Active Learning Literature Survey

Burr Settles

Computer Sciences Technical Report 1648
University of Wisconsin–Madison
Updated on: January 9, 2009

Abstract

The key idea behind *active learning* is that a machine learning algorithm can achieve greater accuracy with fewer labeled training instances if it is allowed to choose the data from which is learns. An active learner may ask *queries* in the form of unlabeled instances to be labeled by an *oracle* (e.g., a human annotator). Active learning is well-motivated in many modern machine learning problems, where unlabeled data may be abundant but labels are difficult, time-consuming, or expensive to obtain.

This report provides a general introduction to active learning and a survey of the literature. This includes a discussion of the scenarios in which queries can be formulated, and an overview of the query strategy frameworks proposed in the literature to date. An analysis of the empirical and theoretical evidence for active learning, a summary of several problem setting variants, and a discussion of related topics in machine learning research are also presented.

Contents

1 Introduction

This report provides a general review of the literature on active learning. There have been a host of algorithms and applications for learning with queries over the years, and this document is an attempt to distill the core ideas, methods, and applications that have been considered by the machine learning community. To make this survey more useful in the long term, an online version will be updated and maintained indefinitely at:

> http://pages.cs.wisc.edu/~bsettles/active-learning/

When referring to this document, I recommend using the following citation:

> Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 2009.

An appropriate BibTeX entry is:

```
@techreport{settles.tr09,
    Author = {Burr Settles},
    Institution = {University of Wisconsin--Madison},
    Number = {1648},
    Title = {Active Learning Literature Survey},
    Type = {Computer Sciences Technical Report},
    Year = {2009}
}
```

This document is written for a machine learning audience, and assumes the reader has a working knowledge of supervised learning algorithms (particularly statistical methods). For a good introduction to general machine learning, I recommend Mitchell (1997) or Duda et al. (2001). This review is by no means comprehensive. My research deals primarily with applications in natural language processing and bioinformatics, thus much of the empirical active learning work I am familiar with is in these areas. Active learning (like so many subfields in computer science) is growing and evolving rapidly, so it is difficult for one person to provide an exhaustive summary. I apologize in advance for any oversights or inaccuracies, and encourage interested readers to submit additions, comments, and corrections to me at: bsettles@cs.wisc.edu.

1.1 What is Active Learning?

Active learning (also called "query learning," or sometimes "optimal experimental design" in the statistics literature) is a subfield of machine learning and, more generally, artificial intelligence. The key hypothesis is that, if the learning algorithm is allowed to choose the data from which it learns—to be "curious," if you will—it will perform better with less training. Why is this a desirable property for learning algorithms to have? Consider that, for any supervised learning system to perform well, it must often be trained on hundreds (even thousands) of labeled instances. Sometimes these labels come at little or no cost, such as the the "spam" flag you mark on unwanted email messages, or the five-star rating you might give to films on a social networking website. Learning systems use these flags and ratings to better filter your junk email and suggest movies you might enjoy. In these cases you provide such labels for free, but for many other more sophisticated supervised learning tasks, labeled instances are very difficult, time-consuming, or expensive to obtain. Here are a few examples:

- *Speech recognition*. Accurate labeling of speech utterances is extremely time consuming and requires trained linguists. Zhu (2005a) reports that annotation at the word level can take ten times longer than the actual audio (e.g., one minute of speech takes ten minutes to label), and annotating phonemes can take 400 times as long (e.g., nearly seven hours). The problem is compounded for rare languages or dialects.

- *Information extraction*. Good information extraction systems must be trained using labeled documents with detailed annotations. Users highlight entities or relations of interest in text, such as person and organization names, or whether a person works for a particular organization. Locating entities and relations can take half an hour for even simple newswire stories (Settles et al., 2008a). Annotations for other knowledge domains may require additional expertise, e.g., annotating gene and disease mentions for biomedical information extraction often requires PhD-level biologists.

- *Classification and filtering*. Learning to classify documents (e.g., articles or web pages) or any other kind of media (e.g., image, audio, and video files) requires that users label each document or media file with particular labels, like "relevant" or "not relevant." Having to annotate thousands of these instances can be tedious and even redundant.

Active learning systems attempt to overcome the labeling bottleneck by asking *queries* in the form of unlabeled instances to be labeled by an *oracle* (e.g., a human annotator). In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learning is well-motivated in many modern machine learning problems where data may be abundant but labels are scarce or expensive to obtain. Note that this kind of active learning is related in spirit, though not to be confused, with the family of instructional techniques by the same name in the education literature (Bonwell and Eison, 1991).

1.2 Active Learning Examples

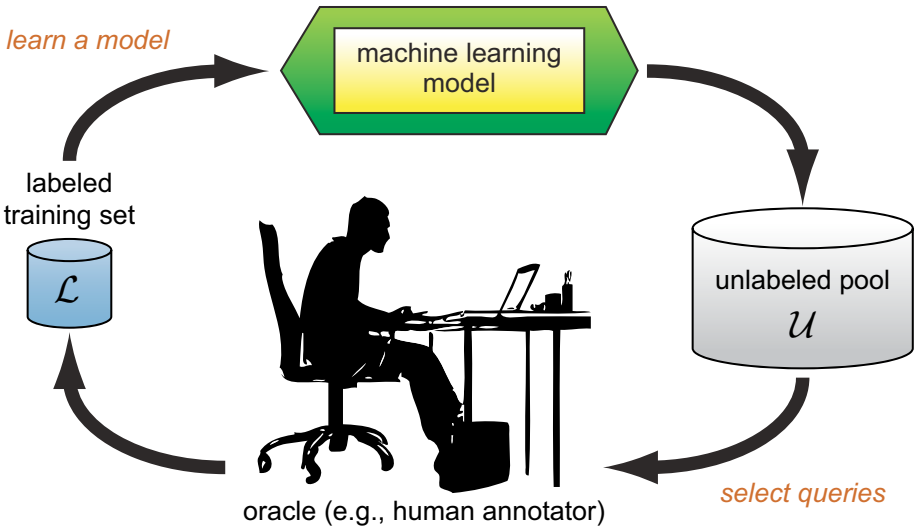


Figure 1: The pool-based active learning cycle.

There are several scenarios in which active learners may pose queries, and there are also several different query strategies that have been used to decide which instances are most informative. In this section, I present two illustrative examples in the *pool-based* active learning setting (in which queries are selected from a large pool of unlabeled instances $\mathcal{U}$) using an *uncertainty sampling* query strategy (which selects the instance in the pool about which the model is least certain how to label). Sections 2 and 3 describe all the active learning scenarios and query strategy frameworks in more detail.

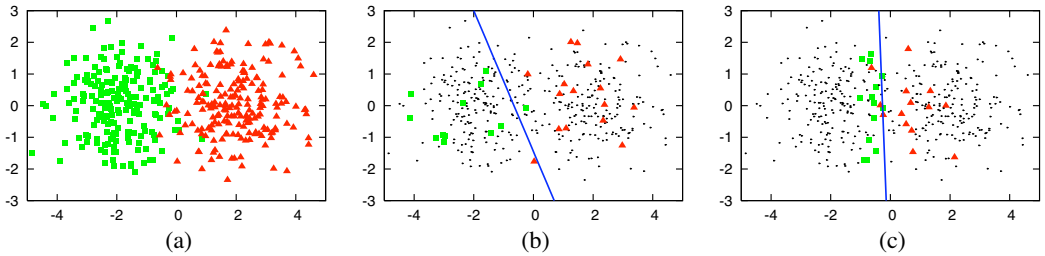


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain. The line represents the decision boundary of the classifier (accuracy = 0.7). (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling (accuracy = 0.9).

Figure 1 illustrates the pool-based *active learning cycle*. A learner may begin with a small number of instances in the labeled training set $\mathcal{L}$, request labels for one or more carefully selected instances, learn from the query results, and then leverage its new knowledge to choose which instances to query next. Once a query has been made, there are usually no additional assumptions on the part of the learning algorithm. The new labeled instance is simply added to the labeled set $\mathcal{L}$, and the learner proceeds from there in a standard supervised way. There are a few exceptions to this, such as when the learner is allowed to make alternative types of queries (Section 5.4), or when active learning is combined with semi-supervised learning (Section 6.1).

