good performance in terms of



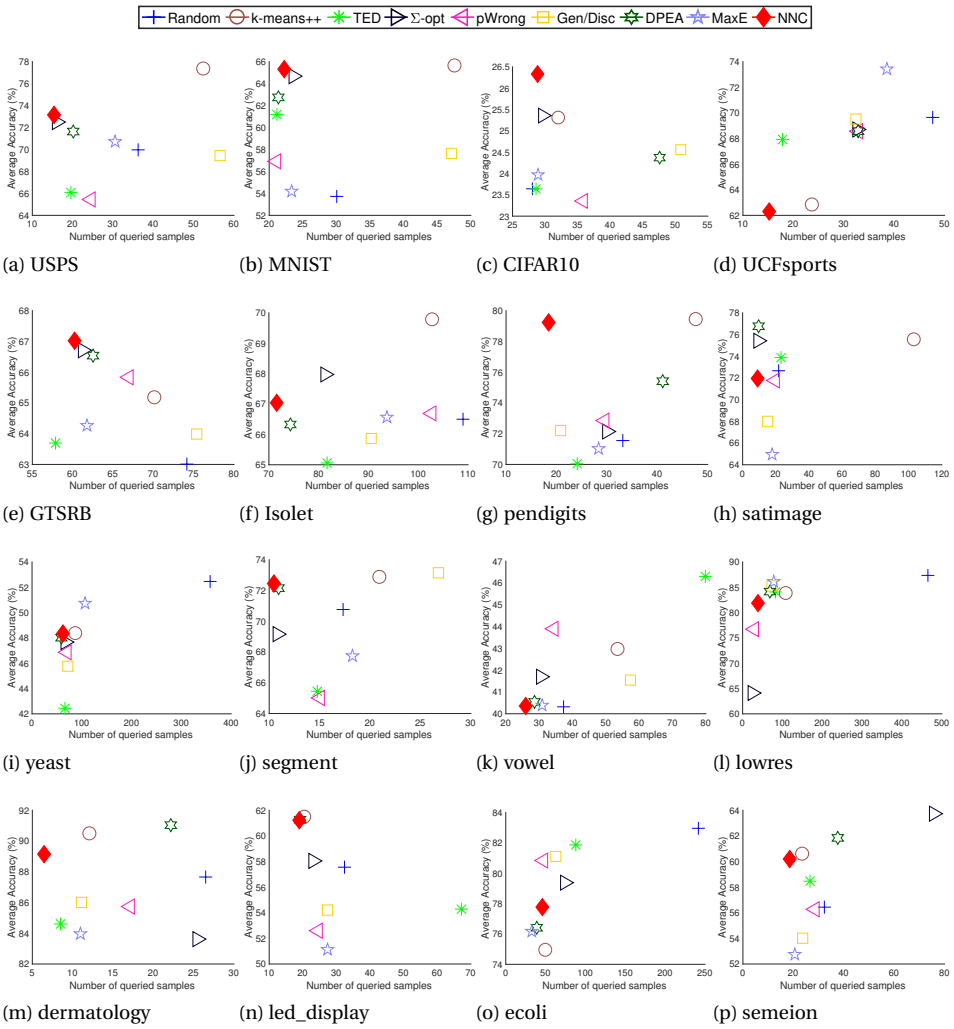
(a) USPS (b) MNIST (c) CIFAR10 (d) UCFsports

(e) GTSRB (f) Isolet (g) pendigits (h) satimage

(i) yeast (j) segment (k) vowel (l) lowres

(m) dermatology (n) led_display (o) ecoli (p) semeion

Figure 6.1: Plots of the exponential of the number of discovered classes *w.r.t* the number of queried samples. The x-axis is the number of queried samples and the y-axis is the exponential of the number of classes. The exponential function is chosen for making the difference of various methods clearer.



(a) USPS (b) MNIST (c) CIFAR10 (d) UCFsports

(e) GTSRB (f) Isolet (g) pendigits (h) satimage

(i) yeast (j) segment (k) vowel (l) lowres

(m) dermatology (n) led_display (o) ecoli (p) semeion

Figure 6.2: Illustration of the average accuracy *w.r.t* the number of queried samples. The x-axis is the average number of queried samples and the y-axis is the average accuracy over 100 trials.

the number of queried samples and also the classification accuracy. DPEA is the second best initialization strategy while Σ-opt slightly performs worse than DPEA.

6.5. DISCUSSION AND CONCLUSION

We investigated how to find a labeled set to initialize active learning algorithms with as few annotations as possible, while at the same time, the initial set consists of at least one instance from each class. For this, we experimented with different methods. Overall, it is interesting to see that no methods can always outperforms random sampling. This

situation is similar to the one we find in active learning (after the initialization phase), where active learning sometimes is surpassed by passive learning [51, 153]. It would be very welcome to understand what causes such behavior and to what extent it can then be avoided.

There are several other questions that may warrant further investigations. It is, for instance, of interest to consider what to do if we cannot know the number of classes in advance? All the initialization criteria presented in this work are under the assumption that we are aware of the total number of classes beforehand. This gives us a clear criterion of when to stop the initialization stage. In real-world applications, however, it is possible that we do not have prior knowledge about how many categories the data contains. In that case, we have to consider finding some, possibly more elaborate criteria that are can also be used to decide on terminating the active initialization process.

Another question is how the chosen initial data set affects the subsequent active learning algorithms. The quality of initial labeled data has an impact on the acquisitions of the subsequently queried samples. Poor selection of the initial set is likely to make active learning methods select uninformative samples and decrease the efficiency of active learning. It is also an open question to what extent active learning algorithms may prefer the one initialization criterion over the other.

As a separate contribution, a new criterion, NNC, was proposed. It selects the samples which minimizes the dissimilarity between unlabeled data and the queried data that has been labeled, where the dissimilarity is measured by the overall nearest neighbor distance. The experiments clearly show that the number of queried samples obtained by this method is much less than that of the algorithms compared to. Furthermore, the initially data selected by our method shows good performance with respect to the classification accuracy in comparison to other approaches and may overall be considered the current method of preference.

7

DISCUSSION

In this chapter, we first summarize the main findings of this thesis and then present possible directions for future research.

7.1. CONCLUSIONS

This thesis focuses on the pool-based active learning and provides new insights for applying active learning to classification tasks. In the following, several important findings are revisited, followed by an analysis of the limitations of the current work. Various possible approaches to address these limitations are also provided.

In Chapter 2, a review of the state-of-the-art active learning algorithms built on logistic regression is presented, followed by an extensive benchmark study of 9 different active learning algorithms on 3 synthetic datasets and 44 real-world datasets. In particular, a visualization technique, what we refer to as preference maps, is proposed to illustrate the characteristic differences of the compared active learning methods. What is somewhat of a surprise is that uncertainty sampling, probably one of the simplest active learning algorithms around, has the best overall performance in comparison with other supposedly more sophisticated methods. Additionally, we find that there is no active learning algorithm that always outperforms random sampling. In some sense, this urges us to seek for safe active learning algorithms which consistently surpass random sampling.

In Chapters 3 and 4, two new retraining-based active learning algorithms are proposed. Chapter 3 proposes to weight the retraining-based criteria with an uncertainty score that is measured by the predictive posterior probability. Chapter 4 views the retraining-based approaches as a variant of classical query-by-committee and proposes to select instances which lead to the largest disagreement among all committee members. We design two particular variances as measures of disagreement. The proposed method in Chapter 4 clearly improves upon the current state of the art on binary and multi-class classification problems. A remaining issue is that these retraining-based methods usually have a large computational cost. Finding proper ways to accelerate these methods can make active learning more practical in real-world use.

