# Chapter 1

# Selective Data Acquisition for Machine Learning

**Josh Attenberg**
*NYU Polytechnic Institute, Brooklyn, NY 11201*
josh@cis.poly.edu

**Prem Melville**
*IBM T.J. Watson Research Center, Yorktown Heights, NY 10598*
pmelvil@us.ibm.com

**Foster Provost**
*NYU Stern School of Business, New York, NY 10012*
fprovost@stern.nyu.edu

**Maytal Saar-Tsechansky**
*Red McCombs School of Business, University of Texas at Austin*
maytal.saar-tsechansky@mccombs.utexas.edu

## 1.1 Introduction

In many applications, one must invest effort or money to acquire the data and other information required for machine learning and data mining. Careful selection of the information to acquire can substantially improve generalization performance per unit cost. The costly information scenario that has received the most research attention (see Chapter X) has come to be called "active learning," and focuses on choosing the instances for which target values (labels) will be acquired for training. However machine learning applications offer a variety of different sorts of information that may need to be acquired.

This chapter focuses on settings and techniques for selectively acquiring information beyond just single training labels (the values of the target variable) for selected instances in order to improve a model's predictive accuracy. The different kinds of acquired information include feature values, feature labels, entire examples, values at prediction time, repeated acquisition for the same data item, and more. For example, Figures 1.1 contrast the acquisition of training labels, feature values, and both. We will discuss all these sorts of information in detail. Broadening our view beyond simple active learning not only expands the set of applications to which we can apply selective acquisition strategies; it also

highlights additional important problem dimensions and characteristics, and reveals fertile areas of research that to date have received relatively little attention.

In what follows, we start by presenting two general notions that are employed to help direct the acquisition of various sorts of information. The first is to prefer to acquire information for which the current state of modeling is *uncertain*. The second is to acquire information that is estimated to be the most valuable to acquire.

After expanding upon these two overarching notions, we discuss a variety of different settings where information acquisition can improve modeling. The purpose of examining various different acquisition settings in some detail is to highlight the different challenges, solutions, and research issues. As just one brief example, distinct from active learning, active acquisition of feature values may have access to additional information, namely instances' labels—which enables different sorts of selection strategies.

More specifically, we examine the acquisition of feature values, feature labels, and prediction-time values. We also examine the specific, common setting where information is not perfect, and one may want to acquire additional information specifically to deal with information quality. For example, one may want to acquire the same data item more than once. In addition, we emphasize that it can be fruitful to expand our view of the sorts of acquisition actions we have at our disposal. Providing specific variable values is only one sort of information "purchase" we might make. For example, in certain cases, we may be able to acquire entire examples of a rare class, or distinguishing words for document classification. These alternative acquisitions may give a better return on investment for modeling.

Finally, and importantly, most research to date has considered each sort of information acquisition independently. However, why should we believe that only one sort of information is missing or noisy? Modelers may find themselves in situations where they need to acquire various pieces of information, and somehow must prioritize the different sorts of acquisition. This has been addressed in a few research papers for pairs of types of information, for example for target labels and feature labels ("active dual supervision"). We argue that a challenge problem for machine learning and data mining research should be to work toward a unified framework within which arbitrary information acquisitions can be prioritized, to build the best possible models on a limited budget.

## 1.2 Overarching principles for selective data acquisition

In general selective data acquisition a learning algorithm can request the value of particular missing data, which is then provided by an oracle at some cost. There may be more than one oracle, and oracles are not assumed to be perfect. The goal of selective data acquisition is to choose to acquire data that is most likely to improve the system's use-time performance on a specified modeling objective in a cost-effective manner. We will use $q$ to refer to the query for a selected piece of missing data. For instance, in traditional active learning this would correspond to querying for the missing label of a selected instance; while in the context of active feature-value acquisition, $q$ is the request for a missing feature value. We will focus primarily (but not exclusively) on *pool-based* selective data acquisition, where we select a query from a pool of available candidate queries, e.g., the set of all missing feature-values that can be acquired on request.

Note that in most cases, selection from the pool is performed in epochs, whereby at each phase, a batch of one or more queries are performed simultaneously. The combinatorial

problem of selecting the most useful such batches (and overall data set) from such a pool of candidates makes direct optimization an NP-hard problem. Typically, a first order Markov relaxation is performed, whereby the most promising data are selected greedily one-at-a-time from the pool. While not guaranteeing a globally optimal result set (regardless of what selection criterion is being used), such sequential data access often works well in practice while making the selection problem tractable.

We begin by discussing two general principles that are applicable for the selective acquisition of many different types of data. These overarching principles must be instantiated specifically to suit the needs of each acquisition setting. While individual instantiations may differ considerably, both principles have advantages and disadvantages that hold in general, which we discuss below. In reviewing these techniques, we often refer to the familiar active learning setting; we will see how the principles do and do not apply to the other settings as the chapter unfolds.

### 1.2.1 Uncertainty Reduction

The most commonly used method for non-uniform sample selection for machine learning is to select data items for which the current model is most uncertain.

This notion is the basis for the most commonly used individual active learning technique, Uncertainty Sampling [44], as well as closely related (and in some cases identical) techniques, such as selecting data points closest to a separating hyperplane, Query-by-Committee and variance reduction [79]. Specifically, with Uncertainty Sampling the active learner requests labels for examples that the currently held model is least certain about how to classify.

Uncertainty reduction techniques are based on the assumption that predictive errors largely occur in regions of the problem space where predictions are most ambiguous. The intent is that by providing supplemental information in these regions, model confidence can be improved, along with predictive performance. Despite the typical reference to the classic 1994 paper [44], even Uncertainty Sampling itself has become a framework within which different techniques are implemented. For example, exactly how one should measure uncertainty is open to interpretation; the following three calculations of uncertainty all have been used [79, 55, 78]:

$$1 - P(\hat{y}|x) \tag{1.1}$$

$$P(\hat{y_1}|x) - P(\hat{y_2}|x) \tag{1.2}$$

$$-\sum_i P(\hat{y_i}|x)\log(P(\hat{y_i}|x)) \tag{1.3}$$

where $P(\hat{y}|x)$ is the highest posterior probability assigned by the classifier to a class, while $P(\hat{y_1}|x)$ and $P(\hat{y_2}|x)$ are the probabilities assigned to the first and second most probable classes as predicted by the classifier.

Uncertainty Reduction is widely used with some success in research literature, though we will discuss situations where Uncertainty Reduction fails to make beneficial selections. The same uncertainty-based heuristic can be applied more broadly to acquiring other forms of data. For instance, when feature-values are missing, one can attempt to impute the missing values from those that are present, and choose to acquire values where the model is least certain of the imputed values.

The advantages of Uncertainty Reduction are:

- Evaluating and selecting queries based on uncertainty is computationally efficient in many settings. For instance, Uncertainty Sampling for training labels only requires

applying the classifier to predict the posterior class probabilities of examples in the unlabeled pool. There is no retraining of models required in order to select queries. Note that, in other settings, such as acquiring feature-values, the complexity may be considerably higher depending on how one choses to measure uncertainty.