Figure 2 shows the potential of active learning in a way that is easy to visualize. This is a toy data set generated from two Gaussians centered at (-2,0) and (2,0) with standard deviation $\sigma = 1$, each representing a different class distribution. Figure 2(a) shows the resulting data set after 400 instances are sampled (200 from each class); instances are represented as points in a 2D feature space. In a real-world setting these instances may be available, but their labels usually are not. Figure 2(b) illustrates the traditional supervised learning approach after randomly selecting 30 instances for labeling, drawn i.i.d. from the unlabeled pool $\mathcal{U}$. The line shows the linear decision boundary of a logistic regression model (i.e., where the posterior equals 0.5) trained using these 30 points. Notice that most of the labeled instances in this training set are far from zero on the horizontal axis, which

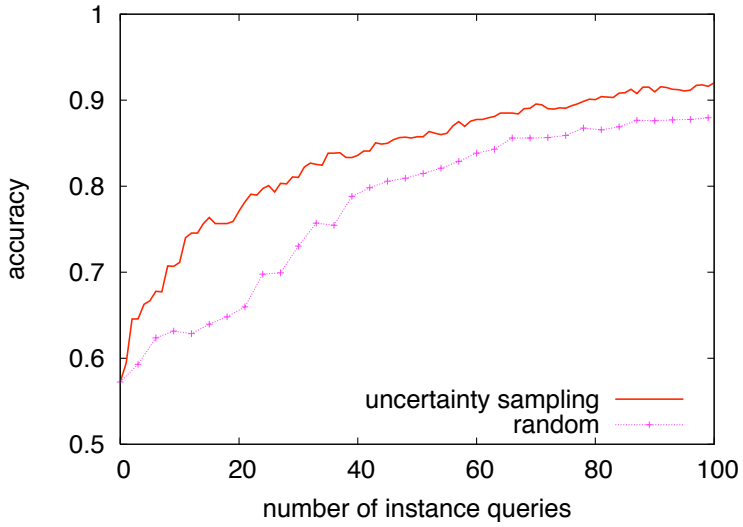


Figure 3: Learning curves for text classification: baseball vs. hockey. Curves plot classification accuracy as a function of the number of documents queried for two selection strategies: uncertainty sampling (active learning) and random sampling (passive learning). We can see that the active learning approach is superior here because its learning curve dominates that of random sampling.

is where the Bayes optimal decision boundary should probably be. As a result, this classifier only achieves accuracy $= 0.7$ on the remaining unlabeled points. Figure 2(c), however, tells a very different story. The active learner uses uncertainty sampling to focus on instances closest to its decision boundary, assuming it can adequately explain those in other parts of the input space characterized by $\mathcal{U}$. As a result, it avoids requesting labels for redundant or irrelevant instances, and achieves accuracy $= 0.9$ with a mere 30 labeled instances. That is a 67% reduction in error compared to "passive" supervised learning (i.e., random sampling), and less than 10% of the data was labeled.

Now let us consider active learning for a real-world learning task: text classification. In this example, a learner must distinguish between baseball and hockey documents from the 20 Newsgroups corpus (Lang, 1995), which consists of 2,000 Usenet documents evenly divided between the two classes. Active learning algorithms are generally evaluated by constructing *learning curves*, which plot the evaluation measure of interest (e.g., accuracy) as a function of the number of

new instance queries that are labeled and added to $\mathcal{L}$. Figure 3 presents learning curves for the first 100 instances labeled using uncertainty sampling and random sampling. The reported results are for a logistic regression model averaged over ten folds using cross-validation. After labeling 30 new instances, the accuracy of uncertainty sampling is 0.810, while the random baseline is only 0.730. As can be seen, the active learning curve dominates the baseline curve for all of the points shown in this figure. We can conclude that an active learning algorithm is superior to some other approach (e.g., a random baseline that represent traditional passive supervised learning) if it dominates the other for most or all of the points along their learning curves.

1.3 Further Reading

To my knowledge, this is the first and only large-scale survey of the active learning literature. One way to view this document is as a heavily annotated bibliography of the field, and the citations within a particular section or subsection of interest serve as good starting points for further investigation. There have also been a few PhD theses over the years dedicated to active learning, with rich related work sections. In fact, this report originated as a chapter in my PhD thesis (Settles, 2008), which focuses on active learning with structured instances and potentially varied annotation costs. Also of interest may be the related work chapters of Tong (2001), which considers active learning for support vector machines and Bayesian networks, and Monteleoni (2006), which considers more theoretical aspects of active learning for classification.

2 Scenarios

There are several different problem scenarios in which the learner may be able to ask queries. The three main settings that have been considered in the literature are (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based active learning. Figure 4 illustrates the differences among these three scenarios. The remainder of this section provides an overview of the different active learning settings.

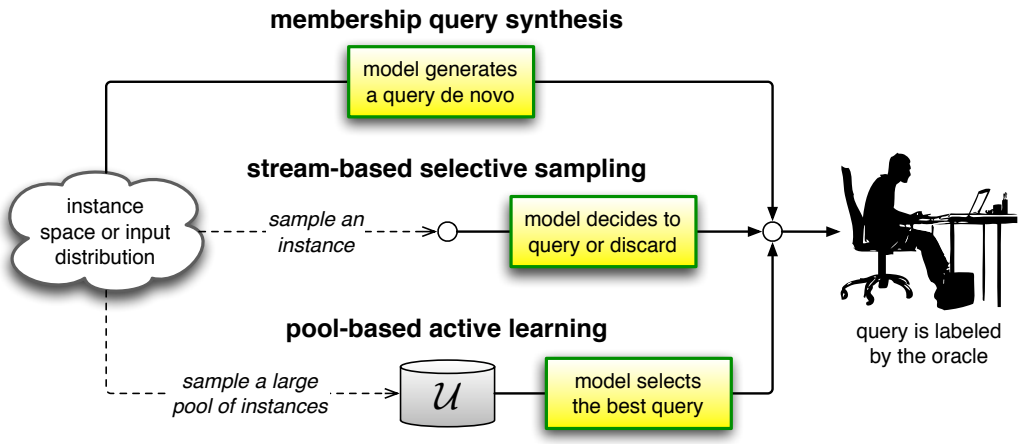


Figure 4: Diagram illustrating the three main active learning scenarios.

2.1 Membership Query Synthesis

One of the first active learning scenarios to be investigated is learning with *membership queries* (Angluin, 1988). In this setting, the learner may request labels for any unlabeled instance in the input space, including (and typically assuming) queries that the learner generates de novo, rather than those sampled from some underlying natural distribution. Efficient query synthesis is often tractable and efficient for finite problem domains (Angluin, 2001). The idea of synthesizing queries has also been extended to regression learning tasks, such as learning to predict the absolute coordinates of a robot hand given the joint angles of its mechanical arm as inputs (Cohn et al., 1996).

Query synthesis is reasonable for many problems, but labeling such arbitrary instances can be awkward if the oracle is a human annotator. For example, Baum and Lang (1992) employed membership query learning with human oracles to train a neural network to classify handwritten characters. They encountered an unexpected problem: many of the query images generated by the learner contained no recognizable symbols, only artificial hybrid characters that had no natural semantic meaning. Similarly, one could imagine that membership queries for natural language processing tasks might create streams of text or speech that amount to gibberish. The stream-based and pool-based scenarios (described in the next sections) have been proposed to address these limitations.

However, King et al. (2004) describe an innovative and promising real-world application of the membership query scenario. They employ a "robot scientist"

which can execute a series of autonomous biological experiments to discover metabolic pathways in the yeast *Saccharomyces cerevisiae*. Here, an instance is a mixture of chemical solutions that constitute a growth medium, as well as a particular yeast mutant. A label, then, is whether or not the mutant thrived in the growth medium. All experiments are autonomously synthesized using an active learning approach based on inductive logic programming, and physically performed using a laboratory robot. This active method results in a three-fold decrease in the cost of experimental materials compared to naïvely running the least expensive experiment, and a 100-fold decrease in cost compared to randomly generated experiments. In domains where labels come not from human annotators, but from experiments such as this, query synthesis may be a promising direction for automated scientific discovery.

2.2 Stream-Based Selective Sampling

An alternative to synthesizing queries is *selective sampling* (Cohn et al., 1994). The key assumption is that obtaining an unlabeled instance is free (or inexpensive), so it can first be sampled from the actual distribution, and then the learner can decide whether or not to request its label. This approach is sometimes called *stream-based* or *sequential* active learning, as each unlabeled instance is typically drawn one at a time from the data source, and the learner must decide whether to query or discard it. If the input distribution is uniform, selective sampling may well behave like membership query learning. However, if the distribution is non-uniform and (more importantly) unknown, we are guaranteed that queries will still be sensible, since they come from a real underlying distribution.

The decision whether or not to query an instance can be framed several ways. One approach is to evaluate samples using some "informativeness measure" or "query strategy" (see Section 3 for examples) and make a biased random decision, such that more informative instances are more likely to be queried (Dagan and Engelson, 1995). Another approach is to compute an explicit *region of uncertainty* (Cohn et al., 1994), i.e., the part of the instance space that is still ambiguous to the learner, and only query instances that fall within it. A naïve way of doing this is to set a minimum threshold on an informativeness measure which defines the region. Instances whose evaluation is above this threshold are then queried. Another more principled approach is to define the region that is still unknown to the overall model class, i.e., to the set of hypotheses consistent with the current labeled training set called the *version space* (Mitchell, 1982). In other words, if any two models of the same model class (but different parameter settings) agree on all

the labeled data, but disagree on some unlabeled instance, then that instance lies within the region of uncertainty. Calculating this region completely and explicitly is computationally expensive, however, and it must be maintained after each new query. As a result, approximations are used in practice (Seung et al., 1992; Cohn et al., 1994; Dasgupta et al., 2008).

The stream-based scenario has been studied in several real-world tasks, including part-of-speech tagging (Dagan and Engelson, 1995), sensor scheduling (Krishnamurthy, 2002), and learning ranking functions for information retrieval (Yu, 2005). Fujii et al. (1998) employ selective sampling for active learning in word sense disambiguation, e.g., determining if the word "bank" means land alongside a river or a financial institution in a given context (only they study Japanese words in their work). The approach not only reduces annotation effort, but also limits the size of the database used in nearest-neighbor learning, which in turn expedites the classification algorithm.

It is worth noting that some authors (e.g., Thompson et al., 1999; Moskovitch et al., 2007) use "selective sampling" to refer to the pool-based scenario described in the next section. Under this interpretation, the term merely signifies that queries are made with a select set of instances sampled from a real data distribution. However, in most of the literature selective sampling refers to the stream-based scenario described here.

