JIMMA UNIVERSITY
INISTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING

ASSIGNMENT OF DATA MINING
ACTIVE LEARNING IN ADVANCED LEARNING TECHNIQUES

GROUP MEMBERS **IDNO**

1.Beri Beriso……. Ru4345/07
2.Habtamu Debasa……...Ru0562/08

Introduction

Active learning is an active area of research in Data Mining and Machine Learning Communities. Active learning methods come from a parallel between active educational methods and learning theory. The learner is from now a statistical model instead of a student. The interactions between the student and the teacher correspond to the opportunity for the model to interact with a human expert. The examples are situations used by the model to generate knowledge on the problem. Active learning methods allow the model to interact with its environment by selecting the more "informative" situations. This paper restricts the active learning domain to the machine learning paradigm

These two scenarios are adaptive sampling and selective sampling. In the first case the examples are instantiations of the input variables of the model. The active strategy is not restricted to a pool of instances and can explore the entire space of variation of the input variables, looking for areas to be sampled

Many modern machine learning techniques require large amounts of training data to reach their full potential. However, annotated data is hard and expensive to obtain, notably in specialized domains where only experts whose time is scarce and precious can provide reliable labels. Active learning(AL) aims to ease the data collection process by automatically deciding which instances an annotator should label to train an algorithm as quickly and effectively as possible. Over the years many AL strategies have been developed for various classification tasks, without any one of them clearly outperforming others in all cases. Consequently, a number of meta-AL approaches have been proposed to automatically select the best strategy. Recent examples include bandit algorithms and reinforcement learning approaches. A common limitation of these methods is that they cannot go beyond combining pre-existing hand-designed heuristics. Besides, they require reliable assessment of the classification performance which is problematic because the annotated data is scarce. In this paper, we overcome these limitations thanks to two features of our approach. First, we look at a whole continuum of AL strategies instead of combinations of pre-specified heuristics.

Second, we bypass the need to evaluate the classification quality from application-specific data because we rely on experience from previous tasks and can seamlessly transfer strategies to new domains. More specifically, we formulate Learning Active Learning (LAL) as a regression problem. Given a trained classifier and its output for a specific sample without a label, we predict the reduction in generalization error that can be expected by adding the label to that data point. that we can train this regression function on synthetic data by using simple features, such as the variance of the classifier output or the predicted probability distribution over possible labels for 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. specific data point. The features for the regression are not domain-specific and this enables to apply the regress or trained on synthetic data directly to other classification problems. Furthermore, if a sufficiently large annotated set can be provided initially, the regress or can be trained on it instead of on synthetic data. The resulting AL strategy is then tailored to the particular problem at hand. We show that LAL works well on real data from several different domains such as

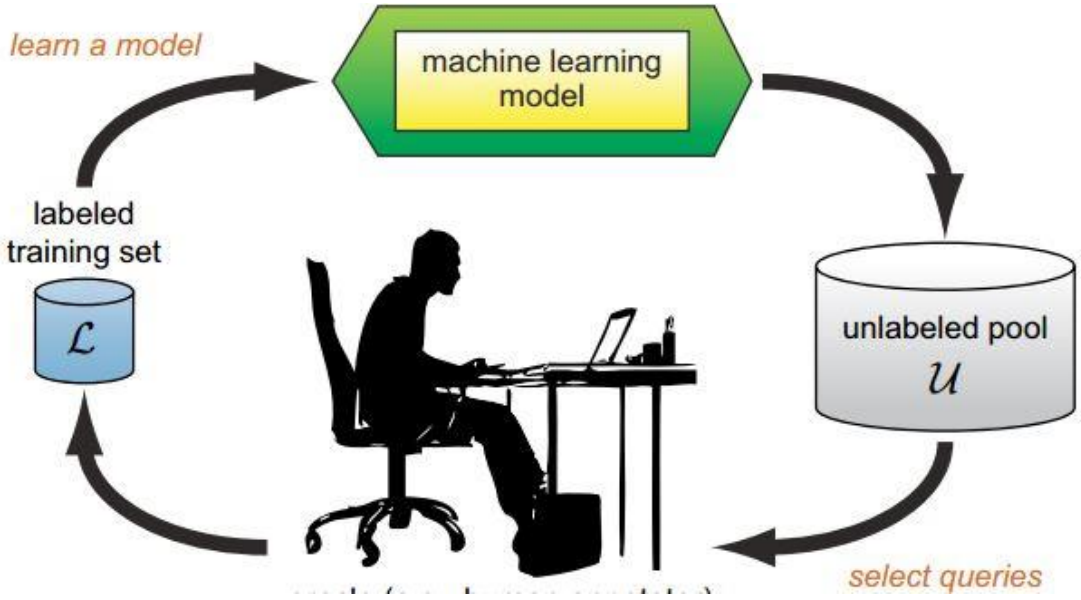
biomedical imaging, economics, molecular biology and high energy physics. This query selection strategy outperforms competing methods without requiring hand-crafted heuristics and at a comparatively low computational cost.

