ACTIVE LEARNING TO MAXIMIZE AREA UNDER THE ROC CURVE

by

Matthew G. Culver

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Stephen D. Scott

Lincoln, Nebraska

August, 2006

ACTIVE LEARNING TO MAXIMIZE AREA UNDER THE ROC CURVE

Matthew G. Culver, M.S.

University of Nebraska, 2006

Advisor: Stephen D. Scott

In active learning, a machine learning algorithm is given an unlabeled set of examples U, and is allowed to request labels for a relatively small subset of U to use for training. The goal of active learning is to judiciously choose which examples in U to have labeled in order to optimize some performance criterion, e.g. generalization accuracy. ROC (Receiver Operating Characteristic) analysis has attracted high attention in machine learning research in the last few years. ROC curves have been advocated and gradually adopted as an alternative to classical machine learning metrics such as misclassification rate. We present several heuristics for active learning designed to optimize area under the ROC curve (AUC) and extensively evaluate them, along with other commonly-used active learning algorithms. One of our algorithms (ESTAUC) was the top performer. When good posterior probability estimates were available, ESTAUC and another of our heuristics (RAR) were by far the best.

ACKNOWLEDGEMENTS

I would like to acknowledge the many people who have supported me throughout this endeavor. First and foremost, I would like to thank Dr. Stephen D. Scott for his guidance, insight and patience. His enthusiasm for both research and teaching is an inspiration. I greatly appreciate the observations, suggestions, and discussions provided by my team member Dun Kun. I also want to thank Nick Roy for his helpful discussions. I want to thank my committee members Dr. Leen-Kiat Soh and Dr. Ashok Samal for reading and providing valuable input. I would also like to thank Dr. Sherri Harms and Dr. John Hastings, whose support and encouragement prompted me to pursue graduate studies. Finally, I am grateful to my parents for their unending encouragement, love and support.

This work was completed in part using the Research Computing Facility at the University of Nebraska.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of machine learning is concerned with the development of algorithms and techniques that allow programs to improve with experience. It is a broad, interdisciplinary field that draws from artificial intelligence, probability and statistics, philosophy, information theory, cognitive science, computational complexity, and control theory. In general, a learning algorithm gains experience by being provided with data composed of examples of a domain to be learned. The algorithm uses this data to discover the underlying rules and patterns that govern the task. Machine learning algorithms can be broken down into three major paradigms, which are supervised, unsupervised, and reinforcement learning. A supervised learning algorithm tries to learn a mapping between an input x and desired output y. For an unsupervised learning algorithm, the task is to understand the relationships between data components. Finally, a reinforcement learning algorithm learns to act in an environment based on delayed rewards or penalties. The work in this thesis falls into the realm of supervised learning; specifically supervised classification.

In supervised classification, a learning problem is defined by a set of examples $D = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in X \times Y$, where x_i is called the *pattern* and y_i is the *label* for x_i. Each pattern (or example) x_i is a vector of attributes $\alpha_1, \ldots, \alpha_\ell$, each of which can be discrete- or continuous-valued. The label space consists of a finite set of discrete values $Y = \{\beta_1, \ldots, \beta_k\}$ that serve to categorize the patterns in X. For simplicity we will only consider binary classification problems for which there are two possible labels $Y = \{+1, -1\}$. A learning algorithm A is trained on D resulting in a hypothesis $h : X \to Y$, which can be used to predict labels on future examples. An example of such a learning problem is to predict whether a tumor is malignant or benign. Each pattern x_i might consist of a series of attributes that describe a tumor such as size, shape, consistency, and the results of lab tests. The label y_i associated with each x_i is either $+1$ for malignant or -1 for benign.

The supervised classification model for learning assumes that D is sufficiently large for learner A to build an adequate hypothesis, which is not always the case. There are many interesting learning problems that offer large amounts of data, but only a relatively small portion of it is labeled. In such a problem the learner has data $D = L \cup U$, where $L = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in X \times Y$ is the set of labeled examples, and $U = \{(x_1, ?), \ldots, (x_n, ?)\}$ is a much larger set of unlabeled examples. In addition, we assume that it is possible to request labels for examples in U from some oracle (e.g. a human expert), which we can then move to L. There is, however, a cost associated with labeling these examples (e.g. the time of the human expert). The learning task for these problems, which is known as *pool-based active learning*, is to judiciously choose a small subset of U to label and move to L. In active learning there are two competing goals: to optimize A's performance relative to some performance measure (e.g. generalization accuracy), and to minimize the cost incurred.

ROC (Receiver Operating Characteristic) analysis has attracted high attention in machine learning research in the last few years. Due to its robustness in imprecise environments, ROC curves have been advocated and gradually adopted as an alternative to classical machine learning metrics such as misclassification rate. The area under the ROC curve (AUC) is of particular interest in that both strong statistical properties and real world experiments have indicated it superior to empirical misclassification rate as an evaluation measure [11]. Despite this, there is a notable absence (to the best of our knowledge) of any research into the affect of active learning on ROC performance, and on heuristics to optimize it. The contribution of this thesis is to fill that gap.

We study how active learning affects AUC. We examine two existing algorithms from the literature: Roy and McCallum's [33] Error Reduction Sampling (ERS) algorithm (designed to directly minimize prediction error) and the "Closest Sampling" method [8,35,38] (sampling closest to the decision boundary; well-known as a simple, fast, high-performance active sampler). We also present our own active learning algorithms designed to maximize the AUC of the hypothesis. In addition, we evaluate two popular base learning algorithms used in active learning (naïve Bayes and support vector machines (SVMs)), and compare their performance in optimizing AUC.

One of our algorithms (ESTAUC) was the top performer, and Closest often came in second behind it. When good posterior probability estimates were available, ESTAUC and another of our heuristics (RAR) were by far the best. We also found that naïve Bayes out-performed the SVM when the labeled set was small, but the SVM generally proved superior as L got larger.

The remainder of the thesis is organized as follows: Chapter 2 formalizes the problem and provides background information about SVMs, naïve Bayes, and ROC analysis. Chapter 3 provides a survey of active learning heuristics from the literature, and Chapter 4 details our heuristics for maximizing AUC. Chapter 5 details our experiments and results. Conclusions and future work are provided in Chapter 6.

Chapter 2

Background

2.1 Problem Formulation

The general algorithm for pool-based active learning is given in Algorithm 1. The key step in this process is in choosing which examples from U to label, which is known as *query selection*. Essentially, the learning algorithm establishes a ranking on U according to how informative it thinks each example will be, and selects the top k examples. Choosing only one example to label at each iteration usually gives the best gain in performance relative to the number of examples labeled [37], but it may be more convenient or less costly for the oracle to label multiple examples at once.

While there are other models for active learning, we will focus on the pool-based model. In the remainder of this thesis, all references to active learning refer to pool-based active learning unless otherwise noted.

Algorithm 1: General algorithm for pool-based active learning

input: learning algorithm A
initial labeled training set L_0
unlabeled pool U

Train A on L_0 to get h_0
$i \leftarrow 0$
while termination condition is not met **do**
  Use A to choose $S \subset U$ according to some strategy
  Request labels for all $x \in S$ from the oracle
  $L_{i+1} \leftarrow \{L_i \cup S\}$
  Remove S from U
  Train A on L_{i+1} to get h_{i+1}
  $i \leftarrow i + 1$
Output h_i

2.2 Applications of Active Learning

Active learning is well-suited for applications where: available labeled training data is insufficient, there exists a large amount of unlabeled data, and manually labeling data is expensive. We will describe several applications where active learning has proven valuable including text classification, image classification, speech recognition and software testing. This will provide motivation for why active learning is important.

A problem where text classification is useful is in email filtering. In this task the active learner is provided with the user's past emails, and is allowed to ask whether individual emails are junk mail or not. Based on the user's answer it refines its hypothesis and brings up another email. This is repeated a given number of times, resulting in an email filter custom tailored to the user. Obviously, the user does not want to invest a lot of time in training the program. This active learning approach minimizes the number of queries the user must answer, while still obtaining a filter that preforms well. Another text classification task is automatic document classification. The goal of this problem is to train a classifier that is able to categorize documents. This would be useful for a user that wants to retrieve certain types of documents from the Web (e.g. published scientific papers). There are an enormous number of documents from which to choose a training set for a learner, and hand labeling them is time consuming. Active learning is a natural solution to minimize this cost. Tong and Koller [38] found that an active learner in some cases reduced the number of documents that needed to be labeled to 10% of the number needed when randomly choosing documents to label.

Luo et al. [26] applied active learning to classifying images of plankton. They had an underwater scanning device that was able to automatically record images of plankton from water samples, and needed to analyze these images while out at sea. It was impossible for human analysts to keep up with the flow of incoming images, so an automated system for classifying images was needed. Their active learning system was able to accomplish this task requiring far fewer training examples than was necessary for a learner trained on randomly selected examples.

Hakkani-Tur et al. [17] used active learning in a speech recognition task for an automatic phone dialog system. Speech recognition systems are typically trained with tens of thousands of transcribed utterances to achieve good accuracy. Their system was able to reduce the number of examples needed by 40% over a system trained on all available data, while performing only 0.1% worse.

Bowring et al. [6] used active learning in software testing where the input data was derived from different execution paths of a program. With their test application they found that only 38% of the training data was needed to achieve the same classifier precision as a learner trained on randomly selected data.

2.3 ROC Analysis

Work in active learning (as well as much general work in machine learning) to date has for the most part centered on optimizing the classification accuracy of the learner. In many applications, however, classification accuracy can be misleading. This is true for problems where the class distribution or cost function is skewed. For example, consider a two-class learning problem that consists of 990 negative examples and 10 positive examples. A trivial hypothesis that always predicts negative will attain a 99% accuracy rate, but this is not reflective of its performance. Another case where optimizing accuracy might not achieve the desired result is when the ranking induced by the learned model is more important than its accuracy. For instance, consider the list of relevant documents returned by a search engine for a specific query. The list may contain thousands of documents, but the user will only examine a small number of them. In this case, the search engine's ranking of the documents is far more important than the accuracy of its classification as being either relevant or not [11].

ROC analysis provides a superior means by which to measure the performance of a learner in these cases. A ROC curve describes the trade-off between the true positive rate (TPR) and false positive rate (FPR) in a two-class classification problem. The plot is generated by first ranking the test examples according the hypothesis h's confidence that they belong to the positive class. Then the threshold θ that is used to convert $h(x)$ to a binary prediction is varied to get the points on the curve. ROC curves are superior to classification accuracy because one can easily observe the effects of the ranking induced by the model, and its performance at different cost distributions. A drawback to using ROC curves as a performance measure is that they come in the form of a graph, and are not as easy to compare as a scalar value like accuracy. However, the area under the ROC curve (AUC) is a scalar performance measure that encapsulates many of the benefits of the ROC curve. AUC is the same as the Wilcoxon-Mann-Whitney statistic [18], which is an unbiased probability estimate that a randomly drawn positive example has a higher $h(\cdot)$ value than a randomly drawn negative example:

$$AUC = \frac{\sum_{i=1}^{|P|} \sum_{j=1}^{|N|} I(h(x_i^+) > h(x_j^-))}{|P| \cdot |N|} \quad , \tag{2.1}$$

where P and N are the set of positive and negative labeled examples, $x_i^+ \in P$ is a positive example, $x_j^- \in N$ is a negative example, and $I(\cdot) = 1$ if its argument is true and 0 otherwise. The range of the AUC is $[0, 1]$, and the higher the value the better the hypothesis.

Cortes and Mohri [11] conducted a study of the relationship between AUC and classification accuracy, and their results argue for optimizing AUC as opposed to accuracy in some cases. They discovered that by fixing the accuracy for a group of classifiers, that the average AUC varies monotonically with accuracy, but the variance is high for uneven class distributions with low accuracy. This means that while in general optimizing accuracy also optimizes AUC, this does not necessarily hold under the conditions noted. This is of particular interest for active learning because in the initial stages the training set is usually small, which in most cases leads to a high error rate. Therefore, heuristics that try to directly minimize expected error may be optimizing a non-optimal performance measure.

2.4 Support Vector Machines

Support Vector Machines (SVMs) were introduced in 1992 by Boser et al. [5, 34]. SVMs find their roots in the Perceptron algorithm, which is a linear learning machine with a similar form. Training data is represented as vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ in some space $\mathcal{X} \subset \mathbb{R}^d$, and each vector $\mathbf{x}_i$ has a binary label $y_i \in \{-1, +1\}$. The goal of Perceptron is to find a weight vector $\mathbf{w}$ that correctly separates the positive and negative training examples $\mathbf{x_i}$ (i.e. correctly classifies the training set) such that:

$$y_i f(\mathbf{x}_i) \geq 0, \qquad \forall i = 1, \ldots, m \quad ,$$

where

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \ , \tag{2.2}$$

and $\langle \mathbf{w}, \mathbf{x} \rangle$ is the dot product between $\mathbf{w}$ and $\mathbf{x}$. During training if an example x_i is misclassified then $\mathbf{w}$ is updated accordingly. Specifically, if $y_i f(\mathbf{x}) < 0$ then

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \, y_i \mathbf{x}_i \ , \tag{2.3}$$

where η is the learning rate parameter that determines how large of an adjustment is made. If the data are linearly separable (i.e. can be split into positive and negative examples with a hyperplane), Perceptron is guaranteed to converge to a solution in a finite amount of time given a sufficiently small, positive learning rate η. If the data are not linearly separable then other techniques such as SVMs must be used.

SVMs can handle non-linearly separable data because they remap the example vectors into a higher-dimensional (possibly infinite-dimensional) feature space through a *kernel function*. A kernel $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ for some nonlinear map Φ is a dot product (i.e. similarity measure) between two vectors when mapped to a higher-dimensional feature space, which can be computed without explicitly performing the dot product. Some of the standard kernels are:

Polynomial	$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d \ ,$
Gaussian radial basis function (RBF)	$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \langle \mathbf{x} - \mathbf{x}', \mathbf{x}' - \mathbf{x} \rangle^2) \ ,$
Sigmoid	$k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \langle \mathbf{x}, \mathbf{x}' \rangle + \vartheta) \ ,$

where $d > 0$, $\gamma > 0$, $\kappa > 0$, and $\vartheta < 0$ are parameters.

In order to use a kernel, we need to rewrite the Perceptron algorithm into its dual form. Let $\mathbf{w}$ initially be $\mathbf{0}$ and $\alpha_i = \eta \times$ *the number of mistakes on* $\mathbf{x}_i$. Then $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$, and Equation 2.2 becomes

$$f(x) = \sum \alpha_i \, y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b \ . \tag{2.4}$$

Then the update from Equation 2.3 changes to

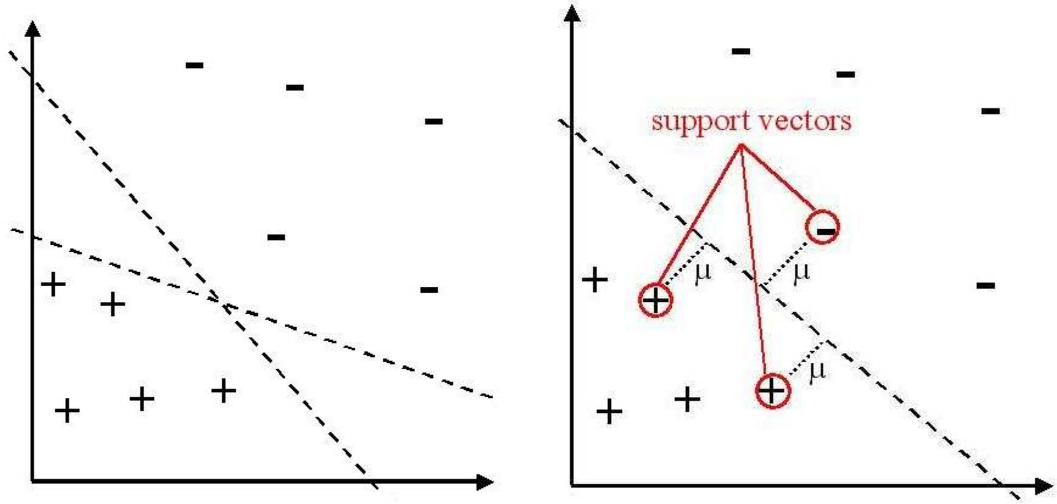
$$\alpha_i \leftarrow \alpha_i + \eta \ .$$

In the dual form, the calculation only depends on the dot products of the example vectors. This is beneficial because we do not have to explicitly maintain the weight vector $\mathbf{w}$, which can be very large for high dimensional feature spaces. We can now substitute any kernel we might like into Equation 2.4 to get a kernelized Perceptron

$$f(\mathbf{x}) = \sum \alpha_i \, y_i \, k(\mathbf{x}_i, \mathbf{x}) + b \ .$$

The use of kernels makes it possible for Perceptron to operate on non-linearly separable data since any finite set of vectors becomes linearly separable in a high enough dimensional feature space. This leads to the next issue that stems from the fact that for any set of vectors that are linearly separable, there are an infinite number of hyperplanes that will separate them. Figure 2.1(a) shows two separating hyperplanes on a simple data set. Intuitively, the optimal hyperplane is the one that exactly splits the closest positive and negative vectors. Another way of saying this is the hyperplane that has the maximal distance μ between itself and the nearest vectors. This distance is called the *margin*, and the vectors that are closest to the hyperplane (i.e. determine its location) are called *support vectors* (SVs). Figure 2.1(b) shows the optimal hyperplane with maximized margin. According to Vapnik-Chervonenkis (VC) theory, maximizing the margin yields the smallest bound on the generalization error (error on future unseen examples) [34].

Maximizing the margin is the same as minimizing the 2-norm of the weight vector



(a) (b)

Figure 2.1: Simple example of separating hyperplanes in a linearly separable data set. (a) Two of the infinite number of separating hyperplanes. (b) Optimal hyperplane with maximized margin μ.

$\mathbf{w}$ [34], which leads to the following constrained optimization problem:

$$\text{minimize}_{\mathbf{w}\in\mathcal{H},\, b\in\mathbb{R}}\ \tau(\mathbf{w}) = \tfrac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \qquad \forall i = 1, \ldots, m \ ,$$

where $\mathcal{H}$ is some dot product space. The equivalent dual optimization problem is [34]:

$$\text{maximize}_{\alpha\in\mathbb{R}^m}\ W(\alpha) = -\sum_{i=1}^{m} \alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{m} \alpha_i\,\alpha_j\,y_i\,y_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle \tag{2.5}$$
$$\text{subject to } \alpha_i \geq 0 \quad \forall i = 1, \ldots, m \qquad \text{and } \sum_{i=1}^{m} \alpha_i\,y_i = 0 \ .$$

Of course, a kernel function can be substituted in place of the dot product $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$.

Equation 2.5 describes a *hard margin* SVM that requires that the data be linearly separable in feature space. In some datasets a separating hyperplane may not exist depending on the choice of kernel. In addition, it is often the case that exactly fitting the training data is undesirable. By fitting the training data too closely we may be conforming our hypothesis to anomalies in the training data that are not representative of the problem as a whole, which is called overfitting. For these reasons a *soft margin* variant of Equation 2.5 is generally used. Using a soft margin means that we allow a certain number of training vectors to infringe upon the margin during training (i.e. margin errors). One soft margin variant called C-SVM is defined as

$$\begin{aligned} &\mathrm{maximize}_{\alpha \in \mathbb{R}^m}\, W(\alpha) = \textstyle\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &\text{subject to } 0 \geq \alpha_i \geq C/m \quad \forall i = 1, \ldots, m \qquad \text{and } \textstyle\sum_{i=1}^{m} \alpha_i\, y_i = 0 \ , \end{aligned} \tag{2.6}$$

where C is a parameter that determines the trade-off between minimizing the training error (i.e. minimizing margin errors), and maximizing the margin. Higher values of C result in a harder margin and lower values a softer margin. Finally, predictions on new examples are made with

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{m} \alpha_i\, y_i\, k(\mathbf{x}, \mathbf{x}_i) + b \right) \ , \tag{2.7}$$

where sgn$(z) = +1$ if $z > 0$ and -1 otherwise.

2.5 Naïve Bayes

The *naïve Bayes classifier* [28] is a simple probabilistic classifier. It is based on *Bayes Theorem*, which is

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)} \ , \tag{2.8}$$

where A and B are random variables. The term $P(B)$ can be read as the probability of B, and $P(B \mid A)$ is read as the probability of B given A. For the purposes of classification we would like to use Bayes Theorem to predict the most probabilistically likely label y^* given a previously unseen example x:

$$
\begin{aligned}
y^* &= \operatorname*{argmax}_{y \in Y} P(y \mid x) \\
&= \operatorname*{argmax}_{y \in Y} \frac{P(x \mid y)P(y)}{P(x)} \quad .
\end{aligned}
\tag{2.9}
$$

Since $P(x)$ is the same for all possible labels $y \in Y$ we can drop the denominator in Equation 2.9 to get

$$
y^* = \operatorname*{argmax}_{y \in Y} P(x \mid y)P(y) \quad . \tag{2.10}
$$

The example x is not a single random variable, but rather a vector of attributes $(\alpha_1, \ldots, \alpha_n)$. The naïve Bayes classifier makes the simplifying assumption that all of the attributes α_i are *conditionally independent* given the label. In other words,

$$
P(\alpha_i \mid \alpha_j, y) = P(\alpha_i \mid y) \qquad \forall i, j \quad i \neq j \quad .
$$

Therefore, we can write the probability of x given y as the product of the conditional probabilities of its attributes, or more specifically,

$$
P(x \mid y) = P(\alpha_1, \ldots, \alpha_n \mid y) = \prod_i P(\alpha_i \mid y) \quad . \tag{2.11}
$$

Equation 2.11 can be substituted into Equation 2.10 to obtain the form of the naïve Bayes classifier

$$
y^* = \operatorname*{argmax}_{y \in Y} P(y) \prod_i P(\alpha_i \mid y) \quad . \tag{2.12}
$$

Despite the fact that the naïve Bayes independence assumption is often violated, it performs well on a variety of problems. This along with the fact that it is simple and fast make it one of the more common base learners used in active learning.

Chapter 3

Related Work

In this chapter, we first explore a couple of problems that are closely related to pool-based active learning, which have important theoretical results. Then several query selection heuristics for pool-based active learning are introduced. Finally, a dual to active learning, called *budgeted learning*, is discussed.

3.1 Membership Query Learning

Active learning is closely related to **membership query learning** (MQL) for which there are many interesting theoretical results. In this problem, the goal is to learn a target concept h_0, chosen from a finite yet exponentially large set of hypotheses $\{h_1, h_2, \ldots\}$. In MQL, a hypothesis h is chosen from the set of hypotheses that are consistent with the current version space. The version space is the set of hypotheses that are consistent with the training set L, or, stated another way, that correctly predict the labels for all $x \in L$. Then h generates a new example x' that is expected to improve the current hypothesis. It is important to note that x' is not chosen from some finite set of unlabeled examples, but from the possibly infinite set of patterns that are consistent with the problem definition. Once selected, x' is labeled by the oracle and added to L, and the process is repeated until $h = h_0$.

In this framework one can prove worst-case bounds on particular concepts. For example, Angluin [2] shows that there is an algorithm that exactly identifies every monotone DNF formula ϕ_* over n variables that contain no negative literals in time polynomial in n and the number of terms of ϕ_*.

While the query learning model is interesting, the assumptions that it is based upon do not allow it to be applied to a large range of problems. This learning model requires the ability to label arbitrary examples, which is reasonable if the concept is pairs of winning poker hands, context-free grammars, etc. There are other problem domains where artificially generated examples have no meaning, however. For example, in document classification it is unlikely that an artificially created text document will make sense to a human. Likewise, in handwritten character recognition an artificially generated character image will probably not look like one of the symbols used in natural language [21].

3.2 Stream-Based Active Learning

Although this thesis focuses on the pool-based active learning model, a stream-based active learning algorithm, **Query by Committee** (QBC), is worth mentioning. The stream-based model differs from the pool-based one in that it does not have access to a pool of unlabeled data U. Rather, the learning algorithm A is presented with a sequence, or stream, of unlabeled examples, one at a time, and must decide whether each one is worth labeling.

One basic idea [10] that is applicable to both stream-based and pool-based active learning is to label examples that effectively shrink the current version space (the set of hypotheses consistent with the training data). As discussed by Tong and Koller [38], such a heuristic would be probabilistically optimal if it could always exactly halve the version space each time A makes a query. Strong theoretical results using this idea are yielded by the QBC algorithm [15, 36].