- Uncertainty Reduction techniques are often adopted because of their ease of implementation. For example, Uncertainty Sampling requires computing one of the uncertainty scores described above which are simply applications of the existing model in order to make predictions on an unlabeled instance.

- In the active learning literature, Uncertainty Reduction techniques have been applied across many problems with reasonable success.

The disadvantages of Uncertainty Reduction are:

- While often effective in practice, Uncertainty Reduction does not directly attempt to optimize a classifier's generalization performance. As such it can often choose queries that may reduce model uncertainty, but not result in improvement on test set predictions. Notably, when applied to obtaining example labels, Uncertainty Sampling is prone to selecting outliers [79]. These could be instances the model is uncertain about, but are not representative of instances in the test set. The selection of outliers can be addressed by using the uncertainty scores to form a sampling distribution [72, 73]; however, sampling also can reduce the effectiveness of Uncertainty Reduction techniques by repeatedly selecting marginally informative examples. In other settings, Uncertainty Reduction may reduce uncertainty about values that are not discriminative, such as acquiring values for a feature that is uncorrelated to the class.

- While Uncertainty Sampling is often favored for ease of implementation, in settings beyond the acquisition of single training labels, estimating uncertainty is often not as straightforward. For example, how should you measure the uncertainty of an instance label, given a current model and many contradictory labels for the same instance from different oracles [82]?

- In general, selective data acquisition assumes that the response to each query comes at a cost. In realistic settings, these costs may vary for each type of acquisition and even for each query. Notably, some examples are more difficult for humans to label than others, and as such may entail a higher cost in terms of annotation time. Similarly, some features-values are more expensive to obtain than others, e.g., if they are the result of a more costly experiment. Uncertainty Reduction methods do not naturally facilitate a meaningful way to trade off costs with potential benefits of each acquisition. One ad hoc approach of attempting this is to divide the uncertainty score with the cost for each query, and make acquisitions in order of the resulting quantity [80, 32]. This is a somewhat awkward approach to incorporating costs, as uncertainty per unit cost is not necessarily proportional to potential benefit to the resulting model.

- As mentioned above, uncertainty can be defined in different ways even for the same type of acquisition, such as a class label. Different types of acquisitions, such as feature values, require very different measures of uncertainty. Consequently, when considering more than one type of acquisition simultaneously, there is no systematic way to compare two different measures of uncertainty, since they are effectively on different scales. This makes it difficult to construct an Uncertainty Reduction technique

that systematically decides which type of information is most beneficial to acquire next.

### 1.2.2 Expected Utility

Selecting data that the current model is uncertain about may result in queries that are not useful in discriminating between classes. An alternative to such uncertainty-based heuristics is to directly estimate the expected improvement in generalization due to each query. In this approach, at every step of selective data acquisition, the next query selected is the one that will result in the highest estimated improvement in classifier performance per unit cost. Since the true values of the missing data are unknown prior to acquisition, it is necessary to estimate the *potential* impact of every query for all possible outcomes. [1] Hence, the decision-theoretic optimal policy is to ask for missing data which, once incorporated into the existing data, will result in the greatest increase in classification performance in *expectation*. If $\omega_q$ is the cost of the query $q$, then its Expected Utility of acquisition can be computed as

$$EU(q) = \int_v P(q = v) \frac{\mathcal{U}(q = v)}{\omega_q} \tag{1.4}$$

where $P(q = v)$ is the probability that query $q$ will take on value $v$, and $\mathcal{U}(q = v)$ is the utility to the model of knowing that $q$ has the value $v$. This utility can be defined in any way to represent a desired modeling objective. For example, $\mathcal{U}$ could be defined as classification error. In this case, this approach is referred to as *Expected Error Reduction*. When applied more generally to arbitrary utility functions we refer to such a selection scheme as *Expected Utility* or *Estimated Risk Minimization*. Note that, in Equation 1.4 the true values of the marginal distribution $P(.)$ and the utility $\mathcal{U}(.)$ on the test set is unknown. Instead, empirical estimates of these quantities are used in practice. When the missing data can only take on discrete values, the expectation can be easily computed by piecewise summation over the possible values. While for continuous values, computation of expected utility can be performed using Monte Carlo methods.

The advantages of Expected Utility are:

- Since this method is directly trying to optimize the objective on which the model will be evaluated, it avoids making acquisitions that do not improve this objective even if it reduces uncertainty or variance in the predictions.

- Incorporating different acquisition costs is also straightforward in this framework. The trade-off of utility versus cost is handled directly, as opposed to relying on an unknown indirect connection between uncertainty and utility.

- This approach is capable of addressing multiple types of acquisition simultaneously within a single framework. Since the measure of utility is independent of the type of acquisition and only dependent on the resulting classifier, we can estimate the expected utility of different forms of acquisitions in the same manner. For instance, we can use such an approach to estimate the utility of acquiring class labels and feature values in tandem [71]. The same framework can also be instantiated to yield a holistic approach to active dual supervision, where the Expected Utility of an instance or

[1] For instance, in the case of binary classification, the possible outcomes are a *positive* or *negative* label for a queried example.

feature label query can be computed and compared on the same scale [2]. By evaluating different acquisitions in the same units, and by measuring utility per unit cost of acquisition, such a framework facilitates explicit optimization of the trade-offs between the costs and benefits of the different types of acquisitions. between the costs and benefits of the different types of acquisitions.

The disadvantages of Expected Utility are:

- A naive implementation of the Expected Utility framework is computationally intractable even for data of moderate dimensionality. The computation of Expected Utility (Equation 1.4) requires iterating over all possible outcomes of all candidate queries. This often means training multiple models for the different values a query may take on. This combinatorial computational cost can often be prohibitive. The most common approach to overcome this, at the cost of optimality, is to sub-sample the set of available queries, and only compute Expected Utility on this smaller candidate set [71]. This method has also been demonstrated to be feasible for classifiers that can be rapidly trained incrementally [70]. Additionally, dynamic programming and efficient data structures can be leveraged to make the computation tractable [8]. The computation of the utility of each outcome of each query, while being the bottleneck, is also fully parallelizable. As such large-scale parallel computing has the potential of making the computational costs little more than that of training a single classifier.

- As mentioned above, the terms $P(.)$ and $\mathcal{U}(.)$ in the Expected Utility computation must be estimated from available data. However, the choice of methods to use for these estimations is not obvious. Making the correct choices can be a significant challenge for a new setting, and can make a substantial difference in the effectiveness of this approach [71].

- Typically, the estimators used for $P(.)$ and $\mathcal{U}(.)$ are based on the pool of available training data, for instance, through cross validation. The available training examples themselves are often acquired through an active acquisition process. Here, due to the preferences of the active process, the distribution of data in the training pool may differ substantially from that of the native data population and as a result, the estimations of $P(.)$ and $\mathcal{U}(.)$ may be arbitrarily inaccurate in the worst case.