The extensive development of AL in the last decade has resulted in various strategies. They include uncertainty sampling query-by-committee expected model change expected error or variance minimization and information gain. Among these, uncertainty sampling is both simple and computationally efficient. This makes it one of the most popular strategies in real applications. In short, it suggests labeling samples that are the most uncertain, i.e., closest to the classifier's decision boundary. The above methods work very well in cases such as the ones depicted in the top but often fail in the more difficult ones depicted in the bottom row.

Among AL methods, some cater to specific classifiers, such as those relying on Gaussian processes or to specific applications, such as natural language processing sequence labeling tasks visual recognition semantic segmentation foreground-background segmentation and preference learning Moreover, various query strategies aim to maximize different performance metrics, as evidenced in the case of multi-class classification. However, there is no one algorithm that consistently outperforms all others in all applications. Meta-learning algorithms have been gaining in popularity in recent years but few of them tackle the problem of learning AL strategies. Abram et al. combine several known heuristics with the help of a bandit algorithm.

Another approach is introduced by Ebert et al. It involves balancing exploration and exploitation in the choice of samples with a Markov decision process.
The two main limitations of these approaches are as follows. First, they are restricted to combining already existing techniques and second, their success depends on the ability to estimate the classification performance from scarce annotated data. The data-driven nature of LAL helps to overcome these limitations.

Active Learning Examples



There are several scenarios in which active learners may pose queries, and there are also several different query strategies that have been used to use to decide which instance are most informative. In this section, I present two illustrative In this section I present two illustrative examples in the pool-based active learning setting (in which queries are selected from a large pool of unlabeled instances ∪) using an uncertainty sampling query strategy (which selects the instance in the pool of unlabeled instance in the pool about which the model is least certain how to label). Sections2and3describeallthe active learning scenarios and query strategy frameworks in more detail.

Description of basic algorithm

In this section we briefly introduce the active leaning framework along with uncertainty sampling(US), the most frequently-used AL heuristic. Then, we motivate why a data-driven approach can improve AL strategies and how it can deal with the situations where US fails. We select US as are preventative method because it is popular and widely applicable, however the behavior that we describe is typical for a wide range of AL strategies.

Given a machine learning model and a pool of unlabeled data, the goal of AL is to select which data should be annotated in order to learn the model as quickly as possible. In practice, this means that instead of asking experts to annotate all the data, we select iteratively and adaptively which data points should be annotated next. In this paper we are interested in classifying data points from a target dataset $Z = \{(x1, y1), \ldots, (xN, yN)\}$, where xi is a D-dimensional feature vector and $yi \in \{0, 1\}$ is its binary label. We choose a probabilistic classifier f that can be trained on some $Lt \dashrightarrow Z$ to map features to labels, $ft(xi) = \hat{}yi$, through the predicted probability $pt(yi = y \mid xi)$. The standard AL procedure unfolds as follows.1. The algorithm starts with a small labeled training dataset $Lt \dashrightarrow Z$ and large pool of un annotated data $Ut = Z \setminus Lt$ with $t = 0$.
A classifier ft is trained using Lt A query selection procedure picks an instance $x \nleftarrow Ut$ to be annotated at the next iteration. $x \nleftarrow$ is given a label $y \nleftarrow$ by an oracle. The labeled and unlabeled sets are updated. t is incremented, and steps 2–5 iterate until the desired accuracy is achieved

Uncertainty Sampling

Perhaps the simplest and most commonly used query framework is *uncertainty sampling* (Lewis and Gale, 1994). In this framework, an active learner queries the instances about which it is least certain how to label. This approach is often straightforward for probabilistic learning models. For example, when using a probabilistic model for binary classification, uncertainty sampling simply queries the instance whose posterior probability of being positive is nearest 0.5 (Lewis and Gale, 1994; Lewis and Catlett, 1994).

