University of Pennsylvania
Dept. of Computer and Information Science
Philadelphia, Pennsylvania, USA

In Partial Fulfillment of the WPEII Requirement

Active Learning for Classification of Medical Signals

Author: Alexander Roederer

Committee Chair: Ben Taskar
Committee Member: Oleg Sokolsky
Committee Member: Camillo J. Taylor

November 2012

Contents

Abstract

Hospitals are increasingly capturing and storing medical signals produced by patients. Machine learning classifiers could be used on these signals to improve patient care, but while unlabeled data is often plentiful, obtaining labels for such a large quantity of data is prohibitively expensive. Active learning attempts to achieve good classification performance with only a minimal number of labelings by choosing which data instances to label.

In this work, we survey the state-of-the-art in applying active learning techniques to medical signals. In particular, we examine five active learning query strategies: (1) expected error reduction sampling, (2) variance reduction sampling, (3) density-weighted sampling, (4) maximal uncertainty sampling, and (5) query-by-committee. Though they are applied to various medical signals (including electrocardiogram, electroencephalogram, and medical images), each provides insight into the viability of active learning in the domain. Taken together they provide a comprehensive overview of current work in active learning. Finally, we compare the performance of these techniques and discuss how active learning could improve real-time machine learning in the medical domain.

1 Introduction

In recent years, a dramatic increase in computing power coupled with reduced costs for digital storage has made it feasible to automatically capture and store large quantities of digital data sourced from a variety of domains [21]. Once this data is captured, it can be used for a variety of tasks, such as billing, quality assurance, and retrospective analysis.

However, the massive quantity of data collected often makes manual analysis of this data impossible. To understand and utilize the data effectively, automated analysis becomes vital. Machine learning can provide algorithms to take large quantities of data and extract useful patterns or predictions from it.

Supervised machine learning algorithms often need to be trained on a large number of labeled data instances to perform well on complex data sets [39]. In many domains, these labeled instances are easy to acquire. For example, emails are routinely labeled as spam by email users when manually removing those emails from their inbox, and these labels can be used to train a classifier that predicts whether an incoming message is a spam message before the user opens it [36, 1]. Likewise, Netflix allows users to view hundreds

of movies, which users can then classify by assigning a number of "stars." Netflix then uses these ratings to predict how much individual users would like movies they have not viewed [29].

In many domains, unlabeled data is abundant, but obtaining labels for those data instances is difficult, time-consuming, or expensive. This review focuses on just such a domain: that of medical data, and in particular medical data that is measured with very high frequency (many times per second). Hospitals are increasingly capturing large amounts of information on their patients, including high-frequency vital sign signals and medical images. The hope is that this data can be used to improve patient care, reduce healthcare costs, and save lives.

Obtaining labels for this sort of data usually requires analysis of the data by a trained expert, such as an electroencephalographer, a cardiologist, or a medical imaging expert. This makes labeling at scale expensive and impractical. Sometimes obtaining labels would require an invasive or expensive medical procedure (*e.g.,* an angiogram) which cannot be performed due to risk to the patient or cost.

Is it possible to reduce the number of labeled training data points required to produce an accurate classifier? In standard machine learning (*i.e., passive learning*), a large training set and label set are blindly provided to the learning algorithm. If instead the learning algorithm is allowed to *choose* the data from which it learns, it may perform better with less required training data.

This is the key insight behind *active learning*. Active learning seeks to overcome the paucity of labeled data by posing *queries* in the form of unlabeled data instances to be labeled by an *oracle* (*e.g.,* a human annotator, such as a medical expert). This way, the active learner tries to achieve high accuracy while minimizing the cost of obtaining labeled data by minimizing the number of used labeled instances.

This paper seeks to provide a general overview of active learning and an investigation of three recent case studies which apply active learning to medical signals. Each of these case studies utilizes a unique variant of active learning, and each claims a significant advantage over passive learning techniques applied to the same data. Investigation and comparison of these techniques will provide insight into the utility of active learning for automated classification of medical signals.

Section 3.2.1 provides an overview of basic supervised learning algorithms and a detailed description of support vector machines, which are used in all

the case studies described. Section 2.2 describes the data under investigation, and how that data is manipulated into a form useable by machine learning algorithms. Section 3 describes the details of active learning and the differences between the broad classes of active learning algorithms, using case studies to illustrate: section 3.2.1 describes the use of active learning to detect seizures in electroencephalograms; section 3.2.2 describes the use of active learning to detect anomalies in medical images; section 3.2.3 describes an application of active learning to detect arrhythmias in electrocardiograms. Section 4 discusses the performance of each of these case studies. Section 5 concludes with a discussion of the significance of the work and offers a discussion of future work to be done in the field.

2 Background

2.1 Machine Learning: Classification

Supervised learning algorithms use labeled training data to infer a model, called a *classifier*, that maps from data instances to classes. This model can then take a data instance (*e.g.,* a particular patient's vital signs) and predict to which of a series of specified classes that data instance belongs (*e.g.,* healthy patients/sick patients).

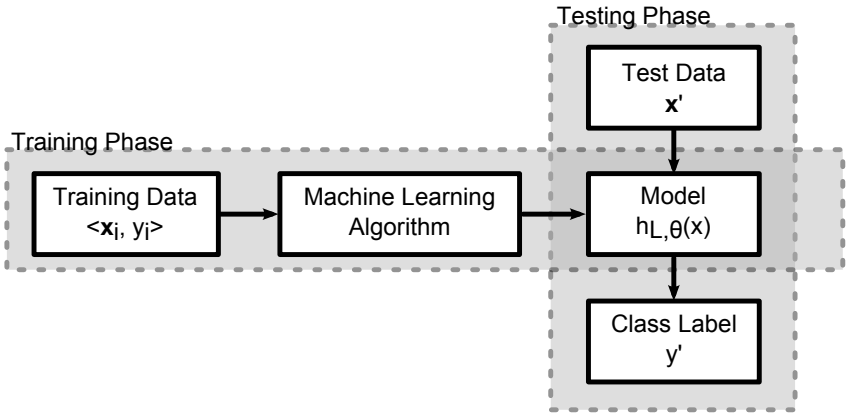


Figure 1: The generic supervised learning process.

Figure 1 illustrates the general machine learning process. As input the algorithm takes a set of labeled *training data* $\mathcal{L} = \langle \mathbf{x}_i, y_i \rangle$, where $\mathbf{x}_i$ are

feature vectors (usually continuous, $\mathbf{x} \in \mathbb{R}^m$) and y_i are *class labels* for one of K classes $y \in \{1, ..., K\}$ (indicating to which class the feature vectors in the training set belong). Binary classification, $K = 2$, is the simplest case. Often, in binary classification, one class is called positive and the other is negative (healthy vs not healthy) and the class labels are encoded as $y \in \pm 1$. The training data $\mathcal{L}$ is used to train a model $h_\theta(\mathbf{x})$ that "fits" that data by minimizing the amount of error in its predictions; that is, we choose values for the parameters θ in the model h such that some measure of error is small. For SVMs, this error is measured as *hinge loss*:

$$Loss = \mathbf{E}[\max(0, 1 - h_\theta(\mathbf{x}) * y)], \tag{1}$$

with expectation taken over some data distribution. Then, given some new data instance $\mathbf{x}_{new}$, the model can be used to predict the class label $h_\theta(\mathbf{x}_{new}) = \hat{y}$ for that data instance [14].

2.1.1 Generative vs Discriminative Classifiers

There are many different types of $h(\mathbf{x})$, often broadly classified into generative classifiers and discriminative classifiers.

Generative models attempt to obtain an estimate of the joint probability, $\hat{P}(\mathbf{x}, y)$ from the training data, often using maximum likelihood estimates, and then produce a classifier

$$h(\mathbf{x}) = \arg\max_y \hat{P}(y|\mathbf{x}). \tag{2}$$

Unfortunately, the joint probability $P(\mathbf{x}, y)$ is usually very complex, which can makes obtaining a good estimate difficult.

Discriminative classifiers avoid this by attempting to model $P(y|\mathbf{x})$ directly (*e.g.,* logistic regression) or by trying to learn a function $h(\mathbf{x})$ that minimizes expected classification error without any probabilistic assumptions at all (*e.g.,* support vector machines).

2.1.2 Support Vector Machines

Support vector machines are a class of non-probabilistic binary linear classifiers that have become extremely popular as they are efficient, generalize well

to diverse datasets, and are extendable to act as non-linear decision functions through use of the kernel trick. [1]

An SVM classifier is defined in terms of its hyperplane

$$\theta^\top \mathbf{x} + b = 0 \tag{3}$$

corresponding to the decision function

$$h_\theta(\mathbf{x}) = sign(\theta^\top \mathbf{x} + b), \theta \in \mathbb{R}^M, b \in \mathbb{R}, \tag{4}$$

given a set of labeled data,

$$\mathcal{L} = \{\langle \mathbf{x}_1, y_1 \rangle, \langle \mathbf{x}_2, y_2 \rangle, \ldots, \langle \mathbf{x}_n, y_n \rangle\}, \quad \mathbf{x}_i \in \mathbb{R}^M, y_i \in \{-1, +1\}. \tag{5}$$

Assuming the data are *separable* [2], the hyperplane separates the two classes of points, while being as far from all points as possible. The region of separation between the plane and the points nearest to the plane is called the *margin*. Hyperplanes parallel to $\theta^\top \mathbf{x} + b = 0$ and passing through these nearest points can be described [3] by

$$y_i(\theta^\top \mathbf{x}_i + b) = 1 \tag{6}$$

with margin $\frac{1}{\|\theta\|_2}$. Maximizing this margin is equivalent to minimizing $\|\theta\|_2$, subject to $y_i(\theta^\top \mathbf{x}_i + b) \geq 1$ (as all the points are on or outside the margin). Substituting $\frac{1}{2}\|\theta\|_2^2$ for $\|\theta\|_2$ converts the problem to a quadratic programming problem, which makes optimization more efficient.

Because the assumption of separability between the two classes rarely holds in practice, *soft margin* classifiers were developed [8]. They introduce slack variables ξ_i which measure the degree of misclassification of $\mathbf{x}_i$,

$$y_i(\theta^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \tag{7}$$

and a cost C which penalizes non-zero ξ_i in the objective function

$$\min_{\theta, \xi, b} \left\{ \frac{1}{2}\|\theta\|_2^2 + C \sum_{i=1}^{n} \xi_i \right\}. \tag{8}$$

[1] All four active learning publications focused on in this paper train an SVM classifier. Thus, the basics of SVMs are presented here.

[2] Two sets of points are separable if they can be completely separated by a single hyperplane.

[3] This is possible because multiplying the classifier $h_\theta(\mathbf{x})$ by a positive constant $\gamma * h_\theta(\mathbf{x}) = sign(\gamma(\theta^\top \mathbf{x} + b)) = sign(\hat{\theta}^\top \mathbf{x} + \hat{b})$ does not change the decision boundary, and thus we can set the scale so that at the points closest to the boundary, $\theta^\top \mathbf{x} + b = \pm 1$

Then, introducing Lagrange multipliers α_i and recasting the problem in terms of the dual, the problem can be expressed as

$$\begin{aligned} &\arg\max_{0\leq\alpha_i\leq C} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ &\text{s.t.} \quad \sum_{i}^{n} \alpha_i y_i = 0. \end{aligned} \tag{9}$$

Those $\mathbf{x}_i$ for which α_i are non-zero are training examples that fall *on* the margin, and thus limit the position of the hyperplane. These are known as *support vectors*. $\mathbf{x}_i$ for which $\alpha_i = C$ are *bound* instances, examples which are incorrectly classified or are within the margin of the hyperplane.