The QBC algorithm creates a committee H consisting of k hypotheses selected at random from all hypotheses that are consistent with the current version space defined by the set of labeled examples. Then H is presented with a stream of randomly chosen unlabeled examples $x_1, x_2, \ldots$ one at a time, and chooses example x_i to label for which exactly half ($k/2$) of the hypotheses in H predict positive and the other half negative. At this point H is again selected from the new version space, and the process is repeated until H evaluates a sufficiently large sequence of unlabeled examples without choosing one to label [15]. Under certain assumptions, the number of queries will be logarithmic in the number of random examples examined (m), and generalization error decreases as $O(1/m)$. In other words, we can expect the same generalization error from a pool of examples with only a logarithmic fraction labeled.

The QBC results are exciting, but they require several unrealistic assumptions. One such assumption is that the data is noise-free, which is almost never true for real world problems. Also, QBC mandates that a perfect deterministic classifier exists, and that it be possible to draw classifiers randomly from the version space, which is computationally intractable for most interesting problems [1, 27].

An active learning approach recently proposed by Balcan et al. [3] has strong theoretical guarantees like QBC, but does away with most of the assumptions. In fact, they only assume that the unlabeled data is drawn i.i.d. (independently and identically distributed) from some underlying distribution. By i.i.d. we mean that seeing one example does not change the probability of seeing any other example. Their approach called **Agnostic Active learning** (A^2) works by halving the set of hypotheses $H \subset \mathcal{H}$ that are considered optimal with respect to the labeled data at each query round. To do this at each round, A^2 samples a a set of examples $S \subset R$, where $R \subset U$ is the region of uncertainty. R is defined as the set of examples $x \in U$ for which there exists a pair of hypotheses in H that disagree on x. Via standard results in statistical learning theory, they calculate upper and lower bounds on the error rate over S for each $h \in H$, and remove any hypothesis from H that has a lower bound that is higher than the minimum upper bound over all other hypotheses from H. S is selected in such a way that H is approximately cut in half at each round. Though these results are promising, A^2 has not been practically applied yet. One obstacle is that H is exponentially large in the size of R, making A^2 computationally expensive.

3.3 Pool-Based Active Learning

There are a number of heuristics for selecting query points within the pool-based active learning model (see Section 2.1). This section provides an overview of some of the more relevant techniques, such as: uncertainty sampling, reducing future error, combined techniques, and others.

3.3.1 Uncertainty Sampling

Uncertainty sampling is a technique that follows from the QBC idea that labeling examples for which the current learner is uncertain in its prediction will shrink the version space. There are several related heuristics that fall under the uncertainty sampling umbrella, including: closest sampling, query by bagging, query by boosting, and the active Decorate method.

Closest sampling [35] selects for labeling the example from U for which A is most uncertain in its prediction, where A is a single classifier. For a classifier that generates posterior probabilities of class membership, like naïve Bayes, this means choosing the examples whose predicted class membership probabilities are closest to 0.5 (assuming a two-class learning problem.) In the context of an SVM this would be the example that is closest to the decision threshold. In a large margin classifier, like an SVM, the current hypothesis lies approximately in the center of the version space, and by choosing the example that is closest to it, the version space is approximately cut in half [38].

Although this heuristic has enjoyed a great deal of success it does have some potential drawbacks. First, the current hypothesis may not lie in the center of the version space, and thus cutting may not be uniform. Second, examples that are close to the decision boundary are ones that are difficult to classify, and might be outliers that do not reflect the true distribution of the data. Third, if a large number of the examples lie near the boundary then better results might be obtained by exploring diversity among these examples rather than always choosing the closest one [4, 31]. Finally, focusing on the decision boundary prevents exploration of the feature space for regions of examples that the current hypothesis misclassifies.

The main problem with applying the QBC algorithm in practice is that for many problems it is hard to generate a committee of classifiers at random that are in the current version space. One attempt at doing this in general is the Kernel Gibbs Sampler [16]. However, due to its complexity it is limited to training set size of no more than 100 examples. Abe and Mamitsuka [1] addressed this problem by using other committee building strategies; namely bagging and boosting.

With their first approach, **Query by Bagging** (QBag), a committee A of k classifiers a^i is trained; each on a different training set L^i of size $m = |L|$. The examples in each L^i are chosen uniformly at random with replacement from L. Predictions on new examples are made by a majority vote of the committee. To select an example for labeling, A evaluates each $x \in U$, and picks the one for which the predictions of the committee members are most evenly split.

Abe and Mamitsuka's second algorithm, **Query by Boosting** (QBoost), is a little more complex. Boosting can be used to 'boost' the performance of a relatively week learning algorithm. It does so by repeatedly re-sampling on the training data to build a classifier, with the sampling distribution varied each time to focus on the examples that the previous classifier got wrong. Predictions on new examples are made by taking a weighted majority of the predictions of all the classifiers in the committee. QBoost uses a slightly different heuristic for choosing the examples to label than QBC and QBag. The margin of an example is defined as the difference in the weighted sum of votes in the current committee for the most popular class label, and that for the second most popular label. Examples with smaller margins are considered to be ones for which the committee is most uncertain in its prediction, and are selected for labeling.

Closely related to Query by Bagging/Boosting is an active learner presented by Melville and Mooney [27] called active Decorate. This learner uses yet another committee building strategy that strives to build a diverse committee in which each classifier a^i is as different as possible, while still maintaining consistency with the current version space. They do this by starting with a classifier a^1 trained on L. Successive classifiers $a^2, \ldots, a^k$ are trained on L plus a set of examples that are randomly generated using an approximation of the training data distribution. These new examples are labeled based on the current committee. They do this by computing the class membership probabilities of the current committee, and then sampling the label from this distribution, such that the probability of selecting a label is inversely proportional to the current committee's prediction. If a new classifier increases the committee's error on L, then it is rejected and another attempt is made. The process continues until the desired committee size is reached or a maximum number of attempts has been made.

Active Decorate uses the same margin-based heuristic for choosing examples to label as (QBoost) with a slight modification. The classifiers a^i provide probabilities of class membership rather than the most likely class label. The margin is then defined as the difference between the highest and second-highest predicted probabilities.

3.3.2 Reducing Future Error

A different approach [15, 20, 24, 33] is to choose examples for labeling that are helpful in building up low future error. For example, Roy and McCallum's **Error-Reduction Sampling** (ERS) algorithm works by directly minimizing expected error on the data set. ERS estimates the loss (either zero-one or log loss) that would result from adding candidate example $x \in U$, with its label, to L. The candidate example that results in the lowest expected error is selected for labeling. Using log loss the ERS algorithm would choose

$$\operatorname*{argmin}_{x \in U} \left\{ - \sum_{x' \in U} \sum_{y' \in Y} P(y' \mid x') \log \hat{P}_{xy_x}(y' \mid x') \right\} \quad , \tag{3.1}$$

where y_x is the true label of example x, Y is the set of labels, $P(y' \mid x')$ is the true posterior probability of label y' given instance x', and $\hat{P}_{xy_x}(y' \mid x')$ is the estimate of the posterior by the candidate model trained on $L \cup \{(x, y_x)\}$. Since $P(y' \mid x')$ is not known, the candidate model's estimate of the posterior is used instead. Since the label for candidate example x is also unknown, the expected loss is calculated for each possible label, and then combined, weighted by the current model's (trained on only L) posterior estimate for x. Then Equation 3.1 becomes

$$\operatorname*{argmin}_{x \in U} \left\{ - \sum_{y \in Y} \hat{P}(y \mid x) \sum_{x' \in U} \sum_{y' \in Y} \hat{P}_{xy}(y' \mid x') \log \hat{P}_{xy}(y' \mid x') \right\} \quad , \tag{3.2}$$

where $\hat{P}(y \mid x)$ is the posterior estimate of the current model. Because the candidate model's posterior estimate is used in place of the true posterior probability in the loss function, ERS selects those examples that maximize the sharpness of the learner's posterior belief about the unlabeled examples [33].

Related to Roy and McCallum's work, Nguyen and Smeulders [30] chose examples that have the largest contribution to the current expected error: they built their classifiers based on centers of clusters and then propagated the classification decision to the other samples via a local noise model. During active learning, the clustering is adjusted using a coarse-to-fine strategy in order to balance between the advantage of large clusters and the accuracy of the data representation. Yet another approach [29] is specific to SVM learning. Conceptually, in SVM learning if we can find all the true support vectors and label all of them, we will guarantee low future error. Mitra et al. assigned a confidence factor c to examples within the current decision boundary and $1 - c$ to examples outside each indicating the confidence of whether they are true support vectors, and then chose those examples probabilistically according to this confidence.

3.3.3 Boosting and Combining

Another category of active learning approaches contains algorithms that try to quickly "boost" or "stabilize" an active learner. Active learning is unstable, especially with limited labeled examples, and the hypothesis may change dramatically each round it sees a new example. One way to boost active learning algorithms is simply combining them in some way. For example, the algorithm COMB by Baramet al. [4] combines three different active learners by finding and fast-switching to the one that currently performs the best. Osugi et al. [31] adopted a similar approach with a simpler implementation, and focused on how to balance the exploration and exploitation of an active learner. In their implementation the empirical difference between the current hypothesis and the previous one is used as a criterion to decide whether exploration should be further encouraged. In other exploration-based active learning, Xiao et al. [40] studied the problem of active learning in extracting useful information in commercial games, in which "decision-boundary refinement sampling" (analogous to closest sampling) and "default rule sampling" (analogous to random sampling) mechanisms are each used half of the time.

3.3.4 Other Approaches

An active learning heuristic that does not fall into any of the above categories is **Relevance sampling** [23]. This technique is almost the opposite of uncertainty sampling, and selects for labeling examples that the current classifier considers most likely to be members of a particular class. This strategy makes sense for problems like document retrieval where the user is more interested in seeing relevant documents than in the effectiveness of the final classifier. As a means to increase classifier performance, however, it does not work very well as it tends to choose redundant examples, and works more poorly as the classifier improves.

The active learning heuristics mentioned so far work best when choosing a single example for labeling at each iteration. As already mentioned, though, it may be more convenient for the oracle to label m examples at once. In this situation many heuristics will tend to choose groups of examples that are similar or redundant. For example with closest sampling, the m examples may all be in the same vicinity, and the resulting cut in the version space would not be much different than if only one point was chosen. **Diverse sampling** [7, 32] addresses this problem for separating hyperplane classifiers like SVMs by picking points that are close to the decision hyperplane, but far away from each other. Hoi et al. [19] took another approach by choosing the subset $S \subset U$ to label that resulted in the largest reduction in the Fisher information matrix. The Fisher matrix measures the overall uncertainty of a classification model. Since the number of candidate subsets in the unlabeled pool is exponential in the size of U they applied a greedy heuristic for finding S.

3.4 Budgeted Learning

Budgeted learning is a problem which serves as a complement to pool-based active learning. In budgeted learning a large set of data $D = \{(?, y_1), \ldots, (?, y_m)\}$ is available where the class labels for each example are known, but the attribute values for the pattern are not. In addition, there is a cost associated with obtaining the value of a particular attribute for a given example. This cost need not be uniform across the attributes, so some attributes may cost more than others. Also, the budgeted learner B is given a fixed budget that it can use to purchase information about the values of attributes for examples in D. B's goal is to effectively spend its fixed budget to obtain a classifier whose performance is optimal. Choosing which attribute values to have labeled is analogous to query selection in active learning. Lizotte et al. [25] proposed several strategies for doing this, and found that the strategy used can have a profound impact on the quality of the final classifier.

An application cited by Lizotte et al. where budgeted learning could be applied is to build a diagnostic classifier of cancer subtypes. In this problem a pool of patients with known cancer subtypes is available, as are various diagnostic tests that can be performed, each with an associated cost. A fixed budget is available with which to pay for tests. Budgeted learning can be used to determine which tests are most discriminating in diagnosing patients.

Chapter 4

Methods

In this section we describe four active learning heuristics for maximizing AUC. The first two loosely follow the closest sampling approach, while the other two are modifications of the ERS [33] algorithm. Throughout this chapter, we let $h(x) \in \mathbb{R}$ denote the current hypothesis's confidence that example x is positive (the larger the value, the higher the confidence in a positive label). This confidence score may come in the form of a probability estimate ranging between 0 and 1, where the decision threshold θ is set at 0.5. Alternatively, it could be any score that indicates the hypothesis's confidence that x is positive like the SVM output function (i.e. Equation 2.7 with the sgn$(\cdot)$ function removed) for which $\theta = 0$. The value of h need not be a probability estimate except in one of our algorithms (ESTAUC).

4.1 Rank Climbing

In our first heuristic, **Rank Climbing** (RANC), we use the current hypothesis h to rank all examples in $L \cup U$ where L is the set of labeled examples used to train h, and U is the unlabeled pool. The examples are ranked in descending order according to the confidences $h(x)$. Let x_ℓ be the lowest ranked example from L with a positive label. We select for labeling the lowest ranked unlabeled example that is ranked higher than x_ℓ:

$$\operatorname*{argmin}_{x \in U : h(x) > h(x_\ell)} \{h(x)\} \ .$$

In the unlikely event that there is no example in U ranked higher than x_ℓ, RANC chooses the highest-ranked example from U. Figure 4.1 shows a simple illustration of how Rank Climbing chooses an example to label.

Member of	h(x)	label	
L	0.98	+	
U	0.87	?	
L	0.75	+	
U	0.65	?	
L	0.61	-	
U	0.57	?	**
L	0.43	+	*
U	0.32	?	
U	0.17	?	
L	0.09	-	
L	0.05	-	

Figure 4.1: Example of query selection by Rank Climbing. The * indicates x_ℓ and ** indicates the example selected for labeling.

4.2 Rank Sampling

In our second heuristic, **Rank Sampling** (RANS), we again use the current hypothesis h to rank all examples in $L \cup U$ in descending order according to their confidences. Let x_u be the highest ranked example from L with a negative label, and x_ℓ be the lowest ranked example from L with a positive label. The example that we choose to label is selected uniformly at random from the set $C = \{x \in U : h(x) > h(x_\ell)$ and $h(x) < h(x_u)\}$. If C is empty then we repeatedly change x_u to be the next highest ranked example from L, and x_ℓ to be the next lowest ranked example from L until C is non-empty. Figure 4.2 illustrates the query selection process for the Rank Sampling heuristic.

Member of	h(x)	Label
L	0.95	+
U	0.89	?
L	0.73	+
L	0.65	-
U	0.58	?
L	0.51	-
U	0.43	?
U	0.32	?
L	0.25	+
L	0.19	-
U	0.05	?
L	0.01	-

Figure 4.2: Example of query selection by Rank Sampling. The example chosen for labeling is selected uniformly at random from the set C.

Since AUC is proportional to the number of negative examples ranked below positive examples, RANC and RANS attempt to find an unlabeled negative example that ranks above positive examples. RANC assumes that the lowest-ranked example above x_ℓ is the most likely to be negative, while RANS makes a random selection to reduce sensitivity to noise.

4.3 Maximizing Estimated AUC

Our next heuristic, **Maximizing Estimated AUC** (ESTAUC), attempts to directly maximize future AUC much in the same way that ERS directly minimizes future error. We first assume that we know the labels of the examples in U. Then the example from U that we choose to label is the one that most improves the AUC on the unlabeled examples, where the AUC is computed by the formula of Hanley et al. [18] given below. More precisely, we would choose

$$\operatorname*{argmax}_{x \in U} \left\{ \frac{\sum_{x' \in U} \sum_{x'' \in U(h_{xy_x}, x')} I(y_{x'} = +)\, I(y_{x''} = -)}{|P|\,|N|} \right\} \quad ,$$

where $y_{x'}$ is the true label of example x', $h_{xy_x}(x')$ is the confidence of the hypothesis trained on $L \cup \{(x, y_x)\}$ and evaluated on x', $U(h_{xy_x}, x') = \{x'' \in U : h_{xy_x}(x'') < h_{xy_x}(x')\}$ is the subset of examples in U that have confidence less than that of x' when evaluated with h_{xy_x}, $P = \{x \in U : y_x = +\}$, $N = \{x \in U : y_x = -\}$, and $I(\cdot) = 1$ if its argument is true and 0 otherwise. Since the denominator is independent of x, we instead can use the unnormalized AUC:

$$\operatorname*{argmax}_{x \in U} \left\{ \sum_{x' \in U} \sum_{x'' \in U(h_{xy_x}, x')} I(y_{x'} = +)\, I(y_{x''} = -) \right\} \quad . \tag{4.1}$$

Since we do not know the true labels of the examples in U, we adapt the approach of Roy and McCallum and use probability estimates derived from the hypothesis h_x in place of the indicator functions. Further, since we do not yet know the label of the candidate point we are considering labeling, we compute Equation 4.1 using each possible label and weight them according to our posterior probability estimates of each label:

$$\begin{aligned} \operatorname*{argmax}_{x \in U} \Bigg\{ & \hat{P}(+ \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x+}, x')} \hat{P}_{x+}(+ \mid x') \hat{P}_{x+}(- \mid x'') \\ & + \hat{P}(- \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x-}, x')} \hat{P}_{x-}(+ \mid x') \hat{P}_{x-}(- \mid x'') \Bigg\} \quad , \end{aligned}$$

where $h_{x+}(x')$ is the confidence of the hypothesis trained on $L \cup \{(x, +)\}$ and evaluated on x'. In addition, $\hat{P}(y \mid x)$ is the probability of predicting $y \in \{+, -\}$ given x by hypothesis h trained on L, and $\hat{P}_{xy}(y' \mid x')$ is the probability of predicting y' given x' by hypothesis h_{xy} trained on $L \cup \{(x, y)\}$. The probability estimates may come from e.g. naïve Bayes or from logistic regression with an SVM.

4.4 Rank Reinforcing

Our fourth heuristic, **Rank Reinforcing** (RAR), focuses on the ranking induced by the hypothesis. RAR uses the current model h to label all examples in U and uses this labeling in the computation of the AUC of U as ranked by h_{x+} and h_{x-}. Specifically, the example we choose is

$$\operatorname*{argmax}_{x \in U} \left\{ \hat{P}(+ \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x+}, x')} f(x', x'') + \hat{P}(- \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x-}, x')} f(x', x'') \right\} \quad , \tag{4.2}$$

where $f(x', x'') = I(h(x') > \theta) I(h(x'') < \theta)$, θ is the threshold used to map $h(\cdot)$ to a binary label (e.g. 0.5 for naïve Bayes and 0 for an SVM). Ties can be broken as follows. Let $T \subseteq U$ be the set of examples involved in the tie. We can break ties either by applying ESTAUC over T or by summing the margins of pairs of relevant examples in T:

$$\operatorname*{argmax}_{x \in T} \left\{ \hat{P}(+ \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x+}, x')} f(x', x'')(h(x') - h(x'')) \right.$$
$$\left. + \hat{P}(- \mid x) \sum_{x' \in U} \sum_{x'' \in U(h_{x-}, x')} f(x', x'')(h(x') - h(x'')) \right\} \quad .$$

RAR amounts to choosing $x \in U$ such that h_x most reinforces h's ranking of the unlabeled examples to the extent that examples that h predicts as positive remain ranked higher than examples that h predicts as negative.

If implemented as stated above, the ERS, ESTAUC, and RAR heuristics could be slow due to the need to repeatedly retrain the hypotheses. There are, however, several techniques that allow for speeding up the execution time of these heuristics without harming performance. The first technique is to filter the set of candidate examples that are under consideration for labeling. This can be accomplished through random sampling, or by using a faster active learning heuristic to rank the examples in the unlabeled pool and choosing the most promising ones as the candidates. Usually, a candidate pool of a few hundred is enough to achieve good results. Using a classifier that is capable of incremental and decremental updates also reduces execution time as it removes the necessity of rebuilding the classifier each time a candidate point is evaluated. For example, both naïve Bayes and SVMs [9] are capable of incremental and decremental updates.

Chapter 5

Experimental Setup and Results

5.1 Experimental Setup

Experiments were carried out on 8 data sets from the UCI Machine Learning Repository [12], and one data set derived from the United States Postal Service (USPS) handwritten digit recognition data set. The USPS data set consists of scanned handwritten digits, where the attributes are the pixel values of the image and the classification is the digit represented. To make this data set binary, we used only examples from the classes '3' and '8' since these numbers are more likely to be confused than others. Attributes for each of the data sets are summarized in Table 5.1 including the ratio of positive to negative examples (P/N). Since AUC was the metric used in evaluating performance, all data sets are two-class.

In addition to the four algorithms of Chapter 4, tests were run with closest sampling (Closest), Roy and McCallum's log loss Error-Reduction Sampling (ERS) method, and a random active learner (Random). The heuristics were evaluated using naïve Bayes and the SVM Sequential Minimal Optimization (SMO) as the base learners. In addition, all experiments were run with 15 and 100 examples in the initial labeled training set L_0. The

Table 5.1: Data set information including: the number of examples, the number of attributes, the ratio of positive to negative labeled examples, and the SMO parameters C and γ. The variable m is the size of the labeled training set L.

Data set	No. of Examples	No. of Attributes	P/N	SMO Parameters C	SMO Parameters γ
Breast cancer	286	9	0.42	$10m$	8.0×10^{-10}
Colic	368	22	0.59	m	0.05
Credit A	690	15	0.80	$10m$	1.0×10^{-5}
Credit G	1000	20	2.33	$3m$	1.0×10^{-5}
Diabetes	768	8	1.87	m	0.01
Ionosphere	351	34	1.77	m	0.5
KR vs. KP	3196	36	1.09	$10m$	0.1
USPS	1416	256	1.00	$10m$	0.01
Vote	435	16	1.21	$3m$	0.0001

larger size for L_0 was used to see if the ESTAUC, RAR and ERS heuristics would benefit from having better probability estimates.

The heuristics were implemented in Java within the Weka machine learning framework [39]. We used the Weka implementations for both SMO and naïve Bayes in our active learning experiments. The Gaussian RBF kernel was used with SMO and the C and γ parameters used on each data set can be found in Table 5.1. The Weka naïve Bayes classifier was modified for incremental and decremental updating, but the standard non-updateable implementation of SMO was used for the SVM results. To get get probability estimates from SMO the SVM output $f(x)$ is applied to a parameterized sigmoid function of the form:

$$P(y|x) = \frac{1}{1 + \exp(-ayf(x) + b)} \ ,$$

where a is a weight, y is the label for which we want a probability of membership for example x, and $f(x)$ is the SVM output function as given in Equation 2.7 with the sgn$(\cdot)$ function removed. The parameters a and b calibrate the probability estimates, and can be trained through maximum likelihood estimation.

We used k-fold cross validation in our tests. Ten folds were used on all of the data sets except Breast Cancer, where seven folds were used due to the small size of the data set. Our testing methodology is summarized in Algorithm 2. For ESTAUC, RAR, and

Algorithm 2: Testing methodology.

input: data set D
Initial labeled set size m
Number of folds k

Split D into k disjoint folds $F_1, \ldots, F_k$
for i from *1* to k **do**
    Reserve F_i as the test set
    $P_i = D \setminus F_i$
    repeat *10* times
        Select an initial labeled set L_0 of size m from P_i uniformly at random
        Unlabeled pool $U = P_i \setminus L_0$
        Run each heuristic on L_0 and U, and evaluate on F_i after each query
Report the average result of all experiments

ERS, the set of instances under consideration for labeling was reduced to 200 candidates to increase speed, however all instances in U were used for estimating AUC (ESTAUC, RAR) or entropy (ERS).

Learning curves were constructed to evaluate the behavior of the algorithms. These curves display the change in performance of the heuristics as they make queries. To construct the curves we plotted the AUC achieved by each heuristic on the test set against the size of L after each query was made. The AUC value plotted was the mean over all tests (10 for each fold). AUC was plotted on the y-axis, and the size of the current labeled set was on the x-axis. Selected learning curves are shown in the results, and the curves for all experiments can be found in Appendix A.

Paired-t tests [28] were performed to establish the significance at which the heuristics differ. Using a paired-t test is valid because AUC approximates a normal distribution when the sample (test set) has more than ten positive and ten negative examples [22]. We compared all heuristics pairwise at each query, and determined the maximum confidence level at which the difference between them was significant. We used cutoffs at the 0.60, 0.70, 0.80, 0.90, 0.95, 0.975, 0.99, and 0.995 confidence levels. We considered the mean AUC of all iterations in each fold a distinct item in the sample population. It is not feasible to report the paired-t results for all experiments, but they will be mentioned where appropriate.