For problems with three or more class labels, a more general uncertainty sampling variant might query the instance whose prediction is the *least confident:*

$x^*_{LC} = \text{argmax } 1 - P_e(y|x)$,

where $y = \text{argmax}_y P_e(y|x)$, or the class label with the highest posterior probability under the model 6. One way to interpret this uncertainty measure is the expected 0/1-loss, i.e., the model's belief that it will

mislabel x. This sort of strategy has been popular, for example, with statistical sequence models in information extraction tasks (Culotta and McCallum, 2005; Settles and Craven, 2008). This is because the most likely label sequence (and its associated likelihood) can be efficiently computed using dynamic programming.

However, the criterion for the least confident strategy only considers information about the most probable label. Thus, it effectively "throws away" information about the remaining label distribution. To correct for this, some researchers use a different multi-class uncertainty sampling variant called *margin sampling* (Scheffer etal., 2001):

x^*_M = argminP_e(fo*i*\x) - P_Q(fo\x),X

where fo and y_2 are the first and second most probable class labels under the model, respectively. Margin sampling aims to correct for a shortcoming in least confident strategy, by incorporating the posterior of the second most likely label. Intuitively, instances with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Instances with small margins are more ambiguous, thus knowing the true label would help the model discriminate more effectively between them. However, for problems with very large label sets, the margin approach still ignores much of the output distribution for the remaining classes.

A more general uncertainty sampling strategy (and possibly the most popular) uses *entropy* (Shannon, 1948) as an uncertainty measure:

x^*_H = argmax - ^ P_e(yi \x) log P_e(y*yi**x*),Xi

where y_i ranges over all possible labelings. Entropy is an information-theoretic measure that represents the amount of information needed to "encode" a distribution. As such, it is often thought of as a measure of uncertainty or impurity in machine learning. For binary classification, entropy-based sampling reduces to the margin and least confident strategies above; in fact all three are equivalent to querying the instance with a class posterior closest to 0.5. However, the entropy- based approach generalizes easily to probabilistic multi-label classifiers and probabilistic models for more complex structured instances, such as sequences (Settles and Craven, 2008) and trees (Hwa, 2004).

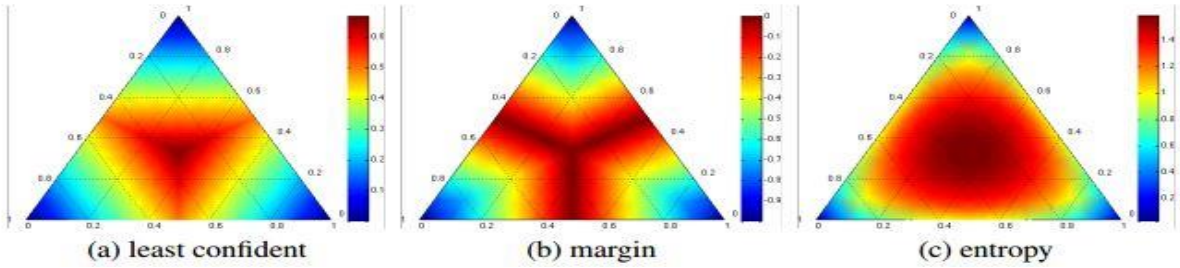


Figure 5: Heatmaps illustrating the query behavior of common uncertainty measures in a three-label classification problem. Simplex corners indicate where one label has very high probability, with the opposite edge showing the probability range for the *other* two classes when that label has very low probability. Simplex centers represent a uniform posterior distribution. The most informative query region for each strategy is shown in dark red, radiating from the centers.