In Chapter 5, a new setting of active learning is addressed, where all the required samples should be chosen in a single shot. Pseudo annotators, which uniformly and randomly annotate queried samples, are introduced to provide standard active learning methods with the ability to explore. The exploratory behavior is further enhanced by selecting the most representative sample via minimizing the nearest neighbor distance between unlabeled samples and queried samples. Excellent performance of the proposed method in comparison with state-of-the-art approaches is demonstrated. Currently, the proposed method assumes that there are a small number of labeled instances available. A natural extension is adapting the proposed method to the cold-start case where no labeled data is available. Another possible future direction is combining transfer learning with single shot active learning when there is a labeled source dataset which shares some similarities to our target dataset.

Chapter 6 studies the initialization problem of active learning: how to find an initial set which contains at least one instance per class with as few annotations as possible. We propose a new active initialization criterion, namely the Nearest Neighbor Criterion (NNC). NNC interactively selects the most representative instance from unlabeled data such that the overall nearest neighbor distance between queried data and unlabeled data achieves its minimum value. Extensive experiments demonstrated that our method requires fewer samples than other compared algorithms. One important assumption made in this work is that we are aware of the number of classes beforehand. An open question is finding new initialization criteria or even new active learning algorithms in the case of the number of classes being unknown. Another future direction is studying the influence of the chosen initial data on the following active learning algorithms.

7

7.2. FUTURE WORK

In this thesis, we make a step towards proposing better performing active learning algorithms, along with fairly evaluating and understanding active learning and solving other related issues. There are still many challenging issues left in the field of active learning. We discuss several research directions in which the work can be extended.

7.2.1. HYPERPARAMETER TUNING

Many machine learning models have a set of hyperparameters, which are usually fixed before the training process starts. These hyperparameters have an influence on the performance of the learning models and may have to be tuned carefully. In general, this is done by means of cross validation.

However, in the field of active learning, hyperparameter tuning encounters several difficulties. First, there is usually no additional labeled set for validation. The goal of active learning is reducing the annotation cost, which means that it is unreasonable to spend annotation budgets to construct a validation set. Second, it is unreliable to execute cross validation on the basis of already labeled examples chosen by an active learner [41, 219]. The reason is that these queried samples are highly biased since they are selected according to the active learner's preference. This means that they cannot represent the true underlying data distribution. Another possibility is to correct the sampling bias by using some technique like importance weighting [220]. However, applying im-

portance weighting usually introduces additional hyperparameters, which makes hyperparameters tuning even more complicated.

This situation is even worse for active learning algorithms as they often introduce a hyperparameter to balance between the informativeness and representativeness of the next instances to query [44, 46, 61, 80, 101, 102]. This kind of trade-off parameter is so important that it has a crucial effect on the performance of an active learner. Only if this parameter is properly chosen, the active learner can obtain a relatively good performance. More efforts should be devoted to studying how to tune hyperparameters for active learning.

7.2.2. STOPPING CRITERION

An important practical issue of applying active learning to real-world applications is to determine when to stop active learning. The simplest way to stop active learning is waiting until the annotation budget is finished. However, it is sensible to terminate active learning when continuously annotating more samples does not increase the learning performance. In other words, if we can find a proper stopping criterion which indicates when to end active learning, we can achieve the best performance with a minimum number of queried samples, reducing the annotation cost.

Two aspects make this problem very challenging. The first one is that we cannot accurately estimate the performance of the model trained on currently queried data since we do not have an extra labeled set to evaluate the accuracy. Several studies proposed to address this issue by estimating some criteria which somehow are related to the model's performance [221–226]. For example, Zhu *et al.* [224] used the maximum uncertainty value of all unlabeled data as an indicator of whether the performance is stable. They argued that if the maximum uncertainty score is small enough, the learning performance may not increase. Other types of stopping criteria are the gradient of a learning curve[221], the gradient of the value of a loss function [226].

The second issue is that it is hard to judge when the performance reaches a peak value. It may happen that the performance remains stable for a long time and suddenly witnesses an increase when some samples are labeled. The current solution is using a predefined threshold on some criteria. How to set the threshold is still an open problem. Overall, identifying an appropriate stopping criterion for active learning is challenging.

7.2.3. ACTIVE TESTING

To evaluate the performance of machine learning algorithms, it is often assumed that there exists a test set which consists of a variety of labeled instances. We propose a new setting in which a pool of unlabeled instances is used as test set and we need to actively query a subset of instances to accurately evaluate the performance of a given trained model with as few annotation as possible [227–229]. We refer to this as active testing.

The aims of active learning and active testing are different. The aim of the former is to actively select instances to learn a good model, while the latter one is to estimate the performance as fast and good as possible. Perhaps, an easiest way to perform the evaluation is just randomly annotating a number of instances, what we refer to as passive testing. Other possibilities are adapting some active learning strategies (i.e. uncertainty sampling) to the problem of active testing. For instance, we can use the given trained

model to predict the labels of test instances and select the most uncertain ones, or even the most certain ones. Clearly, this topic needs to be studied further, e.g. when active testing performs better than passive testing and when to stop active testing.

7.2.4. SAFE ACTIVE LEARNING

Various studies have found that the performance of an active learning algorithm sometimes is worse than that of random sampling [28, 152, 230, 231]. This means that the active learning algorithms we use may be not safe, i.e. failing to outperform traditional passive learning on some datasets. An interesting question is whether or not there exists a safe active learner which always performs better than random sampling. If the answer is no, can we identify when an active learner is worse than random sampling? Currently, there are no studies which address the above question. It would be very valuable, for instance, if we know when to use active learning and when to switch to passive learning.

Motivated by the finding in [152] that introducing some degree of randomization in the selection process can help active learning algorithms achieve relatively better performances, a safe active learner may be obtained by integrating classical active learning strategies with some kind of random selection. But, we may expect a safe but less-performing active learner: its performance is consistently better than passive learning and sometimes worse than the best performing active learning algorithms.

This thesis has shown that active learning has the potential to reduce the annotation cost, i.e. achieving accuracies comparable to passive learning using fewer labeled instances. There are still numerous issues that limit the usage of active learning in practical real-world applications. We hope that the work presented in this thesis motivates researchers to devote efforts to fascinating field of active learning.

REFERENCES

[1] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, *Supervised machine learning: A review of classification techniques,* Emerging artificial intelligence applications in computer engineering **160**, 3 (2007).

[2] M. Loog, *Supervised classification: Quite a brief overview,* arXiv preprint arXiv:1710.09230 (2017).

[3] T. Hastie, R. Tibshirani, and J. Friedman, *Unsupervised learning,* in *The elements of statistical learning* (Springer, 2009) pp. 485–585.

[4] X. Zhu, *Semi-supervised learning literature survey,* (2005).

[5] M. D. Kohli, R. M. Summers, and J. R. Geis, *Medical image data and datasets in the era of machine learning—whitepaper from the 2016 c-mimi meeting dataset session,* Journal of digital imaging **30**, 392 (2017).

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database,* in *CVPR09* (2009).

[7] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, *Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video,* in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017) pp. 7464–7473.

[8] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, *Two-dimensional active learning for image classification,* in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (IEEE, 2008) pp. 1–8.

[9] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, *Multi-class active learning for image classification,* in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (IEEE, 2009) pp. 2372–2379.

[10] A. J. Joshiy, F. Porikli, and N. Papanikolopoulos, *Multi-class batch-mode active learning for image classification,* in *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (IEEE, 2010) pp. 1873–1878.

[11] X. Li and Y. Guo, *Adaptive active learning for image classification,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013) pp. 859–866.

[12] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, *Cost-effective active learning for deep image classification,* IEEE Transactions on Circuits and Systems for Video Technology (2016).

[13] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, *Batch mode active learning and its application to medical image classification,* in *Proceedings of the 23rd International Conference on Machine Learning* (ACM, 2006) pp. 417–424.

[14] P. T. Saito, C. T. Suzuki, J. F. Gomes, P. J. de Rezende, and A. X. Falcão, *Robust active learning for the diagnosis of parasites,* Pattern Recognition **48**, 3572 (2015).

[15] J. Cheng and K. Wang, *Active learning for image retrieval with co-svm,* Pattern recognition **40**, 330 (2007).