Overall significance is established between two heuristics by taking the median confidence level at which they differ across all queries. We use the median because it is insensitive to outliers, and because it requires that a heuristic be significantly better on at least half of the queries for it to be considered significantly better overall. To demonstrate how this is done consider the paired-t results for hypothetical heuristics A, B, and C in Figure 5.1. The three algorithms are compared pairwise over five query iterations. In each row the algorithms being compared are listed in the first two columns. The subsequent values indicate the level of significance at which the two heuristics differ at each iteration. A positive value indicates that the heuristic in the first column is significantly better, and a negative value the heuristic in the second column. Therefore, heuristic C is significantly better than heuristic B on iteration 23 at the 0.60 confidence level. We sort each row in ascending order as shown in Figure 5.2. The middle value (shown in bold) is the median. From these results we say that heuristic A is significantly better than both B and C, while there is no significant difference between B and C.

Query Iteration		20	21	22	23	24
A	B	0.70	0.70	0.80	0.90	0.80
A	C	0.60	0.70	0.70	0.70	0.80
B	C	0.60	0.00	0.00	-0.60	-0.70

Figure 5.1: Example paired-t significance results for heuristics A, B, and C over five query iterations.

Because we have done a broad analysis of active learning heuristics to include using two base learners and two initial labeled set sizes, there are a large number of results to report. Results are generally displayed using learning curves, but with so many it is difficult to get

A	B	0.70	0.70	**0.80**	0.80	0.90
A	C	0.60	0.70	**0.70**	0.70	0.80
B	C	-0.70	-0.60	**0.00**	0.00	0.60

Figure 5.2: Example sorted paired-t significance results for heuristics A, B, and C. The median is displayed in bold type.

a handle on the big picture. To aid in this endeavor we also make use of three summary statistics.

5.1.1 Ranked Performance

The first statistic we use is what we call the *ranked performance* of the heuristics. With this statistic we establish a ranking over heuristics on a data set taking the paired-t tests into account. We rank each heuristic according to how many of the other heuristics it is significantly better than. With n heuristics the best heuristic will receive a rank of 1 and the worst a rank of n. Therefore, if heuristic A performs significantly worse than heuristic B but is significantly better than all others, it gets a rank of 2. It is also possible for a heuristic to have a rank *range* rather than a single value. This occurs when the difference between it and another heuristic is not significant. As an example, if heuristic C is significantly worse than two of the heuristics, and it has no significant difference with two others then it will receive a rank of 3–5. In general, a heuristic A can be considered significantly better than a heuristic B if there is no way for B to be ranked higher than A within the established ranking scheme.

The rank performance statistic is also summarized across all of the data sets by displaying the mean rank and number of wins for each heuristic. An algorithm's mean rank is simply the mean of its lower and upper ranks across all data sets. A win is awarded on each data set for the heuristic that receives a rank of 1. In the case where multiple heuristics have a 1 in their rank range (i.e. there is no significant difference between them), then partial credit is assigned to each weighted by the width of their rank range. Let $Q = q_1, ..., q_n$ be the set of heuristics that have a 1 in their rank range, and $r(q_i)$ be the width of the rank range for heuristic q_i. The win credit earned $W(q_i)$ for heuristic q_i is:

$$W(q_i) = \frac{1/(r(q_i) - 1)}{\sum_{j=1}^{n} 1/(r(q_j) - 1)} \ .$$

For example consider two heuristics A and B that have rank ranges of 1–2 and 1–3, respectively. Hence, we have $r(A) = 2$ and $r(B) = 3$. The win credit earned by heuristic A is

$$W(A) = \frac{1/(2 - 1)}{1/(2 - 1) + 1/(3 - 1)} = 0.67 \ ,$$

and the win credit earned by heuristic B is

$$W(B) = \frac{1/(3 - 1)}{1/(2 - 1) + 1/(3 - 1)} = 0.33 \ .$$

5.1.2 Data Utilization

One of the primary aims of active learning is to reduce the amount of training data needed to induce an accurate model. To measure this we use another summary statistic called *data utilization*. We define the *target AUC* as the mean AUC achieved by Random for the final 20% of the queries. We then report the minimum number of examples needed by each algorithm to achieve the target AUC. We also report the data utilization ratio, which is the number of examples needed by each heuristic to reach the target AUC divided by the number needed by Random. In the event that a heuristic does not reach the target AUC we simply report that the minimum number of examples needed is greater than the size of L after the last query round. This measure reflects how efficiently a heuristic uses the data, but may not reflect large changes in performance in the later query rounds. To summarize over all data sets we also report the median data utilization ratio and number of wins for each heuristic. This metric is similar to one used by Melville et al. [27] and Abe et al. [1].

5.1.3 Area Under the Learning Curve

Our last summary statistic is the area *under the learning curve above Random*, which is the difference between the area under the learning curve for a heuristic and that of Random. A negative value for this statistic indicates that the heuristic on average performed worse than Random. The area under the learning curve for a heuristic is calculated as the sum of the AUC achieved by a heuristic over all query rounds. It is more sensitive to the overall performance of the heuristics throughout the learning process than the previous two statistics. To summarise across all data sets we also report the mean area above Random achieved by a heuristic as well as the number of wins.

5.2 Results

5.2.1 SMO when Starting with 15 Labeled Examples

The learning curves for the active learning heuristics when using SMO as the base learner and starting with 15 examples in the initial labeled set L_0 are shown in Figures A.1, A.2, and A.3 of Appendix A. Overall, Closest does very well as its performance is close to the best on most of the data sets, and does better than all others on Diabetes and KR vs. KP. Figure 5.3 shows the learning curves for the KR vs. KP data set. Here Closest is significantly better than its nearest competitor ERS at the 0.8 confidence level between 53 and 108 labeled points.

The performance of RASP is generally similar to Closest, and is the best of the four algorithms introduced in this thesis on these experiments. This makes sense since both

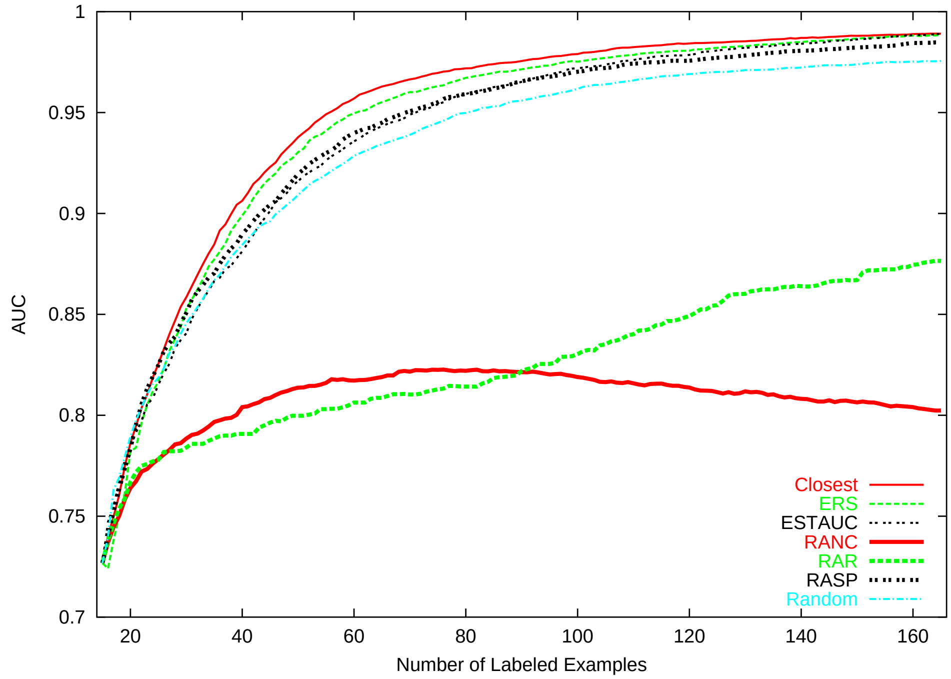


Figure 5.3: Learning curves for the KR vs. KP data set using SMO as the classifier and starting with 15 initial labeled points.

heuristics focus on examples that the current classifier is uncertain about. Although RASP is competitive with Closest, it does not dominate on any of the data sets.

ESTAUC and ERS are also very similar to each other on most of the learning curves. Again, this is probably due to the similarity in their approaches. Neither seems to have a clear advantage over the other as ESTAUC performs better on some data sets (e.g. Ionosphere), and ERS is better on others (e.g. Breast Cancer). Figure 5.4 shows the USPS learning curves on which both heuristics do well. They both have a significant advantage over RASP between 33 and 112 labeled points (ESTAUC ≥ 0.8 and ERS ≥ 0.7 confidence levels). However, there is no significant difference between ESTAUC and ERS between 28 and 145 labeled points.

RANC does not do very well on most of the data sets due to the fact that SVMs do a good job of separating the training set into the positive and negative classes. This means

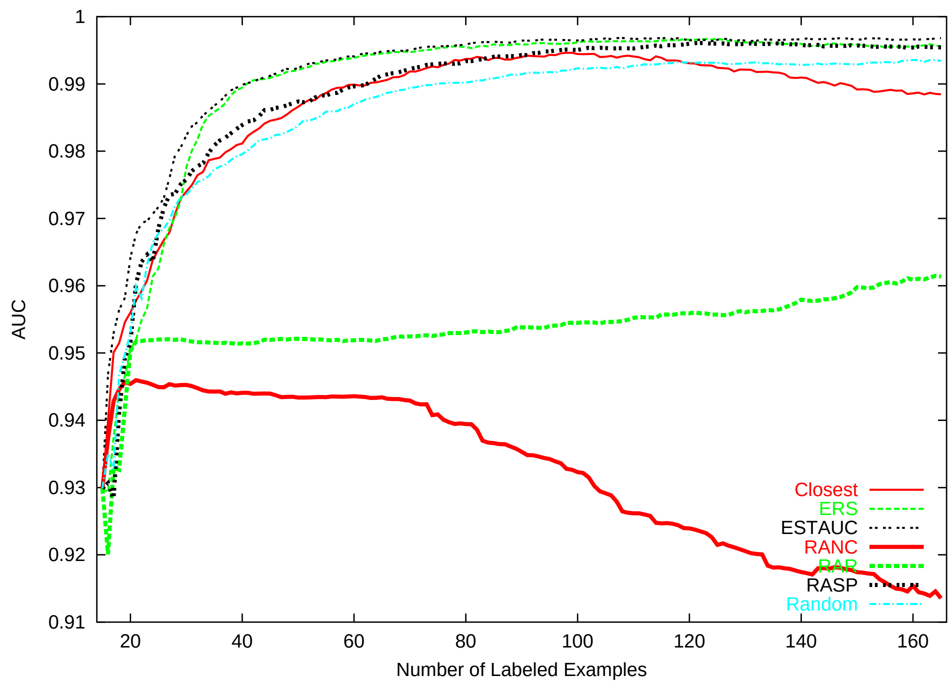


Figure 5.4: Learning curves for the USPS data set using SMO as the classifier and starting with 15 initial labeled points.

that the lowest positive labeled example will likely be ranked fairly high. Therefore, RANC focuses on unlabeled examples for which the classifier is fairly confident in a positive label. As a result its performance is generally worse than Random. One exception to this is the Breast Cancer data set, on which RANC does fairly well. This is probably due to the difficulty of the problem since no heuristic achieves an AUC higher than 0.72. It is likely that the Breast Cancer data is fairly noisy, which causes the classifier to be less confident on the labeled set, and benefits RANC.

The RAR heuristic likewise does worse than Random for the most part. This is due to the fact that it favors examples that change the current hypothesis as little as possible. This results in a behavior that is similar to relevance sampling where the heuristic focuses on examples for which it is already confident in its prediction.

Table 5.2 shows the ranked performance of the heuristics on the experiments with SMO and an initial labeled set with 15 examples. The table supports the conclusions made from the learning curves as Closest out performs the other heuristics, with RASP following close behind. In fact, there is only a difference of 0.1 between the mean rank of these two. The wins tell a different story, however. Closest receives almost twice the win credit as RASP since Closest has two uncontested wins and RASP does not have any.

There are mixed results for ERS and ESTAUC since ERS has a lower mean rank range, but ESTAUC has more win credit. This means that overall ERS ranks slightly better than ESTAUC, but when ESTAUC does well it is more significant than when ERS is at the top. It is also worth noting that Random does not do much worse than these two heuristics.

Table 5.2: Ranked performance of the active learning heuristics on the initial experiments using SMO when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
B. Cancer	3–5	1	5–6	2–4	2–3	4–6	7
Colic	1–4	3–5	2–5	7	6	1–2	2–5
Credit A	4–5	1–3	2–5	6	7	1–4	1–4
Credit G	1–2	3–5	5–6	3–6	7	1–3	2–5
Diabetes	1	3–6	5–6	3–5	7	2	3–5
Ionosphere	2–5	4–5	1–2	7	6	3–5	1–4
KR vs. KP	1	2	3–4	6–7	6–7	3–4	5
USPS	4–5	2	1	7	6	3	4–5
Vote	1–2	4–6	3	5–6	7	1–2	4–5
Mean	2.0–3.3	2.6–3.9	3.0–4.2	5.1–6.1	6.0–6.2	2.1–3.4	3.2–5.0
Wins	3.42	1.43	1.75	0.00	0.00	1.87	0.54

Table 5.3 shows the data utilization of the heuristics. Again, Closest is shown to be superior by achieving the target AUC before any other heuristic on four of the data sets. Interestingly, this metric shows more of a disparity between Closest and RASP since RASP collects only one win on the experiments. Upon closer inspection, though, the data utilization ratio for the two heuristics is often close. Closest simply out performs RASP by a small margin on most of the data sets. ERS and ESTAUC are again even with both garnering two wins, and finally, RANC and RAR remain at the bottom of the pack.

Table 5.4 shows the area under the learning curve above Random (AULC) for this first set of experiments. Closest has a greater AULC than the other heuristics on three of the

Table 5.3: Data utilization for the active learning heuristics on the initial experiments using SMO when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RAND.	TARGET AUC
B. CANCER	80	**41**	93	54	54	73	108	0.69
	(0.74)	**(0.38)**	(0.86)	(0.5)	(0.5)	(0.67)	(1.00)	
COLIC	113	162	121	>165	>165	**94**	133	0.84
	(0.85)	(1.22)	(0.91)	(>1.24)	(>1.24)	**(0.71)**	(1.00)	
CREDIT A	>165	**145**	>165	>165	>165	>165	150	0.91
	(>1.10)	**(0.97)**	(>1.10)	(>1.10)	(>1.10)	(>1.10)	(1.00)	
CREDIT G	**119**	164	>165	160	>165	125	152	0.72
	(0.78)	(1.08)	(>1.09)	(1.05)	(>1.09)	(0.82)	(1.00)	
DIABETES	**89**	148	>165	133	>165	113	147	0.81
	(0.61)	(1.01)	(>1.12)	(0.90)	(>1.12)	(0.77)	(1.00)	
IONOSPHERE	126	>165	**77**	>165	>165	124	143	0.97
	(0.88)	(>1.15)	**(0.54)**	(>1.15)	(>1.15)	(0.87)	(1.00)	
KR VS. KP	**85**	96	106	>165	>165	109	151	0.97
	(0.56)	(0.64)	(0.70)	(>1.09)	(>1.09)	(0.72)	(1.00)	
USPS	78	55	**53**	>165	>165	80	118	0.99
	(0.66)	(0.47)	**(0.44)**	(>1.40)	(>1.40)	(0.68)	(1.00)	
VOTE	**55**	131	84	>165	>165	57	150	0.99
	(0.37)	(0.87)	(0.56)	(>1.10)	(>1.10)	(0.38)	(1.00)	
MEDIAN DUR	0.74	0.97	0.86	>1.10	>1.10	0.72	1.00	
WINS	4	2	2	0	0	1	0	

data sets and has the highest average area. RASP is close behind with the largest AULC on two data sets and a mean area that is fairly close to that of Closest. ERS and ESTAUC are also close to one another with ERS having the greater mean AULC, but ESTAUC obtaining more wins. Finally, the mean areas of RAR and RANC are significantly lower than those of the other heuristics. In fact, they are both significantly worse than Random on average.

Table 5.4: Area under the learning curve above Random for the active learning heuristics on the initial experiments using SMO when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
B. Cancer	2.07	**4.21**	1.28	2.65	2.95	1.51	0.00
Colic	0.39	-0.54	-0.33	-7.07	-5.81	**0.74**	0.00
Credit A	-0.44	-0.35	-0.45	-2.00	-5.70	-0.08	**0.00**
Credit G	**1.03**	-0.38	-1.33	-0.58	-3.35	0.88	0.00
Diabetes	**1.90**	-0.40	-1.05	0.19	-5.78	1.10	0.00
Ionosphere	-1.11	-2.20	**0.48**	-17.40	-3.72	-0.67	0.00
KR vs. KP	**2.66**	1.93	1.14	-18.67	-16.09	1.17	0.00
USPS	0.10	0.62	**0.85**	-8.11	-4.90	0.39	0.00
Vote	1.10	-0.44	0.34	-0.35	-1.04	**1.15**	0.00
Mean	0.856	0.272	0.103	-5.693	-4.827	0.688	0
Wins	3	1	2	0	0	2	1

5.2.2 Naïve Bayes when Starting with 15 Labeled Examples

The learning curves for active learning using naïve Bayes when starting with 15 examples in the initial labeled set can be found in Figures A.1, A.2, and A.3 of Appendix A. With these experiments we again see Closest performing better than the other heuristics in general. Figure 5.5 shows the general trend with Closest on top closely followed by RASP. Here Closest is significantly better than RASP between 18 and 44 labeled point at the 0.7 confidence level or greater. Closest also receives top marks according to the rank performance metric in Table 5.5. The mean rank is even lower than it was for SMO, but this is due to a decrease in performance of the ERS and ESTAUC heuristics, not an increase in Closest's performance. The data utilization statistic in Table 5.6 gives Closest the greatest number of wins at four, and it is able to achieve the target AUC with less than half of the labeled points required by Random for most of the data sets. As for the area under the learning curve in Table 5.7, Closest has the largest area on five of the data sets.

RASP is again competitive with Closest when using naïve Bayes as the classifier. It generally does as well or slightly worse than Closest, and does better on the Credit G data set. However, the gap is slightly more substantial between these heuristics according to the summary statistics.

Table 5.5: Ranked performance of the active learning heuristics on the initial experiments using naïve Bayes when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
B. Cancer	1–6	1–6	1–6	1–6	7	2–6	1–5
Colic	1–3	6	5	1–3	7	1–3	4
Credit A	1–2	5	6	4	7	1–2	3
Credit G	3–4	3–5	6–7	4–5	6–7	1–2	1–2
Diabetes	1–2	4–6	4–6	4–6	7	3	1–2
Ionosphere	1–2	3	5–6	4–6	7	1–2	4–5
KR vs. KP	1	5–7	5–7	3	5–7	2	4
USPS	1	5	4	3	6	2	7
Vote	1–2	7	4	3	5	1–2	6
Mean	1.2–2.6	4.3–5.7	4.4–5.7	3.0–4.3	6.3–6.7	1.6–2.7	3.4–4.2
Wins	4.47	0.14	0.14	0.47	0.00	2.61	1.17

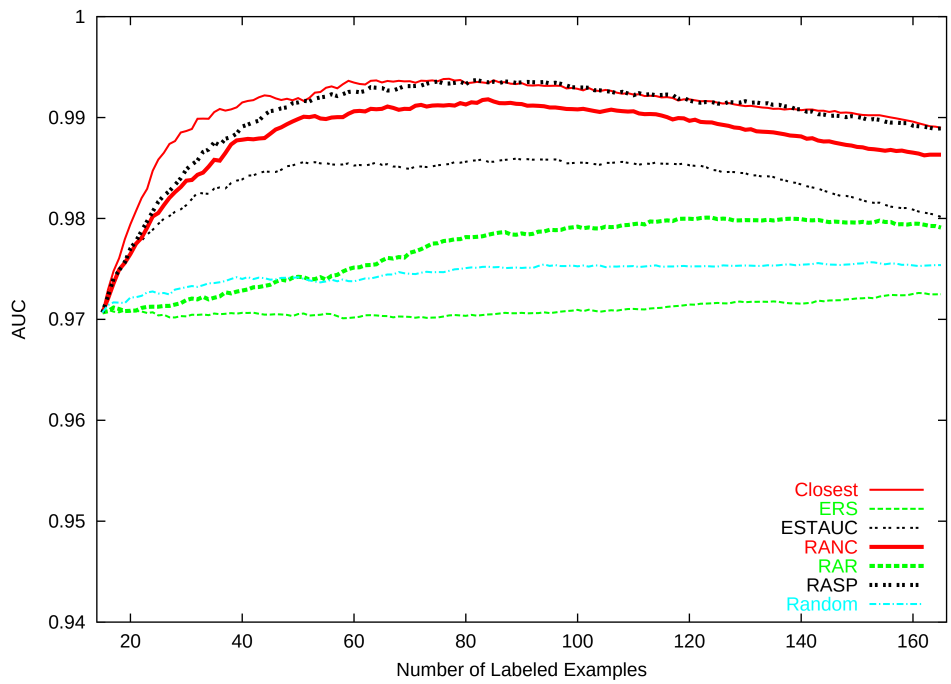


Figure 5.5: Learning curves for the Vote data set using naïve Bayes as the classifier and starting with 15 initial labeled points.

One interesting phenomenon is that Random does relatively well with the naïve Bayes classifier on some data sets. It is at or near the top on the Breast Cancer, Credit G and Diabetes data sets. See Figure 5.6 for an example. This is reflected in the rank performance table as Random receives more win credit than any other heuristic except RASP and Closest. Its mean rank is lower than RANC, however, so its performance varies.

The performance of RANC is very different when using naïve Bayes. Although it does not out perform Closest or RASP overall, it does significantly better than when SMO was used as the classifier. See Figure 5.5 for a good idea of the general trend for RANC's performance. This is due to the fact that naïve Bayes does not directly fit the labeled set, but rather builds a generative model that explains the data. This makes it more likely for a naïve Bayes classifier to predict a negative label for a training set example that is actually positive. This benefits RANC since examples selected will be in the area of uncertainty.

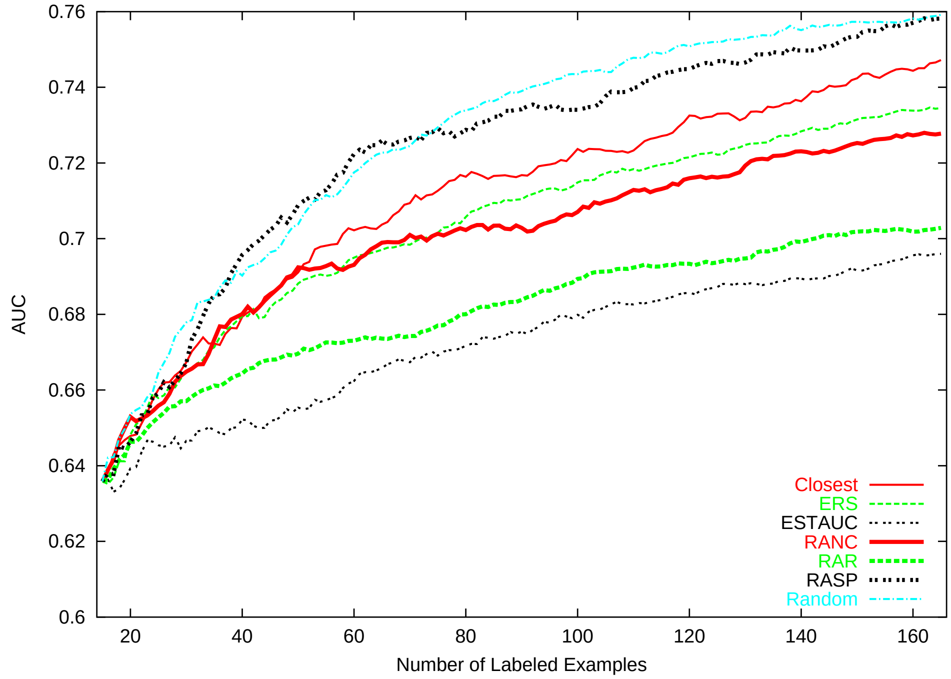


Figure 5.6: Learning curves for the Credit G data set using naïve Bayes as the classifier and starting with 15 initial labeled points.

