Active Learning Literature Survey

Burr Settles

Computer Sciences Technical Report 1648
University of Wisconsin–Madison
Updated on: January 26, 2010

Abstract

The key idea behind *active learning* is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose *queries*, usually in the form of unlabeled data 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 or easily obtained, 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 successful active learning, a summary of problem setting variants and practical issues, 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://active-learning.net/

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). I have strived to make this review as comprehensive as possible, but it is by no means complete. My own 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 rapidly growing and evolving in a myriad of directions, so it is difficult for one person to provide an exhaustive summary. I apologize for any oversights or inaccuracies, and encourage interested readers to submit additions, comments, and corrections to me at: bsettles@cs.cmu.edu.

1.1 What is Active Learning?

Active learning (sometimes called "query learning" or "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 a half-hour or more 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 usually 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

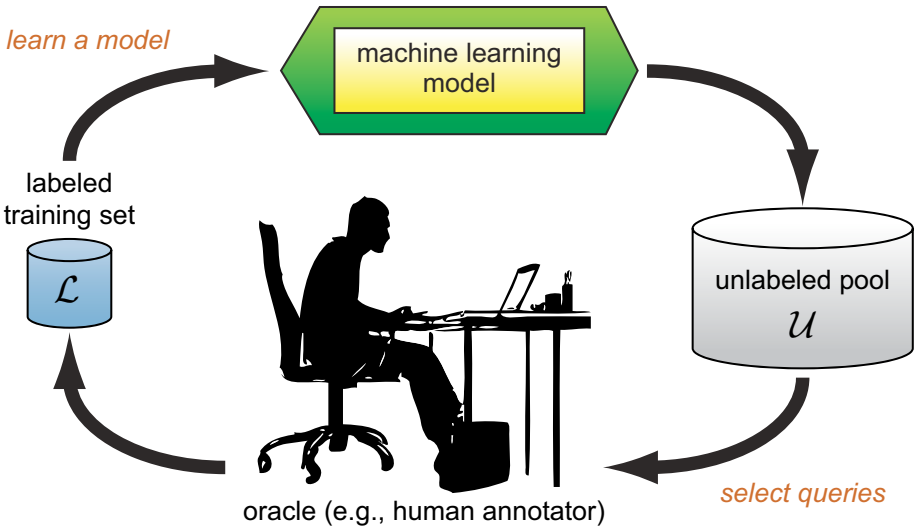


Figure 1: The pool-based active learning cycle.

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.

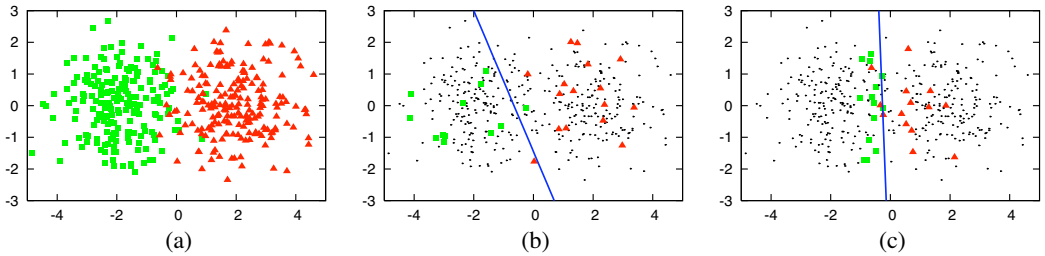


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain. The line represents the decision boundary of the classifier (70% accuracy). (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling (90%).

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 6.4), or when active learning is combined with semi-supervised learning (Section 7.1).

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

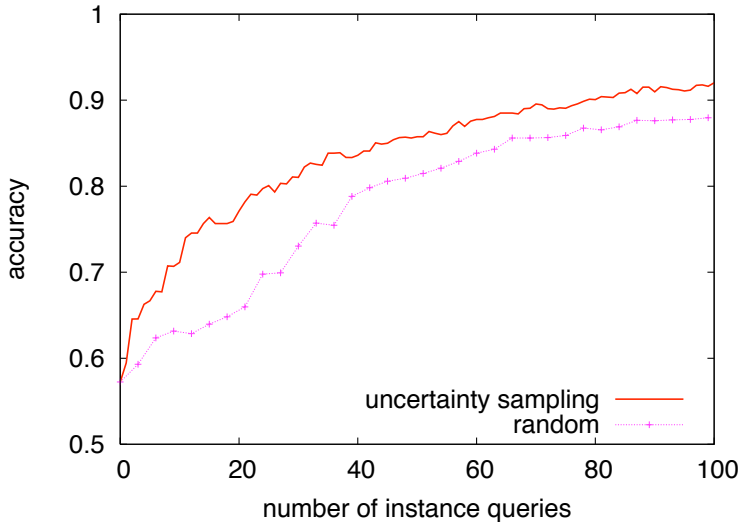


Figure 3: Learning curves for text classification: baseball vs. hockey. Curves plot classification accuracy as a function of the number of documents queried for two selection strategies: uncertainty sampling (active learning) and random sampling (passive learning). We can see that the active learning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

axis, which is where the Bayes optimal decision boundary should probably be. As a result, this classifier only achieves 70% accuracy on the remaining unlabeled points. Figure 2(c), however, tells a 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 90% accuracy with a mere 30 labeled instances.

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 81%, while the random baseline is only 73%. 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 like traditional passive supervised learning) if it dominates the other for most or all of the points along their learning curves.

1.3 Further Reading

This is the first 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, Monteleoni (2006), which considers more theoretical aspects of active learning for classification, and Olsson (2008), which focuses on active learning for named entity recognition (a type of information extraction). Fredrick Olsson has also written a survey of active learning specifically within the scope of the natural language processing (NLP) literature (Olsson, 2009).

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 sampling. Figure 4 illustrates the differences among these three scenarios, which are explained in more detail in this section. Note that all these scenarios (and the lion's share of active learning work to date) assume that queries take the form of unlabeled instances to be labeled by the oracle. Sections 6 and 5 discuss some alternatives to this setting.

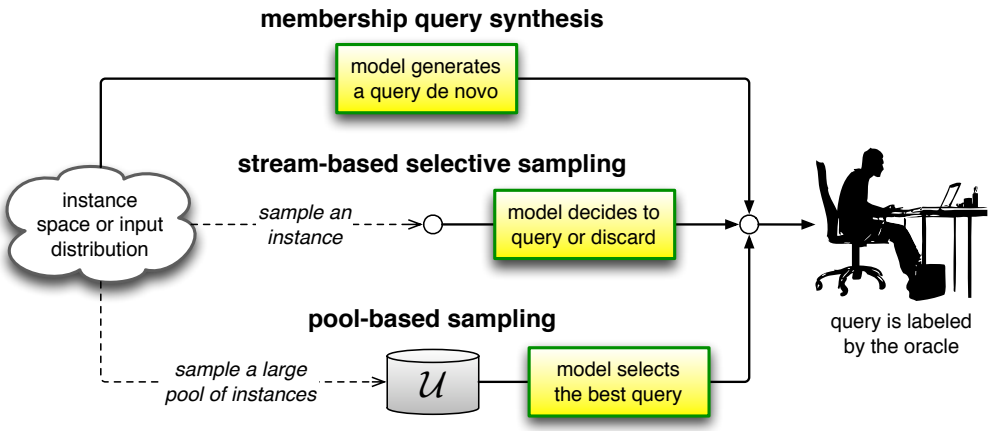


Figure 4: Diagram illustrating the three main active learning scenarios.

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, Lang and Baum (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, 2009) describe an innovative and promising real-world application of the membership query scenario. They employ a "robot scien-

tist" 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., 1990, 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 Sampling

For many real-world learning problems, large collections of unlabeled data can be gathered at once. This motivates *pool-based sampling* (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 (Tür 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 that are used. From this point on, I use the notation x_A^* to refer to the most informative instance (i.e., the best 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, uncertainty sampling simply queries the instance whose posterior probability of being positive is nearest 0.5 (Lewis and Gale, 1994; Lewis and Catlett, 1994).