Supporting nonlinear classification requires application of the kernel trick: assuming a feature map $\phi(\mathbf{x})$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. The kernelized dual optimization problem is then

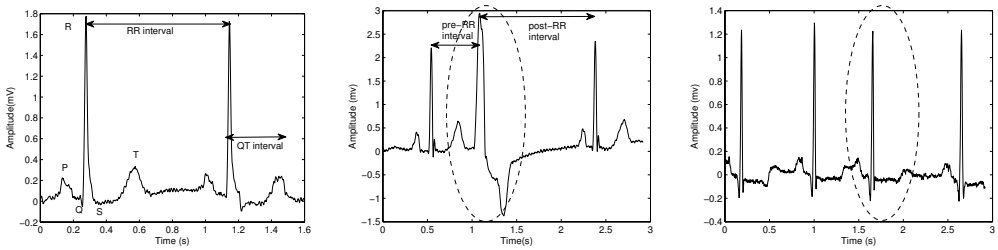
$$\begin{aligned} &\arg\max_{0\leq\alpha_i\leq C} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j}^{n} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i^\top \mathbf{x}_j) \\ &\text{s.t.} \quad \sum_{i}^{n} \alpha_i y_i = 0. \end{aligned} \tag{10}$$

Support vector machines are known to work well in practice on a variety of problems. However, they do have some potential drawbacks:

- They produce only two-class membership labels with no associated class membership probabilities. As will be seen, this makes estimating future error difficult. To correct this, one must employ a *calibration method*, which transforms class labels into membership probabilities. Examples include the popular logistic regression approach by Platt [34].

- They are only applicable to two-class classification. Multi-class classification requires the application of an algorithm to break the problem down into a series of two-class problems.

- While some broad facts about the parameters produced by SVMs can be informative (nonzero α_i means $\mathbf{x}_i$ is a support vector, and under linear kernels the magnitude of the coefficients rougly correspond to the "importance" of the $\mathbf{x}_i$), under non-linear kernels, parameters often have little interpretable meaning.



(a) **(b)** **(c)**

Figure 2: From [47], normal sinus rhythm beats shown in (2a) originate from the pacemaker cells of the sinoatrial node. Premature venctricular contractions (2b) (note the contracted pre-RR interval, overlapping P and T segments between the first and second beats, the dramatic S segment, and the lack of T segment after the second beat) and atrial premature beats (2c) (note the contracted pre-RR interval and overlapping P and T segments between the second and third beats) are two examples of arrhythmias that can be indicative of poor health.

2.2 Medical Signals

In their purest form, medical signals are electrical potentials read from a sensor at a high frequency over time. This produces a continuous waveform. Many of the medical signals most commonly used by doctors and nurses in hospitals are nothing more than graphs of these continuous waveforms, usually with some simple noise reduction. With advances in medical technology, these signals are becoming more widespread in use. Here we discuss three popular signals that are used to monitor patient state.

Electrocardiograms: An *electrocardiogram* (ECG) is a measurement of the electrical potential across the heart. They are one of the primary ways doctors and nurses measure and record the behavior of the heart and assess its health. As they beat, normal, healthy hearts produce electrical signals such as those in Figure 2a. Hearts that are unhealthy can exhibit abberations in electrical behavior, known as *arrhythmias* (Figures 2b and 2c) that are visually distinct from healthy heartbeats. These aberrations can be used by doctors to diagnose specific defects within the heart.

Electroencephalograms: Neural activity in the brain also produces elec-

trical signals, which can be detected by sensors placed on the scalp or implanted in the skull. The resulting signal is known as an electroencephalogram (EEG), and provides information about the brain's behavior and health. Some patients, in particular patients suffering from neurological illnesses such as epilepsy, can experience *seizures*. Seizures are periods of abnormal, excessive, or synchronous neuronal activity. Seizures can be dangerous on their own but can also be indicative of other physiological conditions. Since seizure activity produces corresponding abnormalities in the electrical signals in the brain, they can be viewed in an EEG as a deviation from normal brain electrical activity (see Figure 3). However, it often takes a skilled expert to identify periods of seizure, given an EEG signal.

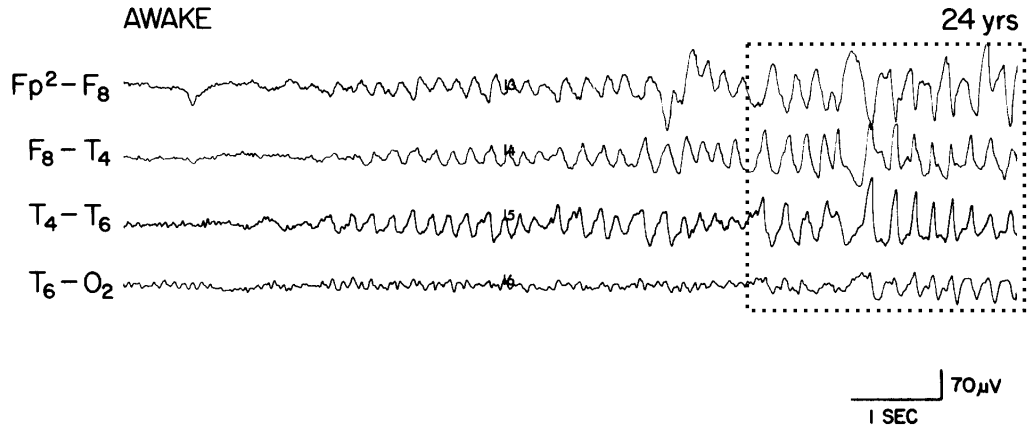


Figure 3: From [6], the EEG signal of a 24-year-old patient with annotated area of seizure activity (dotted box). Increased electrical activity can be seen leading up to and during the seizure.

Medical Imaging: Likewise, due to the rapid development of medical imaging technology, it is becoming more convenient to acquire diagnostic images from a variety of sources, such as x-rays and ultrasounds. Doctors utilize these images to identify illness or deterioration in patients. Many of these images, however, require special expertise to interpret, and most exhibit extreme inter- and intra-patient variability (Figure 4). In all three of these cases, reductions in the cost of sensors have made it possible to capture medical signals continuously on a large number of patients. However, this produces a large quantity of data that is difficult for physicians to handle.

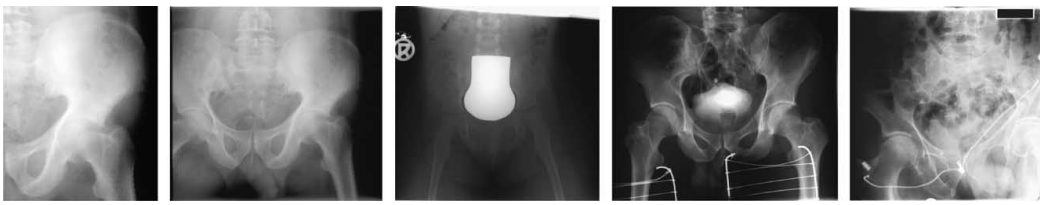


Figure 4: Example of radiographs from the ImageCLEF database (originally from the Aachen University Hospital, Aachen, Germany) [26]. These all belong to the same class (IRMA 1121-120-800-700).

Physicians simply do not have time to constantly watch the signals of every patient in their care to check for aberrations, nor do they have time to retrospectively review the day's data.

Small changes in those patient signals, however, can provide very useful information to doctors. Even infrequent arrhythmias can be an important sign of illness in a patient [47], and timely detection of seizures provides the opportunity for treatment [3]. Clearly, it would be beneficial if the waveforms being produced by a patient could be analyzed automatically, with abnormal or clinically interesting sections of the waveform isolated and presented to the physician for further analysis. As discussed, machine learning classifiers can be employed for just such a purpose. However, as previously noted, classification algorithms often need hundreds or even thousands of labeled training instances to achieve accurate performance. While unlabeled data is abundant in this field, acquiring labeled examples of medical signals is usually difficult, as labeling is either expensive, time consuming, dangerous, or all three. Labeling of EEGs requires the expertise of an electroencephalographer, and labeling a day's worth of EEGs for just one patient can require many hours of work. Likewise, labeling every beat in one day's worth of ECGs for a patient is a laborious task. Medical images can be labeled, but confirming diagnosis of what is seen in the images may require substantial additional tests, which are expensive and put the patient at risk.

2.2.1 Feature Extraction

Most standard machine learning algorithms require data instances (both the training data and test data instances) in the form of a feature vector $\mathbf{x} \in \mathbb{R}^M$. In order to apply these algorithms to medical signals, the signals must first

be "repackaged" as feature vectors. This process is *feature extraction*. Feature extraction techniques are often domain specific (such as R-R interval for ECG, the interval from the peak of one QRS complex to the next). Some techniques, such as approximating the signal with wavelet coefficients, normalized energy measures, and various statistics (such as mean and median) are popular across domains. A full discussion of feature extraction is beyond the scope of this work.

It is important to note that most medical data sets contain many more healthy or "normal" medical signal examples (which are relatively plentiful) than medical signals that belong to the positive class (particular conditions or diseases are often very rare).

3 Active Learning

Active learning is designed to overcome the labeling bottleneck. The key insight is that if the learning algorithm is allowed to choose the data learns from, it will perform better with less required training data. The active learner can choose unlabeled instances it wishes to have labeled by an *oracle* (*e.g.,* a human expert), and in this way can choose to get labels for only those instances which will help improve its ability to classify. Preliminary work has begun applying active learning to medical signals with promising results.

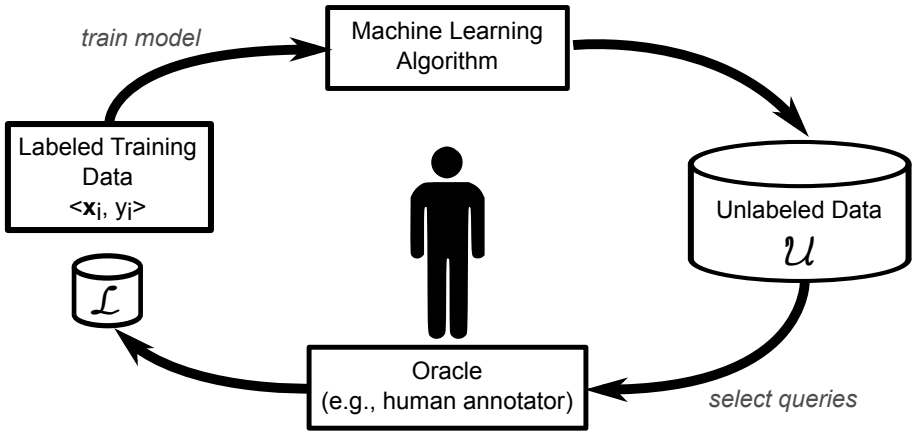


Figure 5: A generic diagram explaining the (pool-based) active learning cycle.