[16] R. Liu, Y. Wang, T. Baba, D. Masumoto, and S. Nagata, *Svm-based active feedback in image retrieval using clustering and unlabeled data,* Pattern Recognition **41**, 2645 (2008).

[17] D. Zhang, F. Wang, Z. Shi, and C. Zhang, *Interactive localized content based image retrieval with multiple-instance active learning,* Pattern Recognition **43**, 478 (2010).

[18] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, *Active learning methods for remote sensing image classification,* IEEE Transactions on Geoscience and Remote Sensing **47**, 2218 (2009).

[19] D. Tuia, J. Muñoz-Marí, and G. Camps-Valls, *Remote sensing image segmentation by active queries,* Pattern Recognition **45**, 2180 (2012).

[20] A. Samat, J. Li, S. Liu, P. Du, Z. Miao, and J. Luo, *Improved hyperspectral image classification by active learning using pre-designed mixed pixels,* Pattern Recognition **51**, 43 (2016).

[21] Z. Wang, B. Du, L. Zhang, and L. Zhang, *A batch-mode active learning framework by querying discriminative and representative samples for hyperspectral image classification,* Neurocomputing **179**, 88 (2016).

[22] C. Deng, X. Liu, C. Li, and D. Tao, *Active multi-kernel domain adaptation for hyperspectral image classification,* Pattern Recognition (2017).

[23] A. McCallum and K. Nigam, *Employing em and pool-based active learning for text classification,* in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98 (1998) pp. 350–358.

[24] S. Tong and D. Koller, *Support vector machine active learning with applications to text classification,* The Journal of Machine Learning Research **2**, 45 (2002).

[25] J. Zhu, H. Wang, B. K. Tsou, and M. Ma, *Active learning with sampling by uncertainty and density for data annotations,* Audio, Speech, and Language Processing, IEEE Transactions on **18**, 1323 (2010).

[26] D. Cai and X. He, *Manifold adaptive experimental design for text categorization,* Knowledge and Data Engineering, IEEE Transactions on **24**, 707 (2012).

[27] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan, *Multi-criteria-based active learning for named entity recognition,* in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (Association for Computational Linguistics, 2004) p. 589.

[28] Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny, and H. Xu, *A study of active learning methods for named entity recognition in clinical text,* Journal of biomedical informatics **58**, 11 (2015).

[29] C. A. Thompson, M. E. Califf, and R. J. Mooney, *Active learning for natural language parsing and information extraction,* in *ICML* (1999) pp. 406–414.

[30] M. Tang, X. Luo, and S. Roukos, *Active learning for statistical natural language parsing,* in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Association for Computational Linguistics, 2002) pp. 120–127.

[31] G. Tur, R. E. Schapire, and D. Hakkani-Tur, *Active learning for spoken language understanding,* in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, Vol. 1 (IEEE, 2003) pp. I–I.

[32] R. Jin and L. Si, *A bayesian approach toward active learning for collaborative filtering,* in *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (AUAI Press, 2004) pp. 278–285.

[33] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, *Non-myopic active learning for recommender systems based on matrix factorization,* in *Information Reuse and Integration (IRI), 2011 IEEE International Conference on* (IEEE, 2011) pp. 299–303.

[34] M. Elahi, M. Braunhofer, F. Ricci, and M. Tkalcic, *Personality-based active learning for collaborative filtering recommender systems,* in *Congress of the Italian Association for Artificial Intelligence* (Springer, 2013) pp. 360–371.

[35] B. Lamche, U. Trottmann, and W. Wörndl, *Active learning strategies for exploratory mobile recommender systems,* in *Proceedings of the 4th Workshop on Context-Awareness in Retrieval and Recommendation* (ACM, 2014) pp. 10–17.

[36] B. Settles, *Active learning literature survey,* University of Wisconsin, Madison **52**, 11 (2010).

[37] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, *Stream-based joint exploration-exploitation active learning,* in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (IEEE, 2012) pp. 1560–1567.

[38] D. Sculley, *Online active learning methods for fast label-efficient spam filtering.* in *CEAS*, Vol. 7 (2007) p. 143.

[39] R. Ganti and A. Gray, *Upal: Unbiased pool based active learning,* in *Artificial Intelligence and Statistics* (2012) pp. 422–431.

[40] N. Roy and A. Mccallum, *Toward optimal active learning through sampling estimation of error reduction,* in *In Proc. 18th International Conf. on Machine Learning* (2001) pp. 441–448.

[41] Y. Baram, R. E. Yaniv, and K. Luz, *Online choice of active learning algorithms,* Journal of Machine Learning Research **5**, 255 (2004).

[42] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, *Semisupervised svm batch mode active learning with applications to image retrieval,* ACM Transactions on Information Systems (TOIS) **27**, 16 (2009).

[43] N. V. Cuong, W. S. Lee, and N. Ye, *Near-optimal adaptive pool-based active learning with general loss,* in *Conference on Uncertainty in Artificial Intelligence (UAI)* (2014).

[44] S. Chakraborty, V. Balasubramanian, Q. Sun, S. Panchanathan, and J. Ye, *Active batch selection via convex relaxations with guaranteed solution bounds,* IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, 1945 (2015).

[45] W. Cai, Y. Zhang, Y. Zhang, S. Zhou, W. Wang, Z. Chen, and C. Ding, *Active learning for classification with maximum model change,* ACM Transactions on Information Systems (TOIS) **36**, 15 (2017).

[46] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, *Exploring representativeness and informativeness for active learning,* IEEE Transactions on Cybernetics **47**, 14 (2017).

[47] P. Juszczak, *Learning to recognise: A study on one-class classification and active learning,* Ph.D. thesis, TU Delft, Delft University of Technology (2006).

[48] M. Wang and X.-S. Hua, *Active learning in multimedia annotation and retrieval: A survey,* ACM Transactions on Intelligent Systems and Technology (TIST) **2**, 10 (2011).

[49] Y. Fu, X. Zhu, and B. Li, *A survey on instance selection for active learning,* Knowledge and Information Systems **35**, 249 (2013).

[50] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, *Active learning: A survey,* in *Data Classification: Algorithms and Applications* (2014) pp. 571–606.

[51] Y. Yang and M. Loog, *A benchmark and comparison of active learning for logistic regression,* Pattern Recognition **83**, 401 (2018).

[52] K. Yu, J. Bi, and V. Tresp, *Active learning via transductive experimental design,* in *Proceedings of the 23rd International Conference on Machine Learning* (ACM, 2006) pp. 1081–1088.

[53] M. Ji and J. Han, *A variance minimization criterion to active learning on graphs,* in *Artificial Intelligence and Statistics* (2012) pp. 556–564.

[54] Y. Ma, R. Garnett, and J. Schneider, *σ-optimality for active learning on gaussian random fields,* in *Advances in Neural Information Processing Systems* (2013) pp. 2751–2759.

[55] E. Elhamifar, G. Sapiro, and S. S. Sastry, *Dissimilarity-based sparse subset selection,* IEEE transactions on pattern analysis and machine intelligence **38**, 2182 (2016).

[56] R. Hu, S. J. Delany, and B. Mac Namee, *Egal: Exploration guided active learning for tcbr,* in *International Conference on Case-Based Reasoning* (Springer, 2010) pp. 156–170.

[57] X. He, W. Min, D. Cai, and K. Zhou, *Laplacian optimal design for image retrieval,* in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (ACM, 2007) pp. 119–126.

[58] H. Wang, L. Du, P. Zhou, L. Shi, Y. Qian, and Y.-D. Shen, *Experimental design with multiple kernels,* in *Data Mining (ICDM), 2015 IEEE International Conference on* (IEEE, 2015) pp. 419–428.

[59] M. Gönen and E. Alpaydın, *Multiple kernel learning algorithms,* Journal of machine learning research **12**, 2211 (2011).

[60] L. Shi and Y.-D. Shen, *Diversifying convex transductive experimental design for active learning.* in *IJCAI* (2016) pp. 1997–2003.

[61] K. Brinker, *Incorporating diversity in active learning with support vector machines,* in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003) pp. 59–66.