- Additionally, it should be noted that despite the *emperical risk minimization* moniker, Expected Utility methods do not in general yield the globally optimal set of selections. This is due to several simplifications: first, the myopic, sequential acquisition policy mentioned above where the benefit for each individual example is taken in isolation, and second, the utilization of empirical risk as a proxy for actual risks.

## 1.3 Active feature-value acquisition

In this section we begin by discussing active feature-value acquisition (AFA), the selective acquisition of single feature values for training. We then review extensions of these policies for more complex settings in which feature values, different sets thereof, as well as class labels can be acquired at a cost simultaneously. We discuss some insights on the challenges and effective approaches for these problems as well as interesting open problems.

As an example setting for active feature-value acquisition, consider consumers' ratings of different products being used as predictors to estimate whether or not a customer is likely to be interested in an offer for a new product. At any given time, only a subset of any given consumer's "true" ratings of her prior purchases are available, and thus many feature values are missing, potentially undermining inference. To improve this inference, it is possible to offer consumers incentives so as to reveal their preferences—for example, rating other prior purchases. Thus, it is useful to devise an intelligent acquisition policy to select which products and which consumers are most cost-effective to acquire. Similar scenarios arise in a variety of other domains, including when databases are being used to estimate the likelihood of success of alternative medical treatments. Often the feature values of some predictors, such as of medical tests, are not available; furthermore, the acquisition of different tests may incur different costs.

This general active feature-value acquisition setting, illustrated in Figure 1.1(b), differs from active learning in several important ways. First, policies for acquiring training labels assign one value to an entire prospective instance acquisition. Another related distinction is that the impact on induction from obtaining an entire instance may require a less precise measure to that required to estimate the impact from acquiring merely a single feature value. In addition, having multiple missing feature values gives rise to a myriad of settings regarding the level of granularity at which feature values can be acquired and the corresponding cost [102, 47, 57, 56, 38]. For example, in some applications, such as when acquiring consumers' demographic and lifestyle data from syndicated data providers, only the complete set of all feature values can be purchased at a fixed cost. In other cases, such as the recommender systems and treatment effectiveness tasks above, it may be possible to purchase the value of a single variable, such as by running a medical test. And there are intermediate cases, such as when different subsets (e.g., a battery of medical tests) can be acquired as a bundle. Furthermore, different sets may incur different costs; thus, while very few policies do so, it is beneficial to consider such costs when prioritizing acquisition. In the remainder of this section, we discuss how we might address some of these problem settings, as well as interesting open challenges.

A variety of different policies can be envisioned for the acquisition of individual feature values at a fixed cost, following the Expected Utility framework we discussed in Section 1.2 and [47, 56, 71]. As such, Expected Utility AFA policies aim to estimate the expected benefit from an acquisition by the change in some loss/gain function in expectation.
For feature value acquisitions the expected utility framework has several important advantages, but also some limitations. Because different features (such as medical tests) are very likely to incur different costs perhaps the most salient advantage for AFA is the ability to incorporate cost information when prioritizing acquisitions. The Expected Utility framework also allows us to prioritize among acquisitions of individual features as well as different sets thereof. However, the expected value framework would guarantee the acquisition of the optimal single feature value in expectation, only if the true distributions of values for each missing feature were known, and the loss/gain function, $\mathcal{U}$, were to capture the actual change in the model's generalization accuracy following an acquisition. In settings where many feature values are missing these estimations may be particularly challenging. For example, empirical estimation of the model's generalization performance over instances with many missing values may not accurately approximate the magnitude or even the direction of change in generalization accuracy. Perhaps a more important consideration regarding the choice of gain function, $\mathcal{U}$, for feature value acquisition is the sequential, myopic nature of these policies. Similar to most information acquisition policies, if multiple features are to be acquired, a myopic policy, which aims to estimate the benefit from each prospective acquisition in isolation, is not guaranteed to identify the optimal *set* of acquisitions, even if the estimations listed above were precise. This is because the

expected contribution of an individual acquisition is estimated with respect to the current training data, irrespective of other acquisitions which will be made. Interestingly, due to this myopic property, selecting the acquisition which yields the best estimated improvement in generalization accuracy often does not yield the best results; rather, other measures have been shown to be empirically more effective [71]—for example, log gain. Specifically, when a model is induced from a training set $T$, let $\hat{P}(c_k|x_i)$ be the probability estimated by the model that instance $x_i$ belongs to class $c_k$; and $\mathbb{I}$ is an indicator function such that $\mathbb{I}(c_k, x_i) = 1$ if $c_k$ is the correct class for $x_i$ and $\mathbb{I}(c_k, x_i) = 0$, otherwise. Log Gain (LG) is then defined as:

$$LG(x_i) = -\sum_{k=1}^{K} \mathbb{I}(c_k, x_i) \log \hat{P}(c_k|x_i) \tag{1.5}$$

Notably, LG is sensitive to changes in the model's estimated probability of the correct class. As such, this policy promotes acquisitions which increase the likelihood of correct class prediction, once other values are acquired.

We have discussed the computational cost which the Expected Utility framework entails and the need to reduce the consideration set to a small subset of all prospective acquisitions. For AFA, drawing from the set of prospective acquisitions uniformly at random [71] may be used; however, using a fast heuristics to identify feature values that are likely to be particularly informative per unit cost can be more effective. For example, a useful heuristic is to draw a subset of prospective acquisitions based on the corresponding features' predictive values [71] or to prefer acquisitions from particularly informative instances [56].

A related setting, illustrated in Figure 1.1(c), is one in which for all instances the same subset of feature values are known, and the subset of all remaining feature values can be acquired at a fixed cost. Henceforth, we refer to this setting as *instance completion*. Under some conditions, this problem bears strong similarity to the active learning problem. For example, consider a version of this problem in which only instances with complete feature values are used for induction [102, 103]. If the class labels of prospective training examples are unknown or are otherwise not used to select acquisition, this problem becomes very similar to active learning in that the value from acquiring a complete training instance must be estimated. For example, we can use measures of prediction uncertainty (cf., section 1.2 to prioritize acquisitions [103].

Note however that the active feature-value acquisition setting may have additional information that can be brought to bear to aid in selection: the known class labels of prospective training examples. The knowledge of class labels can lead to selection policies that prefer to acquire features for examples on which the current model makes mistakes [57]. An extreme version of the instance completion problem is when there are no known feature values and the complete feature set can be acquired as a bundle for fixed cost (see Section 1.7 below).