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

$$x_{LC}^* = \underset{x}{\operatorname{argmax}}\ 1 - P_\theta(\hat{y}|x),$$

where $\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$, or the class label with the highest posterior probability under the model θ. One way to interpret this uncertainty measure is the expected 0/1-loss, i.e., the model's belief that it will mislabel x. This sort of strategy has been popular, for example, with statistical sequence models in information

extraction tasks (Culotta and McCallum, 2005; Settles and Craven, 2008). This is because the most likely label sequence (and its associated likelihood) can be efficiently computed using dynamic programming.

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

$$x_M^* = \operatorname*{argmin}_x P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x),$$

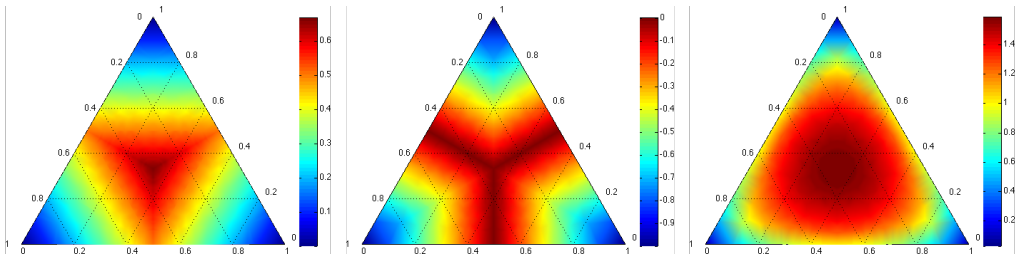
where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable class labels under the model, respectively. Margin sampling aims to correct for a shortcoming in least confident strategy, by incorporating the posterior of the second most likely label. Intuitively, instances with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Instances with small margins are more ambiguous, thus knowing the true label would help the model discriminate more effectively between them. However, for problems with very large label sets, the margin approach still ignores much of the output distribution for the remaining classes.

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

$$x_H^* = \operatorname*{argmax}_x -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x),$$

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

Figure 5 visualizes the implicit relationship among these uncertainty measures. In all cases, the *most* informative instance would lie at the center of the triangle, because this represents where the posterior label distribution is most uniform (and thus least certain under the model). Similarly, the *least* informative instances are at the three corners, where one of the classes has extremely high



(a) least confident (b) margin (c) entropy

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

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

Uncertainty sampling strategies may also be employed with non-probabilistic classifiers. One of the first works to explore uncertainty sampling used a decision tree classifier (Lewis and Catlett, 1994). Similar approaches have been applied to active learning with nearest-neighbor (a.k.a. "memory-based" or "instance-based") classifiers (Fujii et al., 1998; Lindenbaum et al., 2004), by allowing each neighbor to vote on the class label of x, with the proportion of these votes representing the posterior label probability. Tong and Koller (2000) also experiment

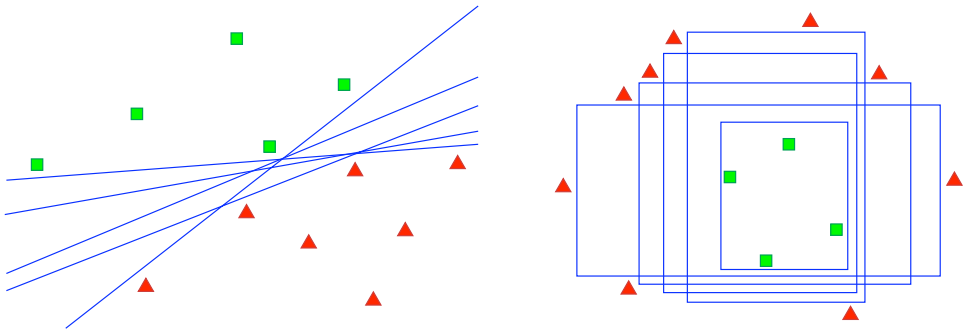
with an uncertainty sampling strategy for support vector machines—or SVMs—that involves querying the instance closest to the linear decision boundary. This last approach is analogous to uncertainty sampling with a probabilistic binary linear classifier, such as logistic regression or naïve Bayes.

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

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 described in Section 2.2) the set of hypotheses that are consistent with the current labeled training data $\mathcal{L}$. Figure 6 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 aims to do, by querying in controversial regions of the input space. In order to implement a QBC selection algorithm, one must:



(a) (b)

Figure 6: 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.

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. Muslea et al. (2000) construct a committee of two models by partitioning the feature space. 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, and C is the committee size. 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:

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

Here $\theta^{(c)}$ represents a particular model in the committee, and $\mathcal{C}$ represents the committee as a whole, thus $P_{\mathcal{C}}(y_i|x) = \frac{1}{C} \sum_{c=1}^{C} P_{\theta^{(c)}}(y_i|x)$ 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. Other information-theoretic approaches like Jensen-Shannon divergence have also been used to measure disagreement (Melville et al., 2005), as well as the other uncertainty sampling measures discussed in Section 3.1, by pooling the model predictions to estimate class posteriors (Körner and Wrobel, 2006). Note also that in the equations above, such posterior estimates are based on committee members that cast "hard" votes for their respective label predictions. They might also cast "soft" votes using their posterior label probabilities, which in turn could be weighted by an estimate of each committee member's accuracy.

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 selective sampling algorithm that uses a committee of two neural networks, 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

margin strategy for SVMs which, as it turns out, attempts to minimize the version space directly. The membership query algorithms of Angluin (1988) and King et al. (2004) can also be interpreted as synthesizing instances de novo that most constrain 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 to serve as a subset approximation.

QBC can also be employed in regression settings, i.e., by measuring disagreement as the variance among the committee members' output predictions (Burbidge et al., 2007). Note, however, that there is no notion of "version space" for models that produce continuous outputs, so the interpretation of QBC in regression settings is a bit different. We can think of $\mathcal{L}$ as constraining the posterior joint probability of predicted output variables *and* the model parameters, $P(Y, \theta|\mathcal{L})$ (note that this applies for both regression and classification tasks). By integrating over a set of hypotheses and identifying queries that lie in controversial regions of the instance space, the learner attempts to collect data that reduces variance over both the output predictions and the parameters of the model itself (as opposed to uncertainty sampling, which focuses only on the output variance of a single hypothesis).

3.3 Expected Model Change

Another general active learning framework uses a decision-theoretic approach, selecting 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 6.4), and has also been applied to probabilistic sequence models like CRFs (Settles and Craven, 2008).

In theory, the EGL strategy can be applied to any learning problem where gradient-based training is used. 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 $\nabla\ell_\theta(\mathcal{L})$ be the gradient of the objective function ℓ with respect to the model parameters θ. Now let $\nabla\ell_\theta(\mathcal{L} \cup \langle x, y\rangle)$ 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:

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

where $\| \cdot \|$ is, in this case, the Euclidean norm of each resulting gradient vector. Note that, at query time, $\|\nabla \ell_\theta(\mathcal{L})\|$ should be nearly zero since ℓ converged at the previous round of training. Thus, we can approximate $\nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \approx \nabla \ell_\theta(\langle x, y_i \rangle)$ for computational efficiency, because training instances are usually 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. Furthermore, the EGL approach can be led astray if features are not properly scaled. That is, the informativeness of a given instance may be over-estimated simply because one of its feature values is unusually large, or the corresponding parameter estimate is larger, both resulting in a gradient of high magnitude. Parameter regularization (Chen and Rosenfeld, 2000; Goodman, 2004) can help control this effect somewhat, and it doesn't appear to be a significant problem in practice.

3.4 Expected Error Reduction

Another decision-theoretic approach aims to measure not how much the model is likely to change, but how much its generalization error is likely to be reduced. The idea it to estimate the expected future error of a model trained using $\mathcal{L} \cup \langle x, y \rangle$ on the remaining unlabeled instances in $\mathcal{U}$ (which is assumed to be representative of the test distribution, and used as a sort of validation set), and query the instance with minimal expected future error (sometimes called *risk*). One approach is to minimize the expected 0/1-loss:

$$x^*_{0/1} = \operatorname*{argmin}_x \sum_i P_\theta(y_i|x) \left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x^{(u)}) \right),$$

where $\theta^{+\langle x, y_i \rangle}$ refers to the the new model after it has been re-trained with the training tuple $\langle x, y_i \rangle$ added to $\mathcal{L}$. Note that, as with EGL in the previous section,

we do not know the true label for each query instance, so we approximate using expectation over all possible labels under the current model θ. The objective here is to reduce the expected total number of incorrect predictions. Another, less stringent objective is to minimize the expected log-loss:

$$x^*_{\log} = \operatorname*{argmin}_x \sum_i P_\theta(y_i|x) \left(-\sum_{u=1}^{U} \sum_j P_{\theta^{+\langle x,y_i \rangle}}(y_j|x^{(u)}) \log P_{\theta^{+\langle x,y_i \rangle}}(y_j|x^{(u)}) \right),$$

which is equivalent to reducing the expected entropy over $\mathcal{U}$. Another interpretation of this strategy is maximizing the expected *information gain* of the query x, or (equivalently) the *mutual information* of the output variables over x and $\mathcal{U}$.