Figure 5 shows the active learning cycle. A learner may begin with a small number of instances in the labeled training set $\mathcal{L}$. The learner then carefully selects one or more unlabeled instances $\mathbf{x}$ from some set of unlabeled instances $\mathcal{U}$, queries their labels from the oracle, learns from the results, and then uses its updated knowledge to choose which instances to query next.

For the purpose of simplicity, there are usually some assumptions placed on the new instances added to the pool. For example, it is assumed that the oracle always provides accurate labels, and that each newly labeled instance can be added directly to the pool of labeled training data already acquired. [4]

There are two major details of the active learning process that must be addressed.

- Where are the unlabeled data instances coming from?
- How does the algorithm select which unlabeled instance to query?

Each of these questions can be answered in several ways, and the answers will determine much of the character of a given active learning algorithm [39].

3.1 Unlabeled Data Sampling

In the broad base of active learning literature, there are three main settings that have been considered in regards to the source of unlabeled instances. Note that all of these settings make the assumption that queries take the form of unlabeled instances to be labeled by the oracle. [5]

3.1.1 Query Synthesis

Query synthesis allows the active learning algorithm to query for any unlabeled instance in the input space. This includes queries that the learner synthesizes from scratch, which frees the algorithm from being limited to instances sampled from some underlying distribution. This scenario has distinct advantages, because the entire input space is available to query, and so the query selection process can be fully optimized. Selecting instances in this way is often tractable and efficient for finite problem domains.

The difficulty, however, comes in presenting these possibly arbitrary instances to the oracle. As the oracle is usually a human annotator, it can

[4] These assumptions will be discussed in Appendix A.

[5] This assumption will also be discussed in Appendix A.

be awkward for them to interpret contrived queries, as they may bear little to no relationship with a "typical" query the human might be trained to respond to. For example, if our input space is all possible EEG waveforms, a waveform synthesized from scratch to optimize the query strategy may not conform to either a "normal" or a "seizure" brainwave pattern, being instead a signal that no biological system could ever produce. This would mean the human expert would be unable to provide a label for such a query.

Because an understanding of the underlying model that produces biological data is often not available, it isn't even possible to check whether a synthesized query would be suitable for the oracle ahead of time. This limitation makes membership query synthesis mostly unsuitable for use on medical data, and thus it will not be further discussed.

3.1.2 Stream-Based Sampling

A slightly more realistic method for obtaining unlabeled data instances is known as *stream-based sampling*, sometimes called *selective sampling* or *sequential sampling*. In this scenario, unlabeled instances are drawn, one at a time, from a data source (which has some underlying distribution), and the learning algorithm decides whether or not to query for the label for that instance.

If the input distribution is uniform across the entire input space (and there is no cost for drawing new unlabeled data points), this technique is functionally equivalent to membership query synthesis. Normally, though, data is sampled from the real world, where the input distribution is non-uniform and unknown. In this scenario, queries are virtually guaranteed to make sense to the oracle, as they are being sampled from a realistic data source.

Stream-based selective sampling may be well adapted for a real-world medical environment, where data samples are often drawn from a patient at regular intervals. Consider data being produced in real-time by a patient, like heartbeats, or brainwaves. An active learner could consider, in real-time, whether to select a certain data instance to be labeled, as that data instance occurs. That said, no work seems to exist employing stream-based selective sampling on real-time medical data, making any evaluation of its effectiveness difficult. This may be due in part to a perceived difficulty in achieving high performance with this technique, and the demand it would place on the oracle (that they would have to be available to provide labels

continuously, in real-time).

3.1.3 Pool-Based Sampling

For many real-world learning problems, large collections of unlabeled data can be gathered at once, which motivates pool-based sampling. *Pool-based sampling* assumes there is some small set of labeled data and a large pool of unlabeled data readily available. The learner selects data from the pool to query. Typically, instances are chosen from the pool in a greedy fashion, according to an informativeness measure used to evaluate all instances in the pool (or some sub-sample if the pool is too big). "Informativeness measures" (query strategies) will be discussed in section 3.2.

The main difference between the previously discussed stream-based sampling and pool-based sampling is that the former scans through the data sequentially and makes query decisions on each data point individually, whereas the latter evaluates and ranks the entire collection of data before selecting the best query.

Due to the aforementioned availability of large amounts of free, cheap, unlabeled data in many scenarios, pool-based sampling is very popular and has been studied extensively, occurring in publications with a much higher frequency than stream-based sampling. Collections of high-frequency medical signals, including ECGs [30], EEGs [2], medical imaging [26], and many others [19] have been assembled to form the basis for these data pools. All of the publications discussed in this report utilize pool-based sampling.

3.2 Query Strategy

Once the data source is established, the second question to answer is how to evaluate the "informativeness" of an unlabeled instance. Many *query strategies* have been developed in the literature to address this question, and some have been tested on medical signals.

3.2.1 Expected Error Reduction Sampling

One possible method query strategy involves estimating the possible reduction in the model's expected future error caused by the addition of an instance. That is, the algorithm can estimate the error of the model trained on $\mathcal{L} \cup \langle \mathbf{x}, y \rangle$ for each possible $\langle \mathbf{x}, y \rangle$ in the remaining unlabeled data set $\mathcal{U}$,

using these unlabeled data points as a validation set. Then, the instance that minimizes the estimated error is chosen.

The error could be formalized using a variety of techniques, *e.g.,* the hinge loss discussed in Section , or the expected 0/1 loss:

$$\mathbf{x}^*_{0/1} = \arg\min_{\mathbf{x}\in\mathcal{U}} \sum_{k=1}^{K} P_\theta(y = k \mid \mathbf{x}) \mathbf{E}_{\mathbf{x}'\sim\mathcal{U}, y'\sim P_{\theta^+\langle\mathbf{x}, y=k\rangle}} [\mathbf{1}(h_\theta(\mathbf{x}') \neq y')]. \quad (11)$$

Here the true label for each candidate query instance is not known, so it is approximated using expectation over all possible labels under the current model θ. $\theta^{+\langle\mathbf{x}, y=k\rangle}$ is the new model after it has been retrained with $\langle\mathbf{x}, y = k\rangle$ added to $\mathcal{L}$.

In theory, this approach could be employed to optimize any generic performance measure of interest, such as precision, recall, F_1-measure, or area under the ROC. The major difficulty here is that expected error reduction computation is often expensive.

Possible solutions are to use a model that is non-parametric (making the training procedure more efficient), or use Monte Carlo sampling from the pool to reduce the number of unlabeled data points to sample, or use approximate training techniques to reduce the number of gradient computations required to retrain the model at each step.

Alternately, a heuristic to efficiently estimate the expected future error could be used. Schohn *et al.* [37] develop just such a heuristic for support vector machine (SVM) classifiers. Balakrishnan and Syed [3] apply this heuristic to detect seizures in EEG signals.

SVMs are non-probabilistic classifiers; that is, they attach no probabilistic estimates to a label's confidence. Only recently have attempts been made to re-introduce probabilities. In order to estimate the expected error, the method proposed by Platt [34] could be used assign probabilities to points in space classified by a support vector machine by projecting all examples onto an axis perpendicular to the dividing hyperplane, and performing logistic regression to extract probabilities. Even so, computing the optimal hyperplane and its support vectors involves a form of quadratic programming, and thus it is usually not feasible to recompute this hyperplane for each candidate query.

Instead, Schohn observes in [37] that choosing data instances which maximally narrow the margin of the classifier would most quickly divide up the feature space and improve the classifier's confidence. This entails choosing

data instances that are nearest to the SVM decision boundary, as the classifier is least confident about the classification of these points. Labeling these points is guaranteed to have an effect on the solution, and is computationally inexpensive; once the hyperplane is computed, only a single dot product is required. The best point to query can be computed via:

$$\arg\max_{\mathbf{x}_u \in \mathcal{U}} \left| \sum_{\mathbf{x}_i \in \mathcal{L}} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_u) + b \right|. \tag{12}$$

In [3], Balakrishnan and Syed employ this technique, using the Gaussian kernel for $k(\mathbf{x}_i, \mathbf{x}_u)$. Their results are presented in Section 4.1.1.

3.2.2 Variance Reduction Sampling

As the last section outlines, minimizing the loss function directly can be very expensive. A possible alternative is to attempt to reduce the generalization error of the model indirectly by *variance reduction*. In contrast to the error estimates, output variance sometimes has a closed-form solution.

Letting a model's predicted output for a given instance $h_{\mathcal{L},\theta}(\mathbf{x})$ be represented as $\hat{y}$, expected future error (of the squared-loss) can be decomposed as the sum of noise, bias, and variance in the following way [18]:

$$\mathbf{E}_T[(\hat{y} - y)^2 \mid \mathbf{x}] = \mathbf{E}[(y - \mathbf{E}[y \mid \mathbf{x}])^2] \tag{13}$$
$$+ (\mathbf{E}_{\mathcal{L}}[\hat{y}] - \mathbf{E}[y \mid \mathbf{x}])^2 \tag{14}$$
$$+ \mathbf{E}_{\mathcal{L}}[(\hat{y} - \mathbf{E}_{\mathcal{L}}[\hat{y}])^2], \tag{15}$$

where $E_{\mathcal{L}}[\cdot]$ is an expectation over the labeled set $\mathcal{L}$, $E[\cdot]$ is an expectation over the conditional density $P(y \mid x)$, and $E_T[\cdot]$ is the expectation over both.

Term (13) on the right hand side of the equation is *noise*, *i.e.,* the variance of the true label y given only x, which does not depend on the model or the training data, and so will be ignored here. [6] Term (14) is *bias*, which represents the error due to the model class itself, *e.g.,* if a linear model is used to learn a function that is only approximately linear. Assuming the class of model used is fixed, the bias is invariant. The final term, (15), is the model's *variance*. Since the learning algorithm itself can do nothing about

[6] Noise may result from stochastic effects of the methods used to obtain the labels, or because the feature representation of the data is inadequate or lossy.

the noise or bias, minimizing the variance is guaranteed to minimize the future generalization error of a given model.

A key insight into the value of the variance is its relation to the *Fisher information* [16]. Fisher information is the variance of the *score*, which itself is the gradient of the log-likelihood with respect to the model parameters. Intuitively, the score indicates how sensitive the likelihood function is to changes in the parameters. Fisher information is the variance of this sensitivity, and is the negative of the *Hessian* of the log-likelihood; that is, it describes the curvature of the log-likelihood function. In essence, the Fisher information tells us how easy it is to learn about a probability distribution by sampling from it. A larger Fisher information means it is possible to make more precise estimates of the parameters of the model from the data.

For a model with a single parameter θ_1, Fisher information measures the amount of information that an observable random variable $\mathbf{x}$ carries about θ_1. Let $P_{\theta_1}(y \mid \mathbf{x})$ be the probability function for y given the data $\mathbf{x}$ and the parameter θ_1 and $P(x)$ is the probability distribution of the data. Then the Fisher information is defined as

$$\mathcal{I}(\theta_1) = -\int_{\mathbf{x}} P(\mathbf{x}) \int_{y} P_{\theta_1}(y \mid \mathbf{x}) \frac{\delta^2}{\delta {\theta_1}^2} \log P_{\theta_1}(y \mid \mathbf{x}) dy \; d\mathbf{x}. \tag{16}$$

This measure is convenient because its inverse $\mathcal{I}(\theta_1)^{-1}$ sets a lower bound on the variance of the model's parameter estimates. For an estimator $\hat{\theta_1}$ with *bias* $b(\theta_1)$:

$$var(\hat{\theta_1}) \geq \frac{[1 + b'(\theta_1)]^2}{\mathcal{I}(\theta_1)}. \tag{17}$$

This result is known as the Cramér-Rao inequality [9, 35], and implies that the sampling distribution $P(x)$ that maximizes $\mathcal{I}(\theta_1)$ is the asymptotically most efficient estimator of θ_1. Intuitively, the Cramér-Rao inequality states that the ability to "pin down" the value of underlying parameters using the estimator is limited by the amount of information gleaned from the data.