**Acquiring feature values and class labels**

We noted earlier that an important benefit of the AFA Expected Utility framework is that it allows comparing among the benefits from acquiring feature values, sets thereof, as well as class labels—and thus can consider selecting these different sorts of data simultaneously [71]. This setting is shown in Figure 1.1(d). Note, however, that considering different types of acquisitions simultaneously also presents new challenges. In particular, recall the need for a *computationally fast* heuristic to select a subset of promising prospective acquisitions to be subsequently estimated by the Expected Utility policy. Assuming

uniform costs, in most cases acquiring a class label is likely to be significantly more cost-effective than acquiring a single feature values. However, if the consideration set were to be sampled uniformly at random, when class labels constitute a minority in this pool, many informative class labels may not even be considered for acquisition. One heuristic that has been shown to perform well is to infer a crude measure of a class label's benefit by the benefits of the corresponding instance's known feature values. Specifically, the probability of drawing a prospective feature-value acquisition is proportional to the cost-normalized variant of the corresponding feature's information gain $IG(F, L)$ [63] for class variable L; the likelihood of considering a class label can then be made proportional to the sum of the cost-normalized information gains of all the instance's missing feature values [71].

One important setting in which arbitrary subsets of feature values can be acquired at different costs, has not been explored extensively. It may be natural to extend the Expected Utility framework to consider sets of categorical features. However, estimating the joint probability distribution of all possible sets of values may render such a policy hopelessly inefficient. To our knowledge, there has been some work on the acquisition of sets of values during inference [8]. However, the complexity of the estimation for induction is substantially more significant.

Lastly, integral to designing and evaluating information acquisition policies is a solid understanding of the best *costless* alternatives for dealing with unknown feature values [45]. For example, unknown feature values may be replaced with estimates via imputation or, in some cases, ignored during induction [28, 75]. For the most part, the literature on selective data acquisition has not developed to consider (systematically) alternative costless solutions. Nonetheless, acquisition policies ought to estimate an acquisition's value as compared to the best costless solution (imputing the value; ignoring that variable all together; taking a Bayesian approach). Perhaps more importantly, the conclusions of empirical comparisons among policies and, consequently, the perceived effectiveness of different policies, may be affected substantively by which (if any) costless solutions are employed.

## 1.4 Labeling features versus examples

In selective data acquisition, we can acquire more information about our data instances as in active feature-value acquisition. However, there are other types of class-indicative data that are informative data that may be useful for building predictive models. In such a setting, where myriad forms supervision can be compiled into building predictive models, it becomes important to examine acquisition costs and benefits, allocating budget to those data most valuable to the task at hand. Consider, for example, the task of *sentiment detection*, where given a piece of text as input, the desired output is a label that indicates whether this text expresses a positive or negative opinion. This problem can be cast as a typical binary text classification task, where a learner is trained on a set of documents that have been labeled based on the sentiment expressed in them [60]. Alternatively, one could provide *labeled features*: for example, in the domain of movie reviews, words that evoke positive sentiment (e.g., "captivating", "suspenseful", etc.) may be labeled positive, while words that evoke negative sentiment (e.g., "predictable", "unimaginative", etc.) may be labeled negative. Through this kind of annotation a human conveys prior linguistic experience with a word by a sentiment label that reflects the emotion that the word evokes.

The setting where individual semantic features provide useful class indicators arises

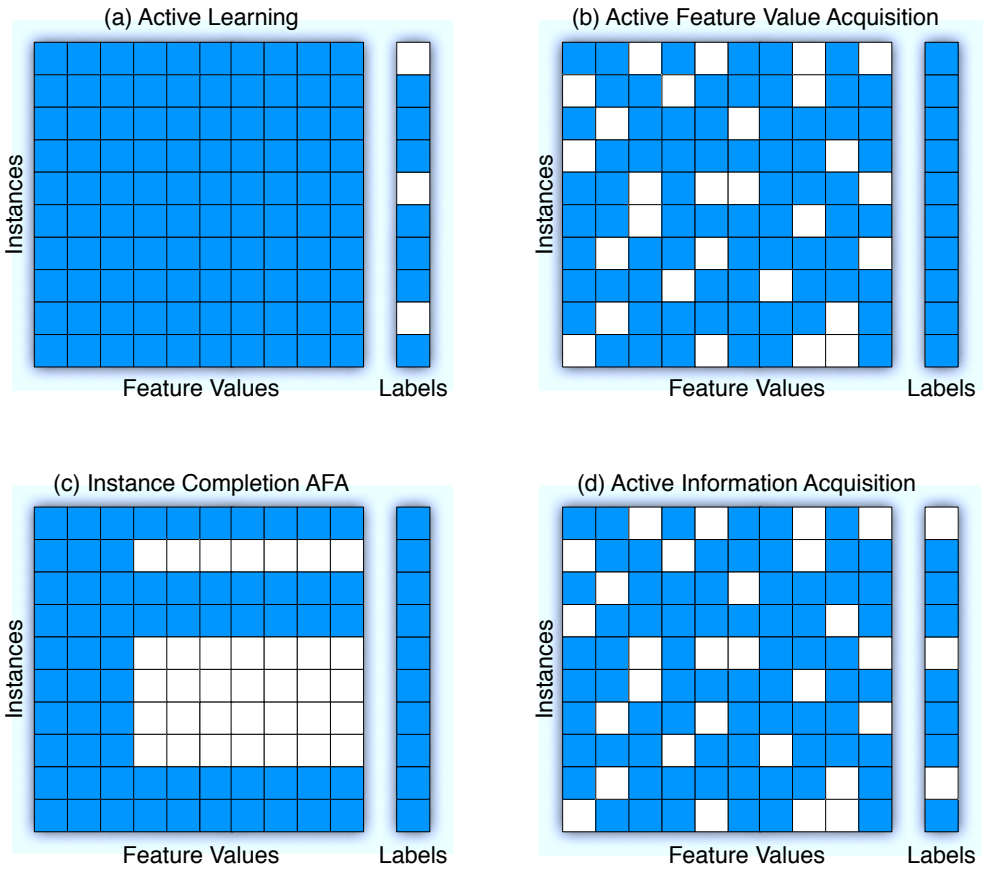


FIGURE 1.1: Different Data Acquisition Settings

broadly, notably in Natural Language Processing tasks where, in addition to labeled documents, it is possible to provide domain knowledge in the form of words or phrases [100] or more sophisticated linguistic features that associate strongly with a class. Such feature supervision can greatly reduce the number of labels required to build high-quality classifiers [24, 84, 54]. In general, example and feature supervision are complementary, rather than redundant. As such they can also be used together. This general setting of learning from both labels on examples and features is referred to as *dual supervision* [84].

In this section we provide a brief overview of learning from labeled features, as well as learning both from labeled features and labeled examples. We will also discuss the challenges of active learning in these settings, and some approaches to overcome them.

### 1.4.1 Learning from feature labels

Providing feature-class associations through labeling features can be viewed as one approach to expressing background, prior or domain knowledge about a particular supervised learning task. Methods to learn from such feature labels can be divided into approaches that use labeled features along with unlabeled examples, and methods that use both labeled features and examples. Since, we focus primarily on text classification in this

section, we will use *words* and *documents* interchangeably with *features* and *examples*. However, incorporating background knowledge into learning has also been studied outside the context of text classification, as in knowledge-based neural networks [90] and knowledge-based SVMs [29, 42].