Roy and McCallum (2001) first proposed the expected error reduction framework for text classification using naïve Bayes. Zhu et al. (2003) combined this framework with a semi-supervised learning approach (Section 7.1), resulting in a dramatic improvement over random or uncertainty sampling. Guo and Greiner (2007) employ an "optimistic" variant that biases the expectation toward the most likely label for computational convenience, using uncertainty sampling as a fallback strategy when the oracle provides an unexpected labeling. This framework has the dual advantage of being near-optimal and not being dependent on the model class. All that is required is an appropriate objective 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). In theory, the general approach can be employed not only to minimize loss functions, but to optimize any generic performance measure of interest, such as maximizing precision, recall, F_1-measure, or area under the ROC curve.

In most cases, unfortunately, expected error reduction is also the most computationally expensive query 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 non-parametric model classes such as Gaussian random fields (Zhu et al., 2003), the incremental training procedure is efficient and exact, making this approach fairly practical[1]. For a many other model classes, this is not the case. For example, a binary logistic regression model would require $O(ULG)$ time complexity simply

[1]The bottleneck in non-parametric models generally not re-training, but inference.

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 expected error reduction framework have mostly only considered simple binary classification tasks. Moreover, because the approach is often still impractical, researchers must resort to Monte Carlo sampling from the pool (Roy and McCallum, 2001) to reduce the U term in the previous analysis, or use approximate training techniques (Guo and Greiner, 2007) to reduce the G term.

3.5 Variance Reduction

Minimizing the expectation of a loss function directly is expensive, and in general this cannot be done in closed form. However, we can still reduce generalization error *indirectly* by minimizing output variance, which sometimes does have a closed-form solution. Consider a regression problem, where the learning objective is to minimize standard error (i.e., squared-loss). We can take advantage of the result of Geman et al. (1992), showing that a learner's expected future error can be decomposed in the following way:

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

where $E_{\mathcal{L}}[\cdot]$ is an expectation over the labeled set $\mathcal{L}$, $E[\cdot]$ is an expectation over the conditional density $P(y|x)$, and E_T is an expectation over both. Here also $\hat{y}$ is shorthand for the model's predicted output for a given instance x, while y indicates the true label for that instance.

The first term on the right-hand side of this equation is *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 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 squared-loss with respect to the target 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 (1994) and Cohn et al. (1996) present the first statistical analyses of active learning for regression in the context of a robot arm kinematics problem, using the estimated distribution of the model's output $\sigma^2_{\hat{y}}$. 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 for some instance x can be approximated by (MacKay, 1992):

$$\sigma^2_{\hat{y}}(x) \approx \left[\frac{\partial \hat{y}}{\partial \theta}\right]^{\mathsf{T}} \left[\frac{\partial^2}{\partial \theta^2} S_\theta(\mathcal{L})\right]^{-1} \left[\frac{\partial \hat{y}}{\partial \theta}\right] \approx \nabla x^{\mathsf{T}} F^{-1} \nabla x,$$

where $S_\theta(\mathcal{L})$ is the squared error of the current model θ on the training set $\mathcal{L}$. In the equation above, the first and last terms are computed using the gradient of the model's predicted output with respect to model parameters, written in shorthand as ∇x. The middle term is the inverse of a covariance matrix representing a second-order expansion around the objective function S with respect to θ, written in shorthand as F. This is also known as the *Fisher information matrix* (Schervish, 1995), and will be discussed in more detail later. An expression for $\langle \tilde{\sigma}^2_{\hat{y}} \rangle^{+x}$ can then be derived, which is the estimated mean output variance across the input distribution after the model has been re-trained on query x and its corresponding label. Given the assumptions that the model's prediction for x is fairly good, that ∇x is locally linear (true for most network configurations), and that variance is Gaussian, variance can be estimated efficiently in closed form so that actual model re-training is not required; more gory details are given by Cohn (1994). The *variance reduction* query selection strategy then becomes:

$$x^*_{VR} = \underset{x}{\operatorname{argmin}} \ \langle \tilde{\sigma}^2_{\hat{y}} \rangle^{+x}.$$

Because this equation represents a smooth function that is differentiable with respect to any query instance x in the input space, gradient methods can be used to search for the best possible query that minimizes output variance, and therefore generalization error. Hence, their approach is an example of query synthesis (Section 2.1), rather than stream-based or pool-based active learning.

This sort of approach is derived from statistical theories of *optimal experimental design*, or OED (Federov, 1972; Chaloner and Verdinelli, 1995). A key

ingredient of these approaches is Fisher information, which is sometimes written $\mathcal{I}(\theta)$ to make its relationship with model parameters explicit. Formally, Fisher information is the variance of the *score*, which is the partial derivative of the log-likelihood function with respect to the model parameters:

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

where there are N independent samples drawn from the input distribution. This measure is convenient because its inverse sets a lower bound on the variance of the model's parameter estimates; this result is known as the Cramér-Rao inequality (Cover and Thomas, 2006). In other words, to minimize the variance over its parameter estimates, an active learner should select data that maximizes its Fisher information (or minimizes the inverse thereof). When there is only one parameter in the model, this strategy is straightforward. But for models of K parameters, Fisher information takes the form of a $K \times K$ covariance matrix (denoted earlier as F), and deciding what exactly to optimize is a bit tricky. In the OED literature, there are three types of optimal designs in such cases:

- *A-optimality* minimizes the *trace* of the inverse information matrix,
- *D-optimality* minimizes the *determinant* of the inverse matrix, and
- *E-optimality* minimizes the maximum *eigenvalue* of the inverse matrix.

E-optimality doesn't seem to correspond to an obvious utility function, and is not often used in the machine learning literature, though there are some exceptions (Flaherty et al., 2006). D-optimality, it turns out, is related to minimizing the expected posterior entropy (Chaloner and Verdinelli, 1995). Since the determinant can be thought of as a measure of volume, the D-optimal design criterion essentially aims to minimize the volume of the (noisy) version space, with boundaries estimated via entropy, which makes it somewhat analogous to the query-by-committee algorithm (Section 3.2).

A-optimal designs are considerably more popular, and aim to reduce the *average* variance of parameter estimates by focusing on values along the diagonal of the information matrix. A common variant of A-optimal design is to minimize $\mathrm{tr}(AF^{-1})$—the trace of the product of A and the inverse of the information matrix F—where A is a square, symmetric "reference" matrix. As a special case, consider a matrix of rank one: $A = \mathbf{c}\mathbf{c}^\mathsf{T}$, where $\mathbf{c}$ is some vector of length

K (i.e., the same length as the model's parameter vector). In this case we have $\text{tr}(AF^{-1}) = \mathbf{c}^\mathsf{T} F^{-1}\mathbf{c}$, and minimizing this value is sometimes called c-optimality. Note that, if we let $\mathbf{c} = \nabla x$, this criterion results in the equation for output variance $\sigma^2_{\hat{y}}(x)$ in neural networks defined earlier. Minimizing this variance measure can be achieved by simply querying on instance x, so the c-optimal criterion can be viewed as a formalism for uncertainty sampling (Section 3.1).