2.3 Pool-Based Active Learning

For many real-world learning problems, large collections of unlabeled data can be gathered at once. This motivates *pool-based active learning* (Lewis and Gale, 1994), which assumes that there is a small set of labeled data $\mathcal{L}$ and a large pool of unlabeled data $\mathcal{U}$ available. Queries are selectively drawn from the pool, which is usually assumed to be closed (i.e., static or non-changing), although this is not strictly necessary. Typically, instances are queried in a greedy fashion, according to an informativeness measure used to evaluate all instances in the pool (or, perhaps if $\mathcal{U}$ is very large, some subsample thereof). The examples from Section 1.2 use this active learning setting.

The pool-based scenario has been studied for many real-world problem domains in machine learning, such as text classification (Lewis and Gale, 1994; McCallum and Nigam, 1998; Tong and Koller, 2000; Hoi et al., 2006a), information extraction (Thompson et al., 1999; Settles and Craven, 2008), image classification and retrieval (Tong and Chang, 2001; Zhang and Chen, 2002), video classification

and retrieval (Yan et al., 2003; Hauptmann et al., 2006), speech recognition (Tur et al., 2005), and cancer diagnosis (Liu, 2004) to name a few.

The main difference between stream-based and pool-based active learning is that the former scans through the data sequentially and makes query decisions individually, whereas the latter evaluates and ranks the entire collection before selecting the best query. While the pool-based scenario appears to be much more common among application papers, one can imagine settings where the stream-based approach is more appropriate. For example, when memory or processing power may be limited, as with mobile and embedded devices.

3 Query Strategy Frameworks

All active learning scenarios involve evaluating the informativeness of unlabeled instances, which can either be generated de novo or sampled from a given distribution. There have been many proposed ways of formulating such *query strategies* in the literature. This section provides an overview of the general frameworks used to date. From this point on, I use the notation x_A^* to refer to the most informative instance (i.e., the optimal query) according to some query selection algorithm A.

3.1 Uncertainty Sampling

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

A more general uncertainty sampling strategy uses *entropy* (Shannon, 1948) as an uncertainty measure:

$$x_{ENT}^* = \underset{x}{\operatorname{argmax}} -\sum_i P(y_i|x;\theta) \log P(y_i|x;\theta),$$

where y_i ranges over all possible labelings. Entropy in an information-theoretic measure that represents the amount of information needed to "encode" a distribution. As such, it is often thought of as a measure of uncertainty or impurity

in machine learning. For binary classification, entropy-based uncertainty sampling is identical to choosing the instance with posterior closest to 0.5. However, the entropy-based approach can be generalized easily to probabilistic multi-label classifiers and probabilistic models for more complex structured instances, such as sequences (Settles and Craven, 2008) and trees (Hwa, 2004). An alternative to entropy in these more complex settings involves querying the instance whose best labeling is the *least confident*:

$$x^*_{LC} = \operatorname*{argmin}_x P(y^*|x;\theta),$$

where $y^* = \operatorname{argmax}_y P(y|x;\theta)$ is the most likely class labeling. This sort of strategy has been shown to work well, for example, with conditional random fields or CRFs (Lafferty et al., 2001) for active learning in information extraction tasks (Culotta and McCallum, 2005; Settles and Craven, 2008). For binary classification, this approach is equivalent to the entropy-based strategy.

Uncertainty sampling strategies may also be employed with non-probabilistic models. One of the first works to explore uncertainty sampling used a decision tree classifier (Lewis and Catlett, 1994) by modifying it to have probabilistic output. Similar approaches have been applied to active learning with nearest-neighbor (a.k.a. "memory-based" or "instance-based") classifiers (Fujii et al., 1998; Lindenbaum et al., 2004), by allowing each neighbor to vote on the class label of x, with the proportion of these votes representing the posterior label probability. Tong and Koller (2000) also experiment with an uncertainty sampling strategy for support vector machines or SVMs (Cortes and Vapnik, 1995), that involves querying the instance closest to the linear decision boundary. This last approach is analogous to uncertainty sampling with a probabilistic binary linear classifier, such as logistic regression or naïve Bayes.

3.2 Query-By-Committee

Another, more theoretically-motivated query selection framework is the *query-by-committee* (QBC) algorithm (Seung et al., 1992). The QBC approach involves maintaining a committee $\mathcal{C} = \{\theta^{(1)}, \ldots, \theta^{(C)}\}$ of models which are all trained on the current labeled set $\mathcal{L}$, but represent competing hypotheses. Each committee member is then allowed to vote on the labelings of query candidates. The most informative query is considered to be the instance about which they most disagree.

The fundamental premise behind the QBC framework is minimizing the version space, which is (as mentioned in Section 2.2) the set of hypotheses that are

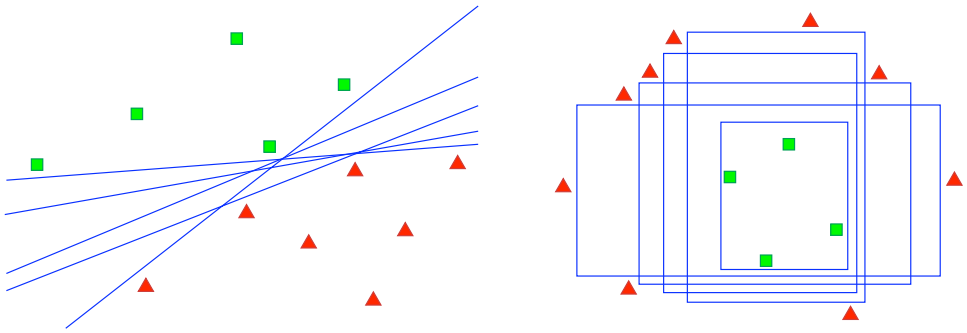


Figure 5: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

consistent with the current labeled training data $\mathcal{L}$. Figure 5 illustrates the concept of version spaces for (a) linear functions and (b) axis-parallel box classifiers in different binary classification tasks. If we view machine learning as a search for the "best" model within the version space, then our goal in active learning is to constrain the size of this space as much as possible (so that the search can be more precise) with as few labeled instances as possible. This is exactly what QBC does, by querying in controversial regions of the input space. In order to implement a QBC selection algorithm, one must:

i. be able to construct a committee of models that represent different regions of the version space, and

ii. have some measure of disagreement among committee members.

Seung et al. (1992) accomplish the first task simply by sampling a committee of two random hypotheses that are consistent with $\mathcal{L}$. For generative model classes, this can be done more generally by randomly sampling an arbitrary number of models from some posterior distribution $P(\theta|\mathcal{L})$. For example, McCallum and Nigam (1998) do this for naïve Bayes by using the Dirichlet distribution over model parameters, whereas Dagan and Engelson (1995) sample hidden Markov models or HMMs by using the Normal distribution. For other model classes, such as discriminative or non-probabilistic models, Abe and Mamitsuka (1998)

have proposed *query-by-boosting* and *query-by-bagging*, which employ the well-known ensemble learning methods boosting (Freund and Schapire, 1997) and bagging (Breiman, 1996) to construct committees. Melville and Mooney (2004) propose another ensemble-based method that explicitly encourages diversity among committee members. There is no general agreement in the literature on the appropriate committee size to use, which may in fact vary by model class or application. However, even small committee sizes (e.g., two or three) have been shown to work well in practice (Seung et al., 1992; McCallum and Nigam, 1998; Settles and Craven, 2008).

For measuring the level of disagreement, two main approaches have been proposed. The first is *vote entropy* (Dagan and Engelson, 1995):

$$x^*_{VE} = \underset{x}{\text{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C},$$

where y_i again ranges over all possible labelings, and $V(y_i)$ is the number of "votes" that a label receives from among the committee members' predictions. This can be thought of as a QBC generalization of entropy-based uncertainty sampling. Another disagreement measure that has been proposed is average *Kullback-Leibler (KL) divergence* (McCallum and Nigam, 1998):

$$x^*_{KL} = \underset{x}{\text{argmax}} \frac{1}{C} \sum_{c=1}^{C} D(P_{\theta^{(c)}} \| P_{\mathcal{C}}),$$

where:

$$D(P_{\theta^{(c)}} \| P_{\mathcal{C}}) = \sum_i P(y_i|x;\theta^{(c)}) \log \frac{P(y_i|x;\theta^{(c)})}{P(y_i|x;\mathcal{C})}.$$

Here $\theta^{(c)}$ represents a particular model in the committee, and $\mathcal{C}$ represents the committee as a whole, thus $P(y_i|x;\mathcal{C}) = \frac{1}{C}\sum_{c=1}^{C} P(y_i|x;\theta^{(c)})$ is the "consensus" probability that y_i is the correct label. KL divergence (Kullback and Leibler, 1951) is an information-theoretic measure of the difference between two probability distributions. So this disagreement measure considers the most informative query to be the one with the largest average difference between the label distributions of any one committee member and the consensus.

Aside from the QBC framework, several other query strategies attempt to minimize the version space as well. For example, Cohn et al. (1994) describe a related selective sampling algorithm for neural networks using a combination of the "most specific" and "most general" models, which lie at two extremes the version

space given the current training set $\mathcal{L}$. Tong and Koller (2000) propose a pool-based query strategy that tries to minimize the version space for support vector machine classifiers directly. The membership query algorithms of Angluin (1988) and King et al. (2004) can also be interpreted as synthesizing de novo instances that limit the size of the version space. However, Haussler (1994) shows that the size of the version space can grow exponentially with the size of $\mathcal{L}$. This means that, in general, the version space of an arbitrary model class cannot be explicitly represented in practice. The QBC framework, rather, uses a committee which is a subset-approximation of the full version space.