Thus, to minimize the variance over the parameter estimates of the model, an active learner need only select data that maximizes the Fisher information score (or, equivalently, minimizes its inverse). This is precisely what Zhang and Oles [50] do to optimize active learning over logistic regression classifiers. Hoi *et al.* [22] then extend their technique to be appropriate for batch mode use, and test it on classification of medical image data.

Let $p(\mathbf{x})$ be the distribution of the unlabeled data, and $q(\mathbf{x})$ be a distribution of unlabeled data chosen for labeling. Then the set of examples that

can most efficiently reduce the variance of the model is found by minimizing the ratio between $\mathcal{I}_{p(\mathbf{x})}(\theta)$ and $\mathcal{I}_{q(\mathbf{x})}(\theta)$. (Minimizing this ratio means the resampled data $q(\mathbf{x})$ has the largest information with respect to the initial distribution $p(\mathbf{x})$.) Note that in the case that θ is not a single parameter, but in fact a vector of K parameters, the Fisher information takes the form of a $K \times K$ covariance matrix, and the question of what to optimize is less obvious. Zhang and Oles choose to optimize:

$$q^* = \arg\min_q tr(\mathcal{I}_{q(\mathbf{x})}(\theta)^{-1}\mathcal{I}_{p(\mathbf{x})}(\theta)). \tag{18}$$

Note that the choice to minimize the trace here (known as an *A-optimal* design) is equivalent to reducing the average variance of the parameter estimates by focusing on values along the diagonal. [7] (For brevity, $\mathcal{I}_{p(\mathbf{x})}(\theta)$ will be shortened to $\mathcal{I}_p(\theta)$, and $\mathcal{I}_{q(\mathbf{x})}(\theta)$ will be shortened to $\mathcal{I}_q(\theta)$.

For logistic regression, the Fisher information matrix $\mathcal{I}_q(\theta)$ is given by

$$\mathcal{I}_q(\theta) = -\int q(\mathbf{x}) \sum_{y=\pm 1} p_\theta(y \mid \mathbf{x}) \frac{\delta^2}{\delta\theta^2} \log p_\theta(y \mid \mathbf{x}) d\mathbf{x} \tag{19}$$

$$= \int \frac{1}{1+exp(\theta^T\mathbf{x})} \frac{1}{1+exp(-\theta^T\mathbf{x})} \mathbf{x}\mathbf{x}^T q(\mathbf{x}) d\mathbf{x} \tag{20}$$

and similarly for $\mathcal{I}_p(\theta)$. Note that the first two terms mean a data point that is close to the margin increases the Fisher information, and the last term $\mathbf{x}^T\mathbf{x}$ means a point with a large size increases the Fisher information (as outliers have a large impact on the variance of the parameters).

In order to estimate $q(\mathbf{x})$, we replace the integration in $\mathcal{I}_p(\theta)$ with a summation over the unlabeled data $\mathcal{U} = \{\mathbf{x}_1, \ldots \mathbf{x}_u\}$, and replace the integration in $\mathcal{I}_q(\theta)$ with the unlabeled data point to be selected $\mathbf{x}_s$. Then the Fisher information becomes:

$$\mathcal{I}_p(\hat{\theta}) = \frac{1}{u} \sum_{\mathbf{x}\in\mathcal{U}} \pi(\mathbf{x})(1-\pi(\mathbf{x}))\mathbf{x}\mathbf{x}^T + \delta I_d \tag{21}$$

$$\mathcal{I}_{\mathbf{x}_s}(\hat{\theta}) = \pi(\mathbf{x}_s)(1-\pi(\mathbf{x}_s))\mathbf{x}_s\mathbf{x}_s^T + \delta I_d \tag{22}$$

[7] D-optimal designs, which minimize the *determinant* of the inverse information matrix, and E-optimal designs, which minimize the maximum *eigenvalue* of the inverse matrix, also appear in the literature. D-optimality is related to minimizing the expected posterior entropy [7].

where (to save space) we represent the logistic regression model as

$$\pi(\mathbf{x}) = p(y = -1 \mid \mathbf{x}) = \frac{1}{1 + exp(\hat{\theta}^T \mathbf{x})}. \tag{23}$$

Note that $\hat{\theta}$ stands for the model parameters estimated from the currently labeled examples $\mathcal{L}$. I_d is the identity matrix of size $d \times d$, and $\delta \ll 1$ is a smoothing parameter. δI_d is added to the estimation of the information matricies to prevent them from being singular (*i.e.,* non-invertible).

Hoi *et al.* [22] take this scheme and adapt it to enable *batch-mode active learning*: rather than choosing a single medical image at a time to query, an optimal subset of the unlabeled images is chosen to be queried at each step (see also Section A.1). The Fisher information matrix for such a subset S then becomes

$$\mathcal{I}_S(\hat{\theta}) = \frac{1}{k} \sum_{\mathbf{x} \in S} \pi(\mathbf{x})(1 - \pi(\mathbf{x}))\mathbf{x}\mathbf{x}^T + \delta \mathcal{I}_d, \tag{24}$$

and the final optimization becomes

$$S^* = \underset{S \subseteq D \wedge |S| = k}{\arg\min}\ tr(\mathcal{I}_S(\hat{\theta})^{-1} \mathcal{I}_p(\hat{\theta})) \tag{25}$$

This can be extended to nonlinear classification by use of the kernel trick to introduce a kernel function $K(\mathbf{x}', \mathbf{x})$, and by rewriting the logistic regression model as

$$p(y \mid \mathbf{x}) = \frac{1}{1 + exp(-yK(\theta, \mathbf{x}))} \tag{26}$$

Using the representer theorem[8] [38], we know that the transformation of the model's parameters under the kernel function $\phi(\theta)$ can be written as a finite linear combination of kernel products $\phi(\mathbf{x})$ for the labeled examples $\mathbf{x} \in \mathcal{L}$, *i.e.,*

$$\phi(\theta) = \sum_{\mathbf{x} \in \mathcal{L}} \alpha(\mathbf{x}) \phi(\mathbf{x}) \tag{27}$$

where $\alpha(\mathbf{x}) \in \mathbb{R}$ is the combination weight for labeled example $\mathbf{x}$. Using this

[8]The representer theorem makes the assumption that regularization is quadratic and the decision boundary of the model is linear (which is the case for logistic regression).

result, we have $K(\theta, \mathbf{x})$ and $p(y \mid \mathbf{x})$ rewritten as

$$K(\theta, \mathbf{x}) = \sum_{\mathbf{x}' \in \mathcal{L}} \alpha(\mathbf{x}') K(\mathbf{x}', \mathbf{x}) \tag{28}$$

$$p(y \mid \mathbf{x}) = \frac{1}{1 + exp(-y \sum_{\mathbf{x}' \in \mathcal{L}} \alpha(\mathbf{x}') K(\mathbf{x}', \mathbf{x}))}. \tag{29}$$

That is, by treating $\{K(\mathbf{x}_1^{\mathcal{L}}, \mathbf{x}), K(\mathbf{x}_2^{\mathcal{L}}, \mathbf{x}), \ldots, K(\mathbf{x}_n^{\mathcal{L}}, \mathbf{x})\}$ (recall $\mathbf{x}^{\mathcal{L}} \in \mathcal{L}$ are the labeled examples) as the new representation for an unlabeled example $\mathbf{x} \in \mathcal{U}$, the result for the linear logistic regression model can be applied directly to the nonlinear case.

Unfortunately, the optimization problem in Eqn. (25) is computationally prohibitive when the number of unlabeled examples is very large. The number of candidate sets for S is exponential in the number of unlabeled examples $|\mathcal{U}| = U$. In order to address this issue, Hoi *et al.* use the properties of submodular functions to establish a greedy algorithm that works well in practice. More detail on submodular functions and greedy optimization is provided in Appendix B.

3.2.3 Density-Weighted Sampling

The previous methods discussed have an advantage over simpler query strategies (see Section 3.2.4) because they consider all the data points in the input space when choosing which data instance to query. This allows the algorithms to avoid querying outliers.

As has been demonstrated, however, these techniques can be computationally expensive. Additionally, as previously established, one of the tenants of good data point selection is that diverse points are queried. This observation motivates the development of *density-weighted sampling*, wherein data is queried on the basis of uncertainty and how "representative" it is of the underlying distribution. Data that inhabits dense regions of the input space, for example, is considered more informative than outliers. Such sampling methods can be achieved by combining one of the query strategies described in this section with a similarity score that establishes how similar a given data point is to all other points in the unlabeled data set.

Wiens and Guttag [47] develop a density-weighted method to classify heartbeats as healthy or as belonging to one of several classes of arrhythmia. They employ a soft-margin SVM to classify the data, but unlike approaches

described in previous sections, they start with a pool of completely unlabeled data. Because of the severe class imbalance in the data (*i.e.,* the number of normal heartbeats greatly outnumbers the number of arrhythmias) and the benefit of choosing representative data instances from each of the classes of data, Wiens and Guttag do not randomly select an initial set of points to query. Instead, they perform hierarchical clustering in an attempt to identify groups of points from each of these classes, and then query the centroid of each of these clusters. Once the initial queries are chosen, the SVM classifier is trained on $\mathcal{L}$ and unlabeled data that falls in the margin of the classifier is re-clustered and the centroids are queried again. The process then repeats.

In hierarchical (agglomerative) clustering, each data point is initially assigned its own cluster. Then, clusters with the smallest inter-cluster distance (as determined by some *linkage criteria*) are combined to form new clusters. This is repeated until the maximum number of clusters κ is no longer exceeded. The maximum number of clusters is incremented after each iteration of the algorithm, in an attempt to achieve a "course to fine" clustering [46].

Wiens and Guttag utilize two "complementary" linkage criteria, in an attempt to reduce the risk of getting stuck in a local solution. The first is *average linkage*:

$$d(q,r) = \frac{1}{(n_q n_r)} \sum_{i=1}^{n_q} \sum_{j=1}^{n_r} dist(\mathbf{x}_{qi}, \mathbf{x}_{rj}). \tag{30}$$

For two clusters q and r, the average linkage defines their distance as the average distance between all pairs of objects in q and r. This linkage is biased toward producing clusters with the same variance, and has a tendency to merge clusters with small variances.

The second linkage criteria used is *Ward's linkage* [45]:

$$d(q,r) = ss(qr) - [ss(q) + ss(r)] \tag{31}$$

where $ss(x)$ is the within-cluster sum of squares of cluster x, which is defined as the sum of squares of the distances between all objects in the cluster and the centroid of the cluster:

$$ss(x) = \sum_{i=1}^{n_x} \left| \mathbf{x}_{xi} - \frac{1}{n_x} \sum_{j=1}^{n_x} \mathbf{x}_{xj} \right|^2 \tag{32}$$

and $ss(qr)$ is the within-cluster sum of squares for the cluster created by combining q and r. Ward's method tends to join clusters with a small number of points, and is biased toward producing clusters with approximately the same number of observations (see Figure 6).