Recall that we are interested in reducing variance across the input distribution (not merely for a single point in the instance space), thus the A matrix should encode the whole instance space. MacKay (1992) derived such solutions for regression with neural networks, while Zhang and Oles (2000) and Schein and Ungar (2007) derived similar methods for classification with logistic regression. Consider letting the reference matrix $A = \mathcal{I}_{\mathcal{U}}(\theta)$, i.e., the Fisher information of the unlabeled pool of instances $\mathcal{U}$, and letting $F = \mathcal{I}_x(\theta)$, i.e., the Fisher information of some query instance x. Using A-optimal design, we can derive the *Fisher information ratio* (Zhang and Oles, 2000):

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

The equation above provides us with a ratio given by the inner product of the two matrices, which can be interpreted as the model's output variance across the input distribution (as approximated by $\mathcal{U}$) that is not accounted for by x. Querying the instance which minimizes this ratio is then analogous to minimizing the future output variance once x has been labeled, thus indirectly reducing generalization error (with respect to $\mathcal{U}$). The advantage here over error reduction (Section 3.4) is that the model need not be retrained: the information matrices give us an approximation of output variance that simulates retraining. Zhang and Oles (2000) and Schein and Ungar (2007) applied this sort of approach to text classification using binary logistic regression. Hoi et al. (2006a) extended this to active text classification in the batch-mode setting (Section 6.1) in which a set of queries $\mathcal{Q}$ is selected all at once in an attempt to minimize the ratio between $\mathcal{I}_{\mathcal{U}}(\theta)$ and $\mathcal{I}_{\mathcal{Q}}(\theta)$. Settles and Craven (2008) have also generalized the Fisher information ratio approach to probabilistic sequence models such as CRFs.

There are some practical disadvantages to these variance-reduction methods, however, in terms of computational complexity. Estimating output 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 processing tasks. Paass

and Kindermann (1995) propose a sampling approach based on Markov chains to reduce the U term in this analysis. For inverting the Fisher information matrix and reducing the K^3 term, 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.

3.6 Density-Weighted Methods

A central idea of the estimated error and variance reduction frameworks is that they focus on the entire input space rather than individual instances. Thus, they are less prone to querying outliers than simpler query strategies like uncertainty sampling, QBC, and EGL. Figure 7 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. By utilizing the unlabeled pool $\mathcal{U}$ when estimating future errors and output variances, the estimated error and variance reduction strategies implicitly avoid these problems. We can also overcome these problems by modeling the input distribution explicitly during query selection.

The *information density* framework described by Settles and Craven (2008), and further analyzed in Chapter 4 of Settles (2008), is a general 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 underlying distribution (i.e., inhabit dense regions of the input space). Therefore, we wish to query instances as follows:

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

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

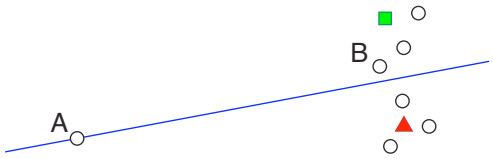


Figure 7: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances in $\mathcal{L}$, and circles represent unlabeled instances in $\mathcal{U}$. Since A is on the decision boundary, it would be queried as the most uncertain. However, querying B is likely to result in more information about the data distribution as a whole.

controls the relative importance of the density term. A variant of this might first cluster $\mathcal{U}$ and compute average similarity to instances in the same cluster.

This formulation was presented 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) least similar to the labeled instances in $\mathcal{L}$, and (ii) most similar to the unlabeled instances in $\mathcal{U}$. Nguyen and Smeulders (2004) 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 6.1) 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). This is advantageous for conducting active learning interactively with oracles in real-time.

4 Analysis of Active Learning

This section discusses some of the empirical and theoretical evidence for how and when active learning approaches can be successful.

4.1 Empirical Analysis

An important question is: "does active learning work?" Most of the empirical results in the published literature suggest that it does (e.g., the majority of papers in the bibliography of this survey). Furthermore, consider that software companies and large-scale research projects such as CiteSeer, Google, IBM, Microsoft, and Siemens are increasingly using active learning technologies in a variety of real-world applications[2]. Numerous published results and increased industry adoption seem to indicate that active learning methods have matured to the point of practical use in many situations.

As usual, however, there are caveats. In particular, consider that a training set built in cooperation with an active learner is inherently tied to the model that was used to generate it (i.e., the class of the model selecting the queries). Therefore, the labeled instances are a biased distribution, not drawn i.i.d. from the underlying natural density. If one were to change model classes—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 (see Section 6.6 for more discussion on this topic). Somewhat surprisingly, Schein and Ungar (2007) showed that active learning can sometimes require more labeled instances than passive learning even when using the *same* model class, in their case logistic regression. Guo and Schuurmans (2008) found that off-the-shelf query strategies, when myopically employed in a batch-mode setting (Section 6.1) are often much worse than random sampling. Gasperin (2009) reported negative results for active learning in an anaphora resolution task. Baldridge and Palmer (2009) found a curious inconsistency in how well active learning helps that seems to be correlated with the proficiency of the annotator (specifically, a domain expert was better utilized by an active learner than a domain novice, who was better suited to a passive learner).

Nevertheless, active learning does reduce the number of labeled instances required to achieve a given level of accuracy in the majority of reported results (though, admittedly, this may be due to the publication bias). This is often true even for simple query strategies like uncertainty sampling. Tomanek and Olsson (2009) report in a survey that 91% of researchers who used active learning in large-scale annotation projects had their expectations fully or partially met. Despite these findings, the survey also states that 20% of respondents opted *not* to use active learning in such projects, specifically because they were "not convinced that [it] would work well in their scenario." This is likely because other

[2]Based on personal communication with (respectively): C. Lee Giles, David "Pablo" Cohn, Prem Melville, Eric Horvitz, and Balaji Krishnapuram.

subtleties arise when using active learning in practice (implementation overhead among them). Section 6 discusses some of the more problematic issues for real-world active learning.

4.2 Theoretical Analysis

A strong 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 a pool-based active learning setting, in which we can acquire the same number of *unlabeled* instances from this distribution for free (or very inexpensively), and only labels incur a cost. 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 discover the location at which the transition occurs while paying for as few labels as possible. 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. Of course, this is a simple, one-dimensional, 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.

There have been some fairly strong 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 (Lang and Baum, 1992) and may result in querying outliers, since they are not created according to the data's underlying natural density. A great many

applications for active learning assume that unlabeled data (drawn from a real distribution) are available, so these results also have limited practical impact.

A stronger early theoretical result in the stream-based and pool-based scenarios is an analysis of the query-by-committee (QBC) algorithm by Freund et al. (1997). They show that, under a Bayesian assumption, 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 can tempered somewhat by the computational complexity of the QBC algorithm in certain practical situations, but Gilad-Bachrach et al. (2006) offer 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 label complexity bounds as reported for QBC. Interestingly, they show that a *standard* perceptron makes a poor active learner in general, requiring $O(1/\epsilon^2)$ labels as a lower bound. The modified training update rule—originally proposed in a non-active setting by Blum et al. (1996)—is key in achieving the exponential savings. The two main differences between QBC and their approach are that (i) QBC is more limited, requiring a Bayesian assumption for the theoretical analysis, and (ii) QBC can be computationally prohibitive, whereas the modified perceptron algorithm is much more lightweight and efficient, even suitable for online learning.

In earlier work, Dasgupta (2004) also provided a variety of theoretical upper and lower bounds for active learning in the more general pool-based setting. In particular, if using linear classifiers the sample complexity can grow to $O(1/\epsilon)$ in the worst case, which offers no improvement over standard supervised learning, but is also no worse. Encouragingly, Balcan et al. (2008) also show that, asymptotically, certain active learning strategies should 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 recent theoretical work in *agnostic active learning* (Balcan et al., 2006), which only requires that unlabeled instances are 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. Dasgupta et al. (2008) propose a somewhat more efficient query selection algorithm, by presenting a polynomial-time reduction from active learning to supervised learning for arbitrary input distributions and model classes. These agnostic active learning approaches explicitly use complexity bounds to determine which hypotheses still "look viable," so to speak, and queries can be assessed by how valuable they are in distinguishing among these viable hypotheses. Methods such as these have attractive PAC-style convergence guarantees and complexity bounds that are, in many cases, significantly better than passive 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 classification problems. In fact, most are limited to binary classification with the goal of minimizing 0/1-loss, and are not easily adapted to other objective functions that may be more appropriate for many applications. Furthermore, some of these methods require an explicit enumeration over the version space, which is not only often intractable (see the discussion at the end of Section 3.2), but difficult to even consider for complex learning models (e.g., heterogeneous ensembles or structured prediction models for sequences, trees, and graphs). However, some recent theoretical work has begun to address these issues, coupled with promising empirical results (Dasgupta and Hsu, 2008; Beygelzimer et al., 2009).