**Labeled features and unlabeled examples**

A simple way to utilize feature supervision is to use the labels on features to label examples, and then use an existing supervised learning algorithm to build a model. Consider the following straightforward approach. Given a representative set of words for each class, create a *representative document* for each class containing all the representative words. Then compute the cosine similarity between unlabeled documents and the representative documents. Assign each unlabeled document to the class with the highest similarity, and then train a classifier using these *pseudo-labeled examples*. This approach is very convenient as it does not require devising a new model, since it can effectively leverage existing supervised learning techniques such as Naïve Bayes [46]. Given that it usually take less time to label a word than it takes to label a document [24], this is a cost-effective alternative.

An alternative to approaches of generating and training with pseudo-labeled examples, is to directly use the feature labels to constrain model predictions. For instance, a label $y$ for feature $x_i$ can be translated into a soft constraint, $P(y|x_i) > 0.9$, in a multinomial logistic regression model [24]. Then the model parameters can be optimized to minimize some distance, e.g. Kullback-Leibler divergence from these reference distributions.

**Dual Supervision**

In dual supervision models, labeled features are used in conjunction with labeled examples. Here too, labeled features can be used to generate pseudo-labeled examples, either by labeling unlabeled examples [97] or re-labeling duplicates of the training examples [77]. These pseudo-labeled examples can be combined with the given labeled examples, using weights to down-weight prior knowledge when more labeled examples are available. Such methods can be implemented within existing frameworks, such as boosting logistic regression [77], and weighted margin support vector machines [97].

Generating pseudo-labeled examples is an easy way to leverage feature labels within the traditional supervised learning framework based on labeled examples. Alternatively one can incorporate both forms of supervision directly into one unified model [54, 84]. Pooling Multinomials is one such classifier, which builds a generative model that explains both labeled features and examples. In Pooling Multinomials unlabeled examples are classified just as in multinomial Naïve Bayes classification [52], by predicting the class with the maximum likelihood, given by $argmax_{c_j} P(c_j) \prod_i P(w_i|c_j)$; where $P(c_j)$ is the prior probability of class $c_j$, and $P(w_i|c_j)$ is the probability of word $w_i$ appearing in a document of class $c_j$. In the absence of background knowledge about the class distribution, the class priors $P(c_j)$ are estimated solely from the training data. However, unlike regular Naïve Bayes, the conditional probabilities $P(w_i|c_j)$ are computed using both the labeled examples and the set of labeled features. Given two models built using labeled examples and labeled features, the multinomial parameters of such models are aggregated through a convex combination, $P(w_i|c_j) = \alpha P_e(w_i|c_j) + (1-\alpha)P_f(w_i|c_j)$; where $P_e(w_i|c_j)$ and $P_f(w_i|c_j)$ represent the probability assigned by using the example labels and feature labels respectively, and $\alpha$ is a weight indicating the level of confidence in each source of information. At the crux of this framework is the generative labeled-features model, which assumes that the feature-class associations provided by human experts are implicitly arrived at by examining many latent documents of each class. This assumption translates in into several constraints on the model parameter, which allows one to exactly derive the conditional distributions $P_f(w_i|c_j)$ that would generate the latent documents [54].

### 1.4.2 Active Feature Labeling

While traditional active learning has primarily focused on selecting unlabeled *instances* to be labeled, the dual-supervision setting adds an additional aspect to active learning where labels may be acquired for features as well. In this section we focus on the task of active learning applied to feature-label acquisition, illustrated by Figure 1.2

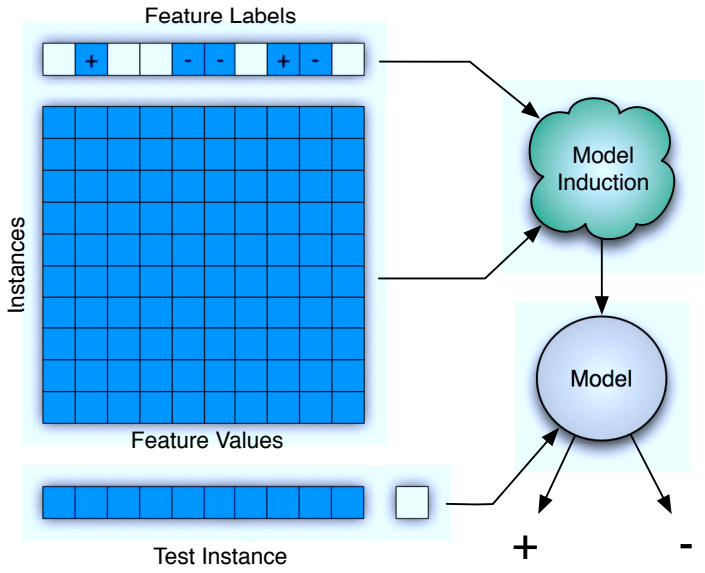


FIGURE 1.2: Active Feature Labeling

**Uncertainty-based approaches**

Feature and instance labels contribute very differently to learning a model, and as such, standard active learning approaches may not be directly applicable to feature label acquisition. Nevertheless, heuristic approaches based on the principle of Uncertainty Sampling have been applied to acquire feature labels, with varying degrees of success. As in traditional Uncertainty Sampling, Feature Uncertainty Sampling requests labels for *features* for which the current model has the highest degree of uncertainty.

Much like instance uncertainty, feature uncertainty can be measured in different ways, depending on the underlying method used for incorporating feature supervision. For instance, when using a learner that produces a linear classifier, we can use the magnitude of the weights on the features as a measure of uncertainty [85] — where lower weights indicate less certainty. In the case of Pooling Multinomials, which builds a multinomial Naïve Bayes model, we can directly use the model's conditional probabilities of each feature $f$ given a class. Specifically, feature uncertainty can be measured by absolute log-odds ratio, $abs\left(\log\left(\frac{P(f|+)}{P(f|-)}\right)\right)$. The smaller this value, the more uncertain the model is about the feature's class association. Then in every iteration of active learning, features with the lowest certainty scores are selected for labeling.

Though Uncertainty Sampling for features seems like an appealing notion, it may not lead to better models. If a classifier is uncertain about a feature, it may have insufficient information about this feature and may indeed benefit from learning its label. However, it

is also quite likely that a feature has a low certainty score because it does not carry much discriminative information about the classes. For instance, in the context of sentiment detection, one would expect that neutral/non-polar words will appear to be uncertain words. For example, words such as "the" which are unlikely to help in discriminating between classes, are also likely to be considered the most uncertain. In such cases, Feature Uncertainty ends up squandering queries on such words ending up with performance inferior to random feature queries. What works significantly better in practice is *Feature Certainty*, that acquires labels for features in *descending* order of the uncertainty scores [85, 58]. Alternative uncertainty-based heuristics have also been used with different degrees of success [25, 31].