3.3 Expected Model Change

Another general active learning framework is to query the instance that would impart the greatest change to the current model *if we knew its label*. An example query strategy in this framework is the "expected gradient length" (EGL) approach for discriminative probabilistic model classes. This strategy was introduced by Settles et al. (2008b) for active learning in the multiple-instance setting (see Section 5.4), and has also been applied to probabilistic sequence models like CRFs (Settles and Craven, 2008).

Since discriminative probabilistic models are usually trained using gradient-based optimization, the "change" imparted to the model can be measured by the length of the training gradient (i.e., the vector used to re-estimate parameter values). In other words, the learner should query the instance x which, if labeled and added to $\mathcal{L}$, would result in the new training gradient of the largest magnitude. Let $\nabla\ell(\mathcal{L};\theta)$ be the gradient of the objective function ℓ with respect to the model parameters θ. Now let $\nabla\ell(\mathcal{L} \cup \langle x, y\rangle;\theta)$ be the new gradient that would be obtained by adding the training tuple $\langle x, y\rangle$ to $\mathcal{L}$. Since the query algorithm does not know the true label y in advance, we must instead calculate the length as an expectation over the possible labelings:

$$x^*_{EGL} = \operatorname*{argmax}_x \sum_i P(y_i|x;\theta) \Big\| \nabla\ell(\mathcal{L} \cup \langle x, y_i\rangle;\theta) \Big\|,$$

where $\|\cdot\|$ is the Euclidean norm of each resulting gradient vector. Note that, at query time, $\|\nabla\ell(\mathcal{L};\theta)\|$ should be nearly zero since ℓ converged at the previous round of training. Thus, we can approximate $\nabla\ell(\mathcal{L} \cup \langle x, y_i\rangle;\theta) \approx \nabla\ell(\langle x, y_i\rangle;\theta)$ for computational efficiency, because the training instances are assumed to be independent.

The intuition behind this framework is that it prefers instances that are likely to most influence the model (i.e., have greatest impact on its parameters), regardless of the resulting query label. This approach has been shown to work well in empirical studies, but can be computationally expensive if both the feature space and set of labelings are very large.

3.4 Variance Reduction and Fisher Information Ratio

Cohn et al. (1996) propose one of the first statistical analyses of active learning, demonstrating how to synthesize queries that minimize the learner's future error by minimizing its variance. They describe a query strategy for *regression* learning problems, in which the output label is a real-valued number (rather than from discrete set of class labels). They take advantage of the result by Geman et al. (1992) showing that a learner's expected future generalization error can be decomposed in the following way:

$$\begin{aligned} E_T\left[(o-y)^2|x\right] = & E\left[(y-E[y|x])^2\right] \\ & + \left(E_{\mathcal{L}}[o]-E[y|x]\right)^2 \\ & + E_{\mathcal{L}}\left[(o-E_{\mathcal{L}}[o])^2\right], \end{aligned}$$

where $E_{\mathcal{L}}[\cdot]$ is an expectation over some labeled set $\mathcal{L}$ of a given size, $E[\cdot]$ is an expectation over the conditional density $P(y|x)$, and E_T is an expectation over both. Here also $o = g(x;\theta)$ is shorthand for the model's predicted output for a given instance x (g is the learned function parameterized by θ), while y indicates the true label of the instance.

The first term on the right-hand side of this equation is the *noise*, i.e., the variance of the true label y given only x, which does not depend on the model or training data. Such noise may result from stochastic effects of the method used to obtain the true labels, for example, or because the feature representation is inadequate. The second term is the *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. This component of the overall error is invariant given a fixed model class. The third term is the model's *variance*, which is the remaining component of the learner's mean squared error with respect to the true regression function. Minimizing the variance, then, is guaranteed to minimize the future generalization error of the model (since the learner itself can do nothing about the noise or bias components).

Cohn et al. (1996) then use the estimated distribution of the model's output to estimate $\tilde{\sigma}_o^2$, the variance of the learner after some new instance $\tilde{x}$ has been labeled and added to $\mathcal{L}$, and then query the instance resulting in the greatest future *variance reduction*:

$$x^*_{VR} = \underset{\tilde{x}}{\operatorname{argmin}}\, \tilde{\sigma}_o^2.$$

They show that this can be done in closed-form for neural networks, Gaussian mixture models, and locally-weighted linear regression. In particular, for neural networks the output variance is approximated by (MacKay, 1992):

$$\sigma_o^2 \approx S(\mathcal{L};\theta) \left[\frac{\partial o}{\partial \theta}\right]^{\mathsf{T}} \left[\frac{\partial^2}{\partial \theta^2} S(\mathcal{L};\theta)\right]^{-1} \left[\frac{\partial o}{\partial \theta}\right],$$

where $S(\mathcal{L};\theta) = \frac{1}{L}\sum_{l=1}^{L}(o^{(l)} - y^{(l)})^2$ is the mean squared error of the current model θ on the training set $\mathcal{L}$. In the equation above, the second and last terms are computed using the gradient of the model's predicted output with respect to model parameters θ. The middle term is the inverse of a covariance matrix representing a second-order expansion around the objective function S with respect to θ. A closed-form expression for $\tilde{\sigma}_o^2$ can then be derived, given the assumptions that $\frac{\partial o}{\partial \theta}$ is locally linear (true for most network configurations) and that variance is Gaussian and constant for all x; further details are given by Cohn (1994). Since the equation is a smooth function and differentiable with respect to any query $\tilde{x}$ in the input space, gradient methods can be used to search for the best possible query that minimizes future variance, and therefore future error. This approach is derived from statistical theories of optimal experimental design (Federov, 1972).

However, the approach of Cohn et al. (1996) applies only to regression tasks, and synthesizes new queries de novo. For many learning problems like text classification, this technique cannot be used. More recently, though, Zhang and Oles (2000) have proposed an analogous approach for selecting optimal queries in a pool-based setting for discriminative classifiers based on *Fisher information*. Formally, Fisher information $\mathcal{I}(\theta)$ is the variance of the *score*, which is the partial derivative of the log-likelihood function with respect to model parameters θ (Schervish, 1995). Fisher information is given by:

$$\mathcal{I}(\theta) = -\int_x P(x) \int_y P(y|x;\theta) \frac{\partial^2}{\partial \theta^2} \log P(y|x;\theta),$$

and can be interpreted as the overall uncertainty about an input distribution $P(x)$ with respect to the estimated model parameters. For a model with multiple parameters, Fisher information takes the form of a covariance matrix. The optimal

instance to query, then, is the one which minimizes the *Fisher information ratio*:

$$x^*_{FIR} = \underset{x}{\operatorname{argmin}} \operatorname{tr}\left(\mathcal{I}_x(\theta)^{-1}\mathcal{I}_{\mathcal{U}}(\theta)\right),$$

where $\mathcal{I}_x(\theta)$ is the Fisher information matrix for an unlabeled query candidate $x \in \mathcal{U}$, and $\mathcal{I}_{\mathcal{U}}(\theta)$ is the analogous matrix integrated over the entire unlabeled pool. The trace function $\operatorname{tr}(\cdot)$ is the sum of the terms along the principal diagonal of a matrix, thus the equation above provides us with a ratio given by the inner product of $\mathcal{I}_x(\theta)$'s inverse matrix and $\mathcal{I}_{\mathcal{U}}(\theta)$.

The key idea behind the Fisher information ratio is that $\mathcal{I}_x(\theta)$ will tell us not only how uncertain the model is about x (e.g., the magnitude of the matrix diagonal), but it also tells us *which* model parameters are most responsible for this uncertainty, as it is encoded in the matrix. Likewise, $\mathcal{I}_{\mathcal{U}}(\theta)$ can tell us the same information about the entire unlabeled pool. By minimizing the ratio above, the learner will tend to query the instance whose model variance is most similar to the overall input distribution approximated by $\mathcal{U}$. A more formal explanation as to why this is the optimal approach stems from the Cramér-Rao lower-bound on asymptotic efficiency, as explained by Zhang and Oles (2000). They apply this method to text classification using binary logistic regression. Hoi et al. (2006a) extend this approach to active text classification in the batch-mode setting (see Section 5.2) in which a set of queries $\mathcal{Q}$ is selected all at once in an attempt to minimize the ratio between $\mathcal{I}_{\mathcal{Q}}(\theta)$ and $\mathcal{I}_{\mathcal{U}}(\theta)$. Settles and Craven (2008) have also generalized the Fisher information ratio approach to probabilistic sequence models such as CRFs.

The query strategies of variance reduction (Cohn et al., 1996) and Fisher information ratio (Zhang and Oles, 2000), while designed for different tasks and active learning scenarios, are grouped together here because they can be viewed as strategies under a more general *variance minimization* framework. Both are grounded in statistics, and both select the optimal query to reduce model variance given the assumptions. There are some practical disadvantages to these methods, however, in terms of computational complexity. In both strategies, estimating the variance requires inverting a $K \times K$ matrix for each new instance, where K is the number of parameters in the model θ, resulting in a time complexity of $O(UK^3)$, where U is the size of the query pool $\mathcal{U}$. This quickly becomes intractable for large K, which is a common occurrence in, say natural language tasks. For variance estimation with neural networks, Paass and Kindermann (1995) propose a sampling approach based on Markov chains to address this problem. For inverting the Fisher information matrix, Hoi et al. (2006a) use principal component

analysis to reduce the dimensionality of the parameter space. Alternatively, Settles and Craven (2008) approximate the matrix with its diagonal vector, which can be inverted in only $O(K)$ time. However, these methods are still empirically much slower than simpler query strategies like uncertainty sampling.