[62] H. S. Seung, M. Opper, and H. Sompolinsky, *Query by committee,* in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92 (1992) pp. 287–294.

[63] N. A. H. Mamitsuka, *Query learning strategies using boosting and bagging,* in *Machine Learning: Proceedings of the Fifteenth International Conference (ICML'98)*, Vol. 1 (Morgan Kaufmann Pub, 1998) pp. 1–9.

[64] L. Breiman, *Bagging predictors,* Machine learning **24**, 123 (1996).

[65] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting,* Journal of computer and system sciences **55**, 119 (1997).

[66] P. Melville and R. J. Mooney, *Diverse ensembles for active learning,* in *Proceedings of the twenty-first international conference on Machine learning* (ACM, 2004) p. 74.

[67] I. Muslea, S. Minton, and C. A. Knoblock, *Selective sampling with redundant views,* in *AAAI/IAAI* (2000) pp. 621–626.

[68] S.-S. Ho and H. Wechsler, *Query by transduction,* IEEE transactions on pattern analysis and machine intelligence **30**, 1557 (2008).

[69] I. Dagan and S. P. Engelson, *Committee-based sampling for training probabilistic classifiers,* in *Machine Learning Proceedings 1995* (Elsevier, 1995) pp. 150–157.

[70] P. Melville, S. M. Yang, M. Saar-Tsechansky, and R. Mooney, *Active learning for probability estimation using jensen-shannon divergence,* in *European Conference on Machine Learning* (Springer, 2005) pp. 268–279.

[71] M. Saar-Tsechansky and F. Provost, *Active sampling for class probability estimation and ranking,* Machine learning **54**, 153 (2004).

[72] A. Culotta and A. McCallum, *Reducing labeling effort for structured prediction tasks,* in *AAAI*, Vol. 5 (2005) pp. 746–751.

[73] D. D. Lewis and 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.

[74] T. Scheffer, C. Decomain, and S. Wrobel, *Active hidden markov models for information extraction,* in *International Symposium on Intelligent Data Analysis* (Springer, 2001) pp. 309–318.

[75] Y. Guo and R. Greiner, *Optimistic active-learning using mutual information.* in *IJCAI*, Vol. 7 (2007) pp. 823–829.

[76] A. I. Schein and L. H. Ungar, *Active learning for logistic regression: an evaluation,* Machine Learning **68**, 235 (2007).

[77] B. Settles, M. Craven, and S. Ray, *Multiple-instance active learning,* in *Advances in neural information processing systems* (2008) pp. 1289–1296.

[78] A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler, *Labeling examples that matter: Relevance-based active learning with gaussian processes,* in *German Conference on Pattern Recognition* (Springer, 2013) pp. 282–291.

[79] A. Freytag, E. Rodner, and J. Denzler, *Selecting influential examples: Active learning with expected model output changes,* in *Computer Vision–ECCV 2014* (Springer, 2014) pp. 562–577.

[80] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, *Semi-supervised svm batch mode active learning for image retrieval,* in *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on* (IEEE, 2008) pp. 1–7.

[81] Y. Yang and M. Loog, *Active learning using uncertainty information,* in *ICPR* (2016) pp. 111–118.

[82] S.-J. Huang, R. Jin, and Z.-H. Zhou, *Active learning by querying informative and representative examples,* IEEE Transactions on Pattern Analysis and Machine Intelligence **10**, 1936 (2014).

[83] C. K. Dagli, S. Rajaram, and T. S. Huang, *Leveraging active learning for relevance feedback using an information theoretic diversity measure,* in *International Conference on Image and Video Retrieval* (Springer, 2006) pp. 123–132.

[84] B. Settles and M. Craven, *An analysis of active learning strategies for sequence labeling tasks,* in *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2008) pp. 1070–1079.

[85] Y. Gu, Z. Jin, and S. C. Chiu, *Active learning with maximum density and minimum redundancy,* in *International Conference on Neural Information Processing* (Springer, 2014) pp. 103–110.

[86] T. He, S. Zhang, J. Xin, P. Zhao, J. Wu, X. Xian, C. Li, and Z. Cui, *An active learning approach with uncertainty, representativeness, and diversity,* The Scientific World Journal **2014** (2014).

[87] J. Zhu, H. Wang, T. Yao, and B. K. Tsou, *Active learning with sampling by uncertainty and density for word sense disambiguation and text classification,* in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (Association for Computational Linguistics, 2008) pp. 1137–1144.

[88] D. Cohn, L. Atlas, and R. Ladner, *Improving generalization with active learning,* Machine learning **15**, 201 (1994).

[89] M.-F. Balcan, A. Beygelzimer, and J. Langford, *Agnostic active learning,* in *Proceedings of the 23rd international conference on Machine learning* (ACM, 2006) pp. 65–72.

[90] M.-F. Balcan, A. Broder, and T. Zhang, *Margin based active learning,* in *International Conference on Computational Learning Theory* (Springer, 2007) pp. 35–50.

[91] S. Hanneke, *A bound on the label complexity of agnostic active learning,* in *Proceedings of the 24th international conference on Machine learning* (ACM, 2007) pp. 353–360.

[92] S. Dasgupta, D. J. Hsu, and C. Monteleoni, *A general agnostic active learning algorithm,* in *Advances in neural information processing systems* (2008) pp. 353–360.

[93] A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang, *Agnostic active learning without constraints,* in *Advances in Neural Information Processing Systems* (2010) pp. 199–207.

[94] S. Dasgupta, *Two faces of active learning,* Theoretical computer science **412**, 1767 (2011).

[95] C. Zhang and K. Chaudhuri, *Beyond disagreement-based agnostic active learning,* in *Advances in Neural Information Processing Systems* (2014) pp. 442–450.

[96] S. Hanneke *et al.*, *Theory of disagreement-based active learning,* Foundations and Trends® in Machine Learning **7**, 131 (2014).

[97] B. Demir, C. Persello, and L. Bruzzone, *Batch-mode active-learning methods for the interactive classification of remote sensing images,* IEEE Transactions on Geoscience and Remote Sensing **49**, 1014 (2011).

[98] S. Patra and L. Bruzzone, *A cluster-assumption based batch mode active learning technique,* Pattern Recognition Letters **33**, 1042 (2012).

[99] Z. Xu, R. Akella, and Y. Zhang, *Incorporating diversity and density in active learning for relevance feedback,* in *European Conference on Information Retrieval* (Springer, 2007) pp. 246–257.

[100] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, *Batch mode active sampling based on marginal probability distribution matching,* ACM Transactions on Knowledge Discovery from Data (TKDD) **7**, 13 (2013).

[101] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, *Multi-class active learning by uncertainty sampling with diversity maximization,* International Journal of Computer Vision **113**, 113 (2015).

[102] Z. Wang and J. Ye, *Querying discriminative and representative samples for batch mode active learning,* ACM Transactions on Knowledge Discovery from Data (TKDD) **9**, 17 (2015).

[103] B. Demir and L. Bruzzone, *A novel active learning method in relevance feedback for content-based remote sensing image retrieval,* IEEE Transactions on Geoscience and Remote Sensing **53**, 2323 (2015).

[104] O. Berger-Tal, J. Nathan, E. Meron, and D. Saltz, *The exploration-exploitation dilemma: a multidisciplinary framework,* PloS one **9**, e95693 (2014).

[105] S. J. Gershman, *Deconstructing the human algorithms for exploration,* Cognition **173**, 34 (2018).

[106] T. Osugi, D. Kim, and S. Scott, *Balancing exploration and exploitation: A new algorithm for active machine learning,* in *Data Mining, Fifth IEEE International Conference on* (IEEE, 2005) pp. 8–pp.

[107] Y. Guo and D. Schuurmans, *Discriminative batch mode active learning,* in *Proceedings of the 20th International Conference on Neural Information Processing Systems* (2007) pp. 593–600.

[108] Y. Guo, *Active instance sampling via matrix partition,* in *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, NIPS'10 (Curran Associates Inc., USA, 2010) pp. 802–810.