5 Problem Setting Variants

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

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 8 illustrates how, for example, an information extraction problem can be viewed as a sequence labeling task. Let $\mathbf{x} = \langle x_1, \ldots, x_T \rangle$

Figure 8: 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.

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 8(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 8(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 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 data that require complex, expensive, or risky medical procedures. Here, the task of the model is to suggest a diagnosis using incomplete patient information as the feature set.

In these domains, *active feature acquisition* seeks to alleviate these problems by allowing the learner to request more complete feature information. The assumption is that 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 during training, rather than randomly or exhaustively acquiring all new features for all training instances. Zheng and Padmanabhan (2002) proposed two "single-pass" approaches for this problem. In the first approach, they attempt to impute the missing values, and then acquire the ones about which the model has least confidence. As an alternative, they also consider imputing these values, training a classifiers on the imputed training instances, and only acquiring feature values for the instances which are misclassified. In contrast, incremental active feature acquisition may acquire values for a few salient features at a time, either by selecting a small batch of misclassified examples (Melville et al., 2004), or by taking a decision-theoretic approach and acquiring the feature values which are expected to maximize some utility function (Saar-Tsechansky et al., 2009).

Similarly, work in *active classification* considers the case in which missing feature values may be obtained during classification (test time) rather than during training. Greiner et al. (2002) introduced this setting and provided a PAC-style theoretical analysis of learning such classifiers given a fixed budget. Variants of

naïve Bayes (Ling et al., 2004) and decision tree classifiers (Chai et al., 2004; Esmeir and Markovitch, 2008) have also been proposed to minimize costs at classification time. Typically, these are evaluated in terms of their total cost (feature acquisition plus misclassification, which must be converted into the same currency) as a function of the number of missing values. This is often flexible enough to incorporate other types of costs, such as delays between query time and value acquisition (Sheng and Ling, 2006). Another approach is to model the feature acquisition task as a sequence of decisions to either acquire more information or to terminate and make a prediction, using an HMM (Ji and Carin, 2007).

The difference between these learning settings and typical active learning is that the "oracle" provides salient feature values rather than training labels. Since feature values can be highly variable in their acquisition costs (e.g., running two different medical tests might provide roughly the same predictive power, while one is half the cost of the other), some of these approaches are related in spirit to cost-sensitive active learning (see Section 6.3).

5.3 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.

5.4 Active Clustering

For most of this survey, we assume that the learner to be "activized" is *supervised*, i.e., the task of the learner is to induce a function that accurately predicts a label y for some new instance x. In contrast, a learning algorithm is called *unsupervised* if its job is simply to organize a large amount of unlabeled data in a meaningful way. The main difference is that supervised learners try to map instances into a predefined vocabulary of labels, while unsupervised learners exploit latent structure

in the data alone to find meaningful patterns[3]. *Clustering* algorithms are probably the most common examples of unsupervised learning (e.g., see Chapter 10 of Duda et al., 2001).

Since active learning generally aims to select data that will reduce the model's classification error or label uncertainty, unsupervised active learning may seem a bit counter-intuitive. Nevertheless, Hofmann and Buhmann (1998) have proposed an active clustering algorithm for proximity data, based on an expected value of information criterion. The idea is to generate (or subsample) the unlabeled instances in such a way that they self-organize into groupings with less overlap or noise than for clusters induced using random sampling. The authors demonstrate improved clusterings in computer vision and text retrieval tasks.

Some clustering algorithms operate under certain constraints, e.g., a user can specify a priori that two instances *must* belong to the same cluster, or that two others *cannot*. Grira et al. (2005) Have explored an active variant of this approach for image databases, where queries take the form of such "must-link" and "cannot-link" constraints on similar or dissimilar images. Huang and Mitchell (2006) experiment with interactively-obtained clustering constraints on both instances and features, and Andrzejewski et al. (2009) address the analogous problem of incorporating constraints on features in *topic modeling* (Steyvers and Griffiths, 2007), another popular unsupervised learning technique. Although these last two works do not solicit constraints in an active manner, one can easily imagine extending them to do so. Active variants for these unsupervised methods are akin to the work on active learning by labeling features discussed in Section 6.4, with the subtle difference that constraints in the (semi-)supervised case are links between features and *labels*, rather than features (or instances) with one another.

6 Practical Considerations

Until very recently, most active learning research has focused on mechanisms for choosing queries from the learner's perspective. In essence, this body of work addressed the question, "can machines learn with fewer training instances if they ask questions?" By and large, the answer to this question is "yes," subject to some assumptions. For example, we often assume that there is a single oracle, or that the oracle is always correct, or that the cost for labeling queries is either free or uniformly expensive.

[3]Note that *semi-supervised learning* (Section 7.1) also tries to exploit the latent structure of unlabeled data, but with the specific goal of improving label predictions.

In many real-world situations these assumptions do not hold. As a result, the research question for active learning has shifted in recent years to "can machines learn *more economically* if they ask questions?" This section describes several of the challenges for active learning in practice, and summarizes some the research that has addressed these issues to date.

6.1 Batch-Mode Active Learning

In most active learning research, queries are selected in *serial*, i.e., one at a time. However, sometimes the 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 set $\mathcal{Q}$. Myopically querying the "Q-best" queries according to some 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 (Section 3.6). Specifically, they query the centroids of clusters of instances that lie closest to the decision boundary. Hoi et al. (2006a,b) extend the Fisher information framework (Section 3.5) 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 (see Section 7.3) to find batches that are guaranteed to be near-optimal. 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 "Q-best" batch construction.

6.2 Noisy Oracles

Another strong assumption in most active learning work is that the quality of labeled data is high. If labels come from an empirical experiment (e.g., in biological, chemical, or clinical studies), then one can usually expect some noise to result from the instrumentation of experimental setting. Even if labels come from human experts, they may not always be reliable, for several reasons. First, some instances are implicitly difficult for people and machines, and second, people can become distracted or fatigued over time, introducing variability in the quality of their annotations. The recent introduction of Internet-based "crowdsourcing" tools such as Amazon's Mechanical Turk[4] and the clever use of online annotation games[5] have enabled some researchers to attempt to "average out" some of this noise by cheaply obtaining labels from multiple non-experts. Such approaches have been used to produce gold-standard quality training sets (Snow et al., 2008) and also to evaluate learning algorithms on data for which no gold-standard labelings exist (Mintz et al., 2009; Carlson et al., 2010).

The question remains about how to use non-experts (or even noisy experts) as oracles in active learning. In particular, when should the learner decide to query for the (potentially noisy) label of a *new* unlabeled instance, versus querying for repeated labels to de-noise an *existing* training instance that seems a bit off? Sheng et al. (2008) study this problem using several heuristics that take into account estimates of both oracle and model uncertainty, and show that data can be improved by selective repeated labeling. However, their analysis assumes that (i) all oracles are equally and consistently noisy, and (ii) annotation is a noisy process over some underlying true label. Donmez et al. (2009) address the first issue by allowing annotators to have different noise levels, and show that both true instance labels and individual oracle qualities can be estimated (so long as they do not change over time). They take advantage of these estimates by querying only the more reliable annotators in subsequent iterations active learning.

There are still many open research questions along these lines. For example, how can active learners deal with noisy oracles whose quality varies over time (e.g., after becoming more familiar with the task, or after becoming fatigued)? How might the effect of payment influence annotation quality (i.e., if you pay a non-expert twice as much, are they likely to try and be more accurate)? What if some instances are inherently noisy regardless of which oracle is used, and repeated labeling is not likely to improve matters? Finally, in most crowdsourcing

[4] http://www.mturk.com
[5] http://www.gwap.com

environments the users are not necessarily available "on demand," thus accurate estimates of annotator quality may be difficult to achieve in the first place, and might possibly never be applicable again, since the model has no real choice over which oracles to use. How might the learner continue to make progress?