The rank performance metric gives RANC the third best mean rank overall, but it is fourth in terms of win credit. It does not do as well on the data utilization, and only achieves the target AUC on four of the nine data sets due to the strength of Random. RANC does slightly better than Random overall to get the third best area under the learning curve.

The performance of the ERS and ESTAUC heuristics is also changed from the SMO results. Both heuristics do much worse. In fact, they are even out performed by Random in most cases. To explain this behavior we need only to look at the quality of the naïve Bayes probability estimates. It is well known that the violated independence assumption causes the naïve Bayes classifier to produce probability estimates that are highly polarized [30,41]. This means that probabilities of class membership provided by the classifier will almost always be very close to 0 or 1. Since ERS and ESTAUC require good probability estimates to determine the expected utility of a candidate example, they both suffer when naïve Bayes

Table 5.6: Data utilization for the active learning heuristics on the initial experiments using naïve Bayes when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Rand.	Target AUC
B. Cancer	>165	>165	>165	>165	>165	>165	**120**	0.70
	(>1.38)	(>1.38)	(>1.38)	(>1.38)	(>1.38)	(>1.38)	**(1.00)**	
Colic	**29**	>165	>165	30	>165	35	57	0.85
	(0.51)	(>2.89)	(>2.89)	(0.53)	(>2.89)	(0.61)	(1.00)	
Credit A	71	>165	>165	>165	>165	**70**	150	0.92
	(0.47)	(>1.10)	(>1.10)	(>1.10)	(>1.10)	**(0.47)**	(1.00)	
Credit G	>165	>165	>165	>165	>165	160	**149**	0.76
	(>1.11)	(>1.11)	(>1.11)	(>1.11)	(>1.11)	(1.07)	**(1.00)**	
Diabetes	>165	>165	>165	>165	>165	>165	**140**	0.83
	(>1.18)	(>1.18)	(>1.18)	(>1.18)	(>1.18)	(>1.18)	**(1.00)**	
Ionosphere	55	99	>165	>165	>165	**44**	146	0.97
	(0.38)	(0.67)	(>1.13)	(>1.13)	(>1.13)	**(0.30)**	(1.00)	
KR vs. KP	**55**	>165	>165	78	>165	58	148	0.92
	(0.37)	(>1.11)	(>1.11)	(0.53)	(>1.11)	(0.39)	(1.00)	
USPS	**17**	21	**17**	18	42	18	147	0.98
	(0.12)	(0.14)	**(0.12)**	(0.12)	(0.29)	(0.12)	(1.00)	
Vote	**18**	>165	19	19	64	19	141	0.98
	(0.13)	(>1.18)	(0.13)	(0.13)	(0.45)	(0.13)	(1.00)	
Median DUR	0.47	>1.11	>1.11	>1.10	>1.11	0.47	1.00	
Wins	4	0	1	0	0	2	3	

is used as the base learner.

The RAR heuristic likewise does poorly on these experiments, and is the worst overall. It is hurt both by its tendency to pick redundant examples for labeling and the flawed probability estimates from the classifier.

We tried a couple of techniques to improve the probability estimates produced by the naïve Bayes classifier, but did not see any marked improvement in the performance of the heuristics. The first technique, used by Roy and McCallum [33], was to use bagging to create a committee of m classifiers. The posterior probability estimate was then an unweighted average of the estimates of the m classifiers. Roy and McCallum obtained good results using this method on a automatic document classification task, but we did not see similar benefits on our data sets. The second approach that we tried was a binning technique proposed by Zadrozny and Elkan [41]. With this technique the training set is

Table 5.7: Area under the learning curve above Random for the active learning heuristics on the initial experiments using naïve Bayes when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
B. CANCER	-0.96	-0.94	-0.27	-0.58	-4.37	-0.83	**0.00**
COLIC	**3.12**	-2.58	-1.61	2.86	-3.28	2.65	0.00
CREDIT A	**0.71**	-1.63	-2.09	-0.51	-4.67	0.52	0.00
CREDIT G	-2.40	-3.46	-8.24	-3.94	-6.85	-0.46	**0.00**
DIABETES	-0.56	-1.81	-2.07	-3.56	-4.30	-1.11	**0.00**
IONOSPHERE	1.09	0.26	-0.68	-0.31	-1.44	**1.27**	0.00
KR VS. KP	**8.59**	-23.18	-23.53	3.46	-23.41	7.33	0.00
USPS	**2.68**	1.17	1.89	2.32	0.87	2.49	0.00
VOTE	**2.45**	-0.54	1.36	2.04	0.36	2.34	0.00
MEAN	1.636	-3.634	-3.916	0.198	-5.232	1.589	0.000
WINS	5	0	0	0	0	1	3

sorted according probability scores provided by the naïve Bayes classifier. The sorted set is divided into b subsets of equal size called bins. For each bin the lower and upper boundary scores are computed, which are used to assign new test examples to the appropriate bin. The probability of class membership for class j in a bin is calculated as the fraction of training examples in that bin which belong to class j. While this is an interesting approach, we found that it does not work well with the small training sets used in active learning.

5.2.3 SMO when Starting with 100 Labeled Examples

For the next batch of experiments we again used SMO as the base learner, but started with 100 examples in the initial labeled training set L_0. A side effect of starting with a large training set is that on some of the data sets the initial classifier performs close to optimal and there is little room for improvement. Consequently, there is little difference between the active learning heuristics in some cases. This is evident on the Credit A and Ionosphere data sets.

The overall trend is the same as for the experiments with $|L_0| = 15$, though the differences between heuristics are smaller due to the fact that there is less room for improvement. Closest does the best overall; performing at or near the top on almost all of the data sets. RASP is again competitive doing only slightly worse than Closest. ESTAUC and ERS are competitive with each other though significantly worse than Closest and RASP. Then RANC, RAR and Random bring up the rear. Figure 5.7 shows the learning curves for the Credit G data set which is representative of the general performance of the heuristics for these experiments. The rank performance, data utilization, and area under the learning curve metrics can be found in Tables 5.8, 5.9, and 5.10 respectively.

Table 5.8: Ranked performance of the active learning heuristics on the initial experiments using SMO when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
B. CANCER	2–7	3–6	2–7	1–4	1–2	4–6	3–7
COLIC	1–6	1–5	1–5	1–6	7	1–6	3–6
CREDIT A	3–6	1–3	4–7	5–7	4–7	1–3	1–4
CREDIT G	1–3	2–6	3–6	3–6	7	1–2	3–6
DIABETES	1–5	1–5	5–6	1–6	7	1–5	1–5
IONOSPHERE	1–3	4–7	1–3	4–5	5–7	1–3	5–7
KR VS. KP	1	2–3	2–3	7	6	4	5
USPS	1–4	1–4	1–4	7	6	1–4	5
VOTE	1–2	3–4	5	2–4	6–7	1–3	6–7
MEAN	1.3–4.1	2.0–4.8	2.7–5.1	3.4–5.8	5.4–6.2	1.7–4.0	3.6–5.8
WINS	2.97	1.06	0.81	0.60	0.75	2.35	0.46

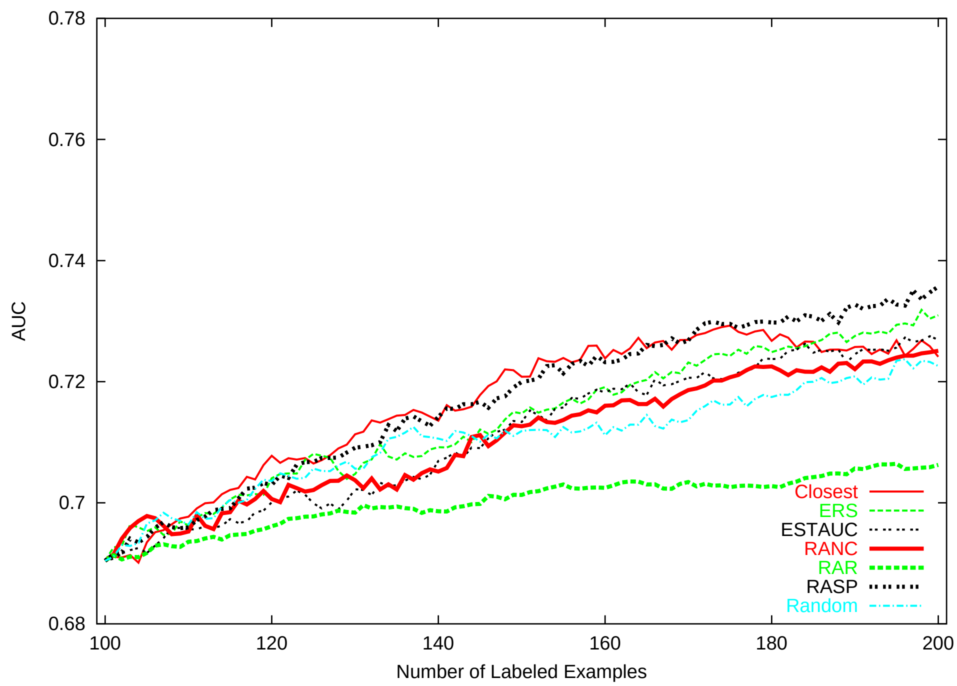


Figure 5.7: Learning curves for the Credit G data set using naïve Bayes as the classifier and starting with 100 initial labeled points.

Table 5.9: Data utilization for the active learning heuristics on the initial experiments using SMO when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Rand.	Target AUC
B. Cancer	136	153	147	**126**	132	177	173	0.72
	(0.79)	(0.88)	(0.85)	**(0.73)**	(0.76)	(1.02)	(1.00)	
Colic	154	149	147	**135**	>200	144	170	0.87
	(0.91)	(0.88)	(0.86)	**(0.79)**	(>1.18)	(0.84)	(1.00)	
Credit A	>200	177	>200	>200	>200	**170**	176	0.91
	(>1.14)	(1.01)	(>1.14)	(>1.14)	(>1.14)	**(0.97)**	(1.00)	
Credit G	**148**	166	172	176	>200	153	190	0.72
	(0.78)	(0.87)	(0.91)	(0.93)	(>1.05)	(0.81)	(1.00)	
Diabetes	177	>200	>200	178	>200	196	**171**	0.82
	(1.04)	(>1.17)	(>1.17)	(1.04)	(>1.17)	(1.15)	**(1.00)**	
Ionosphere	119	160	**111**	166	>200	127	189	0.97
	(0.63)	(0.85)	**(0.59)**	(0.88)	(>1.06)	(0.67)	(1.00)	
KR vs. KP	**126**	133	132	>200	>200	145	191	0.98
	(0.66)	(0.70)	(0.69)	(>1.05)	(>1.05)	(0.76)	(1.00)	
USPS	105	105	**102**	>200	166	112	127	0.99
	(0.83)	(0.83)	**(0.80)**	(>1.57)	(1.31)	(0.88)	(1.00)	
Vote	**111**	113	113	110	143	113	141	0.99
	(0.79)	(0.80)	(0.80)	(0.78)	(1.01)	(0.80)	(1.00)	
Median DUR	0.79	0.87	0.85	0.93	>1.05	0.84	1.00	
Wins	3	0	2	2	0	1	1	

Table 5.10: Area under the learning curve above Random for the active learning heuristics on the initial experiments using SMO when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
B. Cancer	0.28	0.21	0.23	0.55	**0.62**	-0.19	0.00
Colic	0.21	0.24	**0.25**	0.11	-0.70	0.14	0.00
Credit A	-0.12	**0.05**	-0.15	-0.25	-0.20	0.02	0.00
Credit G	0.61	0.34	0.08	0.04	-1.03	**0.67**	0.00
Diabetes	-0.01	-0.03	-0.38	-0.12	-0.97	-0.01	**0.00**
Ionosphere	**0.44**	0.06	0.38	0.18	-0.09	0.35	0.00
KR vs. KP	**1.03**	0.86	0.91	-2.32	-0.61	0.55	0.00
USPS	0.17	**0.19**	0.18	-1.34	-0.04	**0.19**	0.00
Vote	**0.49**	0.32	0.19	0.36	0.06	0.41	0.00
Mean	0.344	0.249	0.188	-0.31	-0.329	0.237	0
Wins	3	2	1	0	1	2	1

5.2.4 Naïve Bayes when Starting with 100 Labeled Examples

The results for the experiments run with naïve Bayes using an initial labeled set of size 100 show the same trends as when starting with 15 examples. With these experiments, however, Closest dominates all of the heuristics, including RASP. In fact, it is out performed on only two of the data sets (Breast Cancer and Diabetes). This can be seen from the summary statistics in Tables 5.11, 5.12, and 5.13. Closest achieves as many or more wins on the rank performance and data utilization metrics than all of the other heuristics combined. In addition, it has the highest area under the learning curve on all of the data sets except the two previously mentioned. Figure 5.8 shows the learning curves for the Colic data set on which Closest does particularly well. Closest is significantly better than all of the other heuristics from 120 labeled examples on at the 0.7 confidence level or greater. The remainder of the heuristics follow the same trend as with $L_0 = 15$. RASP comes in a solid second on all metrics. RANC is third, and Random, ERS, ESTAUC and RAR are found at the bottom.

Table 5.11: Ranked performance of the active learning heuristics using naïve Bayes when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
B. CANCER	3–7	4–6	1–2	3–7	1–2	3–7	3–7
COLIC	1	7	5–6	2–3	5–6	2–3	4
CREDIT A	1	5	7	3–4	6	2	3–4
CREDIT G	1–3	5–7	4–7	4–6	4–7	1–3	1–3
DIABETES	4–6	3–6	4–6	7	3–4	1–2	1–2
IONOSPHERE	1–3	4	7	1–3	5–6	1–3	5–6
KR VS. KP	1	6–7	6–7	3	5	2	4
USPS	1	6	3	4–5	4–5	2	7
VOTE	1	7	3–4	3–4	5	2	6
MEAN	1.6–2.7	5.2–6.1	4.2–5.4	3.3–4.7	4.2–5.1	1.8–2.9	3.8–4.8
WINS	5.67	0.00	0.50	0.33	0.50	1.17	0.83

There are a several conclusions that can be made from these four sets of experiments. First, Closest and RASP perform the best of all the heuristics tested. They do well consistently on almost all of the data sets, and rarely perform significantly worse than any of

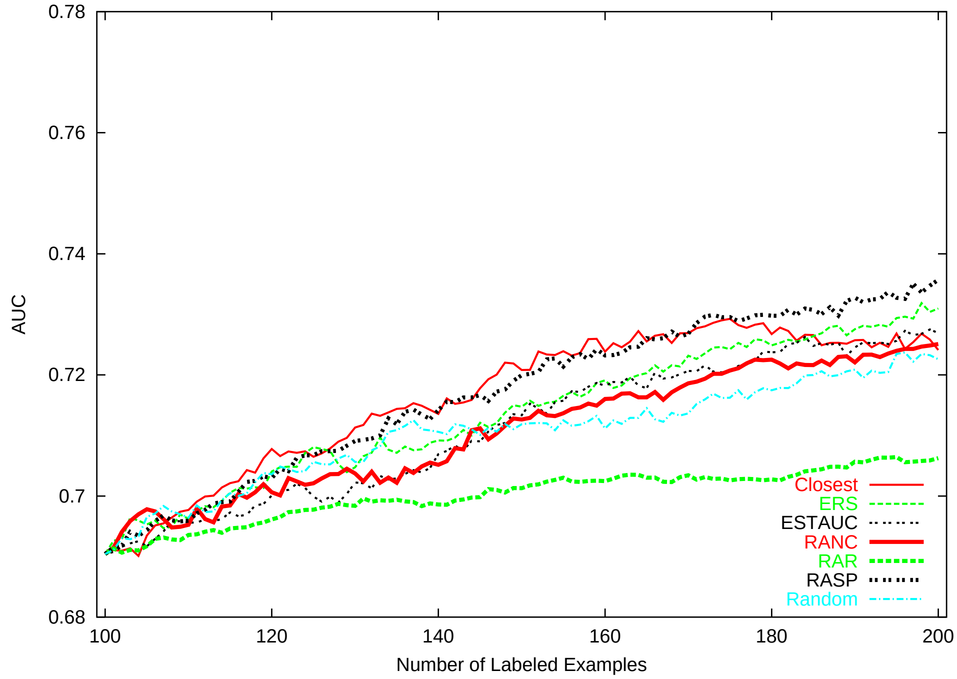


Figure 5.8: Learning curves for the Colic data set using naïve Bayes as the classifier and starting with 100 initial labeled points.

the other heuristics. Also, when a naïve Bayes classifier has an accurate hypothesis, Closest provides the best results for continued improvement. Second, the ERS and ESTAUC heuristics are hampered by poor probability estimates. Both heuristics perform substantially better when using SMO as the base learner than when using naïve Bayes. We attribute this to the higher quality probability estimates provided by SMO. Another thing to note is due to their computational complexity, the ERS, ESTAUC, and RAR heuristics consider as candidates for labeling a relatively small random subset of U rather than all of U. If they were able to instead evaluate all of U, their performance might improve.

The discussion thus far has focused on the performance of the active learning heuristics, but an additional contribution of this thesis is to compare the performance of SMO to naïve Bayes in the active learning task. To facilitate this, the learning curves for the two base learners appear side by side with the same y-axis scale in Appendix A. We use the highest

Table 5.12: Data utilization for the active learning heuristics using naïve Bayes when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RAND.	TARGET AUC
B. CANCER	111 (1.06)	**108** **(1.03)**	113 (1.08)	>200 (>1.90)	109 (1.04)	114 (1.09)	105 (1.00)	0.71
COLIC	100 (1.00)	100 (1.00)	100 (1.00)	100 (1.00)	100 (1.00)	100 (1.00)	100 (1.00)	0.85
CREDIT A	**108** **(0.64)**	>200 (>1.18)	>200 (>1.18)	195 (1.15)	>200 (>1.18)	111 (0.65)	170 (1.00)	0.92
CREDIT G	**174** **(0.91)**	>200 (>1.05)	>200 (>1.05)	>200 (>1.05)	>200 (>1.05)	183 (0.96)	191 (1.00)	0.76
DIABETES	>200 (>1.43)	>200 (>1.43)	>200 (>1.43)	>200 (>1.43)	>200 (>1.43)	167 (1.19)	**140** **(1.00)**	0.84
IONOSPHERE	125 (0.70)	152 (0.85)	>200 (>1.12)	126 (0.70)	>200 (>1.12)	**120** **(0.67)**	179 (1.00)	0.97
KR VS. KP	**114** **(0.60)**	>200 (>1.05)	>200 (>1.05)	133 (0.69)	>200 (>1.05)	116 (0.61)	191 (1.00)	0.92
USPS	**101** **(0.69)**	102 (0.70)	102 (0.70)	102 (0.70)	102 (0.70)	102 (0.70)	146 (1.00)	0.98
VOTE	104 (0.66)	>200 (>1.27)	**103** **(0.66)**	112 (0.71)	106 (0.68)	106 (0.68)	157 (1.00)	0.97
MEDIAN DUR	0.70	>1.05	>1.05	1.00	>1.05	0.70	1.00	
WINS	4	1	1	0	0	1	1	

AUC achieved by any heuristic as the basis for comparing the classifiers.

In the experiments with an initial labeled set of size 15, naïve Bayes shows a distinct advantage. The AUC achieved by naïve Bayes before the first query round (trained only on L_0) is higher than that of the SMO classifier on all of the data sets. However, SMO catches up quickly most of the time, and on the tests with $|L_0| = 100$, SMO starts off with a higher AUC more often than naïve Bayes. There is a tendency for each classifier to do better on certain data sets. For instance, naïve Bayes always does better on Credit A, Credit G, and Diabetes, while SMO does better on Breast Cancer, KR vs. KP, and Vote. Despite this, the general trend suggests that naïve Bayes does better with smaller data sets, while SMO generally catches up and often surpasses naïve Bayes once the labeled set is sufficiently large. Naïve Bayes should be preferred on problems where the labeled set remains small after active learning is complete, and SMO should be used when L will be reasonably large

Table 5.13: Area under the learning curve above Random for the active learning heuristics using naïve Bayes when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
B. CANCER	-0.18	-0.11	0.54	-0.21	**0.64**	-0.12	0.00
COLIC	**1.08**	-0.65	-0.37	0.45	-0.33	0.52	0.00
CREDIT A	**0.73**	-0.39	-0.92	-0.08	-0.62	0.45	0.00
CREDIT G	**0.20**	-1.35	-0.99	-0.66	-1.01	0.01	0.00
DIABETES	-0.74	-0.43	-0.78	-1.21	-0.42	-0.08	**0.00**
IONOSPHERE	**0.37**	0.08	-0.20	0.28	-0.12	0.34	0.00
KR VS. KP	**4.70**	-9.26	-9.29	1.42	-4.72	3.90	0.00
USPS	**1.34**	0.72	0.94	0.81	0.79	1.12	0.00
VOTE	**0.85**	-0.08	0.55	0.61	0.38	0.69	0.00
MEAN	0.928	-1.274	-1.169	0.157	-0.601	0.759	0.000
WINS	7	0	0	0	1	0	1

at the time one wants to use the classifier.

5.2.5 Synthetic Data

Given how ERS, ESTAUC, and RAR are defined with respect to probability estimates, we wanted to see how well they would do if given perfect probability estimates. To do this we developed two synthetic datasets. The synthetic data was created from two overlapping ten-dimensional Gaussian distributions: $\mathcal{N}_+(\mu_+, \sigma^2 I_{10})$ and $\mathcal{N}_-(\mu_-, \sigma^2 I_{10})$, where $\sigma^2 = 12.5$ and I_{10} is the 10×10 identity matrix. For the first dataset (**Gaussian 0,5**) $\mu_+ = (0, \ldots, 0)$ and $\mu_- = (5, \ldots, 5)$, and for the second dataset (**Gaussian 0,10**) $\mu_+ = (0, \ldots, 0)$ and $\mu_- = (10, \ldots, 10)$. The classification of each example generated by $\mathcal{N}_+$ is positive and that of those generated by $\mathcal{N}_-$ is negative. We then ran ESTAUC and RAR with the true Gaussian pdfs P in place of the estimates $\hat{P}$. We also ran ERS with the true pdf in place of the first $\hat{P}(y' \mid x')$ term of Equation 3.2.

Without exception ESTAUC and RAR dominate the other heuristics on the synthetic data when using either base learner, and with either initial labeled set size. Figures 5.9 and 5.10 show the performance of the active learning heuristics for two representative experiments. The gap between ESTAUC and RAR and the other heuristics in both plots is significant at the 0.9 confidence level or greater from 16 to 55 labeled points and 0.8 or better over the entire plot. Interestingly, ERS did not gain a similar benefit from having perfect probability estimates. The summary statistics for these experiments can be found in Tables 5.14–5.25.

These results clearly demonstrate the potential of these two heuristics if given good probability estimates. Unfortunately, that is easier said than done. Any probability model that we might generate will necessarily be based on the labeled training data. In active learning this is generally a relatively small set of data. Obviously, it is difficult to generate high quality probability estimates from such a small training set.

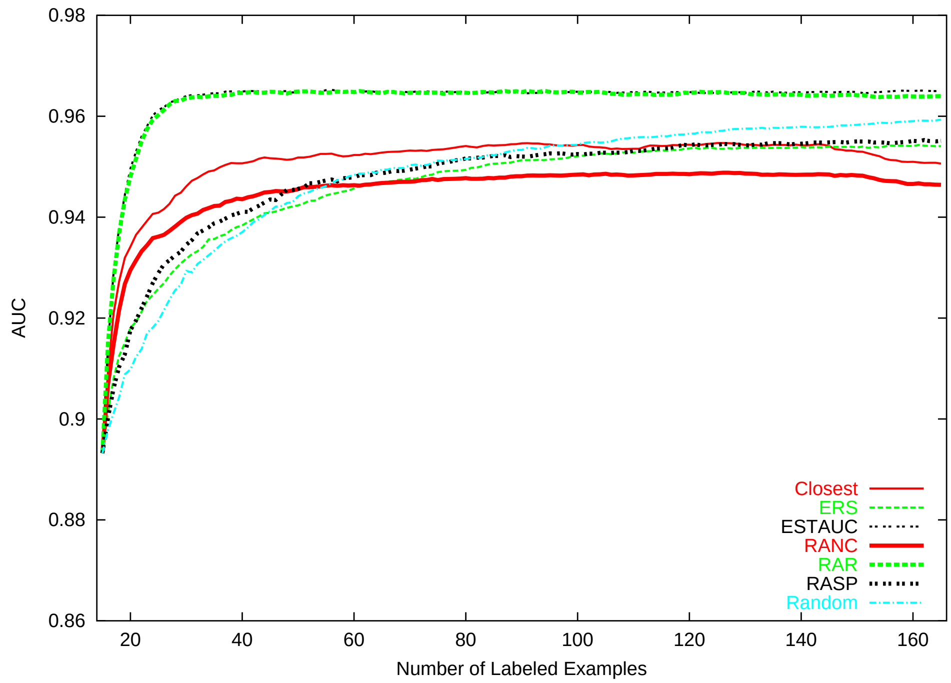


Figure 5.9: Learning curves for the Gaussian 0,10 data set using naïve Bayes as the classifier and starting with 15 initial labeled points.