[109] Y. Chen and A. Krause, *Near-optimal batch mode active learning and adaptive submodular optimization,* in *Proceedings of The 30th International Conference on Machine Learning* (2013) pp. 160–168.

[110] R. Mehrotra and E. Yilmaz, *Representative & informative query selection for learning to rank using submodular functions,* in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 2015) pp. 545–554.

[111] T. Zhang and F. Oles, *The value of unlabeled data for classification problems,* in *Proceedings of the Seventeenth International Conference on Machine Learning,(Langley, P., ed.)* (Citeseer, 2000) pp. 1191–1198.

[112] Q. Gu, T. Zhang, and J. Han, *Batch-mode active learning via error bound minimization,* in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence* (AUAI Press, 2014) pp. 300–309.

[113] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* in *Advances in neural information processing systems* (2012) pp. 1097–1105.

[114] P. Stanitsas, A. Cherian, A. Truskinovsky, V. Morellas, and N. Papanikolopoulos, *Active convolutional neural networks for cancerous tissue recognition,* (ICIP, 2017).

[115] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, *Cost-effective active learning for deep image classification,* IEEE Transactions on Circuits and Systems for Video Technology **27**, 2591 (2017).

[116] O. Sener and S. Savarese, *Active learning for convolutional neural networks: A core-set approach,* stat **1050**, 21 (2018).

[117] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, *Suggestive annotation: A deep active learning framework for biomedical image segmentation,* in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2017) pp. 399–407.

[118] S. Gavves, T. Mensink, T. Tommasi, C. Snoek, and T. Tuytelaars, *Active transfer learning with zero-shot priors: Reusing past datasets for future tasks,* in *Proceedings ICCV 2015* (2015) pp. 2731–2739.

[119] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, *Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally,* in *IEEE conference on computer vision and pattern recognition, Hawaii* (2017) pp. 7340–7349.

[120] S.-J. Huang, J.-W. Zhao, and Z.-Y. Liu, *Cost-effective training of deep cnns with active model adaptation,* arXiv preprint arXiv:1802.05394 (2018).

[121] G. Contardo, L. Denoyer, and T. Artieres, *A meta-learning approach to one-step active learning,* arXiv preprint arXiv:1706.08334 (2017).

[122] P. Bachman, A. Sordoni, and A. Trischler, *Learning algorithms for active learning,* in *Proceedings of the 34th International Conference on Machine Learning* (International Convention Centre, Sydney, Australia, 2017) pp. 301–310.

[123] M. Woodward and C. Finn, *Active one-shot learning,* arXiv preprint arXiv:1702.06559 (2017).

[124] A. Holub, P. Perona, and M. C. Burl, *Entropy-based active learning for object recognition,* in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on* (IEEE, 2008) pp. 1–8.

[125] Y. Yang and M. Loog, *A variance maximization criterion for active learning,* Pattern Recognition **78**, 358 (2018).

[126] C. Käding, A. Freytag, E. Rodner, A. Perino, and J. Denzler, *Large-scale active learning with approximations of expected model output changes,* in *German Conference on Pattern Recognition* (Springer, 2016) pp. 179–191.

[127] S. C. Hoi, R. Jin, and M. R. Lyu, *Batch mode active learning with applications to text categorization and image retrieval,* IEEE Transactions on knowledge and data engineering **21**, 1233 (2009).

[128] T. Kanamori, *Pool-based active learning with optimal sampling distribution and its information geometrical interpretation,* Neurocomputing **71**, 353 (2007).

[129] T. Kanamori and H. Shimodaira, *Active learning algorithm using the maximum weighted log-likelihood estimator,* Journal of Statistical Planning and Inference **116**, 149 (2003).

[130] A. Liu, L. Reyzin, and B. D. Ziebart, *Shift-pessimistic active learning using robust bias-aware prediction.* in *AAAI* (2015) pp. 2764–2770.

[131] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, *An optimization based framework for dynamic batch mode active learning,* Advances in Neural Information Processing (2010).

[132] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, *Generalized batch mode active learning for face-based biometric recognition,* Pattern Recognition **46**, 497 (2013).

[133] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, *Adaptive batch mode active learning,* IEEE Transactions on Neural Networks and Learning Systems (2014).

[134] C.-C. Chang and B.-H. Liao, *Active learning based on minimization of the expected path-length of random walks on the learned manifold structure,* Pattern Recognition **71**, 337 (2017).

[135] C. E. Shannon, *A mathematical theory of communication,* Bell System Technocal Journal **27**, 379 (1948).

[136] Y. Grandvalet and Y. Bengio, *Semi-supervised learning by entropy minimization,* in *Advances in neural information processing systems* (2004) pp. 529–536.

[137] A. Atkinson, A. Donev, and R. Tobias, *Optimum experimental designs, with SAS* (Oxford Univ. Press, UK, 2007).

[138] K. Lu, J. Zhao, and Y. Wu, *Hessian optimal design for image retrieval,* Pattern Recognition **44**, 1155 (2011).

[139] M. Lichman, *UCI machine learning repository,* (2013).

[140] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition,* Proceedings of the IEEE **86**, 2278 (1998).

[141] K. Lang, *Newsweeder: Learning to filter netnews,* in *Proceedings of the Twelfth International Conference on Machine Learning* (1995) pp. 331–339.

[142] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, *Do we need hundreds of classifiers to solve real world classification problems?* The Journal of Machine Learning Research **15**, 3133 (2014).

[143] C.-C. Chang and C.-J. Lin, *Libsvm: a library for support vector machines,* ACM Transactions on Intelligent Systems and Technology (TIST) **2**, 27:1 (2011).

[144] P. W. Frey and D. J. Slate, *Letter recognition using holland-style adaptive classifiers,* Machine Learning **6**, 161 (1991).

[145] X. Zhu, J. Lafferty, and Z. Ghahramani, *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* (2003) pp. 58–65.

[146] T. Tommasi and T. Tuytelaars, *A testbed for cross-dataset analysis,* in *European Conference on Computer Vision* (Springer, 2014) pp. 18–31.

[147] C.-L. Li, C.-S. Ferng, and H.-T. Lin, *Active learning with hinted support vector machine.* in *ACML* (2012) pp. 221–235.

[148] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, *Liblinear: A library for large linear classification,* The Journal of Machine Learning Research **9**, 1871 (2008).

[149] D. J. Cook and N. C. Krishnan, *Activity learning: discovering, recognizing, and predicting human behavior from sensor data* (John Wiley & Sons, 2015).

[150] M. Loog and A. C. Jensen, *Semi-supervised nearest mean classification through a constrained log-likelihood,* Neural Networks and Learning Systems, IEEE Transactions on **26**, 995 (2015).

[151] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, *Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction,* IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 1553 (2004).

[152] I. Guyon, G. C. Cawley, G. Dror, and V. Lemaire, *Results of the active learning challenge.* Active Learning and Experimental Design@ AISTATS **16**, 19 (2011).

[153] M. Loog and Y. Yang, *An empirical investigation into the inconsistency of sequential active learning,* in *Pattern Recognition (ICPR), 2016 23rd International Conference on* (IEEE, 2016) pp. 210–215.

[154] J. Kremer, K. Steenstrup Pedersen, and C. Igel, *Active learning with support vector machines,* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **4**, 313 (2014).

[155] S.-J. Huang, S. Chen, and Z.-H. Zhou, *Multi-label active learning: Query type matters.* in *IJCAI* (2015) pp. 946–952.

[156] E. A. Cherman, Y. Papanikolaou, G. Tsoumakas, and M. C. Monard, *Multi-label active learning: key issues and a novel query strategy,* Evolving Systems , 1 (2017).

[157] A. Harpale, *Multi-Task Active Learning*, Ph.D. thesis, Carnegie Mellon University (2012).

[158] Y. Zhang, *Multi-task active learning with output constraints.* in *AAAI* (2010).