3.5 Estimated Error Reduction

Query strategies that attempt to minimize generalization error directly have also been considered in the literature. The algorithms in the previous section minimize error indirectly by reducing model variance, however, this cannot be done in closed form for all model classes. We can instead estimate the expected future error that would result if some new instance x is labeled and added to $\mathcal{L}$, and then select the instance that minimizes that expectation. The idea is similar in spirit to the EGL strategy (Section 3.3), but differs in that we want to query for minimal expected future error, as opposed to maximal expected model change.

Roy and McCallum (2001) first proposed the estimated error reduction framework for text classification using naïve Bayes. Zhu et al. (2003) combined this framework with a semi-supervised learning approach (Section 6.1), resulting in a dramatic improvement over random or uncertainty sampling. Guo and Greiner (2007) employ an "optimistic" variant[1] that also biases the expectation toward the most likely label, using uncertainty sampling as a fallback strategy when the oracle provides an unexpected labeling. The estimated error reduction framework has the dual advantage of being near-optimal and not dependent on the model class. All that is required is an appropriate loss function and a way to estimate posterior label probabilities. For example, strategies in this framework have been successfully used with a variety of models including naïve Bayes (Roy and McCallum, 2001), Gaussian random fields (Zhu et al., 2003), logistic regression (Guo and Greiner, 2007), and support vector machines (Moskovitch et al., 2007).

Unfortunately, estimated error reduction may also be the most prohibitively expensive query selection framework. Not only does it require estimating the expected future error over $\mathcal{U}$ for each query, but a new model must be incrementally re-trained for each possible query labeling, which in turn iterates over the entire pool. This leads to a drastic increase in computational cost. For some model classes such as Gaussian random fields (Zhu et al., 2003), the incremental training procedure is efficient and exact, making this approach fairly practical. For a many

[1] Guo and Greiner refer to their strategy as maximizing "mutual information." However, their formulation is, in fact, equivalent to minimizing the expected future log-loss.

other model classes, this is not the case. For example, a binary logistic regression model would require $O(ULG)$ time complexity simply to choose the next query, where U is the size of the unlabeled pool $\mathcal{U}$, L is the size of the current training set $\mathcal{L}$, and G is the number of gradient computations required by the by optimization procedure until convergence. A classification task with three or more labels using a MaxEnt model (Berger et al., 1996) would require $O(M^2ULG)$ time complexity, where M is the number of class labels. For a sequence labeling task using CRFs, the complexity explodes to $O(TM^{T+2}ULG)$, where T is the length of an input sequence. Because of this, the applications of the estimated error reduction framework have mostly only considered simple binary classification tasks. Moreover, because the approach is often still impractical, some researchers have resorted to subsampling the pool $\mathcal{U}$ when selecting queries (Roy and McCallum, 2001) or using only approximate training techniques (Guo and Greiner, 2007).

3.6 Density-Weighted Methods

It has been suggested that uncertainty sampling and QBC strategies are prone to querying outliers, which is a main motivating factor behind the Fisher information and estimated error reduction frameworks (Roy and McCallum, 2001; Zhu et al., 2003; Hoi et al., 2006b). Figure 6 illustrates this problem for a binary linear classifier using uncertainty sampling. The least certain instance lies on the classification boundary, but is not "representative" of other instances in the distribution, so knowing its label is unlikely to improve accuracy on the data as a whole. QBC and EGL may exhibit similar behavior, by spending time querying possible outliers simply because they are controversial, or are expected to impart significant change in the model. The Fisher information and estimated error reduction strategies avoid such traps implicitly, by utilizing the unlabeled pool $\mathcal{U}$ when estimating ratios and future errors (respectively). We can also model the input distribution in query selection strategies explicitly.

The *information density* framework presented by Settles and Craven (2008), and further analyzed in Chapter 4 of Settles (2008), is a density-weighting technique. The main idea is that informative instances should not only be those which are uncertain, but also those which are "representative" of the input distribution (i.e., inhabit dense regions of the input space). Therefore, we wish to query instances as follows:

$$x^*_{ID} = \operatorname*{argmax}_x \phi_A(x) \times \left(\frac{1}{U} \sum_{u=1}^{U} \text{sim}(x, x^{(u)}) \right)^{\beta}.$$

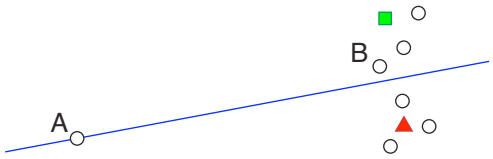


Figure 6: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances in $\mathcal{L}$, and circles represent unlabeled instances in $\mathcal{U}$. Since A is on the decision boundary, it would be queried as the most uncertain. However, querying B is likely to result in more information about the data distribution as a whole.

Here, $\phi_A(x)$ represents the informativeness of x according to some "base" query strategy A, such as an uncertainty sampling or QBC approach. The second term weights the informativeness of x by its average similarity to all other instances in the input distribution (as approximated by $\mathcal{U}$), subject to a parameter β that controls the relative importance of the density term.

This formulation was first published by Settles and Craven (2008), however it is not the only strategy to consider density and representativeness in the literature. McCallum and Nigam (1998) also developed a density-weighted QBC approach for text classification with naïve Bayes, which is a special case of information density. Fujii et al. (1998) considered a query strategy for nearest-neighbor methods that selects queries that are (i) unlike the labeled instances already in $\mathcal{L}$, and (ii) most similar to the unlabeled instances in $\mathcal{U}$. Nguyen and Smeulders (2004) have also proposed a density-based approach that first clusters instances and tries to avoid querying outliers by propagating label information to instances in the same cluster. Similarly, Xu et al. (2007) use clustering to construct sets of queries for batch-mode active learning (Section 5.2) with SVMs. Reported results in all these approaches are superior to methods that do not consider density or representativeness measures. Furthermore, Settles and Craven (2008) show that if densities can be pre-computed efficiently and cached for later use, the time required to select the next query is essentially no different than the base informativeness measure (e.g., uncertainty sampling).

4 Analysis of Active Learning

This section discusses some of the empirical and theoretical evidence for how and when active learning works in practice.

4.1 Empirical Analysis

An important question is: does active learning actually work? Most empirical results in the literature suggest yes (e.g., Cohn et al., 1994; Thompson et al., 1999; Tong and Koller, 2000; Tur et al., 2005; Settles and Craven, 2008), however, there are caveats. First, consider that a training set built in cooperation with an active learner is inherently tied to the learning model that was used to generate it (i.e., the model selecting the queries). Therefore, the labeled instances are not drawn i.i.d. from the underlying data distribution. If one were to change models—as we often do in machine learning when the state of the art advances—this training set may no longer be as useful to the new model class. Baldridge and Osborne (2004) study these effects for several natural language parsers and suggest some techniques for better model generalization. Schein and Ungar (2007) also show that active learning can sometimes require more labeled instances than "passive" supervised learning for the *same* model class, in their case logistic regression. Guo and Schuurmans (2008) demonstrate that general active learning strategies, when employed in a batch-mode setting (Section 5.2) are also often much worse than random i.i.d. sampling. Anecdotally, however, active learning does reduce the number of labeled instances required to achieve a given level of accuracy in most reported cases (though this may be due to the publication bias).

4.2 Theoretical Analysis

A theoretical case for why and when active learning should work remains somewhat elusive, although there have been some recent advances. In particular, it would be nice to have some sort of bound on the number of queries required to learn a sufficiently accurate model for a given task, and theoretical guarantees that this number is less than in the passive supervised setting. Consider the following toy learning task to illustrate the potential of active learning. Suppose that instances are points lying on a one-dimensional line, and our model class is a

simple binary thresholding function g parameterized by θ:

$$g(x;\theta) = \begin{cases} 1 & \text{if } x > \theta, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

According to the *probably approximately correct* (PAC) learning model (Valiant, 1984), if the underlying data distribution can be perfectly classified by some hypothesis θ, then it is enough to draw $O(1/\epsilon)$ random labeled instances, where ϵ is the maximum desired error rate. Now consider an active learning setting, in which we can acquire the same number of *unlabeled* instances from this distribution for free. If we arrange these points on the real line, their (unknown) labels are a sequence of zeros followed by ones, and our goal is to quickly discover the location at which the transition occurs. By conducting a simple binary search through these unlabeled instances, a classifier with error less than ϵ can be achieved with a mere $O(\log 1/\epsilon)$ queries—since all other labels can be inferred—resulting in an exponential reduction in the number of labeled instances required. Of course, this is a one-dimensional, perfectly separable, noiseless, binary toy learning task. Generalizing this phenomenon to more interesting and realistic problem settings is the focus of much theoretical work in active learning.

Unfortunately, little more is known. There have been some fairly strong theoretical results for the membership query scenario, in which the learner is allowed to create query instances de novo and acquire their labels (Angluin, 1988, 2001). However, such instances can be difficult for humans to annotate (Baum and Lang, 1992) and may result in querying outliers, because they are not created according to an underlying natural distribution. Since a great many applications for active learning assume that unlabeled data (drawn from some natural distribution) are available, these results also have limited practical impact.