**Expected feature utility**

Selecting features that the current model is uncertain about may results in queries that are not useful in discriminating between classes. On the other hand, selecting the most certain features is also suboptimal, since queries may be wasted simply confirming confident predictions, which is of limited utility to the model. An alternative to such certainty-based heuristics, is to directly estimate the expected value of acquiring each feature label. This can be done by instantiating the Expected Utility framework described in Section 1.2.2, for this setting. This results in the decision-theoretic optimal policy, which is to ask for feature labels which, once incorporated into the data, will result in the highest increase in classification performance in *expectation*.

More precisely, if $f_j$ is the label of the $j$-th feature, and $q_j$ is the query for this feature's label, then the Expected Utility of a feature query $q_j$ can be computed as:

$$EU(q_j) = \sum_{k=1}^{K} P(f_j = c_k)\mathcal{U}(f_j = c_k) \tag{1.6}$$

Where $P(f_j = c_k)$ is the probability that $f_j$ will be labeled with class $c_k$, and $\mathcal{U}(f_j = c_k)$ is the utility to the model of knowing that $f_j$ has the label $c_k$. As in other applications of this framework, the true values of these two quantities are unknown, and the main challenge is to accurately estimate these quantities from the data currently available.

A direct way to estimate the utility of a feature label is to measure expected classification accuracy. However, small changes in the probabilistic model that result from acquiring a single additional feature label may not be reflected by a change in accuracy. As in active feature-value acquisition (see Section 1.3) one can use a finer-grained measure of classifier performance, such as Log Gain defined in Equation 1.5. Then the utility of a classifier, $\mathcal{U}$, can be measured by summing the Log Gain for all instances in the training set.

In Eq. 1.6, apart from the measure of utility, we also do not know the true probability distribution of labels for the feature under consideration. This too can be estimated from the training data, by seeing how frequently the word appears in documents of each class. For Pooling Multinomials, one can use the model parameters to estimate the feature label distribution, $\hat{P}(f_j = c_k) = \frac{P(f_j|c_k)}{\sum_{k=1}^{K} P(f_j|c_k)}$. Given the estimated values of the feature-label distribution and the utility of a particular feature query outcome, we can now estimate the Expected Utility of each unknown feature, selecting the features with the highest Expected Utility for labeling.

As in other settings, this approach can be computationally intensive if Expected Utility estimation is performed on all unknown features. In the worst case this requires building and evaluating models for each possible outcome of each unlabeled feature. In a setting with $m$ features and $K$ classes, this approach requires training $O(mK)$ classifiers. However, the complexity of the approach can be significantly alleviated by only applying Expected Utility evaluation to a sub-sample of all unlabeled features. Given a large number

of features with no true class labels, selecting a sample of available features uniformly at random may be sub-optimal. Instead one can subsample features based on a fast and effective heuristic like Feature Certainty [85]. Figure 1.3 shows the typical advantage one can see using such a decision-theoretic approach versus uncertainty-based approaches.

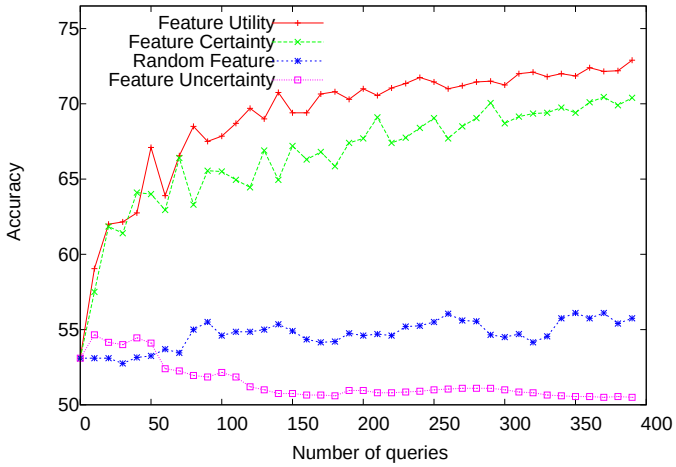


**FIGURE 1.3**: Comparison of different approaches for actively acquiring feature labels, as demonstrated on the Pooling Multinomials classifier applied to the *Movies* [60] data set.

### 1.4.3 Active Dual Supervision

Since dual supervision makes it possible to learn from labeled examples and labeled features simultaneously, one would expect more labeled data of either form to lead to more accurate models. Fig. 1.4 illustrates the influence of increased number of instance labels and feature labels independently, and also in tandem. The figure presents an empirical comparison of three schemes: Instances-then-features, Features-then-instances, and Passive Interleaving. As the name suggests, *Instances-then-features*, provides labels for randomly selected instances until all instances have been labeled, and then switches to labeling features. Similarly, *Features-then-instances* acquires labels for randomly selected features first and then switches to getting instance labels. In *Passive Interleaving* we probabilistically switch between issuing queries for randomly chosen instance and feature labels.

We see from Fig. 1.4 that fixing the number of labeled features, and increasing the number of labeled instances steadily improves classification accuracy. This is what one would expect from traditional supervised learning curves. More interestingly, the results also indicate that we can fix the number of instances, and improve accuracy by labeling more features. Finally, results on Passive Interleaving show that though both feature labels and example labels are beneficial by themselves, dual supervision which exploits the interaction of examples and features does in fact benefit from acquiring both types of labels concurrently.

In the sample results above, we selected instances and/or features to be labeled uniformly at random. Based on previous work in active learning one would expect that we can select instances to be labeled more efficiently, by having the learner decide which instances it is most likely to benefit from. The results in the previous section show that actively selecting features to be labeled is also beneficial. Furthermore, the Passive Interleaving results

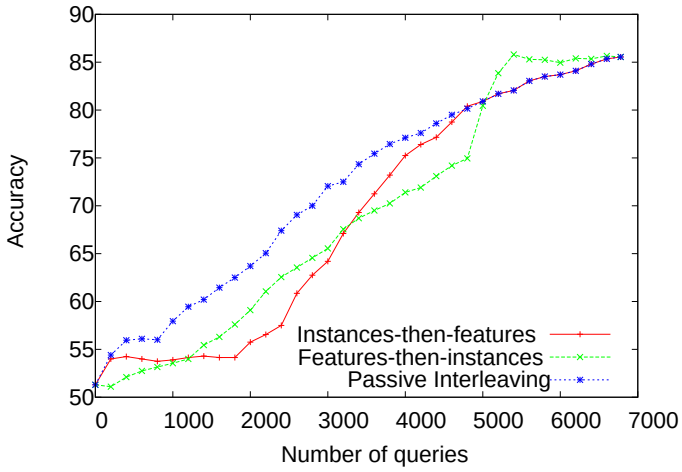


**FIGURE 1.4**: Comparing the effect of instance and feature label acquisition in dual supervision. At the end of the learning curves, each method has labels for all available instances and features; and as such, the last points of all three curves are identical.

suggest that an ideal active dual supervision scheme would actively select both instances and features for labeling. This setting is illustrated in Figure 1.5.