[159] M.-A. Carbonneau, E. Granger, and G. Gagnon, *Bag-level aggregation for multiple instance active learning in instance classification problems,* arXiv preprint arXiv:1710.02584 (2017).

[160] C. Persello, A. Boularias, M. Dalponte, T. Gobakken, E. Naesset, and B. Schoelkopf, *Cost-sensitive active learning with lookahead: Optimizing field surveys for remote sensing data classification,* IEEE Transactions on Geoscience and Remote Sensing **52**, 6652 (2014).

[161] X. Wang, T.-K. Huang, and J. Schneider, *Active transfer learning under model shift,* in *International Conference on Machine Learning* (2014) pp. 1305–1313.

[162] J.-J. Zhu and J. Bento, *Generative adversarial active learning,* arXiv preprint arXiv:1702.07956 (2017).

[163] M. Huijser and J. C. van Gemert, *Active decision boundary annotation with deep generative models,* in *2017 IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2017) pp. 5296–5305.

[164] J. Attenberg and F. Provost, *Inactive learning?: difficulties employing active learning in practice,* ACM SIGKDD Explorations Newsletter **12**, 36 (2011).

[165] M. Loog, J. H. Krijthe, and A. C. Jensen, *On measuring and quantifying performance: Error rates, surrogate loss, and an example in ssl,* in *Handbook of Pattern Recognition and Computer Vision*, edited by C. H. Chen (World Scientific, 2016) 5th ed., Chap. 1.3, pp. 53–68.

[166] B. Settles, *From theories to queries: Active learning in practice,* in *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010* (2011) pp. 1–18.

[167] D. D. Lewis and J. Catlett, *Heterogeneous uncertainty sampling for supervised learning,* in *Proceedings of the eleventh International Conference on Machine Learning* (1994) pp. 148–156.

[168] C. Campbell, N. Cristianini, A. Smola, *et al.*, *Query learning with large margin classifiers,* in *ICML* (2000) pp. 111–118.

[169] W. Cai, Y. Zhang, S. Zhou, W. Wang, C. Ding, and X. Gu, *Active learning for support vector machines with maximum model change,* in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2014) pp. 211–226.

[170] C. Kading, A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler, *Active learning and discovery of object categories in the presence of unnameable instances,* in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (IEEE, 2015) pp. 4343–4352.

[171] S.-J. Huang, R. Jin, and Z.-H. Zhou, *Active learning by querying informative and representative examples,* in *Advances in neural information processing systems* (2010) pp. 892–900.

[172] Y. Yang and M. Loog, *A benchmark and comparison of active learning methods for logistic regression,* arXiv preprint (2016).

[173] J. Platt *et al.*, *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,* Advances in large margin classifiers **10**, 61 (1999).

[174] R. P. Duin and D. M. Tax, *Classifier conditional posterior probabilities,* in *Advances in pattern recognition* (Springer, 1998) pp. 611–619.

[175] Z. Wang, S. Yan, and C. Zhang, *Active learning with adaptive regularization,* Pattern Recognition **44**, 2375 (2011).

[176] H. T. Nguyen and A. Smeulders, *Active learning using pre-clustering,* in *Proceedings of the twenty-first international conference on Machine learning* (ACM, 2004) p. 79.

[177] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, *Representative sampling for text classification using support vector machines,* in *European Conference on Information Retrieval* (Springer, 2003) pp. 393–407.

[178] R. Wang and S. Kwong, *Active learning with multi-criteria decision making systems,* Pattern Recognition **47**, 3106 (2014).

[179] Z. Wang and J. Ye, *Querying discriminative and representative samples for batch mode active learning,* in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2013) pp. 158–166.

[180] L. P. Evans, N. M. Adams, and C. Anagnostopoulos, *Estimating optimal active learning via model retraining improvement,* arXiv preprint arXiv:1502.01664 (2015).

[181] Y. Zhen and D.-Y. Yeung, *Sed: supervised experimental design and its application to text classification,* in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (ACM, 2010) pp. 299–306.

[182] M. Li and I. K. Sethi, *Confidence-based active learning,* IEEE Transactions on Pattern Analysis and Machine Intelligence **28**, 1251 (2006).

[183] R. Wang, S. Kwong, and D. Chen, *Inconsistency-based active learning for support vector machines,* Pattern Recognition **45**, 3751 (2012).

[184] L. Fei-Fei and P. Perona, *A bayesian hierarchical model for learning natural scene categories,* in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 2 (IEEE, 2005) pp. 524–531.

[185] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, *The german traffic sign recognition benchmark: a multi-class classification competition,* in *Neural Networks (IJCNN), The 2011 International Joint Conference on* (IEEE, 2011) pp. 1453–1460.

[186] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images,* (2009).

[187] A. Oliva and A. Torralba, *Modeling the shape of the scene: A holistic representation of the spatial envelope,* International Journal of Computer Vision **42**, 145 (2001).

[188] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection,* in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1 (IEEE, 2005) pp. 886–893.

[189] C. Schuldt, I. Laptev, and B. Caputo, *Recognizing human actions: a local svm approach,* in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 3 (IEEE, 2004) pp. 32–36.

[190] M. D. Rodriguez, J. Ahmed, and M. Shah, *Action mach a spatio-temporal maximum average correlation height filter for action recognition,* in *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on* (IEEE, 2008) pp. 1–8.

[191] S. Sadanand and J. J. Corso, *Action bank: A high-level representation of activity in video,* in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (IEEE, 2012) pp. 1234–1241.

[192] M. Fanty and R. Cole, *Spoken letter recognition,* in *Advances in Neural Information Processing Systems* (1991) pp. 220–226.

[193] T. De Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, and D. Windridge, *An evaluation of bags-of-words and spatio-temporal shapes for action recognition,* in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on* (IEEE, 2011) pp. 344–351.

[194] A. Klaser, M. Marszałek, and C. Schmid, *A spatio-temporal descriptor based on 3d-gradients,* in *BMVC 2008-19th British Machine Vision Conference* (British Machine Vision Association, 2008) pp. 275–1.

[195] W. Liu, J. Wang, Y. Mu, S. Kumar, and S. Chang, *Compact hyperplane hashing with bilinear functions,* in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012* (2012).

[196] K. Wei, R. K. Iyer, and J. A. Bilmes, *Submodularity in data subset selection and active learning,* in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (2015) pp. 1954–1963.

[197] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, *Dermatologist-level classification of skin cancer with deep neural networks,* Nature **542**, 115 (2017).

[198] E. Elhamifar, G. Sapiro, and R. Vidal, *See all by looking at a few: Sparse modeling for finding representative objects,* in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (IEEE, 2012) pp. 1600–1607.

[199] C. Boutsidis, M. W. Mahoney, and P. Drineas, *An improved approximation algorithm for the column subset selection problem,* in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics, 2009) pp. 968–977.

[200] B. J. Frey and D. Dueck, *Clustering by passing messages between data points,* science **315**, 972 (2007).

[201] D. Arthur and S. Vassilvitskii, *k-means++: The advantages of careful seeding,* in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics, 2007) pp. 1027–1035.

[202] C.-L. Li, C.-S. Ferng, and H.-T. Lin, *Active learning using hint information,* Neural computation **27**, 1738 (2015).

[203] J. J. Hull, *A database for handwritten text recognition research,* IEEE Transactions on Pattern Analysis and Machine Intelligence **16**, 550 (1994).

[204] B. Gong, Y. Shi, F. Sha, and K. Grauman, *Geodesic flow kernel for unsupervised domain adaptation,* in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (IEEE, 2012) pp. 2066–2073.

[205] C. Fang, Y. Xu, and D. N. Rockmore, *Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias,* in *International Conference on Computer Vision* (2013).

[206] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan, *Active learning in recommender systems,* in *Recommender systems handbook* (Springer, 2015) pp. 809–846.

[207] D. Kottke, A. Calma, D. Huseljic, G. Krempl, and B. Sick, *Challenges of reliable, realistic and comparable active learning evaluation,* in *Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning* (2017) pp. 2–14.