probability (and thus little model uncertainty). The main differences lie in the rest of the probability space. For example, the entropy measure does not favor instances where only one of the labels is highly unlikely (i.e., along the outer side edges), because the model is fairly certain that it is not the true label. The least confident and margin measures, on the other hand, consider such instances to be useful if the model cannot distinguish between the remaining two classes. Empirical comparisons of these measures (e.g., Korner and Wrobel, 2006; Schein and Ungar, 2007; Settles and Craven, 2008) have yielded mixed results, suggesting that the best strategy may be application-dependent (note that all strategies still generally outperform passive baselines). Intuitively, though, entropy seems appropriate if the objective function is to minimize log-loss, while the other two (particularly margin) are more appropriate if we aim to reduce classification error, since they prefer instances that would help the model better discriminate among specific classes.

Uncertainty sampling strategies may also be employed with non-probabilistic classifiers. One of the first works to explore uncertainty sampling used a decision tree classifier (Lewis and Catlett, 1994). Similar approaches have been applied to active learning with nearest-neighbor (a.k.a. "memory-based" or "instance- based") classifiers (Fujii et al., 1998; Lindenbaum et al., 2004), by allowing each neighbor to vote on the class label of x, with the proportion of these votes representing the posterior label probability. Tong and Koller (2000) also experiment with an uncertainty sampling strategy for support vector machines—or SVMs— that involves querying the instance closest to the linear decision boundary. This last approach is analogous to uncertainty sampling with a probabilistic binary linear classifier, such as logistic regression or naive Bayes.

So far we have only discussed classification tasks, but uncertainty sampling is also applicable in *regression* problems (i.e., learning tasks where the output variable is a continuous value rather than a set of discrete class labels). In this setting, the learner simply queries the unlabeled instance for which the model has the highest output variance in its prediction. Under a Gaussian assumption, the entropy of a random variable is a monotonic function of its variance, so this approach is very much in same the spirit as entropy-based uncertainty sampling for classification. Closed-form approximations of output variance can be computed for a variety of models, including Gaussian random fields (Cressie, 1991) and neural networks (MacKay, 1992). Active learning for regression problems has a long history in the statistics literature, generally referred to as *optimal experimental design* (Federov, 1972). Such approaches shy away from uncertainty sampling in lieu of more sophisticated strategies, which we will explore further in Section 3.5.

Experimental Methodology

This section is separated into three distinct subparagraphs, following the procedure that was respected before we execute our experiments. A short description of each one's content is provided here: how to find the appropriate text datasets, select the most favoring AL approach based on the properties of the collected datasets and finally selecting representative learning algorithms so as to examine their efficacy over the field of classifying text data using AL.

Application area of active learning

Active learning for parameter estimation

We formalize this idea by assuming that some subset C of the variables are controllable. The learner can select a subset of variables QC and a particular instantiation q to Q. The request Q:=qis called a query. The result of such a query is called the response and it is a randomly sampled instance x of all the non-query variables, conditioned on Q:=q. Thus(q; x)is a complete data instance

The interpretation of such a request depends on our active learning setting. In a selective query, we assume that(q;x)is selected at random from instances satisfying the query. ,(q; x)is a random instance from P (Xj Q=q). (The same statement holds for the pool-based variant of selective active learning.) In an interventional query, we assume that our graph G is a causal model and that x is the result of an experiment where we intervene in the model and explicitly set the variables in Q to take the values q.

In the Bayesian network parameter estimation task, an active learner has a querying function that takes G and p(), and selects a query Q:=q. It takes the resulting complete instance(q; x), and uses it to update its distribution p()to obtain a posteriorp0(). It then repeats the process, usingp0forp. We note that the parameter distribution p() summarizes all the relevant aspects of the data seen so far, so that we do not need to maintain the history

of previous instances. To fully specify the algorithm, we need to address two issues: we need to describe how our parameter distribution is updated given that(q;x)is not a random sample, and we need to construct a mechanism for selecting the next query based onp. Updating Using an Actively Sampled Instance Clearly, the answer to the first of these questions depends on the active learning mechanism since the sampling distribution for(q; x)is different in the two cases.