Table 5.14: Ranked performance of the active learning heuristics on synthetic data using SMO when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
GAUSSIAN 0,5	7	3	2	6	1	4–5	4–5
GAUSSIAN 0,10	6–7	3–5	1	6–7	2	3–5	3–5
MEAN	6.5–7.0	3.0–4.0	1.5	6.0–6.5	1.5	3.5–5.0	3.5–5.0
WINS	0.0	0.0	1.0	0.0	1.0	0.0	0.0

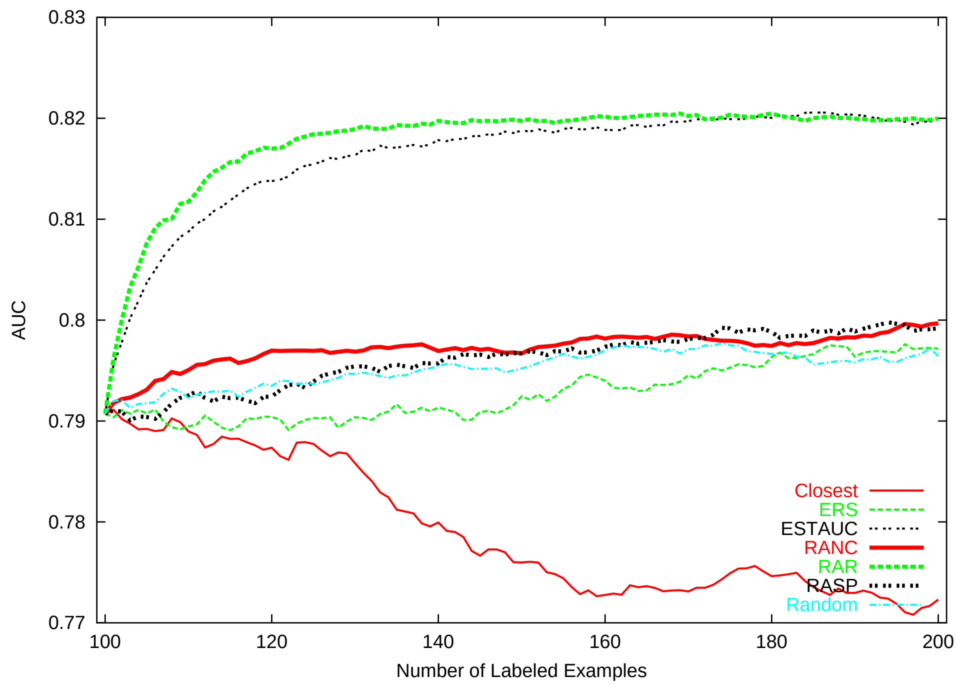


Figure 5.10: Learning curves for the Gaussian 0,5 data set using SMO as the classifier and starting with 100 initial labeled points.

Table 5.15: Data utilization for the active learning heuristics on synthetic data using SMO when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Rand.	Target AUC
Gaussian 0,5	>165 (>1.57)	66 (0.63)	20 (0.19)	>165 (>1.57)	**19 (0.18)**	142 (1.35)	105 (1.00)	0.75
Gaussian 0,10	163 (1.21)	116 (0.86)	19 (0.14)	>165 (>1.22)	**18 (0.13)**	136 (1.01)	135 (1.00)	0.95
Median DUR	>1.39	0.75	0.17	>1.40	0.16	1.18	1.00	
Wins	0	0	0	0	2	0	0	

Table 5.16: Area under the learning curve above Random for the active learning heuristics on synthetic data using SMO when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
Gaussian 0,5	-13.87	1.10	7.96	-10.82	**9.18**	-0.79	0.00
Gaussian 0,10	-5.42	-0.91	**3.36**	-6.13	2.84	-0.37	0.00
Mean	-9.645	0.095	5.660	-8.475	6.010	-0.580	0.000
Wins	0	0	1	0	1	0	0

Table 5.17: Ranked performance of the active learning heuristics on synthetic data using naïve Bayes when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
Gaussian 0,5	3	7	1–2	4–6	1–2	4–5	5–6
Gaussian 0,10	3–4	5–6	1	7	2	4–6	3–5
Mean	3.0–3.5	6.0–6.5	1.0–1.5	5.5–6.5	1.5–2.0	4.0–5.5	4.0–5.5
Wins	0.0	0.0	1.5	0.0	0.5	0.0	0.0

Table 5.18: Data utilization for the active learning heuristics on synthetic data using naïve Bayes when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Rand.	Target AUC
Gaussian 0,5	>165	>165	**25**	>165	**25**	140	150	0.80
	(>1.1)	(>1.1)	**(0.17)**	(>1.1)	**(0.17)**	(0.93)	(1.00)	
Gaussian 0,10	>165	>165	**24**	>165	**24**	>165	151	0.96
	(>1.1)	(>1.1)	**(0.16)**	(>1.1)	**(0.16)**	(>1.1)	(1.00)	
Median DUR	>1.1	>1.1	0.17	(>1.1)	0.17	>1.02	1.00	
Wins	0	0	2	0	2	0	0	

Table 5.19: Area under the learning curve above Random for the active learning heuristics on synthetic data using naïve Bayes when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS	ESTAUC	RANC	RAR	RASP	Random
Gaussian 0,5	1.56	-1.37	4.30	0.60	**4.35**	0.68	0.00
Gaussian 0,10	0.46	-0.25	**2.29**	-0.41	2.24	-0.05	0.00
Mean	1.010	-0.810	3.295	0.095	3.295	0.315	0.000
Wins	0	0	1	0	1	0	0

Table 5.20: Ranked performance of the active learning heuristics on synthetic data using SMO when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
GAUSSIAN 0,5	7	6	2	3–4	1	3–5	4–5
GAUSSIAN 0,10	7	6	1	3	2	4–5	4–5
MEAN	7.0	6.0	1.5	3.0–3.5	1.5	3.5–5.0	4.0–5.0
WINS	0.0	0.0	1.0	0.0	1.0	0.0	0.0

Table 5.21: Data utilization for the active learning heuristics on synthetic data using SMO when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RAND.	TARGET AUC
GAUSSIAN 0,5	>200	180	**102**	119	**102**	141	154	0.80
	(>1.30)	(1.17)	**(0.66)**	(0.77)	**(0.66)**	(0.92)	(1.00)	
GAUSSIAN 0,10	>200	>200	103	124	**102**	171	187	0.96
	(>1.07)	(>1.07)	(0.55)	(0.66)	**(0.55)**	(0.91)	(1.00)	
MEDIAN DUR	>1.19	>1.13	0.61	0.72	0.61	0.92	1.00	
WINS	0	0	1	0	2	0	0	

Table 5.22: Area under the learning curve above Random for the active learning heuristics on synthetic data using SMO when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
GAUSSIAN 0,5	-1.61	-0.24	2.12	0.19	**2.27**	0.09	0.00
GAUSSIAN 0,10	-0.65	-0.2	**0.58**	0.19	0.53	0.03	0.00
MEAN	-1.13	-0.22	1.35	0.19	1.40	0.06	0.00
WINS	0	0	1	0	1	0	0

Table 5.23: Ranked performance of the active learning heuristics on synthetic data using naïve Bayes when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
GAUSSIAN 0,5	3–4	5	1–2	3–4	1–2	7	6
GAUSSIAN 0,10	3–5	3–6	1–2	3–5	1–2	5–6	7
MEAN	3.0–4.5	4.0–5.5	1.0–2.0	3.0–4.5	1.0–2.0	6.0–6.5	6.5
WINS	0.0	0.0	1.0	0.0	1.0	0.0	0.0

Table 5.24: Data utilization for the active learning heuristics on synthetic data using naïve Bayes when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RAND.	TARGET AUC
GAUSSIAN 0,5	108	123	**104**	108	**104**	>200	160	0.80
	(0.68)	(0.77)	**(0.65)**	(0.68)	**(0.65)**	(>1.25)	(1.00)	
GAUSSIAN 0,10	146	122	105	146	**104**	154	193	0.96
	(0.76)	(0.63)	(0.54)	(0.76)	**(0.54)**	(0.80)	(1.00)	
MEDIAN DUR	0.72	0.70	0.60	0.72	0.60	>1.03	1.00	
WINS	0	0	1	0	2	0	0	

Table 5.25: Area under the learning curve above Random for the active learning heuristics on synthetic data using naïve Bayes when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS	ESTAUC	RANC	RAR	RASP	RANDOM
GAUSSIAN 0,5	0.69	0.23	**1.01**	0.69	0.97	-0.20	0.00
GAUSSIAN 0,10	0.15	0.13	**0.55**	0.15	**0.55**	0.10	0.00
MEAN	0.42	0.18	0.78	0.42	0.76	-0.05	0.00
WINS	0	0	2	0	1	0	0

5.2.6 Pre-filtering with Closest Sampling

One problem with the ERS, ESTAUC, and RAR heuristics is that they require longer to execute than other techniques. To partially alleviate this problem they are only allowed to evaluate a reduced set of candidate examples from U. In the initial group of experiments the set of 200 candidate instances was chosen by random sampling. In most cases the set of candidate instances is small compared to the size of the unlabeled pool. As already noted, this places a significant limitation on these heuristics as it makes it unlikely that they will even have the opportunity to pick the 'best' example to label on any given query iteration. The Closest and RASP heuristics performed well in the initial group of experiments, and both of these techniques choose examples for which the current classifier is uncertain in its prediction. Consequently, for our final group of experiments we choose the candidate examples through a closest sampling filter rather than by a random sampling filter. To do this, the unlabeled pool is first ranked by the Closest heuristic from least to most certain. Then the top m most uncertain examples are chosen as the candidate pool for the ERS, ESTAUC and RAR heuristics. The assumption is that this set of candidates is more relevant and the performance of these heuristics will be improved by having a richer set of candidates from which to choose. We tried candidate set sizes m of 50, 100, 150 and 200, and settled on 100 as we found no increase in performance with more examples. This has the added benefit of reducing the run time of the heuristics by half. We repeated all of the initial experiments on the UCI data using the closest sampling pre-filter on ERS, ESTAUC and RAR. All of the heuristics that use the filtered candidate set are postfixed with a '-CS' for clarity. As a control we also introduced a new heuristic called **Random-CS** that selects an example to label uniformly at random from the candidate set. The heuristics that do not use the candidate pool are the same as in the initial group of experiments, and are not covered here except as a basis for comparison. The learning curves for all experiments can be found in Appendix A.3.

In the first set of experiments we used SMO as the base learner and 15 examples in the initial labeled set. All of the heuristics showed marked improvement using the filtered candidate set. ESTAUC-CS benefited the most, and actually performed better than Closest overall. The Random-CS heuristic was also a top performer scoring only slightly worse than ESTAUC-CS, but better than Closest. RAR and ERS, though improved, did not perform better than either Closest or Random-CS. Figure 5.11 shows the learning curves for the Ionosphere data set. ESTAUC-CS was significantly better than all of the other heuristics from 45 to 94 labeled examples at the 0.6 confidence level or greater. Tables 5.26, 5.27, and 5.28 show the summary statistics for these experiments. ESTAUC-CS was clearly the winner on the ranked performance metric with a win credit that was more than twice that of its nearest competitor. On the data utilization table Closest received the same number of wins as ESTAUC-CS, but ESTAUC-CS did better on all of the data sets which Closest didn't win. On the area under the learning curve Random-CS actually achieved a slightly higher mean area than ESTAUC-CS even though ESTAUC-CS had the max area on more data sets.

Table 5.26: Ranked performance of the active learning heuristics on the Closest prefiltering experiments using SMO when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS
B. Cancer	2–6	1–3	2–6	1–5	1–5	7–8	6–8	4–7
Colic	1–6	2–6	1–6	8	7	2–6	2–6	1–3
Credit A	5–7	1–4	1–4	8	6–7	5–6	1–4	1–4
Credit G	1–4	1–6	1–5	4–8	5–7	2–7	7–8	1–5
Diabetes	1–2	3–7	3–5	4–8	7–8	3–6	5–7	1–2
Ionosphere	3–5	6–7	1	8	6–7	2–5	2–4	2–5
KR vs. KP	2–4	2–4	1	8	6	5	7	2–4
USPS	5–6	3–4	1	8	7	2–4	5–6	2–3
Vote	2	4–8	4–5	6–8	5–8	1	5–7	3
Mean	2.4–4.7	2.6–5.4	1.7–3.8	6.1–7.7	5.6–6.9	3.3–5.3	4.4–6.3	1.9–4.0
Wins	1.04	0.94	3.71	0.25	0.25	1.00	0.25	1.55

ESTAUC-CS does well again when using SMO and starting with 100 examples in the initial labeled set. It does have more stiff competition from ERS-CS and Closest, however.

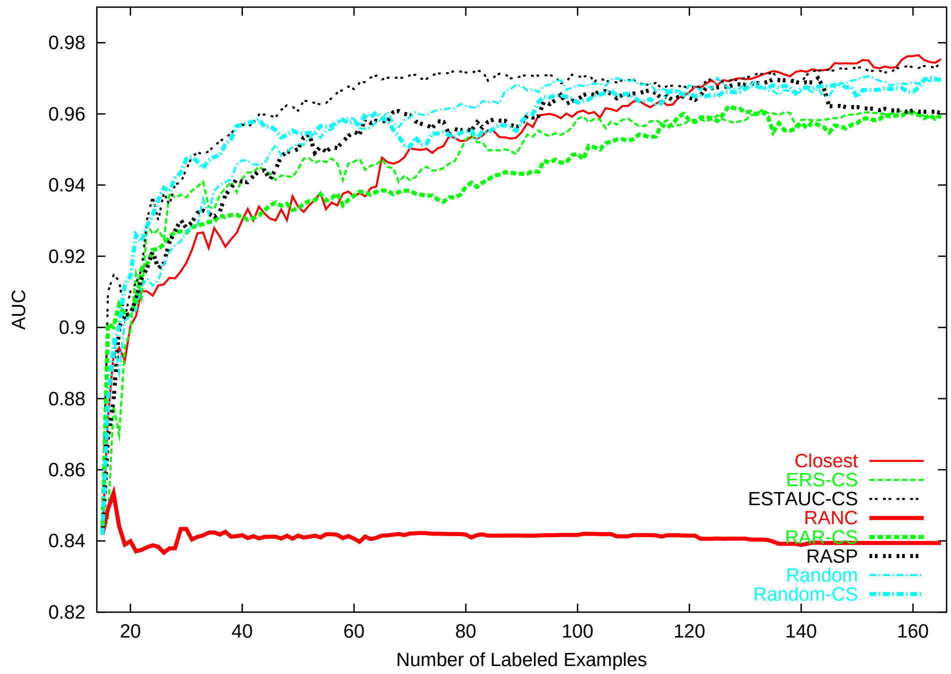


Figure 5.11: Learning curves for the Ionosphere data set using SMO as the classifier and starting with 15 initial labeled points.

Figure 5.12 shows the proximity of these competitors on the KR vs. KP data set. Tables 5.29, 5.30, and 5.31 show the summary statistics for these experiments. ESTAUC-CS is the top performer on all metrics, though by a fairly small margin.

The 'CS' heuristics did not show an improvement on the experiments with naïve Bayes. Having a better candidate set to choose from did not overcome the problems associated with using poor probability estimates. Tables 5.32, 5.33, and 5.34 show the summary statistics for naïve Bayes when starting with 15 labeled examples, and Tables 5.35, 5.36, and 5.37 show the same when starting with 100 examples.

Overall it appears that ESTAUC-CS is the best heuristic that we tested on these data sets when using SMO as the base learner. There is a cost associated with it due to the fact that it is slower than other methods like Closest, so one must take this into consideration when deciding whether to use it. We found that ESTAUC-CS takes approximately one minute

Table 5.27: Data utilization for the active learning heuristics on the Closest pre-filtering experiments using SMO when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS	Target AUC
B. Cancer	98 (0.72)	**66 (0.48)**	75 (0.55)	86 (0.63)	69 (0.50)	161 (1.18)	137 (1.00)	110 (0.80)	0.70
Colic	113 (0.88)	109 (0.84)	112 (0.87)	>165 (>1.28)	160 (1.24)	123 (0.95)	129 (1.00)	**49 (0.38)**	0.84
Credit A	>165 (>1.11)	>165 (>1.11)	**108 (0.72)**	>165 (>1.11)	>165 (>1.11)	>165 (>1.11)	149 (1.00)	>165	0.91
Credit G	**96 (0.64)**	106 (0.71)	106 (0.71)	127 (0.85)	115 (0.77)	119 (0.80)	149 (1.00)	104 (0.70)	0.72
Diabetes	**91 (0.66)**	134 (0.98)	111 (0.81)	133 (0.97)	138 (1.01)	112 (0.82)	137 (1.00)	97 (0.71)	0.81
Ionosphere	123 (1.38)	>165 (>1.13)	**61 (0.69)**	>165 (>1.13)	>165 (>1.13)	128 (1.44)	89 (1.00)	146 (1.64)	0.97
KR vs. KP	83 (0.56)	88 (0.60)	79 (0.54)	>165 (>2.12)	108 (0.73)	108 (0.73)	147 (1.00)	**78 (0.53)**	0.97
USPS	82 (0.55)	74 (0.50)	**44 (0.30)**	>165 (>1.11)	>165 (>1.11)	83 (0.56)	149 (1.00)	63 (0.42)	0.99
Vote	**51 (0.37)**	104 (0.76)	110 (0.80)	>165 (>1.20)	139 (1.01)	53 (0.39)	137 (1.00)	77 (0.56)	0.99
Median DUR	0.66	0.76	0.71	>1.11	1.01	0.82	1.00	0.70	
Wins	3	1	3	0	0	0	0	2	

to choose an example to label in the worst case, which seems like a long time. However, in real world active learning this may be eclipsed by the amount of time it takes an oracle to label the example. Consider for instance how long it would take for a human labeler to classify a web page. In a situation like this, using ESTAUC-CS does not seem unreasonable. Unfortunately, poor probability estimates prevented ESTAUC-CS and the others like it from doing well when naïve Bayes was the base learner, so Closest still remains the recommendation in those situations.

Table 5.28: Area under the learning curve above Random for the active learning heuristics on the Closest pre-filtering experiments using SMO when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	RANDOM	RANDOM-CS
B. CANCER	1.50	**2.37**	1.31	2.18	2.07	-0.40	0.00	0.75
COLIC	0.12	-0.41	0.25	-7.68	-2.28	0.26	0.00	**1.30**
CREDIT A	-0.48	0.10	**0.34**	-2.05	-1.04	-0.33	0.00	0.27
CREDIT G	**2.35**	1.56	1.70	0.74	1.09	1.19	0.00	2.04
DIABETES	**1.78**	0.34	0.75	0.07	-0.90	0.59	0.00	1.33
IONOSPHERE	-0.90	-1.26	**1.06**	-17.34	-1.92	-0.43	0.00	0.11
KR VS. KP	2.63	2.68	**3.29**	-18.71	-0.12	1.29	0.00	2.63
USPS	0.02	0.32	**0.82**	-8.20	-1.18	0.33	0.00	0.45
VOTE	1.11	-0.33	0.03	-0.34	-0.09	**1.21**	0.00	0.83
MEAN	0.903	0.597	1.061	-5.703	-0.486	0.412	0.00	1.079
WINS	2	1	4	0	0	1	0	1

Table 5.29: Ranked performance of the active learning heuristics on the Closest pre-filtering experiments using SMO when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

DATA SET	CLOSEST	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	RANDOM	RANDOM-CS
B. CANCER	1–7	2–7	1–5	1–4	1–5	4–8	6–8	4–8
COLIC	1–7	1–5	1–5	1–7	8	3–7	2–7	2–7
CREDIT A	4–7	1	2–6	8	2–7	2–5	3–7	2–7
CREDIT G	3–5	1–3	3–5	6–8	1–3	6–8	6–8	1–5
DIABETES	1–8	2–8	1–3	2–8	3–8	3–8	3–8	1–5
IONOSPHERE	1–3	3–8	1–3	4–6	6–8	1–4	5–8	4–7
KR VS. KP	1–4	2–4	1–2	8	5	6	7	2–4
USPS	4–5	1–3	1–3	8	6	1–3	7	4–5
VOTE	1	4–6	4–7	2–4	6–7	2–3	8	3–6
MEAN	1.9–5.2	1.9–5.0	1.8–4.3	4.4–4.8	4.2–6.3	3.1–5.8	5.2–7.6	2.6–6.0
WINS	2.15	2.03	2.57	0.53	0.65	0.58	0.00	0.48

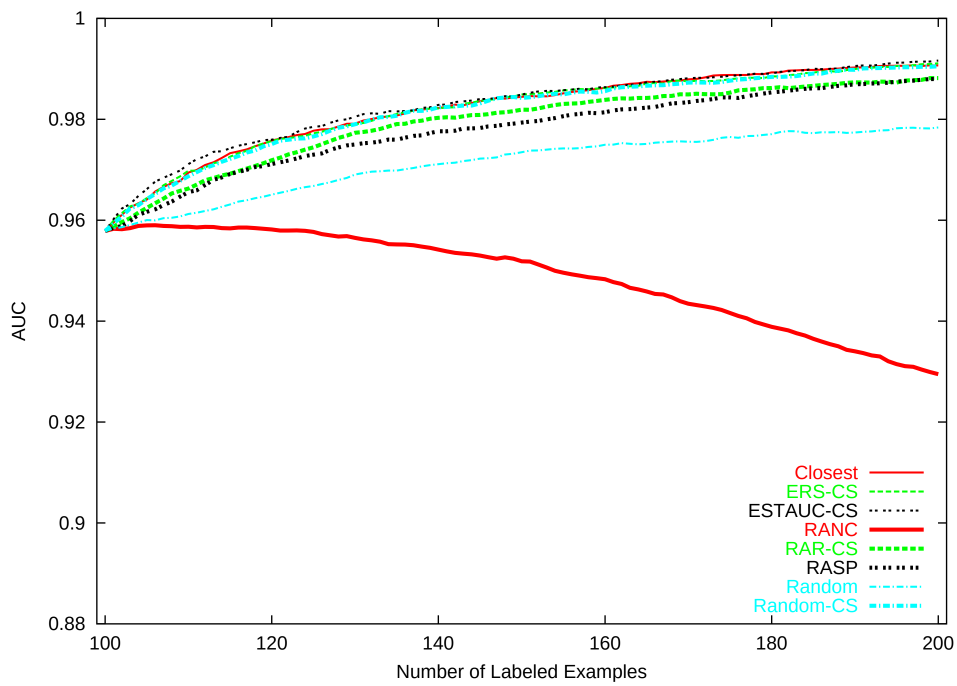


Figure 5.12: Learning curves for the KR vs. KP data set using SMO as the classifier and starting with 100 initial labeled points.