[208] X. You, R. Wang, and D. Tao, *Diverse expected gradient active learning for relative attributes,* IEEE Transactions on Image Processing **23**, 3203 (2014).

[209] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, *Batch mode active sampling based on marginal probability distribution matching,* in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2012) pp. 741–749.

[210] J. Kang, K. R. Ryu, and H.-C. Kwon, *Using cluster-based sampling to select initial training set for active learning in text classification,* in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Springer, 2004) pp. 384–388.

[211] R. Hu, B. Mac Namee, and S. J. Delany, *Off to a good start: Using clustering to select the initial training set in active learning.* in *FLAIRS Conference* (2010).

[212] V. Souza, R. G. Rossi, G. E. Batista, and S. O. Rezende, *Unsupervised active learning techniques for labeling training sets: An experimental evaluation on sequential data,* Intelligent Data Analysis **21**, 1061 (2017).

[213] D. Pelleg and A. W. Moore, *Active learning for anomaly and rare-category detection,* in *Advances in neural information processing systems* (2005) pp. 1073–1080.

[214] J. He and J. G. Carbonell, *Nearest-neighbor-based active learning for rare category detection,* in *Advances in neural information processing systems* (2008) pp. 633–640.

[215] T. Fincham Haines and T. Xiang, *Active learning using dirichlet processes for rare class discovery and classification,* in *British Machine Vision Conference* (University of Bath, 2011).

[216] T. M. Hospedales, S. Gong, and T. Xiang, *A unifying theory of active discovery and learning,* in *European Conference on Computer Vision* (Springer, 2012) pp. 453–466.

[217] T. M. Hospedales, S. Gong, and T. Xiang, *Finding rare classes: Active learning with generative and discriminative models,* IEEE transactions on knowledge and data engineering **25**, 374 (2013).

[218] D. S. Hochbaum and D. B. Shmoys, *A best possible heuristic for the k-center problem,* Mathematics of operations research **10**, 180 (1985).

[219] A. Ali, R. Caruana, and A. Kapoor, *Active learning with model selection.* in *AAAI* (2014) pp. 1673–1679.

[220] M. Sugiyama, M. Krauledat, and K.-R. MÃžller, *Covariate shift adaptation by importance weighted cross validation,* Journal of Machine Learning Research **8**, 985 (2007).

[221] F. Laws and H. Schätze, *Stopping criteria for active learning of named entity recognition,* in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (Association for Computational Linguistics, 2008) pp. 465–472.

[222] A. Vlachos, *A stopping criterion for active learning,* Computer Speech & Language **22**, 295 (2008).

[223] M. Bloodgood and K. Vijay-Shanker, *A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping,* in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning* (Association for Computational Linguistics, 2009) pp. 39–47.

[224] J. Zhu, H. Wang, E. Hovy, and M. Ma, *Confidence-based stopping criteria for active learning for data annotation,* ACM Transactions on Speech and Language Processing (TSLP) **6**, 3 (2010).

[225] M. Ghayoomi, *Using variance as a stopping criterion for active learning of frame assignment,* in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing* (Association for Computational Linguistics, 2010) pp. 1–9.

[226] W. Wang, W. Cai, and Y. Zhang, *Stability-based stopping criterion for active learning,* in *2014 IEEE International Conference on Data Mining (ICDM)* (IEEE, 2014) pp. 1019–1024.

[227] C. Sawade, N. Landwehr, S. Bickel, and T. Scheffer, *Active risk estimation,* in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (Citeseer, 2010) pp. 951–958.

[228] C. Sawade, N. Landwehr, and T. Scheffer, *Active estimation of f-measures,* in *Advances in Neural Information Processing Systems* (2010) pp. 2083–2091.

[229] N. Katariya, A. Iyer, and S. Sarawagi, *Active evaluation of classifiers on large datasets,* in *Data Mining (ICDM), 2012 IEEE 12th International Conference on* (IEEE, 2012) pp. 329–338.

[230] G. C. Cawley, *Baseline methods for active learning.* in *Active Learning and Experimental Design@ AISTATS* (2011) pp. 47–57.

[231] R. L. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. P. Wiechmann, *Active learning for clinical text classification: is it better than random sampling?* Journal of the American Medical Informatics Association **19**, 809 (2012).

SUMMARY

In recent decades, the availability of a large amount of data has propelled the field of machine learning enormously. Machine learning, however, relies heavily on the availability of annotated data, typically labels indicating to which class a data instance belongs. With the huge amounts of data, this raises the question of how to efficiently annotate data, certainly when having limited resources. This thesis addresses the particular challenge of using as few annotations as possible, while at the same time, maintaining a good learning performance. For that we utilize active learning, which iteratively chooses the most valuable instances as to obtain the labels from an oracle (e.g. a human expert). Though many studies have demonstrated that active learning can reduce the annotation cost, there are still several issues that limit its practical use. This thesis makes a further step forwards making active learning more practical for real-world applications.

We first provide a benchmark and comparison of six different categories of active learning algorithms built on logistic regression. This work provides a better understanding of the underlying characteristics of various active learners and illustrates the potential benefits of using such techniques, but it also provides many cases for which active learning fails to outperform passive learning (i.e. randomly selecting instances for labeling). Those failed cases motivate us to propose two novel active learning methods that show a clear advantage over passive learning. The first one proposes to weight the so-called retraining-based criteria with an uncertainty score that is measured by the estimated posterior probability. The second one measures the usefulness of unlabeled instances according to the variance of the predictive probability. This method takes an additional step towards practical active learning, clearly outperforming current state of the art on binary and multi-class classification tasks.

We further consider two realistic issues when applying active learning to real-world problems. One is how to find an initial set that contains at least one instance per class to start the active labeling cycle. The other one is dealing with the absence of human annotators in the interactive labeling loop. We propose new approaches to tackle the above problems and observe good performance compared to existing methods. This thesis concludes with an analysis of the contributions and limitations of our work, as well as research directions that deserve further studies.

We hope that this thesis also inspires others to make active learning more suitable for real-world applications.

SAMENVATTING

In de afgelopen decennia heeft de beschikbaarheid van grote hoeveelheden gegevens het gebied van de *machine learning* tot enorme hoogten gedreven. De mogelijkheid tot machinaal leren is echter sterk afhankelijk van de beschikbaarheid van geannoteerde data—meestal labels die aangeven tot welke klasse een gegevensinstantie behoort. Gegeven de soms enorme hoveelheden input data die beschikbaar is, roept dit de vraag op hoe men op een efficiënte manier deze gegevens kan annoteren, zeker indien men beperkte middelen heeft. Dit proefschrift gaat in op de specifieke uitdaging om zo weinig mogelijk annotaties te gebruiken en tegelijkertijd een zo goed mogelijke leerprestatie te bewerkstelligen. Hiervoor gebruiken we actief leren, dat iteratief de meest waardevolle instanties kiest om de labels aan een orakel (bijvoorbeeld een menselijke expert) te vragen. Hoewel veel onderzoek heeft aangetoond dat actief leren de annotatiekosten kan verlagen, zijn er nog steeds verschillende problemen die het praktische gebruik ervan beperken. Dit proefschrift maakt een verdere stap voorwaarts om actief leren praktischer te maken voor toepassingen in de echte wereld.

We bieden eerst een benchmark en vergelijking van zes verschillende categorieën van actieve-leeralgoritmen gebaseerd op logistische regressie. Dit werk biedt een beter begrip van de onderliggende kenmerken van verschillende actieve leerlingen en illustreert de potentiële voordelen van het gebruik van dergelijke technieken. Het laat echter ook veel gevallen zien waarvoor actief leren niet beter presteert dan passief leren (dat wil zeggen waarbij het selecteren van instanties ter labeling gebeurt op basis van willekeur). Deze tekortkoming motiveert ons om twee nieuwe actieve-leermethoden voor te stellen die een duidelijk voordeel hebben ten opzichte van passief leren. De eerste suggereert de zogenaamde hertrainingscriteria te wegen met een onzekerheidsscore die wordt gemeten aan de hand van de geschatte posterieure waarschijnlijkheid. De tweede meet het nut van niet-gelabelde instanties volgens de variantie van deze voorspellende waarschijnlijkheid. Deze methode neemt een extra stap in de richting van praktisch actief leren en is duidelijk beter dan de huidige stand van de techniek wat binaire en multiklasse classificatietaken betreft.