One could apply an Uncertainty Sampling approach to this problem. However, though uncertainty scores can be used to order examples or features by themselves, there is no principled way to compare an uncertainty score computed for an example with a score for a feature. This is because these scores are based on different heuristics for examples and features, and are not in comparable units. One alternative is to apply uncertainty-based active learning schemes to select labels for examples and features separately. Then, at each iteration of active dual supervision, randomly choose to acquire a label for either an example or feature, and probe the corresponding active learner. Such an Active Interleaving approach is in general more effective than the active learning of either instances or features in isolation [85]. While easy to implement, and effective in practice, this approach is dependent on the ad hoc selection of the *interleave probability* parameter, which determines how frequently to probe for an example versus a feature label. This approach is indeed quite sensitive to the choice of this interleave probability [85]. An ideal active scheme should, instead, be able to assess if an instance or feature would be more beneficial at each step, and select the most informative instance or feature for labeling.

Fortunately, the Expected Utility method is very flexible, capable of addressing both types of acquisition within a single framework. Since the measure of utility is independent of the type of supervision and only dependent on the resulting classifier, we can estimate the expected utility of different forms of acquisitions in the same manner. This yields a holistic approach to active dual supervision, where the Expected Utility of an instance or feature label query, $q$, can be computed as

$$EU(q) = \sum_{k=1}^{K} P(q = c_k) \frac{\mathcal{U}(q = c_k)}{\omega_q} \tag{1.7}$$

where $\omega_q$ is the cost of the query $q$, $P(q = c_k)$ is the probability of the instance or feature queried being labeled as class $c_k$, and utility $\mathcal{U}$ can be computed as in Eq. 1.5. By evaluating

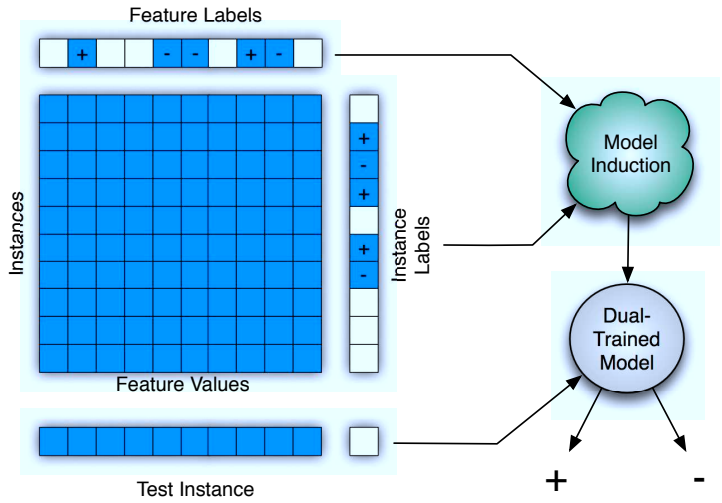


FIGURE 1.5: Active Dual Supervision

instances and features in the same units, and by measuring utility per unit cost of acquisition, such a framework facilitates explicit optimization of the trade-offs between the costs and benefits of the different types of acquisitions. As with Feature Utility, query selection can be sped up by sub-sampling both examples and features, and evaluating the Expected Utility on this candidate set. It has been demonstrated that such a holistic approach does indeed effectively manage the trade-offs between the costs and benefits of the different types of acquisitions, to deterministically select informative examples or features for labeling [2]. Figure 1.6 shows the typical improvements of this unified approach to active dual supervision over active learning for only example or feature labels.

## 1.5 Dealing with noisy acquisition

We have discussed various sorts of information that can be acquired (actively) for training statistical models. Most of the research on active/selective data acquisition either has assumed that the acquisition sources provide perfect data, or has ignored the quality of the data sources. Let's examine this more critically. Here, let's call the values that we will acquire "labels". In principle these could be values of the dependent variable (training labels), feature labels, feature values, etc., although the research to which we refer has focused exclusively on training labels.

In practical settings, it may well be that the labeling is not 100% reliable—due to imperfect sources, contextual differences in expertise, inherent ambiguity, noisy information channels, or other reasons. For example, when building diagnostic systems, even experts are found to disagree on the "ground truth": "no two experts, of the 5 experts surveyed, agreed upon diagnoses more than 65% of the time. This might be evidence for the differences that exist between sites, as the experts surveyed had gained their expertise at different locations. If not, however, it raises questions about the correctness of the expert

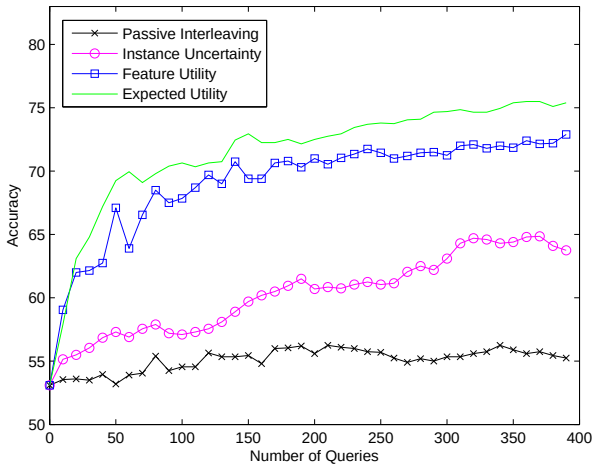


**FIGURE 1.6**: The effectiveness of Expected Utility instantiated for active dual supervision, compared to alternative label acquisition strategies.

data" [62]. The quality of selectively acquired data recently has received greatly increased attention, as modelers increasingly have been taking advantage of low-cost human resources for data acquisition. Micro-outsourcing systems, such as Amazon's Mechanical Turk (and others) are being used routinely to provide data labels. The cost of labeling using such systems is much lower than the cost of using experts to label data. However, with the lower cost can come lower quality.

Surveying all the work on machine learning and data mining with noisy data is beyond the scope of this chapter. The interested reader might start with some classic papers [86, 51, 83]. We will discuss some work that has addressed strategies for the *acquisition* of data specifically from noisy sources to improve data quality for machine learning and data mining (the interested reader should also see work on information fusion [18].