Table 5.30: Data utilization for the active learning heuristics on the Closest pre-filtering experiments using SMO when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS	Target AUC
B. Cancer	135	171	127	125	**117**	173	165	150	0.72
	(0.82)	(1.04)	(0.77)	(0.76)	**(0.71)**	(1.05)	(1.00)	(0.91)	
Colic	167	150	**144**	>200	>200	>200	183	178	0.88
	(0.91)	(0.82)	**(0.79)**	(>1.09)	(>1.09)	(>1.09)	(1.00)	(0.97)	
Credit A	172	**147**	154	>200	>200	164	181	>200	0.92
	(0.95)	**(0.81)**	(0.85)	(>1.10)	(>1.10)	(0.9)	(1.00)	(>1.10)	
Credit G	152	**145**	166	195	151	184	187	159	0.72
	(0.81)	**(0.78)**	(0.89)	(1.04)	(0.81)	(0.98)	(1.00)	(0.85)	
Diabetes	177	174	170	179	191	>200	186	**160**	0.82
	(0.95)	(0.94)	(0.91)	(0.96)	(1.03)	(>1.08)	(1.00)	**(0.86)**	
Ionosphere	114	148	**111**	123	>200	113	165	146	0.97
	(0.69)	(0.90)	**(0.67)**	(0.75)	(>1.21)	(0.68)	(1.00)	(0.88)	
KR vs. KP	125	127	**124**	>200	132	143	193	128	0.98
	(0.65)	(0.66)	**(0.64)**	(>1.04)	(0.68)	(0.74)	(1.00)	(0.66)	
USPS	105	**102**	**102**	>200	112	109	133	104	0.99
	(0.79)	**(0.77)**	**(0.77)**	(>1.50)	(0.84)	(0.82)	(1.00)	(0.78)	
Vote	**112**	133	123	**112**	133	114	184	121	0.99
	(0.61)	(0.72)	(0.67)	**(0.61)**	(0.72)	(0.62)	(1.00)	(0.66)	
Median DUR	0.81	0.81	0.77	1.04	0.84	0.90	1.00	0.86	
Wins	1	3	4	1	1	0	0	1	

Table 5.31: Area under the learning curve above Random for the active learning heuristics on the Closest pre-filtering experiments using SMO when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS
B. Cancer	0.52	0.41	0.62	**0.79**	0.65	0.09	0.00	0.23
Colic	0.10	0.19	**0.20**	0.00	-0.50	-0.09	0.00	-0.01
Credit A	-0.01	**0.23**	0.10	-0.15	0.05	0.12	0.00	0.02
Credit G	0.47	0.94	0.39	-0.10	**0.95**	-0.02	0.00	0.63
Diabetes	0.19	0.07	**0.37**	0.07	-0.04	0.02	0.00	0.26
Ionosphere	**0.42**	0.13	0.36	0.17	-0.04	0.34	0.00	0.14
KR vs. KP	1.08	1.06	**1.14**	-2.27	0.80	0.66	0.00	1.03
USPS	0.18	0.22	**0.24**	-1.32	0.10	0.22	0.00	0.18
Vote	**0.48**	0.24	0.21	0.35	0.14	0.41	0.00	0.29
Mean	0.381	0.388	0.403	-0.273	0.234	0.194	0.00	0.308
Wins	2	1	4	1	1	0	0	0

Table 5.32: Ranked performance of the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 15 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS
B. Cancer	2–8	1–7	3–8	1–7	5–8	1–7	1–7	1–5
Colic	1–3	7–8	5–6	1–4	5–6	1–3	7–8	3–4
Credit A	1–3	6–8	5–8	5–6	6–8	1–3	4	1–3
Credit G	3–6	7–8	4–6	7–8	4–6	1	2–3	2–4
Diabetes	2–4	6–7	5–7	5–6	8	2–4	1	2–4
Ionosphere	1–2	3–4	5–7	7–8	5–7	1–2	5–8	3–4
KR vs. KP	1–2	6	5	4	8	3	7	1–2
USPS	1–2	1–3	2–3	6–7	6–7	4–5	8	4–5
Vote	1–2	7	4–5	3	6	1–2	8	4–5
Mean	1.4–3.6	4.9–5.7	4.2–6.1	3.9–5.9	5.9–7.1	1.7–3.3	4.8–6.0	2.3–4.0
Wins	2.88	0.52	0.00	0.43	0.00	2.89	1.18	1.11

Table 5.33: Data utilization for the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 15 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS	Target AUC
B. Cancer	157	127	>165	121	>165	121	149	**109**	0.70
	(1.05)	(0.85)	(>1.11)	(0.81)	(>1.11)	(0.81)	(1.00)	**(0.73)**	
Colic	29	70	32	31	43	**26**	125	30	0.85
	(0.23)	(0.56)	(0.26)	(0.25)	(0.34)	**(0.20)**	(1.00)	(0.24)	
Credit A	78	165	>165	>165	>165	77	139	**73**	0.92
	(0.56)	(1.19)	(>1.19)	(>1.19)	(>1.19)	(0.55)	(1.00)	**(0.53)**	
Credit G	>165	>165	>165	>165	>165	**132**	151	165	0.75
	(>1.09)	(>1.09)	(>1.09)	(>1.09)	(>1.09)	**(0.87)**	(1.00)	(1.09)	
Diabetes	>165	>165	>165	>165	>165	>165	**145**	>165	0.84
	(>1.14)	(>1.14)	(>1.14)	(>1.14)	(>1.14)	(>1.14)	**(1.00)**	(>1.14)	
Ionosphere	54	**31**	87	>165	>165	47	151	32	0.97
	(0.36)	**(0.21)**	(0.58)	(>1.09)	(>1.09)	(0.31)	(1.00)	(0.21)	
KR vs. KP	**51**	134	93	72	>165	57	151	**51**	0.91
	(0.34)	(0.89)	(0.62)	(0.48)	(>1.09)	(0.38)	(1.00)	**(0.34)**	
USPS	**17**	**17**	**17**	18	18	18	26	**17**	0.98
	(0.65)	**(0.65)**	**(0.65)**	(0.69)	(0.69)	(0.69)	(1.00)	**(0.65)**	
Vote	**18**	30	19	19	36	**18**	134	20	0.98
	(0.13)	(0.22)	(0.14)	(0.14)	(0.27)	**(0.13)**	(1.00)	(0.15)	
Median DUR	0.56	0.85	0.65	0.81	>1.09	0.55	1.00	0.53	
Wins	3	2	1	0	0	3	1	4	

Table 5.34: Area under the learning curve above Random for the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 15 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS
B. Cancer	-0.50	**0.50**	-1.04	-0.11	-1.81	0.04	0.00	0.16
Colic	3.27	0.09	1.37	3.01	1.01	**3.32**	0.00	2.42
Credit A	0.79	-0.96	-0.72	-0.44	-1.51	0.58	0.00	**0.85**
Credit G	-0.69	-3.08	-0.85	-2.23	-1.41	**0.94**	0.00	-0.17
Diabetes	-1.04	-4.11	-3.74	-4.05	-6.44	-1.18	**0.00**	-1.33
Ionosphere	1.05	0.96	-0.08	-0.36	0.13	**1.22**	0.00	0.92
KR vs. KP	**10.06**	1.35	3.91	4.93	-1.29	8.40	0.00	9.98
USPS	**2.59**	2.56	2.56	2.23	2.11	2.37	0.00	2.46
Vote	**2.48**	0.41	1.84	2.06	1.12	2.40	0.00	1.72
Mean	2.001	-0.253	0.361	0.560	-0.899	2.010	0.000	1.890
Wins	3	1	0	0	0	3	1	1

Table 5.35: Ranked performance of the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 100 labeled points. The mean rank range and total win credit summarize across all data sets.

Data set	Closest	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	Random	Random-CS
B. Cancer	3–8	4–8	1–2	4–8	1–6	3–8	2–4	2–8
Colic	1	7–8	2–6	2–6	3–6	2–6	7–8	2–5
Credit A	1–2	7–8	3–4	7–8	5–6	3–4	5–6	1–2
Credit G	1–4	3–5	1–3	7–8	5–8	4–7	5–7	1–3
Diabetes	4–7	3–7	3–7	8	3–7	1–2	1–2	3–6
Ionosphere	1–3	3–5	7–8	1–5	6–7	1–3	6–8	3–5
KR vs. KP	1–2	6	5	7	4	3	8	1–2
USPS	1–2	3	1–2	7	5–6	5–6	8	4
Vote	1	7	2–3	4–5	4–5	2–3	8	6
Mean	1.6–3.3	4.8–6.3	2.8–4.4	5.3–6.9	4–6.1	2.7–4.7	5.6–6.6	2.6–4.6
Wins	4.15	0.00	1.71	0.20	0.17	0.90	0.50	1.38

Table 5.36: Data utilization for the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 100 labeled points. The data utilization ratio (DUR) appears in parentheses below the minimum number of training examples needed to achieve the target AUC. Winners for each data set are displayed in boldface. The median DUR and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	RANDOM	RANDOM-CS	TARGET AUC
B. CANCER	>200	>200	137	>200	**107**	114	173	172	0.71
	(>1.16)	(>1.16)	(0.79)	(>1.16)	**(0.62)**	(0.66)	(1.00)	(0.99)	
COLIC	**104**	112	**104**	129	111	109	109	**104**	0.85
	(0.95)	(1.03)	**(0.95)**	(1.18)	(1.02)	(1.00)	(1.00)	**(0.95)**	
CREDIT A	**118**	>200	124	>200	147	136	160	121	0.92
	(0.74)	(>1.25)	(0.78)	(>1.25)	(0.92)	(0.85)	(1.00)	(0.76)	
CREDIT G	169	>200	164	>200	>200	179	188	**160**	0.76
	(0.90)	(>1.06)	(0.87)	(>1.06)	(>1.06)	(0.95)	(1.00)	**(0.85)**	
DIABETES	>200	>200	>200	>200	>200	>200	**155**	>200	0.84
	(>1.29)	(>1.29)	(>1.29)	(>1.29)	(>1.29)	(>1.29)	**(1.00)**	(>1.29)	
IONOSPHERE	117	114	>200	119	137	**113**	164	116	0.97
	(0.71)	(0.70)	(>1.22)	(0.73)	(0.84)	**(0.69)**	(1.00)	(0.71)	
KR VS. KP	114	117	117	132	120	116	189	**113**	0.92
	(0.60)	(0.62)	(0.62)	(0.70)	(0.63)	(0.61)	(1.00)	**(0.60)**	
USPS	**101**	**101**	**101**	102	102	102	120	102	0.98
	(0.84)	**(0.84)**	**(0.84)**	(0.85)	(0.85)	(0.85)	(1.00)	(0.85)	
VOTE	104	105	**103**	112	104	105	187	104	0.97
	(0.56)	(0.56)	**(0.55)**	(0.60)	(0.56)	(0.56)	(1.00)	(0.56)	
MEDIAN DUR	0.84	1.03	0.84	>1.06	0.85	0.85	1.00	0.85	
WINS	3	1	3	0	1	1	1	3	

Table 5.37: Area under the learning curve above Random for the active learning heuristics on the Closest pre-filtering experiments using naïve Bayes when starting with 100 labeled points. Winners for each data set are displayed in boldface. The mean area above Random and total wins summarize across all data sets.

DATA SET	CLOSEST	ERS-CS	ESTAUC-CS	RANC	RAR-CS	RASP	RANDOM	RANDOM-CS
B. CANCER	-0.39	-0.34	**0.28**	-0.42	0.18	-0.34	0.00	-0.14
COLIC	**1.02**	0.02	0.47	0.39	0.36	0.47	0.00	0.57
CREDIT A	**0.51**	-0.25	0.25	-0.30	0.07	0.21	0.00	**0.51**
CREDIT G	0.51	0.25	0.64	-0.35	-0.21	0.21	0.00	**0.66**
DIABETES	-0.68	-0.58	-0.67	-1.15	-0.52	-0.15	**0.00**	-0.46
IONOSPHERE	**0.41**	0.29	-0.06	0.33	0.03	0.39	0.00	0.28
KR VS. KP	4.68	2.35	2.51	1.40	2.74	3.96	0.00	**4.69**
USPS	**1.33**	1.28	1.31	0.80	1.10	1.10	0.00	1.14
VOTE	**0.86**	0.13	0.71	0.62	0.58	0.72	0.00	0.39
MEAN	0.917	0.350	0.604	0.147	0.481	0.730	0.000	0.849
WINS	5	0	1	0	0	0	1	3

5.2.7 Discussion

Our initial results suggested that Closest was the preferred heuristic as it was almost always among the top performers when using both naïve Bayes and SMO as the base learners. In addition, it is a simple heuristic that runs fast. We also found that Closest was even more dominant when the learner already has a good hypothesis. RASP also did consistently well, and was almost always competitive with Closest. This suggests that, though choosing examples that the current learner is uncertain about gives good results, choosing the most uncertain example as Closest does may not always give the best gains. The ESTAUC and ERS heuristics also performed fairly well on the initial experiments when SMO was the base learner, though not as well as Closest or RASP. They did not do well when naïve Bayes was the base learner because it generates notoriously bad probability estimates. We also noted that ESTAUC, ERS and RAR were at a disadvantage on the initial experiments because they were only able to evaluate a fraction of the unlabeled pool at each query iteration.

The experiments on the synthetic data showed that ESTAUC and RAR can do very well when provided with high quality probability estimates. However, obtaining good estimates from the small training sets used in active learning can be difficult.

On our final set of experiments we utilized the lessons learned on the initial experiments by pre-filtering the candidate sets for ESTAUC, RAR and ERS with a closest sampling filter. This gave these three heuristics a richer pool from which to choose, and all three showed improved performance. In fact, ESTAUC-CS out performed all of the heuristics, including Closest, when SMO was the base learner. This reinforces our belief that choosing examples that the learner is uncertain about gives the best results, but choosing the most uncertain example is not optimal. When naïve Bayes was used as the base learner on the filtered experiments, Closest remained the top performer since ESTAUC, ERS and RAR were still handicapped by poor probability estimates.

We also compared the performance of SMO and naïve Bayes as base learners. We found that naïve Bayes is superior when the labeled set is small, but SMO typically does better when the labeled set is sufficiently large. The point at which SMO passes naïve Bayes varies from data set to data set, but in our experiments this usually occurred when the labeled set had between 100 and 200 examples. However, we found that each of the learners was predisposed to doing better on certain data sets, so the choice of which base learner to use may depend on the particular learning problem.

Overall, we suggest using ESTAUC with a closest sampling filter on the candidate set and an SVM as the base learner for most applications. An SVM should be used as the base learner if enough queries will be made to result in a sufficiently large labeled set (100-200 examples) before the generated hypothesis will be used. For applications where the labeled set will remain small, using the Closest or RASP heuristics with naïve Bayes as the base learner is to be preferred. We also suggest using Closest or RASP in situations where execution speed is a chief concern.

Chapter 6

Conclusions and Future Work

Area under the ROC curve is an important measure of learning performance. We studied how active learning affects AUC. We studied two algorithms from the literature (Closest and ERS), and introduced four new algorithms with the express intent of maximizing AUC (RANC, RASP, ESTAUC, and RAR). We evaluated all these algorithms using both SMO and naïve Bayes as the base classifier. They were evaluated on eight learning problems from the UCI Machine Learning Repository and the USPS handwritten digit recognition problem.

Our conclusions agree with much of the previous work in the field [1,2,15,27,35,36,38] in that choosing an example from U to label that the current learner is uncertain about works well. However, we found that choosing the most uncertain example, as many strategies do (e.g. Closest, QBC, QBag, MQL, active Decorate), does not necessarily give the best gains.

We found ESTAUC out-performed all of the other heuristics when its candidate set was pre-filtered with closest sampling and SMO was used as the base learner. Closest and RASP produced strong results regardless of the base learner. The performance of ERS, ESTAUC, and RAR depended on good probability estimates. In addition, the results on the synthetic data suggest that ESTAUC, and RAR may perform exceptionally well if provided with high quality probability estimates. We also compared SMO to naïve Bayes in optimizing AUC and found the latter to be superior when the labeled set was small. However, SMO tended to do better with larger labeled sets.

Overall, we advocate using ESTAUC with a closest sampling filter and SMO except when the size of the labeled set remains small, or when execution speed is paramount. In these cases we suggest using Closest or RASP with naïve Bayes as the base learner.

Future work includes extending this work to multiclass problems (e.g. maximize the "Volume Under Surface" (VUS) [14]). Other future work is to apply our results to minimize the lower envelopes of cost curves [13], an alternative to ROC curves.

Bibliography

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)*, pages 1–10, 1998.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2(3):319–342, 1988.

[3] M. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proc. 23rd International Conf. on Machine Learning*, pages 65–72, 2006.

[4] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5(Mar):255–291, 2004.

[5] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

[6] J.. Bowring, J. Rehg, and M. Harrold. Active learning for automatic classification of software behavior. In *procedings of International Symposium on Software Testing and Analysis*, pages 195–205, 2004.

[7] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning*, pages 59–66, 2003.

[8] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the 17th Intl. Conf. on Machine Learning*, pages 111–118, 2000.

[9] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Neural Information Processing Systems*, pages 409–415, 2000.

[10] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[11] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems*, volume 16. Cambridge MA: MIT Press, 2004.

[12] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2006.

[13] C. Drummond and R. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, to appear.

[14] C. Ferri et al. Volume under the ROC surface for multi-class problems. In *ECAI 2003*, pages 108–120, 2003.

[15] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[16] Thore Graepel and Ralf Herbrich. The kernel gibbs sampler. In *NIPS*, pages 514–520, 2000.

[17] D. Hakkani-Tur, M. Rahim, and G. Riccardi. Unsupervised and active learning in automatic speech rcognition for call classification. In *procedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 429–432, 2004.

[18] J. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, pages 29–36, 1982.

[19] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Batch mode active learning and its application to medical image classification. In *Proc. 23rd International Conf. on Machine Learning*, pages 417–424, 2006.

[20] V. S. Iyengar, C. Apte, and T. Zhang. Active learning using adaptive resampling. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 91–98, 2000.

[21] K. J. Lang and E. B. Baum. Query learning can work poorly when a human oracle is used. In *International Joint Conference on Neural Networks*, 1992.

[22] E. L. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, 1975.

[23] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In W. Bruce Croft and Cornelis J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, IE, 1994. Springer Verlag, Heidelberg, DE.

[24] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54:125–152, 2004.

[25] D. Lizotte, O. Madani, and R. Greiner. Budgeted learning of naive-bayes classifiers. In *Proceedings of 19th Conference on Uncertainty in Artificial Intelligence (UAI-03)*, Acapulco, Mexico, 2003.

[26] Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(Apr):589–613, 2005.

[27] P. Melville and R. Mooney. Diverse ensembles for active learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*, pages 584–591, Banff, Canada, 2004.

[28] Tom Mitchell. *Machine Learning*. McGraw-Hill, Boston, 1997.

[29] P. Mitra, C. A. Murthy, and S. K. Pal. A probabilistic active support vector learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):413–418, 2004.

[30] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *Proceedings of the 21st International Conference on Machine Learning*, pages 623–630, 2004.

[31] T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 330–337, 2005.

[32] J. M. Park. Convergence and application of online active sampling using orthogonal pillar vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1197–1207, 2004.

[33] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pages 441–448, 2001.

[34] B. Schoelkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

[35] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the 17th Intl. Conf. on Machine Learning*, pages 839–846, 2000.

[36] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294, 1992.

[37] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proc. 16th International Conf. on Machine Learning*, pages 406–414. Morgan Kaufmann, San Francisco, CA, 1999.

[38] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2(Nov):45–66, 2001.

[39] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.

[40] G. Xiao, F. Southey, R. C. Holte, and D. Wilkinson. Software testing by active learning for commercial games. In *AAAI-05: Twentieth National Conference on Artificial Intelligence*, 2005.

[41] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proc. 18th International Conf. on Machine Learning*, pages 609–616. Morgan Kaufmann, San Francisco, CA, 2001.

Appendix A

Additional Learning Curves

A.1 Initial Experiments

A.1.1 Starting with 15 Labeled Examples

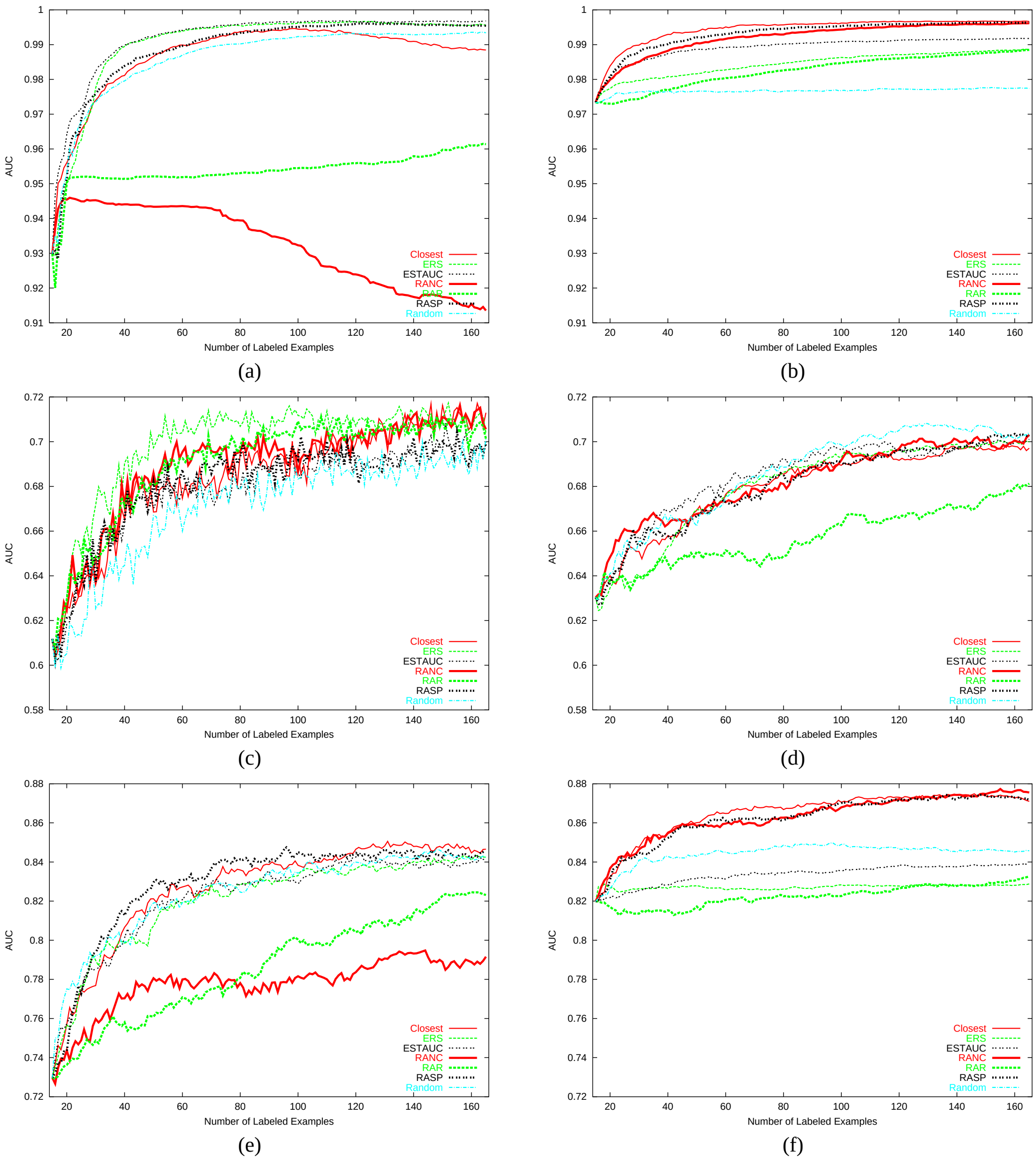


Figure A.1: Learning curves for the initial experiments with $L_0 = 15$. (a) USPS with SMO (b) USPS with naïve Bayes (c) Breast Cancer with SMO (d) Breast Cancer with naïve Bayes (e) Colic with SMO (f) Colic with naïve Bayes