We beschouwen verder twee praktische vraagstukken betreffende het toepassen van actief leren op problemen in de echte wereld. De eerste betreft het vinden van een initiële set die ten minste één instantie per klasse bevat om de actieve-labelcyclus te starten. De andere heeft te maken met de afwezigheid van menselijke annotators in de interactieve labellus. We stellen nieuwe benaderingen voor om bovenstaande problemen aan te pakken en rapporteren goede resultaten in vergelijking met bestaande methoden. Dit proefschrift wordt afgesloten met een analyse van de bijdragen en beperkingen van ons werk, evenals onderzoeksrichtingen die verdere studie behoeven.

We hopen dat dit proefschrift inspireert om actief leren meer geschikt te maken voor toepassingen in de echte wereld.

ACKNOWLEDGEMENTS

The past four years of pursuing a PhD is a wonderful journey with full of challenges, hard work, guidance, as well as supports from supervisors, friends, and family. I would like to acknowledge all who have contributed to this work.

First and foremost, my highest gratitude goes to my supervisor Marco Loog. Dear Marco, this thesis is greatly attributable to your extensive guidance and strong support. Thank you for your valuable suggestions and discussions in our regular meetings, which always improve my ideas and motivate me go one step further. Thank you for always guiding me to think critically, not only on others' work but also on our own work. Thank you for correcting my scientific writing and criticizing on my presentation from which I learned a lot. I do appreciate that you help me translate the summary of my thesis into Dutch language. Marco, it is a great pleasure to work with you.

Secondly, I would like to thank my promotor Marcel Reinders for giving me the opportunity to join in the PRB group at TU Delft. Thank you for your helpful lecture on how to prepare for propositions in the PRB retreat and carefully helping me revise this thesis. I am also grateful to Bob Duin, David Tax, Jan van Gemert, Hayley Hung, for your fascinating coffeetalks, insightful discussions, and constructive suggestions.

Next, I would express my sincere gratitude to all the members in the Pattern Recognition and Bioinformatics group. I really enjoy the four years spent with all of you, for all the discussions, (crazy) Thursday drinking, sitting-together-lunches, wonderful social activities, especially the PR&CV retreat. My thanks to you all, Ahmed, Alexander, Alexey, Alex S, Amin, Amogh, Arlin, Bart, Christine, Christian, Erdogan, Ekin, Gorkem, Hamdi, Jesse, Joana, John, Julian, Laura, Lorenzo, Lu, Nora, Osman, Reza, Robbert, Ruud, Saskia, Seyran, Silvia, Sjoerd, Stavros, Stephanie, Tamim, Thies, Thomas, Tom, Taygun, Veronika, Wenjie, Wouter, Yan, Yancong, Yanxia, Yuan, Yunqiang and everyone else, for maintaining a harmonious working atmosphere. Particular thanks to Wenjie, for being my intimate friend and always supporting me no matter what. I thank Jesse and Wouter for always being willing to answer my questions. Thanks to Laura and Erik, you are always the source of our happiness and contentment. My gratitude to Taygun for being my intimate officemate for almost four years.

Life is becoming fantastic because of the companionship of many friends who stay in the Netherlands. Thanks to Cai Jie, Wang Minchang, Qu Wenhua, for becoming roommates over years and taking care of each other. Special thanks to Wenjie and Xiangrong, for always supporting me and taking trips with me. Many thanks for all you have done for me. Thank Yuanhao for your nice dinners which takes you hours to prepare. I owe thanks to Chen Yue, Chen Linying, Chen Jiao, Fang Haixing, Huang Yamin, Hongzhi, Ma Mingxiao, Shi Chunsheng, Tang Jinyu, Wang Meng, Wang Pengling, Xu Fei, Zhang Rong, Zhang Tian. We started to pursue a PhD at the same time and supported each other for four years. I am also grateful to Chang Xin, Chen Zhanglin, Cheng Dan, Guo Wenjing, Han Dun, He Zhidong, Lao Lingling, Li Xinchao, Lin Xiao, Lin Qin, Liu Qiang, Liu Zishun,

Liu Yueting, Liu Yu, Ma Wenbin, Pan Sining, Qu Bo, Ren Zhijie, Sui Congbiao, Wu Long, Yan Fei, Yang Jie, Yang Song, Ye Qingqing, Yu Jingtao, Yuan Shuai, Zhan Xiuxiu, Zong Haohua, for all your kindness and friendship.

Thanks to all the NUDTers: Cao Yang, Fang Jian, Fu Xiang, Guo Yanming, He Lei, Huang Xu, Li Guangming, Li Yuan, Li Yunlong, Li Xinyi, Liu Liang, Qiu Sihang, Ren Shanshan, Shi Peiteng, Wang Xiaohui, Xie Xu, Xin Yu, Yang Zhiwei, Yin Jiapeng, Zhang Laobing, Zhang Mingxing, Zhang Xiaoke, Zhao Yue, Zhu Baozhou. It was a splendid memory that we went ice-skating and had a nice dinner together.

I would like to thank all members of the doctoral committee for their careful reading and helpful comments.

Last but not least, I also want to thank my master supervisors Dr. Tu Dan and Prof. Li Guohui. Thank you for leading me to the fascinating world of scientific research. Thanks to Prof. Zhang Jun for giving me so much encouragement during these years. I am also grateful to those former colleagues at NUDT, Cai Fei, Guo Qiang, Hou Jinxing, Huang Kuihua, Li Bo, Li Shuohao, Qu Zhiqiang, Ren Weiya, Sun Boliang, Tang Yu, Tang Min, Wang Fenglei, Zhang Hui. I would like to express my thanks to you for all your friendships. Special thanks to Lei Jun and Jiao Wen, for your great friendship and endless help. I am deeply grateful to Yin Xiaoqing, Li Weili, You Hanlin, Luo Tinjing, Hu Chi, for your unconditional support.

Lastly, I would like to express my deepest gratitude to my family, without whom I would have not come this far. My sincere gratitude goes to my parents for their selfless support and truly love. Many thanks to my older sister, Yaping Yang, for her endless love and encouragement. I deeply wish my family health and happiness.

Yazhou
September, 2018
Delft, the Netherlands

CURRICULUM VITÆ

Yazhou Yang received the B.S. degree in information system engineering, the M.E degree of control science and engineering from the National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively. He is currently a PhD candidate in the Pattern Recognition Laboratory at Delft University of Technology, Delft, the Netherlands. His current research interests include active learning, semi-supervised learning and image classification.

LIST OF PUBLICATIONS

- **Yazhou Yang**, and Marco Loog. "*A Variance Maximization Criterion for Active Learning.*" Pattern Recognition 78 (2018): 358-370.
- **Yazhou Yang**, and Marco Loog. "*A Benchmark and Comparison of Active Learning for Logistic Regression.*" Pattern Recognition 83C (2018): 401-415.
- **Yazhou Yang**, and Marco Loog. "*Single Shot Active Learning using Pseudo Annotators.*" under review, major revision, 2018.
- **Yazhou Yang**, and Marco Loog. "*To Actively Initialize Active Learning.*" (under review, 2018).
- **Yazhou Yang**, and Marco Loog. "*Active Learning using Uncertainty Information.*" In *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 2646-2651, 2016.
- Marco Loog, and **Yazhou Yang**. "*An Empirical Investigation into the Inconsistency of Sequential Active Learning.*" In *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 210-215, 2016.
- **Yazhou Yang**, Dan Tu, and Guohui Li. "*Gait Recognition using Flow Histogram Energy Image.*" In *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 444-449, 2014.