The main theoretical result to date in the stream-based and pool-based scenarios seems to be an analysis of the query-by-committee (QBC) algorithm by Freund et al. (1997). They show that, under certain assumptions, it is possible to achieve generalization error ϵ after seeing $O(d/\epsilon)$ unlabeled instances, where d is the *Vapnik-Chervonenkis (VC) dimension* (Vapnik and Chervonenkis, 1971) of the model space, and requesting only $O(d \log 1/\epsilon)$ labels. This, like the toy example above, is an exponential improvement over the typical $O(d/\epsilon)$ sample complexity of the supervised setting. This result is tempered somewhat by the computational complexity of the QBC algorithm in practice, although Gilad-Bachrach et al. (2006) suggest some improvements by limiting the version space via kernel functions.

Dasgupta et al. (2005) propose a variant of the perceptron update rule which can achieve the same sample complexity bounds as reported for QBC, but for a single linear classifier. In earlier work, Dasgupta (2004) also provided a variety of theoretical upper and lower bounds for active learning in more general pool-based settings. In particular, if using linear classifiers the sample complexity can explode to $O(1/\epsilon)$ in the worst case, which offers no improvement over standard supervised learning, but is also no worse. However, Balcan et al. (2008) also show that, under an asymptotic setting, active learning is always better than supervised learning in the limit.

Most of these results have used theoretical frameworks similar to the standard PAC model, and necessarily assume that the learner knows the correct concept class in advance. Put another way, they assume that *some* model in our hypothesis class can perfectly classify the instances, and that the data are also noise-free. To address these limitations, there has been some more recent theoretical work in *agnostic active learning* (Balcan et al., 2006), which only requires that the unlabeled instances be drawn i.i.d. from a fixed distribution, and even noisy distributions are allowed. Hanneke (2007) extends this work by providing upper bounds on query complexity for the agnostic setting, and Dasgupta et al. (2008) propose a somewhat more efficient query selection algorithm. Cesa-Bianchi et al. (2005) have also shown that active learning is possible in the "regret" framework, also known as online adversarial learning.

However, most positive theoretical results to date have been based on intractable algorithms, or methods otherwise too prohibitively complex and particular to be used in practice. The few analyses performed on efficient algorithms have assumed uniform or near-uniform input distributions (Balcan et al., 2006; Dasgupta et al., 2005), or severely restricted hypothesis spaces. Furthermore, these studies have largely only been for simple (often binary) classification problems, with few implications for more complex models (e.g., that label structured instances like sequences and trees), which are central to many large-scale information management tasks addressed by the machine learning community today.

5 Problem Setting Variants

This section discusses some of the generalizations and extensions of traditional active learning work into more complex problem settings.

Figure 7: An information extraction example viewed as a sequence labeling task. (a) A sample input sequence $\mathbf{x}$ and corresponding label sequence $\mathbf{y}$. (b) A sequence model represented as a finite state machine, illustrating the path of $\langle \mathbf{x}, \mathbf{y} \rangle$ through the model.

5.1 Active Learning for Structured Outputs

Active learning for classification tasks has been widely studied (e.g., Cohn et al., 1994; Zhang and Oles, 2000; Guo and Greiner, 2007). However, many important learning problems involve predicting structured outputs on instances, such as sequences and trees. Figure 7 illustrates how, for example, an information extraction problem can be viewed as a sequence labeling task. Let $\mathbf{x} = \langle x_1, \ldots, x_T \rangle$ be an observation sequence of length T with a corresponding label sequence $\mathbf{y} = \langle y_1, \ldots, y_T \rangle$. Words in a sentence correspond to *tokens* in the input sequence $\mathbf{x}$, which are mapped to labels in $\mathbf{y}$. Figure 7(a) presents an example $\langle \mathbf{x}, \mathbf{y} \rangle$ pair. The labels indicate whether a given word belongs to a particular entity class of interest (org and loc in this case, for "organization" and "location," respectively) or not (null).

Unlike simpler classification tasks, each instance $\mathbf{x}$ in this setting is not represented by a single feature vector, but rather a structured sequence of feature vectors: one for each token (i.e., word). For example, the word "Madison" might be described by the features WORD=*Madison* and CAPITALIZED. However, it can variously correspond to the labels person ("The fourth U.S. President James Madison..."), loc ("The city of Madison, Wisconsin..."), and org ("Madison defeated St. Cloud in yesterday's hockey match..."). The appropriate label for a token often depends on its context in the sequence. For sequence-labeling problems like information extraction, labels are typically predicted by a *sequence model*

based on a probabilistic finite state machine, such as CRFs or HMMs. An example sequence model is shown in Figure 7(b).

Settles and Craven (2008) present and evaluate a large number of active learning algorithms for sequence labeling tasks using probabilistic sequence models like CRFs. Most of these algorithms can be generalized to other probabilistic sequence models, such as HMMs (Dagan and Engelson, 1995; Scheffer et al., 2001) and probabilistic context-free grammars (Baldridge and Osborne, 2004; Hwa, 2004). Thompson et al. (1999) also propose query strategies for structured output tasks like semantic parsing and information extraction using inductive logic programming methods.

5.2 Batch-Mode Active Learning

In most active learning research, queries are selected in *serial*, i.e., one at a time. However, sometimes the training time required to induce a model is slow or expensive, as with large ensemble methods and many structured prediction tasks (see Section 5.1). Consider also that sometimes a distributed, parallel labeling environment may be available, e.g., multiple annotators working on different labeling workstations at the same time on a network. In both of these cases, selecting queries in serial may be inefficient. By contrast, *batch-mode* active learning allows the learner to query instances in groups, which is better suited to parallel labeling environments or models with slow training procedures.

The challenge in batch-mode active learning is how to properly assemble the optimal query batch $\mathcal{Q}$. Myopically querying the "N-best" queries according to a given instance-level query strategy often does not work well, since it fails to consider the overlap in information content among the "best" instances. To address this, a few batch-mode active learning algorithms have been proposed. Brinker (2003) considers an approach for SVMs that explicitly incorporates diversity among instances in the batch. Xu et al. (2007) propose a similar approach for SVM active learning, which also incorporates a density measure. Specifically, they query cluster centroids for instances that lie close to the decision boundary. Hoi et al. (2006a,b) extend the Fisher information framework (Section 3.4) to the batch-mode setting for binary logistic regression. Most of these approaches use greedy heuristics to ensure that instances in the batch are both diverse and informative, although Hoi et al. (2006b) exploit the properties of submodular functions to find near-optimal batches. Alternatively, Guo and Schuurmans (2008) treat batch construction for logistic regression as a discriminative optimization problem, and attempt to construct the most informative batch directly. For the most

part, these approaches show improvements over random batch sampling, which in turn is generally better than simple "N-best" batch construction.

5.3 Active Learning With Costs

In some learning problems, the cost of acquiring labeled data can vary from one instance to the next. If our goal in active learning is to minimize the overall cost of training an accurate model, then reducing the number of labeled instances does not necessarily guarantee a reduction in overall labeling cost. One proposed approach for reducing annotation effort in active learning involves using the current trained model to assist in the labeling of query instances by pre-labeling them in structured learning tasks like parsing (Baldridge and Osborne, 2004) or information extraction (Culotta and McCallum, 2005). However, such methods do not actually represent or reason about labeling costs. Instead, they attempt to reduce cost indirectly by minimizing the number of annotation actions required for a query that has already been selected.

Another group of cost-sensitive active learning approaches explicitly accounts for varying label costs in active learning. Kapoor et al. (2007) propose an approach called *value of information* (VOI) that takes into account both labeling costs and estimated misclassification costs. In this setting, each candidate query is evaluated by summing the labeling cost for the instance and the expected future misclassification costs that would be incurred if the instance were added to the training set. Instead of using real costs, however, their experiments make the simplifying assumption that the cost of labeling a voicemail message is a linear function of its length (e.g., ten cents per second). King et al. (2004) use a similar active learning approach in an attempt to reduce actual labeling costs. They describe a "robot scientist" which can execute a series of autonomous biological experiments to discover metabolic pathways, with the objective of minimizing the cost of materials used (i.e., the cost of an experiment plus the expected total cost of future experiments until the correct hypothesis is found).

In the settings above, however, the cost of annotating an instance is assumed to be fixed and known to the learner before querying. Settles et al. (2008a) propose a novel approach to cost-sensitive active learning in settings where annotation costs are variable and *not* known, e.g., when the labeling cost is a function of elapsed annotation time. They learn a regression cost-model alongside the active task-model which tries to predict the real (unknown) annotation cost based on a few simple "meta features" on the instances. An analysis of four data sets using real-world human annotation costs reveals the following:

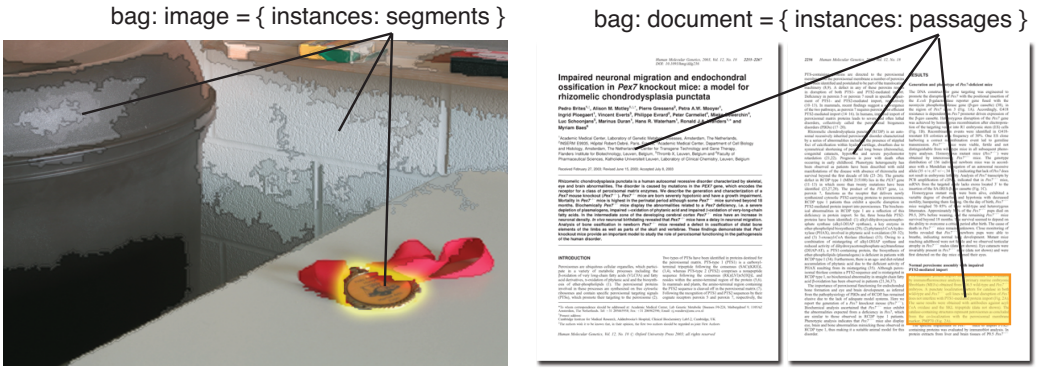
- In some domains, annotation costs are not (approximately) constant across instances, and can vary considerably.
- Consequently, active learning approaches which ignore cost may perform no better than random selection (i.e., passive learning).
- The cost of annotating an instance may not be intrinsic, but may instead vary based on the person doing the annotation.
- The measured cost for an annotation may include stochastic components. In particular, there are at least two types of noise: *jitter* (minor variations due to annotator fatigue, latency, etc.) and *pause* (major variations that should be shorter under normal circumstances).
- In some domains, unknown annotation costs can be accurately predicted, even after seeing only a few training instances. Moreover, these learned cost-models are significantly more accurate than simple cost heuristics (e.g., a linear function of document length).