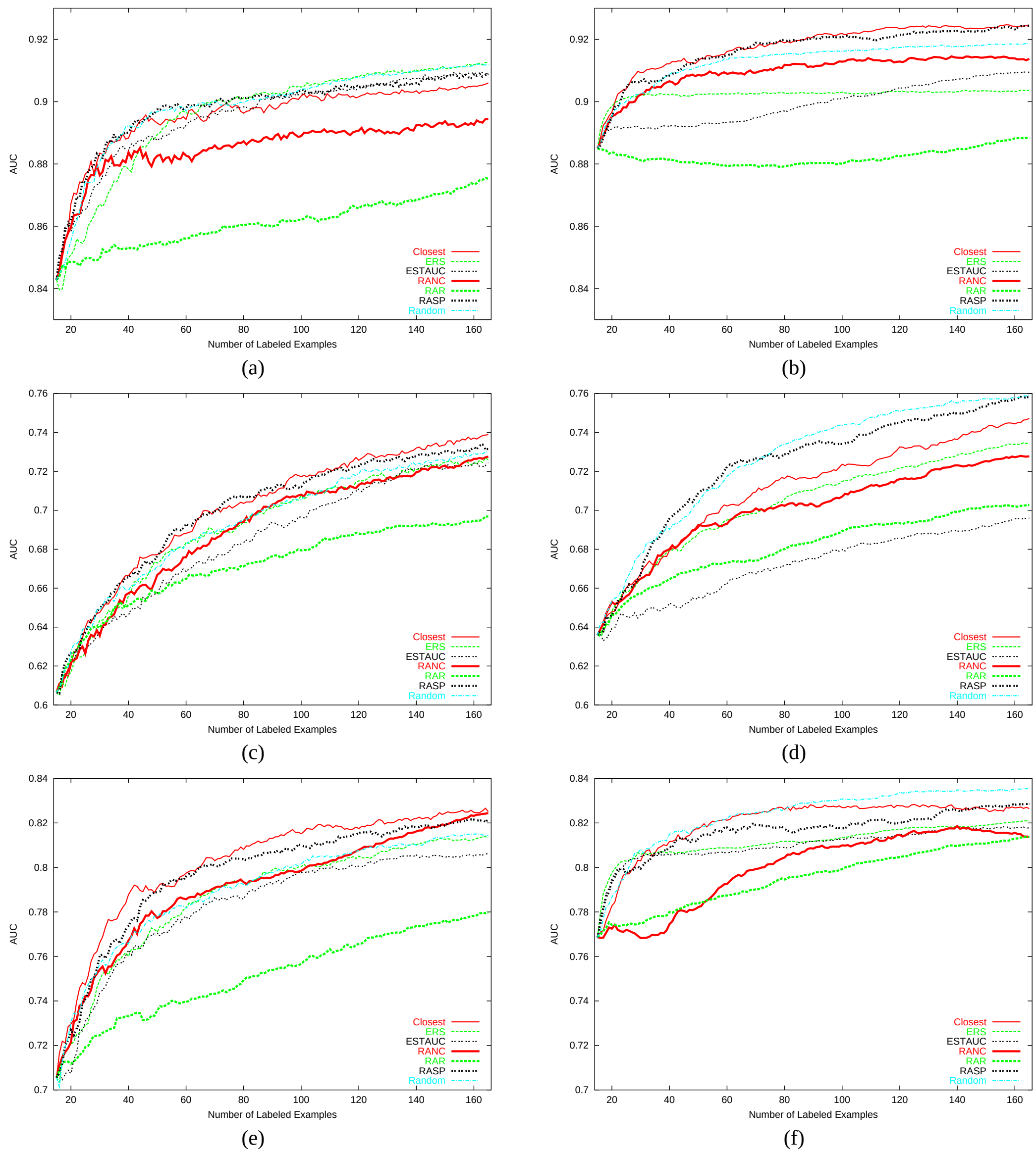


Figure A.2: Learning curves for the initial experiments with $L_0 = 15$. (a) Credit A with SMO (b) Credit A with naïve Bayes (c) Credit G with SMO (d) Credit G with naïve Bayes (e) Diabetes with SMO (f) Diabetes with naïve Bayes

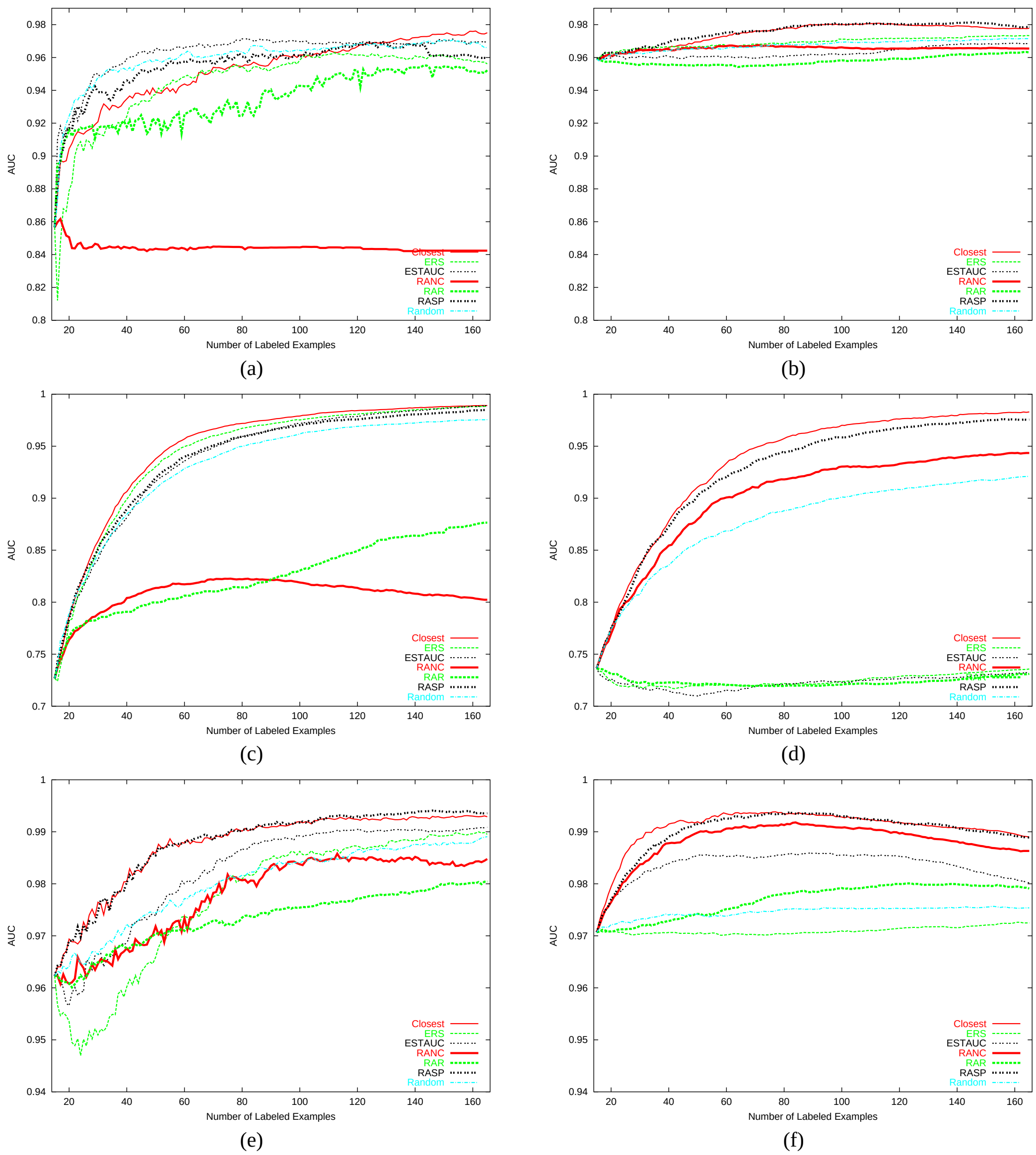


Figure A.3: Learning curves for the initial experiments with $L_0 = 15$. (a) Ionosphere with SMO (b) Ionosphere with naïve Bayes (c) KR vs. KP with SMO (d) KR vs. KP with naïve Bayes (e) Vote with SMO (f) Vote with naïve Bayes

A.1.2 Starting with 100 Labeled Examples

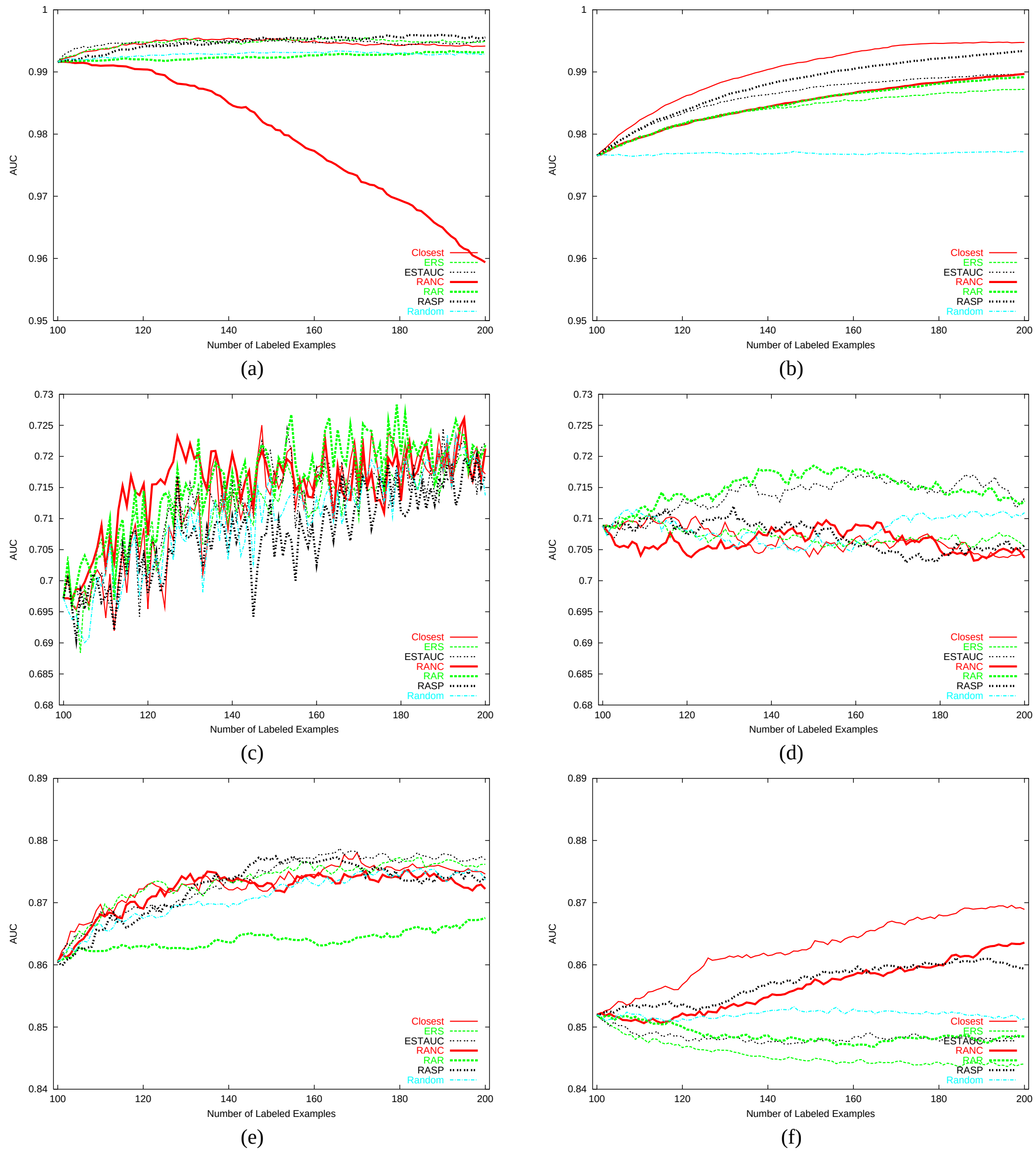
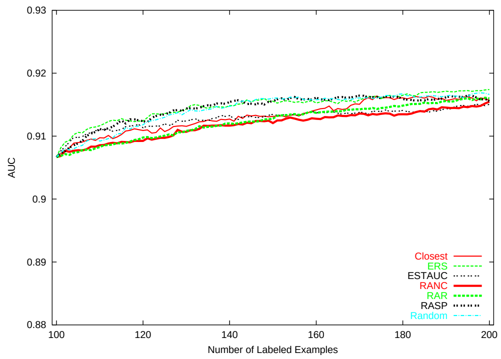
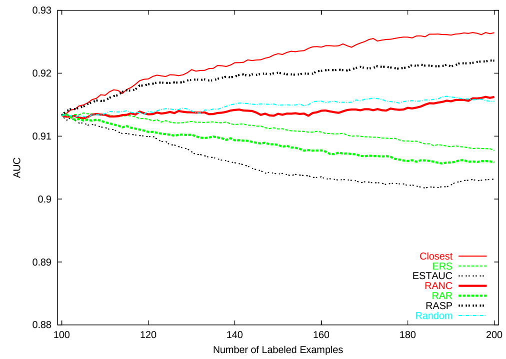


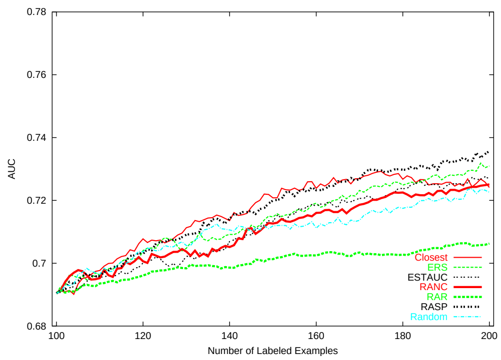
Figure A.4: Learning curves for the initial experiments with $L_0 = 100$. (a) USPS with SMO (b) USPS with naïve Bayes (c) Breast Cancer with SMO (d) Breast Cancer with naïve Bayes (e) Colic with SMO (f) Colic with naïve Bayes



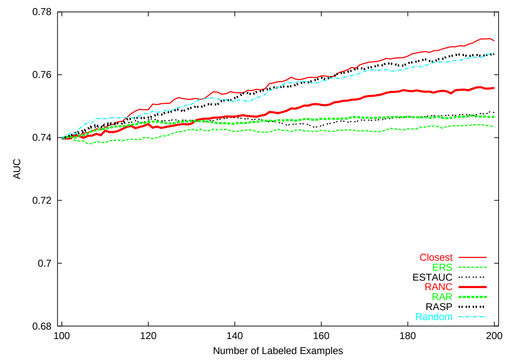
(a)



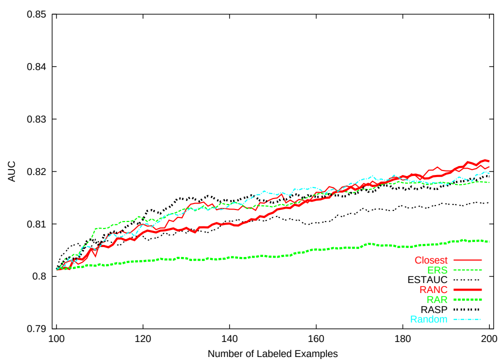
(b)



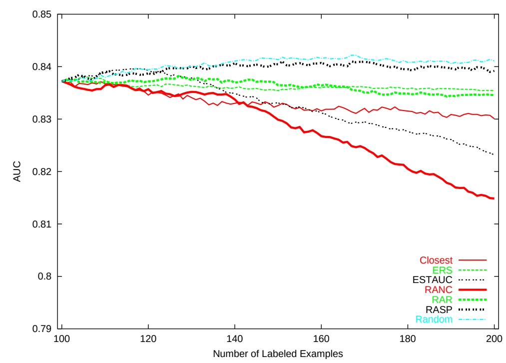
(c)



(d)



(e)



(f)

Figure A.5: Learning curves for the initial experiments with $L_0 = 100$. (a) Credit A with SMO (b) Credit A with naïve Bayes (c) Credit G with SMO (d) Credit G with naïve Bayes (e) Diabetes with SMO (f) Diabetes with naïve Bayes

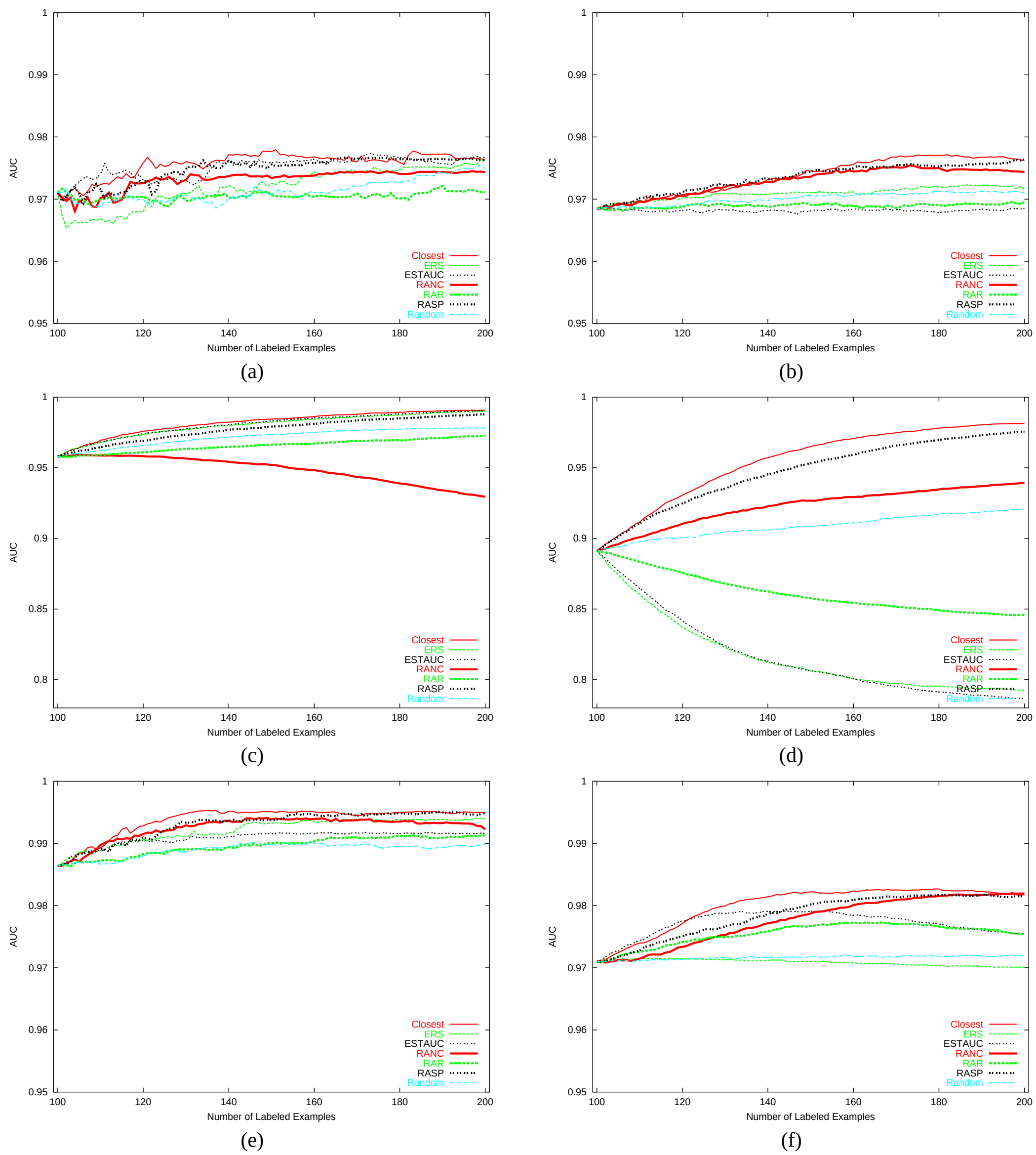


Figure A.6: Learning curves for the initial experiments with $L_0 = 100$. (a) Ionosphere with SMO (b) Ionosphere with naïve Bayes (c) KR vs. KP with SMO (d) KR vs. KP with naïve Bayes (e) Vote with SMO (f) Vote with naïve Bayes

A.2 Synthetic Data

A.2.1 Starting with 15 Labeled Examples

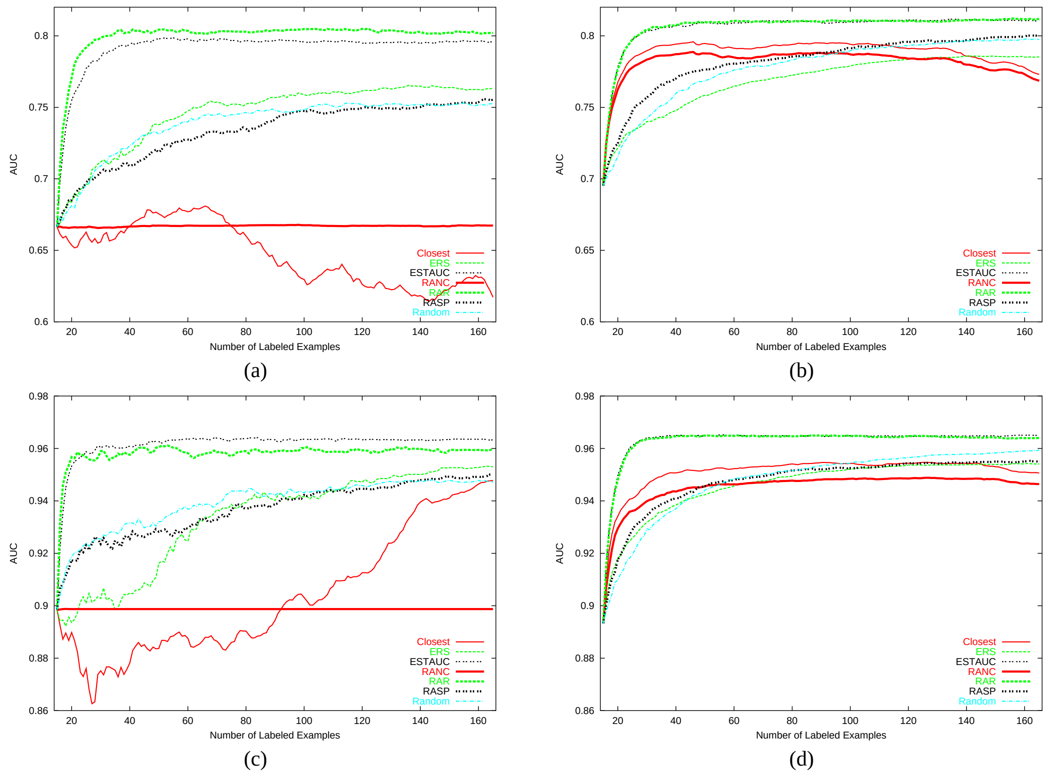


Figure A.7: Learning curves on synthetic data with $L_0 = 15$. (a) Gaussian 0,5 with SMO (b) Gaussian 0,5 with naïve Bayes (c) Gaussian 0,10 with SMO (d) Gaussian 0,10 with naïve Bayes

A.2.2 Starting with 100 Labeled Examples

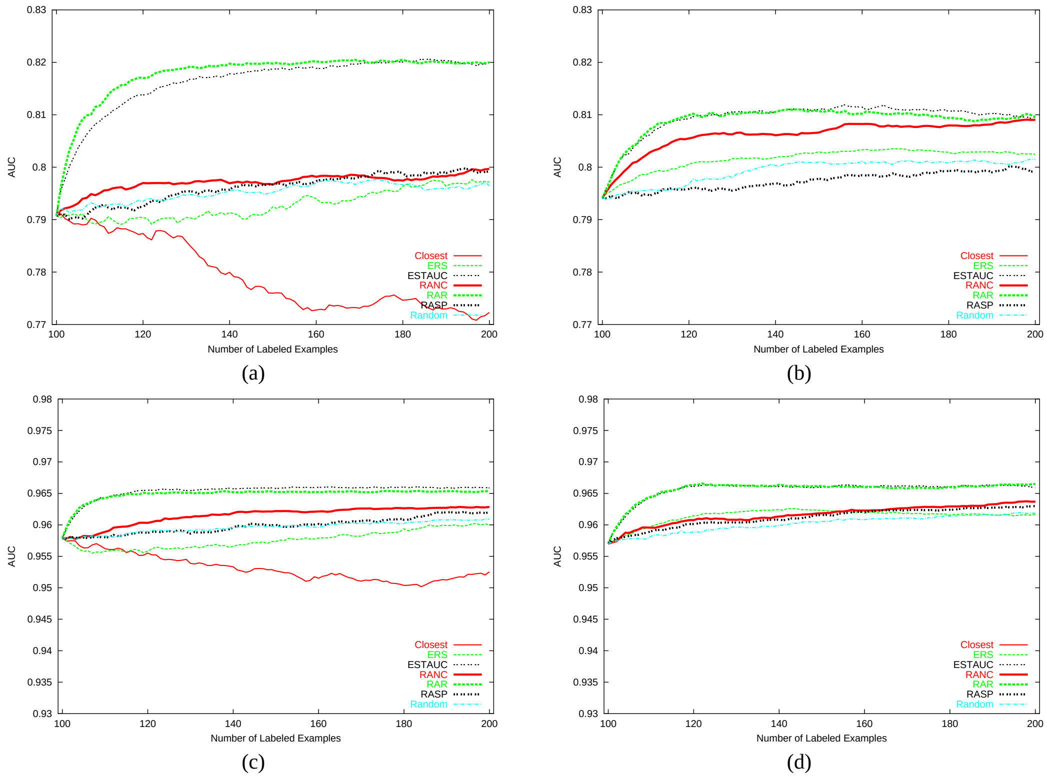


Figure A.8: Learning curves on synthetic data with $L_0 = 100$. (a) Gaussian 0,5 with SMO (b) Gaussian 0,5 with naïve Bayes (c) Gaussian 0,10 with SMO (d) Gaussian 0,10 with naïve Bayes