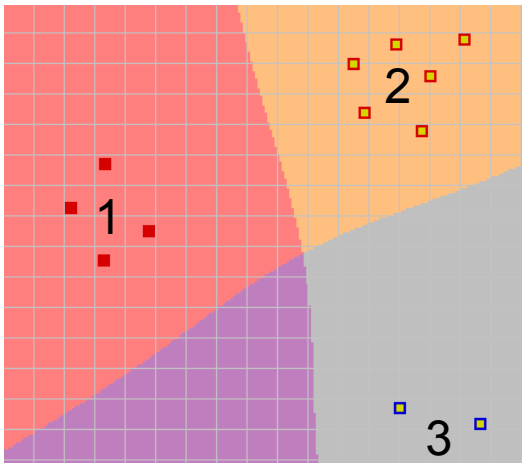


Figure 6: A demonstration of the differences between clustering using average linkage and Ward's linkage. Twelve points are clustered into two clusters. Point borders are colored based on average linkage; centers are colored using Ward's linkage. Under average linkage, the points near (3) are made into the blue cluster, with the remaining points (1 and 2) are placed into one large red cluster. In contrast, under Ward's linkage, the points near (1) are clustered together (red), while remaining points (2 and 3) are clustered into a yellow cluster. Shaded areas show the cluster in which a hypothetical thirteenth point would be added. The only section that would change the clustering is the lower left purple section, where a new point would cause Ward's linkage to reclassify points near (3) into the red cluster with (1), rather than the yellow cluster with (2).

When presented with an outlier, the average linkage tends to assign it to the densest cluster, where it will have the smallest impact on the maximum variance, whereas Ward's method tends to assign it to the cluster with the closest centroid, which minimizes the within-cluster sum of squares.

As each of the linkage criteria can produce κ clusters, between κ and 2κ data instances are queried after each iteration of the algorithm. The

algorithm would normally halt when no unlabeled data lay on or within the margin of the SVM classifier. However, because in this domain there are very likely to be *fusion beats* (that is, arrhythmatic heartbeats that are a fusion of two normal classes of beats), beats are expected to lie within the margin of the classifier. To avoid labeling beats that provide little useful information, the algorithm also terminates when the change in the margin between iterations is within some ϵ. [9]

If densities can be pre-computed efficiently and cached for later use, density-weighted sampling can require very little time to select the next query, which is advantageous for interactive active learning applications. Additionally, the flexibility of this framework (which allows the user to choose different linkage criteria) allows it to be tailored to domain-specific development.

3.2.4 Maximal Uncertainty Sampling

Perhaps the simplest way of selecting queries is by allowing the algorithm to query instances about which it is least certain. This is known as *uncertainty sampling*, and has several variants.

Least Confident: For probabilistic binary learning models, this technique is fairly straightforward, requiring the algorithm to simply look at the posterior probabilities for each $\mathbf{x}$ and choose the one closest to 0.5. In the case of three or more class labels, this can be generalized by choosing the instance whose prediction is *least confident*, that is:

$$\mathbf{x}^*_{LC} = \arg\max_{\mathbf{x}} (1 - P_\theta(\hat{y} \mid \mathbf{x})) \tag{33}$$

where $\hat{y} = \arg\max_y P_\theta(y \mid \mathbf{x})$. That is, we choose the data point with the smallest $P_\theta(\hat{y} \mid \mathbf{x})$ (maximal posterior probability over all possible class labellings). This uncertainty measures the expected 0/1 loss, *i.e.,* the model's belief that it will mislabel x.

The disadvantage of the least confident strategy is that it only considers information about the most probable label, while neglecting information

[9] Nguyen and Smeulders [32] proposed a similar density-based approach, though they went a step further and propagated label information to instances in the same cluster as the queried centroid. They do not test their technique on medical data, so it will not be discussed further here.

about the rest of the label distribution.

Margin Sampling: *Margin sampling* aims to more effectively take advantage of information about the second best label by choosing the instance with the largest margin:

$$\mathbf{x}_M^* = \arg\min_{\mathbf{x}}(P_\theta(\hat{y}_1 \mid \mathbf{x}) - P_\theta(\hat{y}_2 \mid \mathbf{x})) \tag{34}$$

where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable class labels, respectively. Intuitively, if you have a large margin (a big difference in the confidence between the first and second most probable labels), that data point was "easy" to classify, because the most probable label was much more probable than the second most probable label. Thus, by finding the $\mathbf{x}$ with the smallest margin, the algorithm selects the most "uncertain" data instance.

Note that for binary classification, margin sampling is equivalent to least confident sampling: for a probabilistic binary linear classifier, since $0 \leq P_\theta(\hat{y} \mid \mathbf{x}) \leq 1$, the data instance with the smallest margin necessarily has $P_\theta(\hat{y}_1 \mid \mathbf{x})$ closest to 0.5.

If the data being considered has a large label set, margin sampling ignores much of the information from the other label classes.

Entropy: Arguably the most popular maximal uncertainty sampling technique utilizes *entropy* [40]. Calculating the entropy over the distribution of possible class labels results in a value that represents the amount of information needed to "encode" that distribution of labels. In some sense, this can be thought of as a measure of uncertainty: the more entropy in the distribution, the more uncertain the choice of class label for that data value, and the more informative that query would be. The data point with the largest entropy is calculated as

$$\mathbf{x}_H^* = \arg\max_{\mathbf{x}} \sum_{k=1}^{K} -P_\theta(y = k \mid \mathbf{x}) \log P_\theta(y = k \mid \mathbf{x}). \tag{35}$$

Like margin sampling, for binary classification, entropy is equivalent to least confident sampling. In all three cases, the instance with a posterior closest to 0.5 is chosen. However, for data sets with large label sets, measuring the entropy of the distribution of posteriors is usually computationally efficient and generalizes well. The best uncertainty sampling technique may

be application-dependent, though it is clear that across domains all seem to outperform passive learning baselines [39].

Pasolli and Melgani [33] apply margin sampling and maximal entropy sampling to ECG data from the MIT-BIH database (the same data used by Wiens and Guttag in section 3.2.3). Though the problem is multiclass (distinguishing normal beats, PVCs, and SPVCs) they use (binary) SVM classifiers by adopting the one-against-one strategy, in which, for $y \in \{1, \ldots, K\}$, an ensemble of $\frac{K(K-1)}{2}$ parallel SVM classifiers is trained. Each classifier aims to solve a binary classification problem defined by the discrimination between two different classes, and to decide the final label, the class with the maximum number of votes is chosen.

For SVM classifiers, margin sampling is approximated by choosing those data instances closest to the margin. [10] To compute the posterior probabilities required for entropy maximization, the authors utilize the approach of Wu *et al.* [48] in which the pairwise probabilities from the $\frac{K(K-1)}{2}$ parallel SVM classifiers are combined into class probabilities.

3.2.5 Query-by-Committee

Another query selection framework is query-by-committee. In this scenario, a committee of models is trained on the current label set. Each committee member is then allowed to vote on the labellings of query candidates. The instance about which they most disagree is chosen for labeling.

Query-by-committee can be envisioned as minimizing the version space, that is, the set of hypotheses that are consistent with the current labeled training data L. To illustrate, consider the linearly separable binary data set in Figure 7a. A linear classifier which is attempting to split this data set may choose from many possible separating lines, all of which perfectly split the data (Figure 7b). These lines would make up the committee of models, and the point about which they most disagree is the point that minimizes the space of possible models once its label is revealed.

If we view machine learning as a search for the best model within the version space, our goal in active learning is to constrain the size of this space as much as possible (so the search can be more precise) with as few labeled instances as possible.

[10] As we have previously seen, this is equivalent to the error reduction heuristic developed in [37].

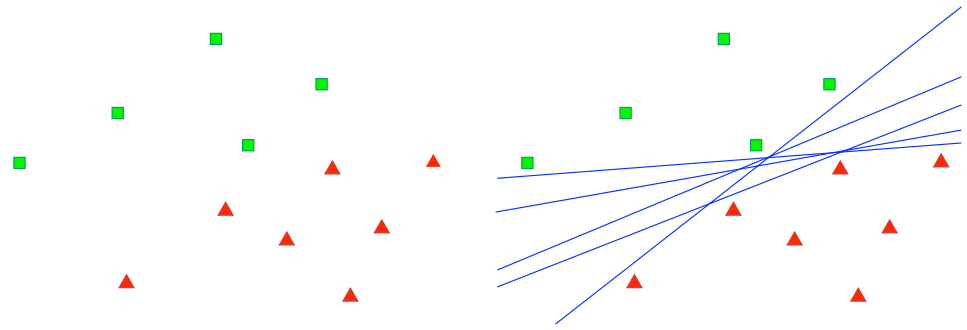


Figure 7: Version space example for a linear classifier. The labeled training data in $\mathcal{L}$ (indicated by the shaded polygons (a)), are all consistant with the different hypotheses (indicated by the lines (b)) but each line represents a different model in the version space [39].

More specifically, query-by-committee involves constructing a committee of possible models, and measuring, for some data instance, the disagreement among those models. There are many ways to construct a committee of possible models, and many ways of measuring disagreement (most commonly vote-entropy or Kullback-Leibler divergence).

Pasolli and Melgani [33] also apply query-by-committee sampling to ECG data from the MIT-BIH database. Choosing a sampling factor $s > 1$, they construct s unlabled data sets $\{U_1, U_2, \ldots, U_s\}$, where each U_i contains only the features $f \in \{1, \ldots, m\}$ that satisfy the condition $(f-1) \mod (s) = (i-1)$. The number of samples of each subset is equal to the original number of samples, but with a number of features reduced by a factor of s. The initial labeled data L is also split in this way, producing $\{L_1, L_2, \ldots, L_s\}$. Then, each of these s data subsets is used to train an ensemble of c SVM classifiers, each using a different kernel function. This produces a committee of $c * s$ classifiers.

The instance with maximal vote entropy among the classifiers is chosen to query:

$$\mathbf{x}^*_{HQ} = \arg\max_{\mathbf{x}} \sum_{i=1}^{K} -r_{y_i,\mathbf{x}} log(r_{y_i,\mathbf{x}}) \tag{36}$$

where $r_{y_i,\mathbf{x}}$ is the relative frequency of class y_i for sample $\mathbf{x}$.

4 Performance and Analysis

4.1 Practical Performance

As each of the different machine learning techniques above was tested on different medical signal data sets, direct comparison is difficult. Instead, a comparison can be made within each domain between active and non-active (*i.e.,* passive) learning to ascertain the benefit granted by active learning.

4.1.1 EEG Seizure Detection

In [3], Balakrishnan *et al.* evaluated the seizure detection performance of an expected error reduction sampling active learner using an SVM classifier on EEG data. The authors utilized the previously published Flint Hills Scientific Public ECOG Dataset [2], which contains 1419 hours of multichannel intracranial EEG data collected from ten anonymous subjects with epilepsy. The dataset contained a total of 59 clinical seizures, with onset and offset of each seizure labeled by expert visual review. [11] The data was then split up into two second epochs.