6.3 Variable Labeling Costs

Continuing in the spirit of the previous section, in many applications there is variance not only in label quality from one instance to the next, but also in the *cost* of obtaining that label. If our goal in active learning is to minimize the overall cost of training an accurate model, then simply 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 while selecting queries. Kapoor et al. (2007) propose a decision-theoretic approach that takes into account both labeling costs and misclassification costs. In this setting, each candidate query is evaluated by summing its labeling cost with the future misclassification costs that are expected to 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 an instances is a linear function of its length (e.g., one cent per second for voicemail messages). Furthermore, labeling and misclassification costs must be mapped into the same currency (e.g., \$0.01 per second of annotation and \$10 per misclassification), which may not be appropriate or straightforward for some applications. King et al. (2004) use a similar decision-theoretic approach 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). But here again, the cost of materials is fixed and known at the time of experiment (query) selection.

In all the settings above, and indeed in most of the cost-sensitive active learning literature (e.g., Margineantu, 2005; Tomanek et al., 2007), the cost of annotating an instance is still 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, for example, 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 (Settles et al., 2008a):

- In some domains, annotation costs are not (approximately) constant across instances, and can vary considerably. This result is also supported by the subsequent findings of others, working on different learning tasks (Arora et al., 2009; Vijayanarasimhan and Grauman, 2009a).
- 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. This result is also supported by the findings of Ringger et al. (2008) and Arora et al. (2009).
- The measured cost for an annotation may include stochastic components. In particular, there are at least two types of noise which affect annotation speed: *jitter* (minor variations due to annotator fatigue, latency, etc.) and *pause* (major variations that should be shorter under normal circumstances).
- Unknown annotation costs can *sometimes* be accurately predicted, even after seeing only a few training instances. This result is also supported by the findings of Vijayanarasimhan and Grauman (2009a). Moreover, these learned cost-models are significantly more accurate than simple cost heuristics (e.g., a linear function of document length).

While empirical experiments show that learned cost-models can be trained to predict accurate annotation times, 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. show that simply dividing the informativeness measure (e.g., entropy) by the cost is not necessarily an effective

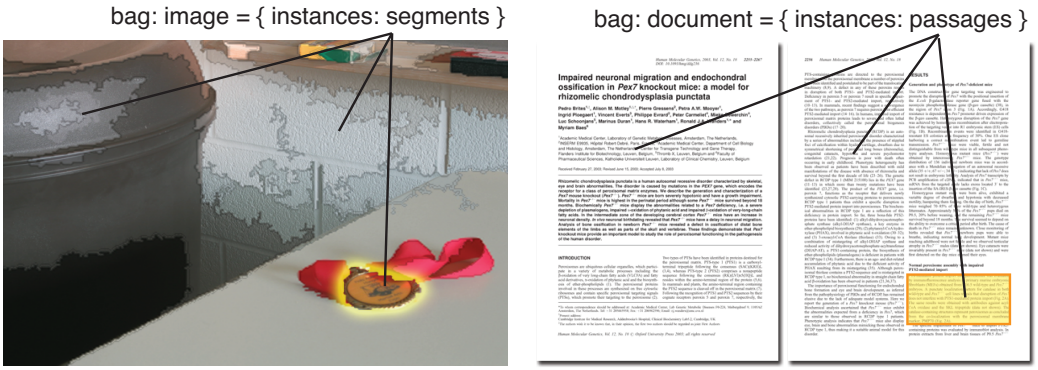
cost-reducing strategy for several natural language tasks when compared to random sampling (even if *true* costs are known). However, results from Haertel et al. (2008) suggest that this heuristic, which they call *return on investment* (ROI), is sometimes effective for part-of-speech tagging, although like most work they use a fixed heuristic cost model. Vijayanarasimhan and Grauman (2009a) also demonstrate potential cost savings in active learning using predicted annotation costs in a computer vision task using a decision-theoretic approach. It is unclear whether these disparities are intrinsic, task-specific, or simply a result of differing experimental assumptions.

Even among methods that do not explicitly reason about annotation cost, several authors have found that alternative query types (such as labeling features rather than instances, see Section 6.4) can lead to reduced annotation costs for human oracles (Raghavan et al., 2006; Druck et al., 2009; Vijayanarasimhan and Grauman, 2009a). Interestingly, Baldridge and Palmer (2009) used active learning for morpheme annotation in a rare-language documentation study, using two live human oracles (one expert and one novice) interactively "in the loop." They found that the best query strategy differed between the two annotators, in terms of reducing both labeled corpus size and annotation costs. The domain expert was a more efficient oracle with an uncertainty-based active learner, but semi-automated annotations—intended to assist in the labeling process—were of little help. The novice, however, was more efficient with a passive learner (selecting passages at random), but semi-automated annotations were in this case beneficial.

6.4 Alternative Query Types

Most work in active learning assumes that a "query unit" is of the same type as the target concept to be learned. In other words, if the task is to assign class labels to text documents, the learner must query a document and the oracle provides 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 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). A naïve approach to MI learning is to view it as supervised learning with one-sided noise (i.e., all negative instances are truly negative, but some positives are actually negative). However, special MI learning algorithms have been developed



(a) (b)

Figure 9: 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.

to learn from labeled bags despite this ambiguity. 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 9 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 9(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 mixed-granularity active learning with a multiple-instance generalization of logistic regression. Vijayanarasimhan and Grauman (2009a,b) have extended the idea to SVMs for the image retrieval task, and also explore an approach that interleaves queries at varying levels of granularity and cost.

Another alternative setting is to query on *features* rather than (or in addition to) instances. Raghavan et al. (2006) have proposed one such approach, *tandem learning*, which can incorporate feature feedback in traditional classification problems. In their work, a text classifier may interleave instance-label queries with feature-salience queries (e.g., "is the word *puck* a discriminative feature for classifying sports documents?"). Values for the salient features are then amplified in instance feature vectors to reflect their relative importance. Raghavan et al. reported that interleaving such queries is very effective for text classification, and also found that words (or features) are often much easier for human annotators to label in empirical user studies. Note, however, that these "feature labels" only imply their discriminative value and do not tie features to class labels directly.

In recent years, several new methods have been developed for incorporating feature-based domain knowledge into supervised and semi-supervised learning (e.g., Haghighi and Klein, 2006; Druck et al., 2008). In this line of work, users may specify a set of constraints between features and labels, e.g., "95% of the time, when the word *puck* is observed in a document, the class label is hockey." The learning algorithm then tries to find a set of model parameters that match expected label distributions over the unlabeled pool $\mathcal{U}$ against these user-specified priors (for details, see Druck et al., 2008; Mann and McCallum, 2008). Interestingly, Mann and McCallum found that specifying many imprecise constraints is more effective than fewer more precise ones, suggesting that human-specified feature labels (however noisy) are useful if there are enough of them. This begs the question of how to *actively* solicit these constraints.

Druck et al. (2009) propose and evaluate a variety of active query strategies aimed at gathering useful feature-label constraints. They show that active feature labeling is more effective than either "passive" feature labeling (using a variety of strong baselines) or instance-labeling (both passive and active) for two information extraction tasks. These results held true for both simulated and interactive human-annotator experiments. Liang et al. (2009) present a more principled approach to the problem, grounded in Bayesian experimental design (see Section 3.5). However, this method is intractable for most real-world problems, and they also resort to heuristics in practice. Sindhwani et al. (2009) have also explored interleaving class-label queries for both instances and features, which they refer to as *active dual supervision*, in a semi-supervised graphical model.

6.5 Multi-Task Active Learning

The typical active learning setting assumes that there is only one learner trying to solve a single task. In many real-world problems, however, the same data instances may be labeled in multiple ways for different subtasks. In such cases, it is likely most economical to label a single instance for all subtasks simultaneously. Therefore, *multi-task active learning* algorithms assume that a single query will be labeled for multiple tasks, and attempt to assess the informativeness of a query with respect to all the learners involved.

Reichart et al. (2008) study a two-task active learning scenario for natural language parsing and named entity recognition (NER), a form of information extraction. They propose two methods for actively learning both tasks in tandem. The first is *alternating selection*, which allows the parser to query sentences in one iteration, and then the NER system to query instances in the next. The second is *rank combination*, in which both learners rank the query candidates in the pool independently, and instances with the highest combined rank are selected for labeling. In both cases, uncertainty sampling is used as the base selection strategy for each learner. As one might expect, these methods outperform passive learning for both subtasks, while learning curves for each individual subtask are not as good as they would have been in the single-task active setting.