A.3 Pre-filtering with Closest Sampling

A.3.1 Starting with 15 Labeled Examples

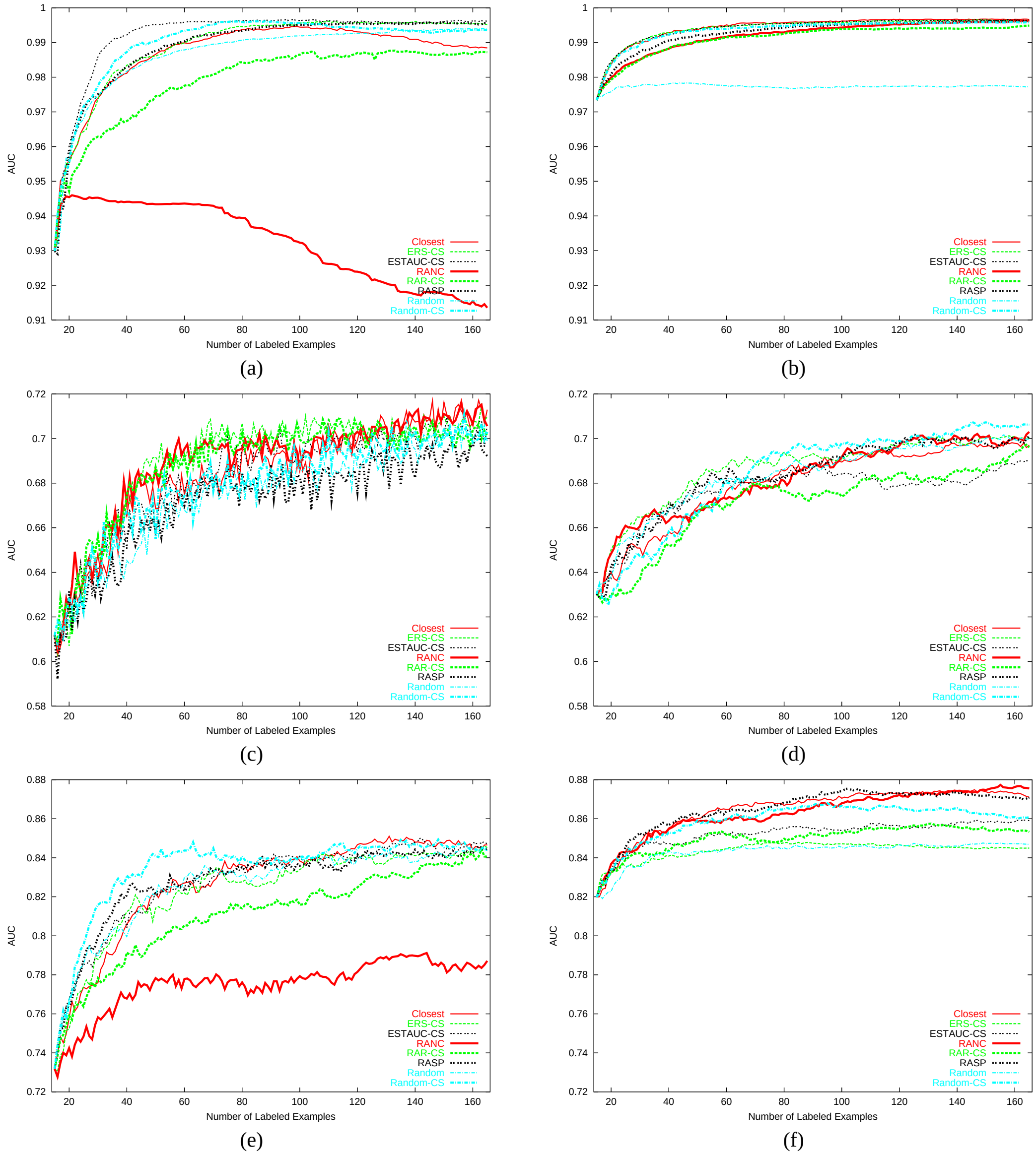


Figure A.9: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 15$. (a) USPS with SMO (b) USPS with naïve Bayes (c) Breast Cancer with SMO (d) Breast Cancer with naïve Bayes (e) Colic with SMO (f) Colic with naïve Bayes

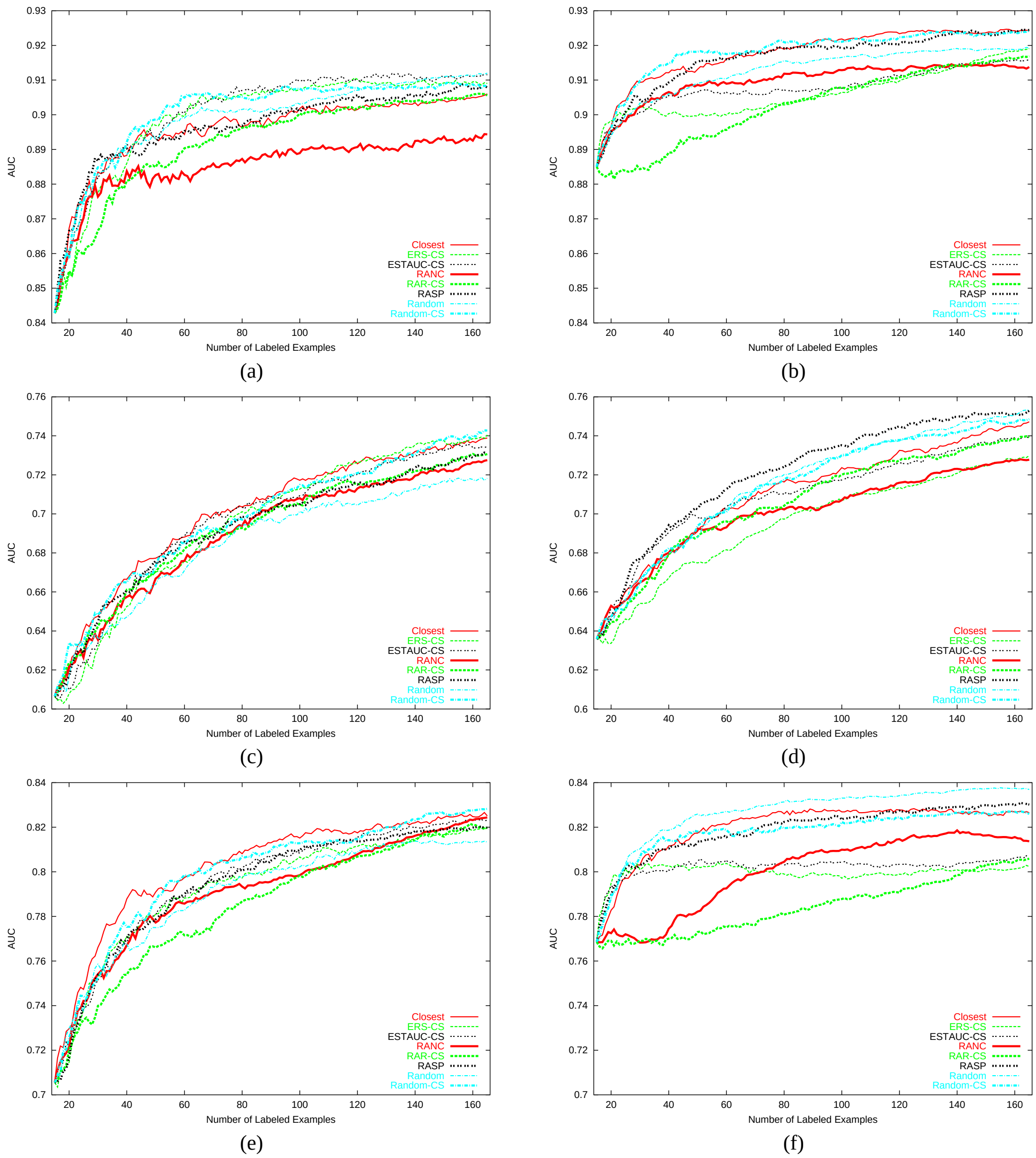


Figure A.10: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 15$. (a) Credit A with SMO (b) Credit A with naïve Bayes (c) Credit G with SMO (d) Credit G with naïve Bayes (e) Diabetes with SMO (f) Diabetes with naïve Bayes

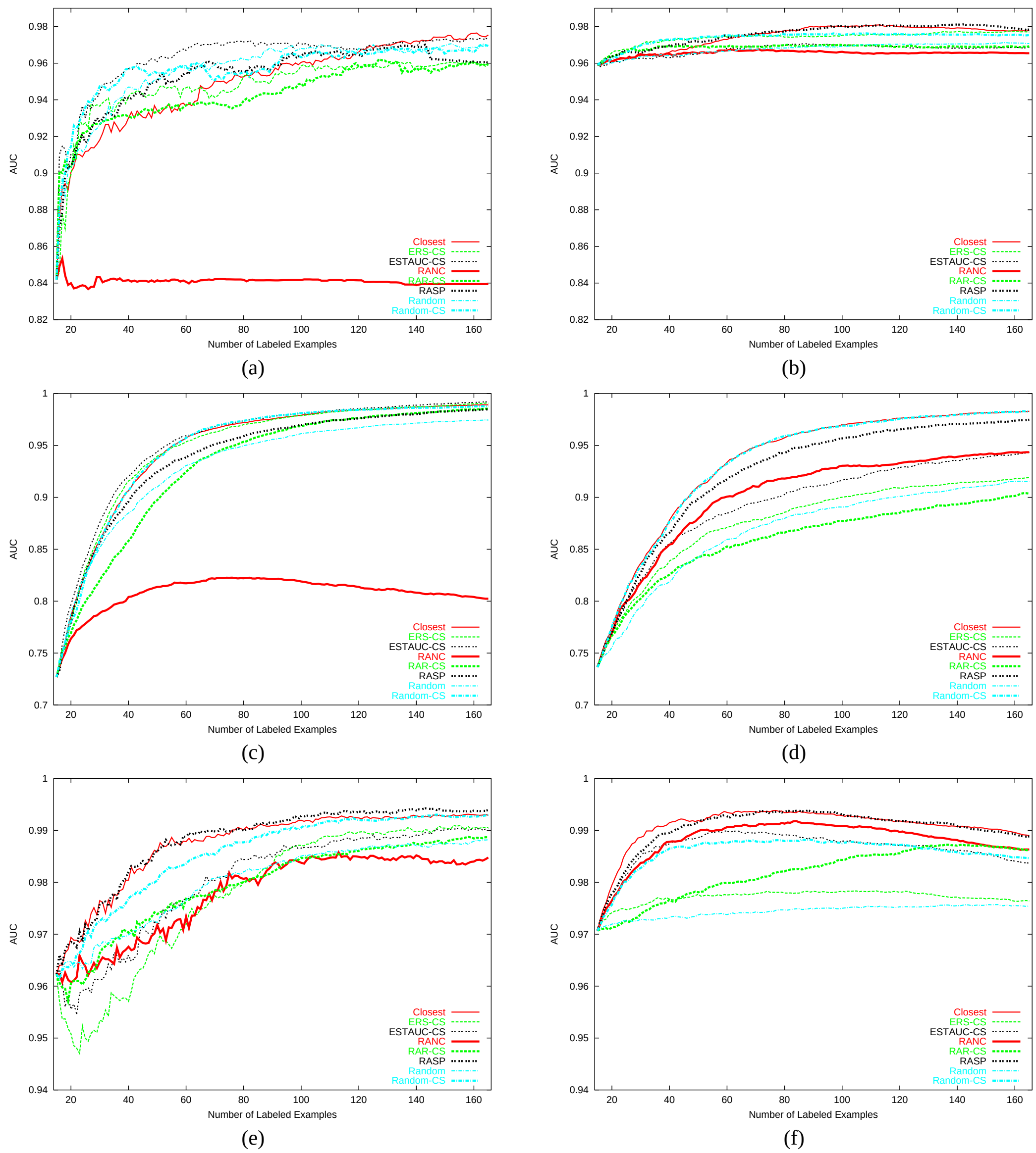


Figure A.11: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 15$. (a) Ionosphere with SMO (b) Ionosphere with naïve Bayes (c) KR vs. KP with SMO (d) KR vs. KP with naïve Bayes (e) Vote with SMO (f) Vote with naïve Bayes

A.3.2 Starting with 100 Labeled Examples

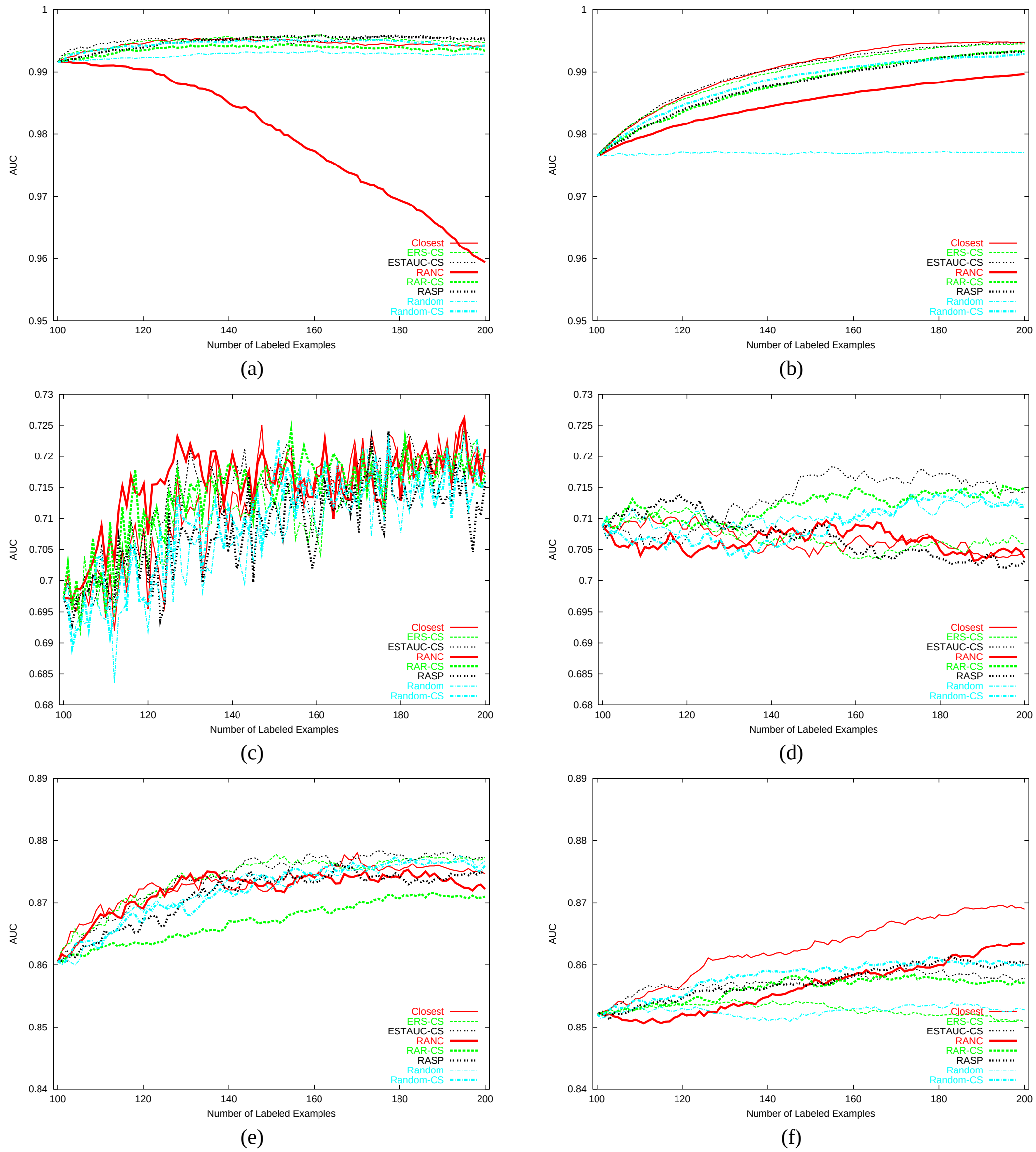


Figure A.12: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 100$. (a) USPS with SMO (b) USPS with naïve Bayes (c) Breast Cancer with SMO (d) Breast Cancer with naïve Bayes (e) Colic with SMO (f) Colic with naïve Bayes

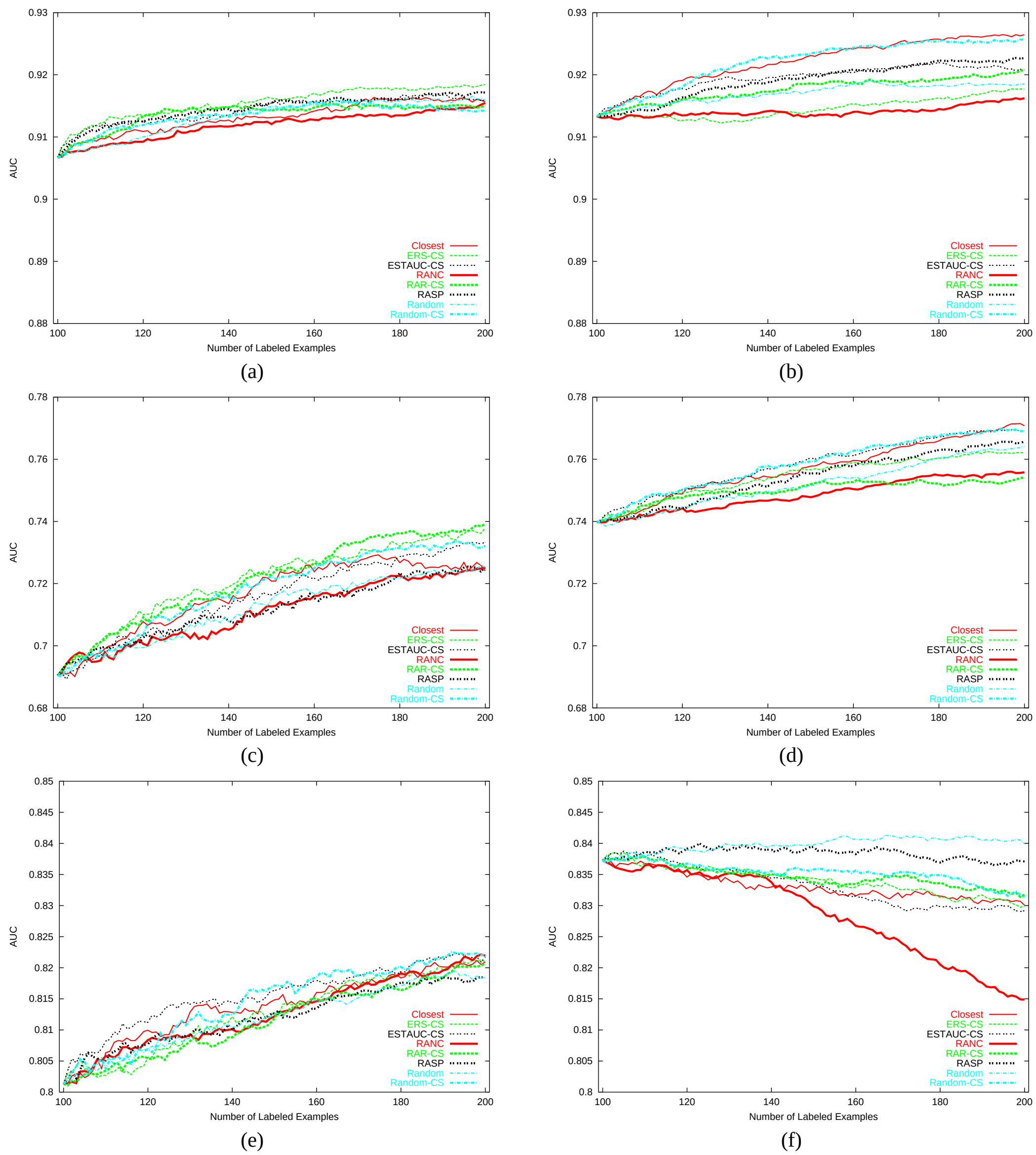


Figure A.13: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 100$. (a) Credit A with SMO (b) Credit A with naïve Bayes (c) Credit G with SMO (d) Credit G with naïve Bayes (e) Diabetes with SMO (f) Diabetes with naïve Bayes

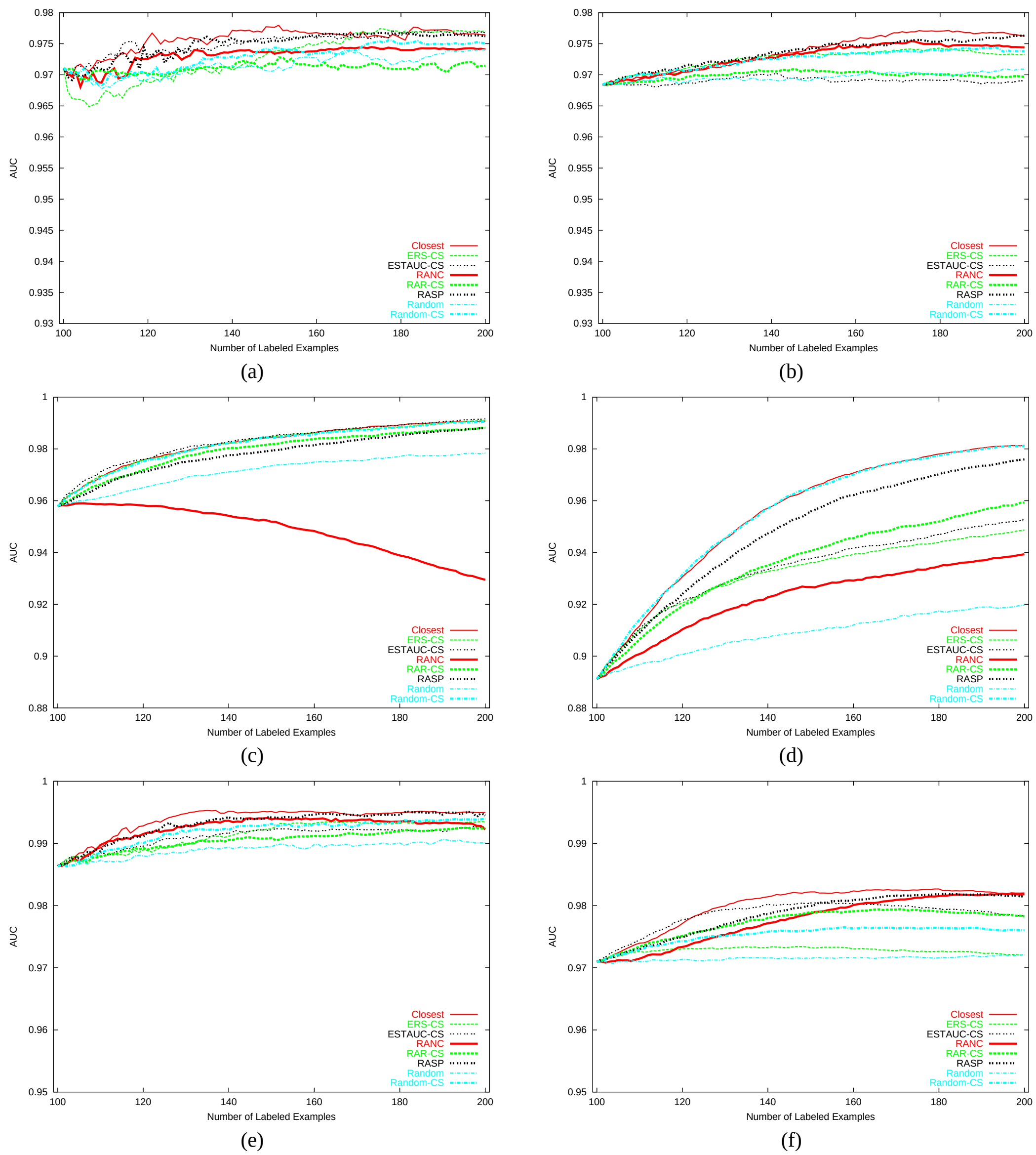


Figure A.14: Learning curves with candidate sets selected by Closest Sampling and $L_0 = 100$. (a) Ionosphere with SMO (b) Ionosphere with naïve Bayes (c) KR vs. KP with SMO (d) KR vs. KP with naïve Bayes (e) Vote with SMO (f) Vote with naïve Bayes