In order to assess the effectiveness of the active learning, and specifically the query method, the authors utilized three algorithms to train three separate SVMs, a method described by Shoeb *et al.* [42]. The first algorithm ("D_O") utilized all the data and all the labels at once. The second ("D_R") naively choose examples at random from the data pool to "query" (as all the labels were available, this process was simulated). The third ("D_A") used the proposed error reduction query approach.

To evaluate performance, the authors calculated the sensitivity of the classifiers when performing leave-one-out cross-validation (each seizure event was left out once). They also calculated specificity by choosing three non-seizure files as a hold-out test set. The authors justified the disparity between these two techniques by noting the "relative sparsity of seizure activity:" the pool of training examples consisted of an average of 26,970 non-seizure epochs and 338 seizure epochs on average per patient. (Note that a seizure can go on for multiple epochs.) This dramatic imbalance (which is characteristic of medical data sets) makes evaluating performance difficult, as a classifier

[11] This database was created in 2005, and Balakrishnan's paper was published in 2006. The database is no longer available online; the Flint Hills Scientific LLC webpage displays only their address in Lawrence, Kansas, and the email address of their lawyer.

that always guesses "no seizure" will get most of the labels correct. Such a classifier will not generalize well.

Detector	Queries	Sens.	Spec.	FP/hr
D_{AT}	**13,580**	**58%**	98.8%	1.20
D_O	27,2632	56%	**99.0%**	**1.14**

Table 1: Comparison of performance of error reduction active learning classifier and passive classifier. FP is false positives.

As seen in Table 1, while all three classifiers achieved comparable sensitivity, specificity, and false positive rates, D_A required an order of magnitude less data to converge to the results provided by D_O. In fact, with the stopping criteria outlined in the paper, D_A queried only 5% of the data used by D_O to obtain comparable performance. The paper also notes that the active learning technique in conjunction with the suggested stoping criteria ("D_{AT}") utilized an order of magnitude less data than D_R, and the results of D_R were so poor in comparison they were not included in the paper. It is clear that utilizing expected error reduction as a query strategy does indeed result in equal performance with less labeled data.

That said, this study does leave many questions unanswered. Sensitivity for all classifiers was fairly low (maximally 0.58) and active learning did not achieve a significant performance improvement over random querying or passive learning; improvement was entirely in the number of labeled examples required to achieve that performance. Additionally, the number of false positives produced by the classifier might be too large to be used in practice (about one per hour per patient). The feature vector used was very limited in scope, encompassing only the total energy for four frequency ranges, and the data set used to learn on was quite small, so it is possible that significant improvements to the results observed in this paper are possible.

4.1.2 ECG Arrhythmia Detection

Wiens and Guttag [47] evaluate the performance of their density-weighted sampling scheme on the MIT-BIH arrhythmia database [30], a widly used labeled ECG arrhythmia dataset developed in 2003. This allows the performance of their technique to be directly compared with many other passive techniques in the literature which also use this dataset. There are two main

classification tasks set out by the Association for the Advancement of Medical Instrumentation (AAMI) in conjunction with the MIT-BIH database: the detection of ventricular ectopic beats (VEBs) and the detection of supraventricular ectopic beats (SVEBs).

Work on this specific problem was initiated with the work of Hu *et al.* [23] and has seen considerable effort since the establishment of the database, including (in recent years) investigations into the application of active learning [33].

Table 2 compares the performance of Wiens' classifier on the two classification problems with previous work in the field. For classifying VEBs, the proposed technique obtains results considerably better than those achieved in Chazal *et al.* [13] and Ince *et al.* [24], but with far fewer fewer labeled data points on average (45 beats per patient record, versus 350 for [24] and 500 for [13]). Likewise, the proposed technique performs far better than [13] or [24] on SVEB classification.

	VEB				SVEB			
Classifier	Sens.	Spec.	PPV	F-Score	Sens.	Spec.	PPV	F-Score
Ince *et al.*	84.6%	98.7%	87.4%	86.0%	63.5%	99.0%	53.7%	58.2%
Proposed (1)	99.0%	**99.9%**	99.2%	99.1%	88.3%	**100.0%**	99.2%	93.4%
Chazal *et al.*	94.3%	99.7%	96.2%	95.2%	87.7%	96.2%	47.0%	61.2%
Proposed (2)	**99.6%**	**99.9%**	**99.3%**	**99.5%**	**92.0%**	**100.0%**	**99.5%**	**95.6%**

Table 2: Comparison of performance of density-weighted active learning classifier and previously published passive classifiers. Row Proposed (1) is the proposed technique run using the 44 records used by Ince *et al.*, and Proposed (2) the proposed technique run on the 22 records used by Chazal *et al.*.

The proposed method also performs nearly perfectly at detecting premature ventricular contractions (PVCs), and achieved far better performance than a rule-based classifier specifically designed to detect PVCs [20].

In order to isolate the impact of active learning from the effect of their improved feature vector design, Wiens and Guttag ran an experiment that directly compared the effect of actively versus passively selecting the training set, while keeping all other parameters the same. The results show both improvement in F-score [12] over passive training and a dramatic reduction in

[12]F-score is calculated as

$$F = \frac{2 * SE * PPV}{SE + PPV} \tag{37}$$

required training data. These results emphasize the impact of active learning, but also indicate that the design of their feature vector did result in some improvement, as the passive classifier with the improved feature vector still achieves better performance than [24], and performed almost as well as [13] (see Table 3).

VEB Classifier	Queries	TP	TN	FP	FN
Proposed Active	**2148**	**7169**	**102573**	**47**	**66**
Passive	24000	6540	102427	193	695

Table 3: Density-weighted active learning versus passive learning on ECG data using equivalent feature extraction. Active learning outperforms a passive approach, and uses 90% less data. TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives.

In order to ascertain whether their technique was feasible for use in a hospital scenario, the authors asked two cardiologists to label the data required for a run of the classifier. Their results were compared against each other and against the output of the algorithm developed by Hamilton *et al.* [20]. Both cardiologists were able to use the tool developed with minimal training, and achieved high classification results with a small amount of labor per record.

Pasolli and Melgani [33] also utilized the MIT-BIH ECG dataset, and tested margin sampling, maximal entropy sampling, and query-by-committee. Their results are, in general, poorer than those achieved by [47], [24], and [13], and require far more labeled samples, but their active learning techniques do show improvement over reference implementations which select random points to label. [13]

4.1.3 Medical Image Classification

The variance reduction sampling scheme utilized in Hoi *et al.*. [22] was tested on five standard (discrete non-medical) datasets from the UCI ma-

where SE is the *sensitivity*, and PPV is the *positive predictive value*, defined as the number of true positives divided by the number of positives. F-score is a commonly-accepted performance evaluation measure in medicine, where one data class (often the positive class) is more important than the other [43].

[13]Their paper does not provide the details of their implementation, which makes it difficult to compare their performance with other similar work.

chine learning repository [17], and then on 2,785 medical images randomly selected from the ImageCLEF [26] that belong to 15 different medical image categories. Images are represented by 2560 visual features that are extracted by a Gabor wavelet transform.

Two large margin classifiers (kernel logistic regression [KLR] and SVM) were trained on a small initial training set of l data instances chosen randomly, then s additional training examples were chosen. As a control, additional data instances were chosen randomly (**KLR-Rand** and **SVM-Rand**), using margin sampling (**KLR-AL**), using entropy-based uncertainty sampling (**SVM-AL**), and using the proposed batch-mode variance reduction technique (**KLR-BMAL**).

All three active learning techniques perform considerably better than the two reference models, both across the UCI datasets and across all categories of medical data, as seen in Table 4. Furthermore, comparing the proposed batch mode active learning algorithm with the two non-batch active learning algorithms reveals that the proposed algorithm for batch mode active learning always improves classification performance, and in some case does so at a statistically significant level. Varying the batch size from 10 to 50 does not change this result.

Because all classifiers tested utilized only $l+s$ labeled data instances, classification performance when labeling all samples is unavailable, and thus it is impossible to ascertain the reduction in the number of samples necessary for labeling to achieve comparable results with fully passive classifiers. Additionally, it is unclear what impact of the sophisticated wavelet-transform-based feature vector had on classification performance. It is also not clear what the possible class bias in the medical image dataset was, nor how the small sample size and high variance of some of the image categories (some of which contained fewer than 100 images) might have impacted results. These are avenues for future exploration.

4.2 Theoretical Performance

Initial experimental results applying active learning techniques to medical signals show promise, as the above studies have demonstrated. Indeed, active learning has proven to be useful across a wide number of domains, and is increasingly used in private industry and large-scale research projects for real-world applications.

If machine learning is viewed as a search for the "best" model within

Dataset	Active Learning Iteration-1					Active Learning Iteration-2				
	SVM-Rand	KLR-Rand	SVM-AL	KLR-AL	**KLR-BMAL**	SVM-Rand	KLR-Rand	SVM-AL	KLR-AL	**KLR-BMAL**
Cat-1	90.42 ±0.64	91.49 ±0.52	94.33 ±0.39	95.07 ±0.25	**95.63 ±0.22**	91.72 ±0.63	93.56 ±0.42	96.44 ±0.25	96.96 ±0.21	**97.32 ±0.10**
Cat-2	66.68 ±1.94	70.68 ±1.92	74.92 ±1.48	77.54 ±1.29	**79.40 ±1.52**	73.96 ±1.39	78.93 ±1.13	84.27 ±0.52	86.17 ±0.51	**86.72 ±0.52**
Cat-3	41.33 ±1.37	44.11 ±1.40	50.51 ±1.93	54.94 ±1.77	**56.81 ±1.72**	50.08 ±1.69	56.72 ±1.45	68.67 ±1.34	76.52 ±0.68	**77.06 ±0.91**
Cat-4	68.09 ±1.70	69.34 ±1.65	77.08 ±1.40	79.22 ±1.40	**80.83 ±1.23**	74.88 ±1.42	78.27 ±1.11	88.00 ±0.87	89.51 ±0.74	**89.80 ±0.72**
Cat-5	62.10 ±0.82	64.12 ±0.66	63.61 ±0.65	64.52 ±0.68	**65.76 ±0.71**	66.27 ±0.87	67.79 ±0.75	68.18 ±0.61	69.64 ±0.74	**69.73 ±0.55**
Cat-6	49.47 ±2.19	48.97 ±2.72	54.80 ±1.79	56.66 ±2.29	**60.20 ±2.03**	56.10 ±2.40	55.02 ±2.36	70.49 ±0.92	71.17 ±1.09	**71.99 ±1.19**
Cat-7	33.26 ±1.43	35.04 ±1.51	34.73 ±1.63	33.92 ±1.01	**34.12 ±0.98**	41.13 ±1.54	42.75 ±1.48	48.27 ±1.71	49.95 ±1.47	**51.50 ±1.43**
Cat-8	30.85 ±0.82	32.79 ±0.91	32.85 ±1.20	36.87 ±1.31	**37.46 ±1.46**	37.17 ±1.05	42.54 ±1.36	45.76 ±1.37	51.69 ±1.71	**54.51 ±1.42**
Cat-9	25.72 ±0.98	26.09 ±1.01	29.50 ±1.16	29.28 ±1.29	**30.17 ±1.32**	32.70 ±1.05	34.07 ±1.13	42.06 ±1.53	43.27 ±1.89	**44.76 ±1.62**
Cat-10	47.19 ±1.75	47.72 ±1.38	51.23 ±2.18	51.03 ±1.68	**53.27 ±1.56**	57.60 ±1.66	56.45 ±1.53	57.88 ±2.46	62.62 ±1.70	**63.64 ±2.21**
Cat-11	74.40 ±1.82	79.65 ±1.85	80.81 ±1.51	83.99 ±1.15	**85.69 ±0.75**	79.43 ±1.28	84.53 ±0.86	87.54 ±0.55	89.72 ±0.40	**90.21 ±0.42**
Cat-12	34.11 ±1.05	35.81 ±1.03	35.91 ±0.84	36.37 ±0.93	**37.54 ±0.92**	38.93 ±1.04	40.30 ±0.95	43.57 ±1.20	43.79 ±1.14	**45.59 ±1.23**
Cat-13	65.00 ±1.61	64.11 ±1.94	71.97 ±0.88	74.89 ±1.34	**76.37 ±1.17**	71.14 ±0.70	73.01 ±1.07	78.66 ±0.92	84.40 ±0.66	**85.24 ±0.39**
Cat-14	60.06 ±1.19	60.50 ±1.30	66.45 ±1.22	68.96 ±1.18	**69.82 ±1.06**	66.42 ±0.97	67.46 ±0.94	74.84 ±0.54	77.98 ±0.72	**78.71 ±0.47**
Cat-15	30.90 ±1.43	32.58 ±1.57	33.96 ±2.36	33.73 ±2.07	**34.37 ±2.20**	40.69 ±2.17	43.88 ±2.38	52.37 ±2.25	53.63 ±2.32	**55.31 ±2.30**