Empirical experiments show that learned cost-models can predict annotation times accurately, however further work is warranted to determine how such approximate, predicted labeling costs can be utilized effectively by cost-sensitive active learning systems. Settles et al. (2008a) show that simply dividing informativeness (e.g., uncertainty sampling or QBC scores) by cost is not an effective cost-reducing strategy for information extraction tasks. However, results from Haertel et al. (2008) suggest that this simple approach, which they call *return on investment* (ROI), can be effective for part-of-speech tagging, although like most previous work they use fixed heuristic cost models.

5.4 Alternative Query Types

To date, most work in active learning has assumed that a query unit is of the same type as the target concept to be learned. For example, if the task is to assign class labels to documents, the learner must query a document and obtains its label. What other forms might a query take?

Settles et al. (2008b) introduce an alternative query scenario in the context of *multiple-instance active learning*. In multiple-instance (MI) learning, instances are grouped into *bags* (i.e., multi-sets), and it is the bags, rather than instances, that are labeled for training. A bag is labeled negative if and only if all of its



(a) (b)

Figure 8: Multiple-instance active learning. (a) In content-based image retrieval, images are represented as bags and instances correspond to segmented image regions. An active MI learner may query which segments belong to the object of interest, such as the gold medal shown in this image. (b) In text classification, documents are bags and the instances represent passages of text. In MI active learning, the learner may query specific passages to determine if they are representative of the positive class at hand.

instances are negative. A bag is labeled positive, however, if at least one of its instances is positive (note that positive bags may also contain negative instances). The MI setting was formalized by Dietterich et al. (1997) in the context of drug activity prediction, and has since been applied to a wide variety of tasks including content-based image retrieval (Maron and Lozano-Perez, 1998; Andrews et al., 2003; Rahmani and Goldman, 2006) and text classification (Andrews et al., 2003; Ray and Craven, 2005).

Figure 8 illustrates how the MI representation can be applied to (a) content-based image retrieval (CBIR) and to (b) text classification. For the CBIR task, images are represented as bags and instances correspond to segmented regions of the image. A bag representing a given image is labeled positive if the image contains some object of interest. The MI paradigm is well suited to this task because only a few regions of an image may represent the object of interest, such as the gold medal in Figure 8(a). An advantage of the MI representation here is that it is significantly easier to label an entire image than it is to label each segment, or even a subset of the image segments. For the text classification task, documents can be represented as bags and instances correspond to short passages (e.g., paragraphs)

that comprise each document. The MI representation is compelling for classification tasks for which document labels are freely available or cheaply obtained (e.g., from online indexes and databases), but the target concept is represented by only a few passages.

For MI learning tasks such as these, it is possible to obtain labels both at the bag level and directly at the instance level. Fully labeling all instances, however, is expensive. Often the rationale for formulating the learning task as an MI problem is that it allows us to take advantage of coarse labelings that may be available at low cost, or even for free. In MI active learning, however, the learner is sometimes allowed to query for labels at a finer granularity than the target concept, e.g., querying passages rather than entire documents, or segmented image regions rather than entire images. Settles et al. (2008b) focus on this type of active learning with a generalization of logistic regression. Vijayanarasimhan and Grauman (2009) have extended the idea to SVMs for the image retrieval task, and also explore an approach that interleaves queries at varying levels of granularity.

Raghavan et al. (2006) have proposed a related idea for traditional classification problems called *tandem learning*, in which the learner is allowed to query for the labels of *features* as well as entire instances. They report not only that interleaving document-level and word-level queries are very effective for a text classification problem, but also that words (features) are often much easier for human annotators to label in user studies.

6 Related Research Areas

Research in active learning is driven by two key ideas: (i) the learner should be allowed to ask questions, and (ii) unlabeled data are often readily available or easily obtained. There are a few related research areas with rich literature as well.

6.1 Semi-Supervised Learning

Active learning and *semi-supervised learning* (for a good introduction, see Zhu, 2005b) both traffic in making the most out of unlabeled data. As a result, there are a few conceptual overlaps between the two areas that are worth considering. For example, a very basic semi-supervised technique is self-training (Yarowsky, 1995), in which the learner is first trained with a small amount of labeled data, and then used to classify the unlabeled data. Typically the *most* confident unlabeled instances, together with their predicted labels, are added to the training set, and

the process repeats. A complementary technique in active learning is uncertainty sampling (see Section 3.1), where the instances about which the model is *least* confident are selected for querying.

Similarly, multi-view learning (de Sa, 1994) and co-training (Blum and Mitchell, 1998) use ensemble methods for semi-supervised learning. Initially, separate models are trained with the labeled data (usually using separate, conditionally independent feature sets), which then classify the unlabeled data, and "teach" the other models with a few unlabeled examples (with predicted labels) about which they are most confident. This helps to reduce the size of the version space, i.e., the models must agree on the unlabeled data as well as the labeled data. Query-by-committee (see Section 3.2) is an active learning compliment here, as the committee represents different parts of the version space, and is used to query the unlabeled instances about which they do *not* agree.

Through these illustrations, we begin to see that active learning and semi-supervised learning attack the same problem from opposite directions. While semi-supervised learning exploits what the learner thinks it already knows about the unlabeled data, active learning attempts to explore the unknown aspects. It is therefore natural to think about combining the two. Some example formulations of semi-supervised active learning include McCallum and Nigam (1998), Muslea et al. (2000), Zhu et al. (2003), Zhou et al. (2004), and Tur et al. (2005).

6.2 Reinforcement Learning

In *reinforcement learning* (Sutton and Barto, 1998), the learner interacts with the world via "actions," and tries to find an optimal policy of behavior with respect to "rewards" it receives from the environment. For example, consider a machine that is learning how to play chess. In a supervised setting, one might provide the learner with board configurations from a database of chess games along with labels indicating which moves ultimately resulted in a win or loss. In a reinforcement setting, however, the machine actually plays the game against real or simulated opponents (Baxter et al., 2001). Each board configuration (state) allows for certain moves (actions), which result in rewards that are positive (e.g., capturing the opponent's queen) or negative (e.g., having its own queen taken). The learner aims to improve as it plays more games.

The relationship with active learning is that, in order to perform well, the learner must be proactive. It is easy to converge on a policy of actions that have worked well in the past but are sub-optimal or inflexible. In order to improve, a reinforcement learner must take risks and try out actions for which it is uncer-

tain about the outcome, just as an active learner requests labels for instances it is uncertain how to label. This is often called the "exploration-exploitation" trade-off in the reinforcement learning literature. Furthermore, Mihalkova and Mooney (2006) consider an explicitly active reinforcement learning approach with aims to reduce the number of actions required to find an optimal policy.

6.3 Equivalence Query Learning

Another closely related research area is learning with *equivalence queries* (Angluin, 1988). Similar to membership query learning (see Section 2.1), here the learner is allowed to synthesize queries de novo. However, instead of generating an *instance* to be labeled by the oracle, the learner instead generates a *hypothesis* of the target concept class, and the oracle either confirms or denies that the hypothesis is correct. If it is incorrect, the oracle should provide a counter-example, i.e., an instance that would be labeled differently by the true concept and the query hypothesis.

There seem to be few practical applications of equivalence query learning, because the oracle often does not know (or cannot provide) an exact description of the concept class for real-world problems. Otherwise, it would be sufficient to create an "expert system" by hand and machine learning is not required. However, it is an interesting intellectual exercise, and learning from combined membership and equivalence queries is in fact the basis of a popular inductive logic game called Zendo[2].

6.4 Active Class Selection

Active learning assumes that instances are freely or inexpensively obtained, and it is the *labeling* process that incurs a cost. Imagine the opposite scenario, however, where a learner is allowed to query a known class label, and obtaining each *instance* incurs a cost. This fairly new problem setting is known as *active class selection*. Lomasky et al. (2007) propose several active class selection query algorithms for an "artificial nose" task, in which a machine learns to discriminate between different vapor types (the class labels) which must be chemically synthesized (to generate the instances). Some of their approaches show significant gains over uniform class sampling, the "passive" learning equivalent.

[2] http://www.wunderland.com/icehouse/Zendo/

6.5 Active Feature Acquisition and Classification

In some learning domains, instances may have incomplete feature descriptions. For example, many data mining tasks in modern business are characterized by naturally incomplete customer data, due to reasons such as data ownership, client disclosure, or technological limitations. Consider a credit card company that wishes to model its most profitable customers; the company has access to data on client transactions using their own cards, but no data on transactions using cards from other companies. Here, the task of the model is to classify a customer using incomplete purchase information as the feature set. Similarly, consider a learning model used in medical diagnosis which has access to some patient symptom information, but not other symptoms that require complex or expensive procedures. Here, the task of the model is to suggest a diagnosis using incomplete symptom information as the feature set.