Let us first consider the case of selective active learning. Assume for simplicity that our query is Q=q for a single node Q. First, it is clear that we cannot use the resulting instance(q; x)to update the parameters of the node Q itself: the fact that we deliberately sought and found an instance where Q=q does not tell us anything about the overall

probability of such instances within the population. However, we also have a more subtle problem. Consider a parent U of Q. Although(q; x)does give us information about the distribution of U, it is not information that we can conveniently use. Intuitively, P(UjQ=q)is sampled from a distribution specified by a complex formula involving multiple parameters. For example, consider the Smoking network: Cancer! Smoking where we must choose the value of Smoking in advance. It then hard to get a coherent idea of the prior probability of cancer. We sidestep this problem simply by ignoring the information provided by(q; x)on nodes that are “upstream” of Q.

Discussion

We now motivate the need for LAL by presenting two toy examples. In the first one, US is empirically observed to be the best greedy approach, but in the second it makes suboptimal decisions. Let us consider simple two-dimensional datasets *Z* and *Z 0* drawn from the same distribution with an equal number of points in each class (Fig. 1, left). The data in each class comes

from a Gaussian distribution with a different mean and the same isotropic covariance. We can initialize the AL procedure of Sec. 3.1 with one sample from each class and its respective label: $L0 = \{(x1, 0), (x2, 1)\} \dashrightarrow Z$ and $U0 = Z \setminus L0$. Here we train a simple logistic regression classifier f on $L0$ and then test it on $Z\ 0$. If $|\ Z\ 0\ |$ is large, the test error can be considered as a good approximation of the generalization error: $`0 = P(x\ 0, y\ 0)\ 2\ Z\ 0\ `(\hat{}y, y\ 0)$, where $y\hat{} = f0(x\ 0)$. Let us try to label every point x from $U0$ one by one, form a new labeled set $Lx = L0\ [\ (x, y)$

and check what error a new classifier fx yields on $Z\ 0$, that is, $`x = P(x\ 0, y\ 0)\ 2\ Z\ 0\ `(\hat{}y, y\ 0)$, where $y\hat{} = fx(x\ 0)$. The difference between errors obtained with classifiers constructed on $L0$ and Lx indicates how much the addition of a new data point x reduces the generalization error: $x = `0\ `x$. We plot x for the 0/1 loss function, averaged over 10 000 experiments as a function of the predicted probability $p0$ (Fig. 1, left). By design, US would select a data point with probability of class 0 close to 0.5.

We observe that in this experiment, the data sample with $p0$ closest to 0.5 is indeed the one that yields the greatest error reduction.

The goal in active feature acquisition is to select the most informative features to obtain during training, rather than randomly or exhaustively acquiring all new features for all training instances

There have been multiple approaches to solving this problem of feature acquisition, e.g:

Zheng and Padmanabhan proposed two “single-pass” approaches

1. Attempt to impute the missing values, and then acquire the ones about which the model has least confidence
2. An alternative, imputing these values, training a classifier on the imputed training instances, and only acquiring feature values for the instances which are misclassified

Incremental active feature acquisition may acquire values for a few salient features at a time

Most active learning research are oriented to work well when the queries are selected in serial

In the presence of multiple annotators, this method slows down the learning process

Batch-Mode active learning allows the learner to query the instances in groups, this approach is more suited for a parallel labeling

Challenge is to form the query set. Querying the “Q-best” queries according to the instance-level strategy usually causes information overlap and is not helpful

Research is oriented at approaches that specifically incorporate diversity in the queries

Conclusion

The goal of machine learning is to extract patterns from the world which can then be used to forward scientific understanding, create automated processes, assist with labor intensive tasks, and much more besides. However, much of machine learning relies on data, and

gathering data is typically expensive and time consuming. We have demonstrated that, in a variety of widely applicable scenarios, active learning can be used to ask targeted, poignant and informative questions thereby vastly reducing the amount of data that needs to be gathered while, at the same time, increasing the quality of the resulting models, classifiers and conclusions.

We have tackled active learning by first creating a general approach whereby we define a model and its quality. We then myopically choose the next query that most improves the expected or minimax quality. We then have applied this general decision theoretic approach to the task at hand. In particular, we have addressed three different tasks: classification using support vector machines, parameter estimation and causal discovery using