Table 4: Evaluation of F1 classification performance for batch mode variance minimization active learning on medical image datasets.

the *version space* (the set of possible hypothesis models that fit the current restrictions on the model, *i.e.,* the training data), active learning provides a mechanism for the methodical constraint of the version space, so the search can be more precise [39].

Consider, as a toy example, a set of instances that are points on the real line, and a model class g_θ, where

$$g_\theta(x) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

Under the *probably approximately correct* (PAC) learning model [44], making the strong assumption that the underlying data distribution can be perfectly classified by some hypothesis θ, it is enough to draw $O(\frac{1}{\epsilon})$ random labeled instances, where ϵ is the maximum desired error rate. On the other hand, in a pool-based active learning setting, we can acquire the same number of unlabeled instances from the distribution for free, and only labels incur a cost. If these points are sorted on the real line, a binary search through these unlabeled instances (querying each point the search decides on) pro-

duces a classifier with error less than ϵ with $O(\log \frac{1}{\epsilon})$ queries, an exponential reduction.

A variety of theoretical upper and lower bounds for active learning in the general pool-based setting have been established [10]. For example, if using linear classifiers, the sample complexity can grow to $O(\frac{1}{\epsilon})$ in the worst case, which offers no improvement over standard supervised learning, but is also no worse. A general agnostic active learning algorithm developed by Dasgupta was shown to never exceed the label complexity of supervised learning [11]. Balcan *et al.* [4] also show that, asymptotically, certain active learning strategies should always be better than supervised learning in the limit.

It is important to note that a labeled training set built in conjunction with an active learner is inherently tied to the class of model that was used to select queries. The labeled instances selected are drawn from a biased distribution. As such, if the model class is changed, the selected data may no longer be as useful, and cannot be considered to have been independently sampled from the underlying distribution [39].

It is also must be noted that active learning is only as good as the underlying model being used to fit the data. If the algorithm is inappropriate for the data, active learning will not be able to overcome this deficiency. Additionally, active learning can only exceed the performance of passive learning techniques when there is a bound or cost associated with the number of labeled data instances available. A passive learning technique that is allowed to draw enough data instances from the underlying distribution will eventually meet the performance of an active learner.

5 Conclusion

This paper aims to highlight recent work in applying active learning techniques to the promising field of medical signal classification. While medical signals are plentiful, the cost of obtaining labels for them is high, which makes traditional machine learning approaches expensive, and fully labeled data sets rare. Active learning offers a promising solution to this problem by analyzing unlabeled data instances and requesting labels only for those that would be most useful. In this way, an active learner aims to achieve high accuracy using as few labeled data instances as possible.

This paper has surveyed the initial forays into the application of active

learning to various medical signals currently in widespread use: EEGs, ECGs, and medical images. Each of these papers applied either a novel or recently developed active learning querying strategy to a preexisting medical signal database. Though each technique had advantages and disadvantages, all produced promising results.

There is clear evidence that significant work remains in the application and refinement of these techniques. Query-by-committee strategies offer many possible options in terms of committee makeup and voting techniques, but are only beginning to be applied to medical data. Many variations of batch-mode active learning are possible, and the framework developed in [47] provides for the possibility of improvement for many classification tasks by using different clustering techniques.

One particularly promising observation is the recent swell in publications aiming to combine global machine learning classifiers with a local classifier that is trained on data generated from an individual patient. While for a normal machine learning classifier this would be too cumbersome to achieve (as each patient would have to have a significant amount of data labeled), active learning may make this feasible, and there is significant literature to suggest that this may provide for a dramatic increase in classification accuracy [23, 12, 13, 15].

Regardless, the potential for active learning to have an impact on medical signal classification is profound. Maintaining the high accuracy of current classifiers while requiring much less labeled data would make use of machine learning in routine patient care much more viable. This could lead to dramatic improvements prognostic and diagnostic abilities, which could help to reduce healthcare costs and potentially save lives.

A Active Learning Assumptions

Standard active learning makes assumptions about certain aspects of the problem scenario that are often not true in practice. This appendix briefly explores what can be done when these assumptions are violated. As most of these topics are active areas of research and deserve full survey papers, these descriptions will be necessarily protracted. More information can be found in citesettles.

A.1 Batch-Mode Active Learning

In most active learning research, queries are selected one at a time. If the time required to induce a model is slow or expensive, or if a distributed labeling environment is available (*e.g.,* multiple annotators), it is advantageous to be able to select groups of data instances to query together. This is known as *batch-mode* active learning. The main challenge in batch-mode active learning is how to create the optimal query set $\mathcal{Q}$. Simply querying the Q "best" queries according to some one at a time query strategy often performs poorly, since it fails to consider the overlap in information content among the "best" instances.

This paper describes in detail the process used by Hoi *et al.* [22], in which a submodular function is developed that approximates the reduction in variance for possible subsets of unlabeled data instances. That submodular function is maximized, which produces the subset of data points which would produce the largest reduction in variance if labeled. Other techniques for batch mode active learning include that of Xu *et al.* [49] which strives to incorporate diversity among instances in the chosen batch by querying the centroids of clusters of instances that lie closest to the decision boundary of an SVM.

For the most part, batch-query approaches show improvement over random batch sampling, which in turn is generally better than simple Q "best" batch construction.

A.2 Noisy Oracles

Another assumption in most active learning is that labels produced by the oracle are very acurate. In the medical setting, labels usually come from human experts, like doctors and nurses, which may not always be reliable. Some data instances (patients) are difficult for people to label, and often experience level will factor considerably into the way a doctor views a patient's data. Additionally, experts tasked with labeling data can become fatigued over time. Even if labels come from an empirical experiment, such as a diagnostic medical test, some noise will result from the instrumentation or the test materials themselves.

This uncertainty provides a new question to the learning algorithm: when should the learner decide to query for the label of a *new* unlabeled instance, versus repeating a query on an instance that has *already* been labeled to

increase the confidence in that instance's label? Sheng *et al.* [41] establish heuristics that take into account oracle and model uncertainty to determine when repeated labeling is appropriate.

A.3 Variable Labeling Cost

In many applications there is variance not only in the quality of labels from one query to the next, but also in the cost of obtaining those labels. If the goal of active learning is to minimize the overall cost of training an accurate model, but labels have variable cost, simply reducing the number of labeled instances required does not necessarily guarantee a reduction in overall labeling cost. Kapoor *et al.* [25] propose an approach that sums labeling costs and misclassification costs to evaluate instances.

A.4 Changing Model Classes

As mentioned in 4.2, a labeled traininng set built in conjunction with an active learner is inherently tied to the class of model that was used to select queries. This can be an issue if the most appropriate class of model is not known ahead of time, or we wish to reuse this training data later with a different class of model. Occasionally this is not a problem: Lewis and Catlett [27] show that decision tree classifiers benefit significantly from a training set constructed by an active naive Bayes learner using uncertainty sampling. In the case that it is a problem, however, heterogeneous ensembles of classifiers can be used during active learning [5, 28].

B Submodular Functions for Greedy Batch Active Learning

As seen in section 3.2.2, optimization problems for batch active learning often involve selecting among a number of candidate subsets of a set of unlabeled data instances $\mathcal{U}$. As noted above, the number of candidate subsets is the size of the power set of $\mathcal{U}$, which is exponential in the number of elements in the set:

$$|\mathcal{P}(\mathcal{U})| = 2^{|\mathcal{U}|} \tag{39}$$

As the size of the unlabeled data set is often very large, choosing a subset thus becomes computationally prohibitive.

B.1 Submodular Functions

To address this problem, we turn to *submodular functions* [31]. A submodular function satisfies:

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \quad \forall X \subseteq Y \text{ and } x \notin Y \tag{40}$$

where X, Y, and $\{x\}$ are subsets of some ground set Ω. More informally, a submodular function is a set function that experiences "diminishing returns;" the difference in the value of the function that an element makes when added to the input set decreases as the size of the input set increases. This makes them well suited for use in greedy algorithms.

If in addition to being a submodular function, f is nondecreasing, and $f(\emptyset) = 0$, then a greedy algorithm that searches for a subset S with k elements to maximize $f(S)$, *i.e.,*

$$\max_{|S|=k} f(S) \tag{41}$$

will guarantee a performance $(1 - \frac{1}{e})f(S^*)$, where $S^* = \arg\max_{|S|=k} f(S)$ is the optimal subset.