Qi et al. (2008) study a different multi-task active learning scenario, in which images may be labeled for several binary classification tasks in parallel. For example, an image might be labeled as containing a beach, sunset, mountain, field, etc., which are not all mutually exclusive; however, they are not entirely independent, either. The beach and sunset labels may be highly correlated, for example, so a simple rank combination might over-estimate the informativeness

of some instances. They propose and evaluate a novel Bayesian approach, which takes into account the mutual information among labels.

6.6 Changing (or Unknown) Model Classes

As mentioned in Section 4.1, a training set built via active learning comes from a biased distribution, which is implicitly tied to the class of the model used in selecting the queries. This can be an issue if we wish to re-use this training data with models of a different type, or if we do not even know the appropriate model class (or feature set) for the task to begin with. Fortunately, this is not always a problem. For example, Lewis and Catlett (1994) showed that decision tree classifiers can still benefit significantly from a training set constructed by an active naïve Bayes learner using uncertainty sampling. Tomanek et al. (2007) also showed that information extraction data gathered by a MaxEnt model using QBC can be effectively re-used to train CRFs, maintaining cost savings compared with random sampling. Hwa (2001) successfully re-used natural language parsing data selected by one type of parser to train other types of parsers.

However, Baldridge and Osborne (2004) encountered the exact opposite problem when re-using data selected by one parsing model to train a variety of other parsers. As an alternative, they perform active learning using a heterogeneous ensemble composed of different parser types, and also use semi-automated labeling to cut down on human annotation effort. This approach helped to reduce the number of training examples required for each parser type compared with passive learning. Similarly, Lu and Bongard (2009) employed active learning with a heterogeneous ensemble of neural networks and decision trees, when the more appropriate model was not known in advance. Their ensemble approach is able to simultaneously select informative instances for the overall model, as well as bias the constituent weak learners toward the more appropriate model class as it learns. Sugiyama and Rubens (2008) have experimented with an ensemble of linear regression models using different feature sets, to study cases in which the appropriate feature set is not yet decided upon.

This section brings up a very important issue for active learning in practice. If the best model class and feature set happen to be known in advance—or if these are not likely to change much in the future—then active learning can probably be safely used. Otherwise, random sampling (at least for pilot studies, until the task can be better understood) may be more advisable than taking one's chances on active learning with an inappropriate learning model. One viable active approach

seems to be the use of heterogeneous ensembles in selecting queries, but there is still much work to be done in this direction.

6.7 Stopping Criteria

A potentially important element of interactive learning applications in general is knowing when to *stop* learning. One way to think about this is the point at which the cost of acquiring new training data is greater than the cost of the errors made by the current model. Another view is how to recognize when the accuracy of a learner has reached a plateau, and acquiring more data is likely a waste of resources. Since active learning is concerned with improving accuracy while remaining sensitive to data acquisition costs, it is natural to think about devising a "stopping criterion" for active learning, i.e., a method by which an active learner may decide to stop asking questions in order to conserve resources.

Several such stopping criteria for active learning have been proposed (Vlachos, 2008; Bloodgood and Shanker, 2009; Olsson and Tomanek, 2009). These methods are all fairly similar, generally based on the notion that there is an intrinsic measure of stability or self-confidence within the learner, and active learning ceases to be useful once that measure begins to level-off or degrade. Such self-stopping methods seem like a good idea, and may be applicable in certain situations. However, in my own experience, the real stopping criterion for practical applications is based on economic or other external factors, which likely come well before an intrinsic learner-decided threshold.

7 Related Research Areas

Research in active learning is driven by two key ideas: (i) the learner should not be strictly passive, and (ii) unlabeled data are often readily available or easily obtained. There are a few related research areas with rich literature as well.

7.1 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, co-training (Blum and Mitchell, 1998) and multi-view learning (de Sa, 1994) 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 (using 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 see that active learning and semi-supervised learning attack the same problem from opposite directions. While semi-supervised methods exploit what the learner thinks it knows about the unlabeled data, active methods attempt to explore the unknown aspects[6]. 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), Tür et al. (2005), Yu et al. (2006), and Tomanek and Hahn (2009).

7.2 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., cap-

[6]One might make the argument that active methods also "exploit" what is known rather than "exploring," by querying about what isn't known. This is a minor semantic issue.

turing 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 uncertain 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 which aims to reduce the number of actions required to find an optimal policy.

7.3 Submodular Optimization

Recently, there has been a growing interest in *submodular functions* (Nemhauser et al., 1978) in machine learning research. Submodularity is a property of set functions that intuitively formalizes the idea of "diminishing returns." That is, adding some instance x to the set $\mathcal{A}$ provides more gain in terms of the target function than adding x to a larger set $\mathcal{A}'$, where $\mathcal{A} \subseteq \mathcal{A}'$. Informally, since $\mathcal{A}'$ is a superset of $\mathcal{A}$ and already contains more information, adding x will not help as much. More formally, a set function F is submodular if it satisfies the property:

$$F(\mathcal{A} \cup \{x\}) - F(\mathcal{A}) \geq F(\mathcal{A}' \cup \{x\}) - F(\mathcal{A}'),$$

or, equivalently:

$$F(\mathcal{A}) + F(\mathcal{B}) \geq F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}),$$

for any two sets $\mathcal{A}$ and $\mathcal{B}$. The key advantage of submodularity is that, for monotonically non-decreasing submodular functions where $F(\emptyset) = 0$, a greedy algorithm for selecting N instances guarantees a performance of $(1 - 1/e) \times F(\mathcal{S}_N^*)$, where $F(\mathcal{S}_N^*)$ is the value of the optimal set of size N. In other words, using a greedy algorithm to optimize a submodular function gives us a lower-bound performance guarantee of around 63% of optimal; in practice these greedy solutions are often within 90% of optimal (Krause, 2008).

In learning settings where there is a fixed budget on gathering data, it is advantageous to formulate (or approximate) the objective function for data selection as a submodular function, because it guarantees near-optimal results with signif-

icantly less computational effort[7]. The relationship to active learning is simple: both aim to maximize some objective function while minimizing data acquisition costs (or remaining within a budget). Active learning strategies do not optimize to submodular functions in general, but Guestrin et al. (2005) show that maximizing mutual information among sensor locations using Gaussian processes (analogous to active learning by expected error reduction, see Section 3.4) can be approximated with a submodular function. Similarly, Hoi et al. (2006b) formulate the Fisher information ratio criterion (Section 3.5) for binary logistic regression as a submodular function, for use with batch-mode active learning (Section 6.1).

7.4 Equivalence Query Learning

An area closely related to active learning is learning with *equivalence queries* (Angluin, 1988). Similar to membership query learning (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 (or any other kind of learning constraint), 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 an oracle often does not know (or cannot provide) an exact description of the concept class for most 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[8].

7.5 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

[7] Many interesting set optimization problems are NP-hard, and can thus scale exponentially. So greedy approaches are usually more efficient.

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

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) proposed the TREPAN (Trees Parroting Networks) algorithm to extract highly accurate decision trees from trained artificial neural networks (or similarly opaque model classes, such as ensembles), providing comprehensible, symbolic interpretations. Several others (Buciluǎ et al., 2006; Liang et al., 2008) have adapted this idea to "compress" large, computationally expensive model classes (such as complex ensembles or structured-output models) into smaller, more efficient model classes (such as neural networks or simple linear classifiers).

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.

8 Conclusion and Final Thoughts

Active learning is a growing area of research in machine learning, no doubt fueled by the reality that data is increasingly easy or inexpensive to obtain but difficult or costly to label for training. Over the past two decades, there has been much work in formulating and understanding the various ways in which queries are selected from the learner's perspective (Sections 2 and 3). This has generated a lot of evidence that the number of labeled examples necessary to train accurate models can be effectively reduced in a variety of applications (Section 4).

Drawing on these foundations, the current surge of research seems to be aimed at applying active learning methods in practice, which has introduced many important problem variants and practical concerns (Sections 5 and 6). So this is an interesting time to be involved in machine learning and active learning in particular, as some basic questions have been answered but many more still remain. These issues span interdisciplinary topics from learning to statistics, cognitive science, and human-computer interaction to name a few. It is my hope that this survey is an effective summary for researchers (like you) who have an interest

in active learning, helping to identify novel opportunities and solutions for this promising area of science and technology.