In these domains, *active feature acquisition* (Zheng and Padmanabhan, 2002; Melville et al., 2004) seeks to alleviate these problems by allowing the learner to request more complete feature information. The assumption is that some additional features can be obtained at a cost, such as leasing transaction records from other credit card companies, or running additional diagnostic procedures. The goal in active feature acquisition is to select the most informative features to obtain, rather than randomly or exhaustively acquiring all new features for all training instances. The difference between this learning setting and typical active learning is that these models request salient feature values rather than instance labels. Similarly, work in *active classification* (Greiner et al., 2002) considers the case in which features may be obtained during classification rather than training.

6.6 Model Parroting and Compression

Different machine learning algorithms possess different properties. In some cases, it is desirable to induce a model using one type of model class, and then "transfer" that model's knowledge to a model of a different class with another set of properties. For example, artificial neural networks have been shown to achieve better generalization accuracy than decision trees for many applications. However, decision trees represent symbolic hypotheses of the learned concept, and are therefore much more comprehensible to humans, who can inspect the logical rules and understand what the model has learned. Craven and Shavlik (1996) propose the TREPAN (Trees Parroting Networks) algorithm to extract highly accurate decision trees from trained artificial neural networks (or other incomprehensible

model classes), providing comprehensible, symbolic interpretations. Buciluǎ et al. (2006) have adapted this idea to "compress" very large and computationally expensive model classes, such as complex ensembles, into smaller and more efficient model classes, such as neural networks.

These approaches can be thought of as active learning methods where the oracle is in fact another machine learning model (i.e., the one being parroted or compressed) rather than, say, a human annotator. In both cases, the "oracle model" can be trained using a small set of the available labeled data, and the "parrot model" is allowed to query the the oracle model for (i) the labels of any unlabeled data that is available, or (ii) synthesize new instances de novo. These two model parroting and compression approaches correspond to the pool-based and membership query scenarios for active learning, respectively.

Acknowledgements

I would like to thank Mark Craven, Jerry Zhu, Jude Shavlik, David Page, Andrew McCallum, Rong Jin, John Langford, Aron Culotta, Greg Druck, Steve Hanneke, Robbie Haertel, Ashish Kapoor, Clare Monteleoni, and the other colleagues who have discussed active learning with me, both online and in person.

References

N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1–9. Morgan Kaufmann, 1998.

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 561–568. MIT Press, 2003.

D. Angluin. Queries revisited. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 12–31. Springer-Verlag, 2001.

D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

M.F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 65–72. ACM Press, 2006.

M.F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 45–56. Springer, 2008.

J. Baldridge and M. Osborne. Active learning and the total cost of annotation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9–16. ACL Press, 2004.

E.B. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1992.

J. Baxter, A. Tridgell, and L. Weaver. Reinforcement learning and chess. In J. Furnkranz and M. Kubat, editors, *Machines that Learn to Play Games*, pages 91–116. Nova Science Publishers, 2001.

A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 92–100. Morgan Kaufmann, 1998.

C. Bonwell and J. Eison. *Active Learning: Creating Excitement in the Classroom*. Jossey-Bass, 1991.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 59–66. AAAI Press, 2003.

C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 535–541. ACM Press, 2006.

N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni. Worst-case analysis of selective sampling for linear-threshold algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 233–240. MIT Press, 2005.

D. Cohn. Neural network exploration using optimal experiment design. In *Advances in Neural Information Processing Systems (NIPS)*, volume 6, pages 679–686. Morgan Kaufmann, 1994.

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

D. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

C. Cortes and V.N. Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

M. Craven and J. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 8, pages 24–30. MIT Press, 1996.

A. Culotta and A. McCallum. Reducing labeling effort for stuctured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 746–751. AAAI Press, 2005.

I. Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 150–157. Morgan Kaufmann, 1995.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, pages 337–344. MIT Press, 2004.

S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 249–263. Springer, 2005.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 353–360. MIT Press, 2008.

V.R. de Sa. Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 6, pages 112–119. MIT Press, 1994.

T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

V. Federov. *Theory of Optimal Experiments*. Academic Press, 1972.

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective samping using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

R. Gilad-Bachrach, A. Navot, and N. Tishby. Query by committee made real. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, pages 443–450. MIT Press, 2006.

R. Greiner, A. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139:137–174, 2002.

Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 823–829. AAAI Press, 2007.

Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Advances in Neural Information Processing Systems (NIPS)*, number 20, pages 593–600. MIT Press, Cambridge, MA, 2008.

R. Haertel, K. Seppi, E. Ringger, and J. Carroll. Return on investment for active learning. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 353–360. ACM Press, 2007.

A. Hauptmann, W. Lin, R. Yan, J. Yang, and M.Y. Chen. Extreme video retrieval: joint maximization of human and computer performance. In *Proceedings of the ACM Workshop on Multimedia Image Retrieval*, pages 385–394. ACM Press, 2006.

D. Haussler. Learning conjunctive concepts in structural domains. *Machine Learning*, 4(1):7–40, 1994.

S.C.H. Hoi, R. Jin, and M.R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the International Conference on the World Wide Web*, pages 633–642. ACM Press, 2006a.

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 International Conference on Machine Learning (ICML)*, pages 417–424. ACM Press, 2006b.

R. Hwa. Sample selection for statistical parsing. *Computational Linguistics*, 30 (3):73–77, 2004.

A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning,. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 877–882. AAAI Press, 2007.

R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–52, 2004.

V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 282–289. Morgan Kaufmann, 2001.

K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 331–339. Morgan Kaufmann, 1995.

D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 148–156. Morgan Kaufmann, 1994.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. ACM/Springer, 1994.

M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54(2):125–152, 2004.

Y. Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, 44:1936–1941, 2004.

R. Lomasky, C.E. Brodley, M. Aernecke, D. Walt, and M. Friedl. Active class selection. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 640–647. Springer, 2007.

D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 10, pages 570–576. MIT Press, 1998.

A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 359–367. Morgan Kaufmann, 1998.

P. Melville and R. Mooney. Diverse ensembles for active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 584–591. Morgan Kaufmann, 2004.

P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*, pages 483–486. IEEE Press, 2004.

L. Mihalkova and R. Mooney. Using active relocation to aid reinforcement learning. In *Proccedings of the Florida Artificial Intelligence Research Society (FLAIRS)*, pages 580–585. AAAI Press, 2006.

T. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

C. Monteleoni. *Learning with Online Constraints: Shifting Concepts and Active Learning*. PhD thesis, Massachusetts Institute of Technology, 2006.

R. Moskovitch, N. Nissim, D. Stopel, C. Feher, R. Englert, and Y. Elovici. Improving the detection of unknown computer worms activity using active learning. In *Proceedings of the German Conference on AI*, pages 489–493. Springer, 2007.

I. Muslea, S. Minton, and C.A. Knoblock. Selective sampling with redundant views. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 621–626. AAAI Press, 2000.

H.T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 79–86. ACM Press, 2004.

G. Paass and J. Kindermann. Bayesian query construction for neural network models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 7, pages 443–450. MIT Press, 1995.

H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

R. Rahmani and S.A. Goldman. MISSL: Multiple-instance semi-supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 705–712. ACM Press, 2006.

S. Ray and M. Craven. Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 697–704. ACM Press, 2005.

N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 441–448. Morgan Kaufmann, 2001.

T. Scheffer, C. Decomain, and S. Wrobel. Active hidden Markov models for information extraction. In *Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA)*, pages 309–318. Springer-Verlag, 2001.

A.I. Schein and L.H. Ungar. Active learning for logistic regression: An evaluation. *Machine Learning*, 68(3):235–265, 2007.

M.J. Schervish. *Theory of Statistics*. Springer, 1995.

B. Settles. *Curious Machines: Active Learning with Structured Instances*. PhD thesis, University of Wisconsin–Madison, 2008.

B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078. ACL Press, 2008.

B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1–10, 2008a.

B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 1289–1296. MIT Press, 2008b.

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

C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

R. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

C.A. Thompson, M.E. Califf, and R.J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 406–414. Morgan Kaufmann, 1999.

S. Tong. *Active Learning: Theory and Applications*. PhD thesis, Stanford University, 2001.

S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 107–118. ACM Press, 2001.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 999–1006. Morgan Kaufmann, 2000.

G. Tur, D. Hakkani-Tür, and R.E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.

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

V.N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

S. Vijayanarasimhan and K. Grauman. Multi-level active prediction of useful image annotations for recognition. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21. MIT Press, 2009.

Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the European Conference on IR Research (ECIR)*, pages 246–257. Springer-Verlag, 2007.

R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *Proceedings of the International Conference on Computer Vision*, pages 516–523. IEEE Press, 2003.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 189–196. ACL Press, 1995.

H. Yu. SVM selective sampling for ranking with application to data retrieval. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 354–363. ACM Press, 2005.

C. Zhang and T. Chen. An active learning framework for content based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, 2002.

T. Zhang and F.J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1191–1198. Morgan Kaufmann, 2000.

Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*, pages 562–569. IEEE Press, 2002.

Z.H. Zhou, K.J. Chen, and Y. Jiang. Exploiting unlabeled data in content-based image retrieval. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 425–435. Springer, 2004.

X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005a.

X. Zhu. Semi-supervised learning literature survey. Computer Sciences Technical Report 1530, University of Wisconsin–Madison, 2005b.

X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the ICML Workshop on the Continuum from Labeled to Unlabeled Data*, pages 58–65, 2003.

Index