B.2 Formulating Variance Reduction as a Submodular Function

B.2.1 Approximation

To utilize the above theorem, Hoi *et al.* [22] attempt to approximate the variance reduction objective function established in 3.2.2 as a submodular function. First, they simplify the objective function as follows [14]:

$$tr(\mathcal{I}_S^{-1}\mathcal{I}_p) = tr(\mathcal{I}_S^{-1}\left(\mathcal{I}_S\frac{k}{n} + \frac{n-k}{n}\delta I + \frac{1}{n}\sum_{\mathbf{x}\notin S}\pi(\mathbf{x})(1-\pi(\mathbf{x}))\mathbf{x}\mathbf{x}^\top\right)) \tag{42}$$

$$= \frac{k}{n} + \delta\frac{n-k}{n}tr(\mathcal{I}_S^{-1}) \tag{43}$$

$$+ \frac{1}{n}\sum_{x\notin S}\pi(x)(1-\pi(\mathbf{x}))\mathbf{x}^\top\mathcal{I}_S^{-1}\mathbf{x}. \tag{44}$$

Note that the second term in the above expression $(\frac{k}{n} + \delta\frac{n-k}{n}tr(\mathcal{I}_S^{-1}))$ is proportional to the smoothing parameter δ, which is usually set to be small,

[14] Throughout this section $\mathcal{I}_p(\theta)$ will be written as $\mathcal{I}_p$, to save space.

so it can be ignored. They then focus on the last term, *i.e.,* $\sum_{x \notin S} \mathbf{x}^\top \mathcal{I}_S^{-1} \mathbf{x}$. To approximate the term $\mathbf{x}^\top \mathcal{I}_S^{-1} \mathbf{x}$, we decompose the matrix into its eigenvalues. Let $\{(\lambda_k, \mathbf{v}_k)\}_{k=1}^d$ be the eigenvectors of matrix I_S. Then, for any $\mathbf{x}$, we have

$$\mathbf{x}^\top \mathcal{I}_S^{-1} \mathbf{x} = \sum_{k=1}^{d} \lambda_k^{-1} (\mathbf{x}^\top \mathbf{v}_k)^2 \tag{45}$$

$$\approx \frac{\|\mathbf{x}\|_2^2}{\sum_{k=1}^{d} \frac{\lambda_k (\mathbf{x}^\top \mathbf{v}_k)^2}{\|\mathbf{x}\|_2^2}} \tag{46}$$

$$= \frac{(\sum_{i=1}^{d} x_i^2)^2}{\mathbf{x}^\top \mathcal{I}_S \mathbf{x}}. \tag{47}$$

In the above, the harmonic mean of the eigenvalues λ_i is being approximated by their arithmetic mean, *i.e.,* $\left(\sum_{i=1}^{d} \lambda_i^{-1} p_i\right)^{-1} \approx \sum_{i=1}^{d} \lambda_i p_i$, where $p_i = \frac{(\mathbf{x}^\top \mathbf{v}_i)^2}{\left(\sum_{i=1}^{d} (\mathbf{x}^\top \mathbf{v}_i)\right)^2} = \frac{(\mathbf{x}^\top \mathbf{v}_i)^2}{\|\mathbf{x}\|_2^2}$ is a PDF. Note that this approximation will make the optimal solution more stable than the original objective function. This is because $tr(\mathcal{I}_S^{-1} \mathcal{I}_p)$ is proportional to λ_i^{-1} and therefore is sensitive to the small eigenvalues of $\mathcal{I}_S$ while the approximate one considers the larger eigenvalues.

To further simplify this expression, we assume each example $\mathbf{x}$ is normalized as 1 (*i.e.,* $\|\mathbf{x}\|_2^2 = 1$). Then:

$$\sum_{\mathbf{x} \notin S} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) \mathbf{x}^\top \mathcal{I}_S^{-1} \mathbf{x} \approx \sum_{\mathbf{x} \notin S} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\mathbf{x}^\top \mathcal{I}_S \mathbf{x}} \tag{48}$$

$$= \sum_{\mathbf{x} \notin S} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))k}{\delta + \sum_{\mathbf{x}' \in S} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x}')^2}. \tag{49}$$

Thus, the entire optimization problem in 41 is simplified to

$$\min_{|S|=k \wedge S \subseteq D} \sum_{\mathbf{x} \notin S} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\delta + \sum_{\mathbf{x}' \in S} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x}')^2}. \tag{50}$$

In this form, the optimization problem can be reformulated as the submodular function

$$f(S) = \frac{1}{\delta} \sum_{\mathbf{x} \in D} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) \tag{51}$$

$$- \sum_{\mathbf{x} \notin S} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\delta + \sum_{\mathbf{x}' \in S} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x}')^2}.$$

It is easy to see that $f(\emptyset) = 0$, since Equation 51 becomes

$$
\begin{aligned}
f(\emptyset) =& \frac{1}{\delta} \sum_{\mathbf{x} \in D} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) && (52)\\
& - \sum_{\mathbf{x} \notin \emptyset} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\delta + \sum_{\mathbf{x}' \in \emptyset} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x}')^2} \\
=& \frac{1}{\delta} \sum_{\mathbf{x} \in D} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) - \sum_{\mathbf{x} \in D} \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\delta + 0} && (53)\\
=& \frac{1}{\delta} \sum_{\mathbf{x} \in D} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) - \frac{1}{\delta} \sum_{\mathbf{x} \in D} \pi(\mathbf{x})(1 - \pi(\mathbf{x})) && (54)\\
=& 0. && (55)
\end{aligned}
$$

It is also not difficult to show that $f(S)$ is a nondecreasing submodular function. Using the sufficient and necessary condition for submodular functions citeparker, *i.e.,* for any two sets $A \subseteq B$, for any element $\mathbf{x} \notin B$, $f(S)$ is a submodular function if and only if:

$$f(A \cup \mathbf{x}) - f(A) \geq f(B \cup \mathbf{x}) - f(B). \tag{56}$$

To show the above property, the authors compute $f(S \cup \mathbf{x}) - f(S)$ for $\mathbf{x} \notin S$. For compactness, first let

$$g(\mathbf{x}, S) = \frac{\pi(\mathbf{x})(1 - \pi(\mathbf{x}))}{\delta + \sum_{\mathbf{x}' \in S} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x}')^2}. \tag{57}$$

Then,

$$
\begin{aligned}
f(S \cup \mathbf{x}) - f(S) =& g(\mathbf{x}, S) + \sum_{\mathbf{x}' \notin (S \cup \mathbf{x})} g(\mathbf{x}', S) g(\mathbf{x}, S \cup \mathbf{x})(\mathbf{x}^\top \mathbf{x}')^2 && (58)\\
\geq& 0 && (59)
\end{aligned}
$$

since $g(\mathbf{x}, S) \geq 0$ for any $\mathbf{x}$ and S. This means that $f(S \cup \mathbf{x}) \geq f(S)$, which means $f(S)$ is a nondecreasing function, and $f(S \cup \mathbf{x}) - f(S)$ is a monotonically decreasing function. As a result, when $A \subseteq B$, $f(A \cup \mathbf{x}) - f(A) \geq f(B \cup \mathbf{x}) - f(B)$. Thus, $f(S)$ is a nondecreasing submodular function.

B.2.2 Result

Based on the above approximation, we can see what type of examples will be chosen by the greedy algorithm.

- $f(S \cup \mathbf{x}) - f(S) \propto \pi(\mathbf{x})(1 - \pi(\mathbf{x}))$, which indicates that examples with large classification uncertainty are more likely to be selected than examples with small classification uncertainty.
- The first term in $f(S \cup \mathbf{x}) - f(S)$, $g(\mathbf{x}, S)$, is inverse to $sum_{\mathbf{x}' \in S} \pi(\mathbf{x}')(1 - \pi(\mathbf{x}'))(\mathbf{x}^\top \mathbf{x})^2$. This indcates that the optimal choice of example $\mathbf{x}$ should not be similar to examples in S (the set of selected instances).
- The second term in $f(S \cup \mathbf{x}) - f(S)$ is proportional to $(\mathbf{x}'\mathbf{x})^2$ for all examples $\mathbf{x}' \notin S$. This indcates that the optimal choice of example $\mathbf{x}$ should be similar to the unselected examples.

Intuitively, these three properties are desireable for batch mode active learning.

References

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. *Computing Research Repository*, cs.CL/0006013, 2000.

[2] R. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

[3] G. Balakrishnan and Z. Syed. Scalable personalized medicine with active learning: Detecting seizures with minimum labeled data. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 83–90, November 2010.

[4] M. Balcan, S. Hanneke, and J. Vaughan. The true sample complexity of active learning. *Machine learning*, 80(2):111–139, 2010.

[5] J. Baldridge and M. Osborne. Active learning and the total cost of annotation. In *Proceedings of EMNLP*, pages 9–16, 2004.

[6] W. Blume, G. Bryan Young, and J. Lemieux. EEG morphology of partial epileptic seizures. *Electroencephalography and clinical neurophysiology*, 57(4):295–302, 1984.

[7] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] H. Cramér. *Mathematical methods of statistics*. Princeton mathematical series. Princeton University Press, 1946.

[10] S. Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17:337–344, 2005.

[11] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20:353–360, 2007.

[12] P. de Chazal, M. O'Dwyer, and R. B. Reilly. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, 51(7):1196–1206, July 2004.

[13] P. de Chazal and R. B. Reilly. A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, 53(12):2535–2543, December 2006.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

[15] M. Faezipour, A. Saeed, S. C. Bulusu, M. Nourani, H. Minn, and L. Tamil. A patient-adaptive profiling scheme for ECG beat classification. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1153–1165, September 2010.

[16] R. Fisher. Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 22, pages 700–725. Cambridge Univ Press, 1925.

[17] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[18] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[20] P. Hamilton. Open source ECG analysis. In *Computers in Cardiology, 2002*, pages 101–104. IEEE, 2002.

[21] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, April 2011.

[22] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 417–424, 2006.

[23] Y. H. Hu, S. Palreddy, and W. J. Tompkins. A patient-adaptable ECG beat classifier using a mixture of experts approach. *IEEE Transactions on Biomedical Engineering*, 44(9):891–900, September 1997.

[24] T. Ince, S. Kiranyaz, and M. Gabbouj. A generic and robust system for automated patient-specific classification of ECG signals. *Biomedical Engineering, IEEE Transactions on*, 56(5):1415–1426, 2009.

[25] A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, page 15, 2007.

[26] T. Lehmann, M. Güld, T. Deselaers, D. Keysers, H. Schubert, K. Spitzer, H. Ney, B. Wein, et al. Automatic categorization of medical images for

content-based retrieval and data mining. *Computerized Medical Imaging and Graphics*, 29(2-3):143–155, 2005.

[27] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156, 1994.

[28] Z. Lu and J. Bongard. Exploiting multiple classifier types with active learning. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1905–1906. ACM, 2009.

[29] L. W. Mackey, D. Weiss, and M. I. Jordan. Mixed membership matrix factorization. In *ICML*, pages 711–718, 2010.

[30] G. Moody and R. Mark. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45 –50, may-june 2001.

[31] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[32] H. Nguyen and A. Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[33] E. Pasolli and F. Melgani. Active learning methods for electrocardiographic signal classification. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1405–1416, November 2010.

[34] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[35] R. C. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–91, 1945.

[36] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–105. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998.

[37] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 839–846. Morgan Kaufmann, 2000.

[38] B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In D. Helmbold and B. Williamson, editors, *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 416–426. Springer Berlin / Heidelberg, 2001.

[39] B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison Computer Sciences Department, January 2010.

[40] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, July 1948.

[41] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[42] A. Shoeb, H. Edwards, J. Connolly, B. Bourgeois, S. T. Treves, and J. Guttag. Patient-specific seizure onset detection. *Epilepsy & Behavior*, 5(4):483–498, 2004.

[43] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021, 2006.

[44] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[45] J. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[46] J. Wiens. Machine learning for patient-adaptive ectopic beat classification. Master's thesis, Massachusetts Institute of Technology, June 2010.

[47] J. Wiens and J. V. Guttag. Active learning applied to patient-adaptive heartbeat classification. In *NIPS*, pages 2442–2450, 2010.

[48] T. Wu, C. Lin, and R. Weng. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, 5:975–1005, 2004.

[49] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. *Advances in Information Retrieval*, pages 246–257, 2007.

[50] T. Zhang and F. J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.