Acknowledgements

This survey began as a chapter in my PhD thesis. During that phase of my career, I am indebted to my advisor Mark Craven and committee members Jude Shavlik, Xiaojin "Jerry" Zhu, David Page, and Lewis Friedland, who offered valuable feedback and encouraged me to expand this into a general resource for the machine learning community. My own research and thinking on active learning has also been shaped by collaborations with several others, including Andrew McCallum, Gregory Druck, and Soumya Ray.

The insights and organization of ideas in the survey are not wholly my own, but draw from the conversations I've had with numerous researchers in the field. After putting out the first draft of this document, I received nearly a hundred emails with additions, corrections, and new perspectives, which have all been woven into the fabric of this revision; I thank everyone who took (and continues to take) the time to share your thoughts. In particular, I would like to thank (in alphabetical order): Jason Baldridge, Aron Culotta, Pinar Donmez, Russ Greiner, Carlos Guestrin, Robbie Haertel, Steve Hanneke, Ashish Kapoor, John Langford, Percy Liang, Prem Melville, Tom Mitchell, Clare Monteleoni, Ray Mooney, Foster Provost, Eric Ringger, Teddy Seidenfeld, Katrin Tomanek, and 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. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 25–32. ACM Press, 2009.

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

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

S. Arora, E. Nyberg, and C.P. Rosé. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing*, pages 18–26. ACL Press, 2009.

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.

J. Baldridge and A. Palmer. How well does active learning *actually* work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 296–305. ACL Press, 2009.

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. Beygelzimer, S. Dasgupta, and J. Langford. Importance-weighted active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 49–56. ACM Press, 2009.

M. Bloodgood and V. Shanker. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 39–47. ACL Press, 2009.

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.

A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the IEEE Symposium on the Foundations of Computer Science*, page 330. IEEE Press, 1996.

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.

R. Burbidge, J.J. Rowland, and R.D. King. Active learning for regression based on query by committee. In *Proceedings of Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 209–218. Springer, 2007.

A. Carlson, J. Betteridge, R. Wang, E.R. Hruschka Jr, and T. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. ACM Press, 2010.

X. Chai, L. Deng, Q. Yang, and C.X. Ling. Test-cost sentistive naive Bayes classification. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*, pages 51–5. IEEE Press, 2004.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):237–304, 1995.

S.F. Chen and R. Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

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, R. Ladner, M. El-Sharkawi, R. Marks II, M. Aggoune, and D. Park. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems (NIPS)*. Morgan Kaufmann, 1990.

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.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2006.

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.

N. Cressie. *Statistics for Spatial Data*. Wiley, 1991.

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 and D.J. Hsu. Hierarchical sampling for active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 208–215. ACM Press, 2008.

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.

P. Donmez, J. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268. ACM Press, 2009.

G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–602. ACM Press, 2008.

G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 81–90. ACL Press, 2009.

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

S. Esmeir and S. Markovitch. Anytime induction of cost-sensitive trees. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 425–432. MIT Press, 2008.

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

P. Flaherty, M. Jordan, and A. Arkin. Robust design of biological experiments. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, pages 363–370. MIT Press, 2006.

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.

C. Gasperin. Active learning for anaphora resolution. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing*, pages 1–8. ACL Press, 2009.

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.

J. Goodman. Exponential priors for maximum entropy models. In *Proceedings of Human Language Technology and the North American Association for Computational Linguistics (HLT-NAACL)*, pages 305–312. ACL Press, 2004.

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

N. Grira, M. Crucianu, and N. Boujemaa. Active semi-supervised fuzzy clustering for image database categorization. In *Proceedings the ACM Workshop on Multimedia Information Retrieval (MIR)*, pages 9–16. ACM Press, 2005.

C. Guestrin, A. Krause, and A.P. Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 265–272. ACM Press, 2005.

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.

A. Haghighi and D. Klein. Prototype-driven learning for sequence models. In *Proceedings of the North American Association for Computational Linguistics (NAACL)*, pages 320–327. ACL Press, 2006.

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.

T. Hofmann and J.M. Buhmann. Active data clustering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 10, pages 528—534. Morgan Kaufmann, 1998.

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.

Y. Huang and T. Mitchell. Text clustering with extended user feedback. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 413–420. ACM Press, 2006.

R. Hwa. On minimizing training corpus for parser acquisition. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 1–6. ACL Press, 2001.

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

S. Ji and L. Carin. Cost-sensitive feature acquisition and classification. *Pattern Recognition*, 40:1474–1485, 2007.

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.

R.D. King, J. Rowland, S.G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L.N. Soldatova, A. Sparkes, K.E. Whelan, and A. Clare. The automation of science. *Science*, 324(5923):85–89, 2009.

C. Körner and S. Wrobel. Multi-class ensemble-based active learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 687–694. Springer, 2006.

A. Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, 2008.

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.

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

K. Lang and E. Baum. Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 335–340. IEEE Press, 1992.

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.

P. Liang, H. Daume, and D. Klein. Structure compilation: Trading structure for features. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 592–599. ACM Press, 2008.

P. Liang, M.I. Jordan, and D. Klein. Learning from measurements in exponential families. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 641–648. ACM Press, 2009.

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

C.X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 483–486. ACM Press, 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.

Z. Lu and J. Bongard. Exploiting multiple classifier types with active learning. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1905–1906. ACM Press, 2009.

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

G. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of the Association for Computational Linguistics (ACL)*. ACL Press, 2008.

D. Margineantu. Active cost-sensitive learning. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1622–1623. AAAI Press, 2005.

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.

P. Melville, S.M. Yang, M. Saar-Tsechansky, and R. Mooney. Active learning for probability estimation using Jensen-Shannon divergence. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 268–279. Springer, 2005.

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.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2009.

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.

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

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.

F. Olsson. *Bootstrapping Named Entity Recognition by Means of Active Machine Learning*. PhD thesis, University of Gothenburg, 2008.

F. Olsson. A literature survey of active machine learning in the context of natural language processing. Technical Report T2009:06, Swedish Institute of Computer Science, 2009.

F. Olsson and K. Tomanek. An intrinsic stopping criterion for committee-based active learning. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 138–146. ACL Press, 2009.

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.

G.J. Qi, X.S. Hua, Y. Rui, J. Tang, and H.J. Zhang. Two-dimensional active learning for image classification. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

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.

R. Reichart, K. Tomanek, U. Hahn, and A Rappoport. Multi-task active learning for linguistic annotations. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 861–869. ACL Press, 2008.

E. Ringger, M. Carmen, R. Haertel, K. Seppi, D. Lonsdale, P. McClanahan, J. Carroll, and N. Ellison. Assessing the costs of machine-assisted corpus annotation through a user study. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association, 2008.

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.

M. Saar-Tsechansky, P. Melville, and F. Provost. Active feature-value acquisition. *Management Science*, 55(4):664–684, 2009.

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.

V.S. Sheng and C.X. Ling. Feature value acquisition in testing: A sequential batch test algorithm. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 809–816. ACM Press, 2006.

V.S. Sheng, F. Provost, and P.G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM Press, 2008.

V. Sindhwani, P. Melville, and R.D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 953–960. ACM Press, 2009.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263. ACM Press, 2008.

M. Steyvers and T. Griffiths. Probabilistic topic models. In T. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*. Erlbaum, 2007.

M. Sugiyama and N. Rubens. Active learning with model selection in linear regression. In *Proceedings of the SIAM International Conference on Data Mining*, pages 518–529. SIAM, 2008.

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.

K. Tomanek and U. Hahn. Semi-supervised active learning for sequence labeling. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1039–1047. ACL Press, 2009.

K. Tomanek and F. Olsson. A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing*, pages 45–48. ACL Press, 2009.

K. Tomanek, J. Wermter, and U. Hahn. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 486–495. ACL Press, 2007.

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. Tür, 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. What's it going to cost you? Predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Press, 2009a.

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, pages 1705–1712. MIT Press, 2009b.

A. Vlachos. A stopping criterion for active learning. *Computer Speech and Language*, 22(3):295–312, 2008.

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.

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1081–1087. ACM Press, 2006.

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