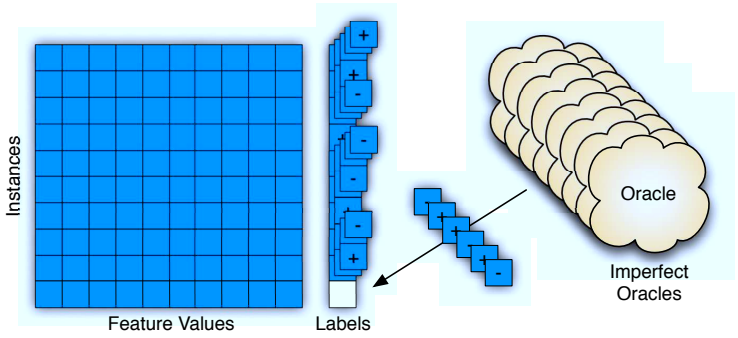


FIGURE 1.7: Multiple Noisy Oracles

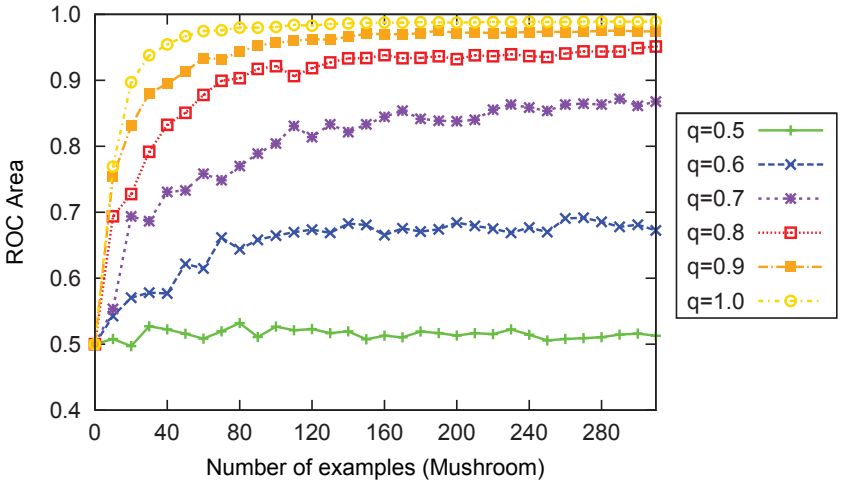


**FIGURE 1.8**: Learning curves under different quality levels of training data ($q$ is the probability of a label being correct).

### 1.5.1 Multiple label acquisition: round-robin techniques

If data acquisition costs are relatively low, a natural strategy for addressing data quality is to acquire the same data point multiple times, as illustrated in Figure 1.7. This assumes that there is at least some independence between repeated acquisitions of the same data point, with the most clean-cut case being that each acquisition is an independent random draw of the value of the variable, from some noisy labeling process. We can define the class of *generalized round-robin* (GRR) labeling strategies. [82]: request a label for the data point that currently has the fewest labels. Depending on the process by which the rest of the data are acquired/selected, generalized round-robin can be instantiated differently. If there is one fixed set of data points to be labeled, GRR becomes the classic *fixed round-robin*: cycle through the data points in some order obtaining another label for each. If a new data point is presented every $k$ labels, then GRR becomes the common strategy of acquiring a fixed number of labels on each data point [88, 87].

Whether one ought to engage in round-robin repeated labeling depends on several factors [82], which are clarified by Figure 1.8. Here we see a set of learning curves[2] for the classic mushroom classification problem from the UCI repository [12]. The mushroom domain provides a useful illustration because with perfect labels and a moderate amount of training, predictive models can achieve perfect classification. Thus, we can examine the effect of having noisy labels and multiple labels. The several curves in the figure show learning curves for different labeling qualities—here simply the probability that a labeler will give the correct label for this binary classification problem. As the labeler quality deteriorates, not only does the generalization performance suffer, but importantly the rate-of-change of the generalization performance as a function of the number of labeled data suffers markedly.

Thus, we can summarize the conditions under which we should consider repeated labeling [82]:

1. What is the relative cost of acquiring labels for data points, as compared to the cost of

[2]Generalization accuracy estimated using cross-validation with a classification tree model.

acquiring the "unlabeled" part of the data item? The cheaper the labels are, relatively speaking, the more one can afford to acquire for each unlabeled data point.

2. How steep is the (single-label) learning curve? Specifically, if the rate of change of generalization performance (however measured) is high, as a function of labeling new instances, it may instead be best to label new instances rather than to get additional labels for already-labeled instances. As illustrated in Figure 1.8, learning curves generally are steeper when relatively few data points have been labeled, and when the labeling is relatively good.

3. How steep is the gradient of generalization performance as a function of increasing the number of labelers? This is a complicated function of the individual labeler quality and the independence of the labelers. For the former, note that at the extremes one will see no improvement with increasing the number of labelers—when the labelers are either perfect or they are completely random. The largest improvement-per-label comes for mid-range quality. With respect to independence, obviously repeated labeling is wasted in the trivial scenario that the labelers all provide the same labels for the same data points, and maximal value will come when the errors the labelers make are completely independent. In Figure 1.8, getting additional (non-trivial) labels for the already-labeled data points would correspond to moving up to a higher-quality curve.

### 1.5.2 Multiple label acquisition: selecting examples to re-label

The round-robin techniques described above repeatedly label (re-label) data instances indiscriminantly. However, if we have the opportunity to monitor the repeated labeling process, intuitively it seems that we would want to select cases carefully for re-labeling. For example, all else equal we would rather get another label on a case with label multiset {+,-,-,+} than one with label multiset {+,+,+,+}. Why? Because we are less certain about the true label in the former case than in the latter. As in other settings, we can formulate our strategy as one of uncertainty reduction (cf., Section 1.2); a difference is that here we examine the uncertainty embodied by the current label multiset. It is important to distinguish here between how mixed up the label set is, as measured for example by its entropy, and our uncertainty in the underlying label. For example, a label multiset with 600 +'s and 400 -'s has high entropy, but if we are expecting high-noise labelers we may be fairly certain that this is a +. Thus, we should compute careful statistical estimates of the certainty of the label multiset, in order to select the highest uncertainty sets for re-labeling; doing so gives consistently better performance than round-robin repeated labeling [82, 35]. Let's call that the *label uncertainty*.

For selective repeated labeling, the label uncertainty is only one sort of information that can be brought to bear to help select cases to re-label. Alternatively, one can learn a predictive model from the current set of re-labeled data, and then examine the uncertainty in the *model* for each case, for example how close the probability estimated by a model or ensemble of models is to the classification threshold, using the measures introduced earlier for uncertainty sampling (Section 1.2). Let's call this *model uncertainty* or MU. Model uncertainty also can identify important cases to relabel; however, its performance is not as consistent as using label uncertainty [82]. Interestingly, these two notions of uncertainty are complementary and we can combine them and prefer selecting examples for which both the label multiset and the model are uncertain of the true class, for example by using the geometric mean of the two measures [82]. This *label-and-model uncertainty* is significantly superior to using either sort of uncertainty alone and therefore also is superior to round-robin repeated labeling [82].

Although model uncertainty uses the same measures as Uncertainty Sampling, there is a major difference. The difference is that to compute MU the model is applied back to the cases from which it was trained; mislabeled cases get systematically higher model uncertainty scores [35]. Thus, model uncertainty actually is more closely akin to methods for finding labeling errors [14]: it selects cases to label because they are mislabeled, in a "self-healing" process, rather than because the examples are going to improve a model in the usual active-learning sense. This can be demonstrated by instead using cross-validation, applying the model to held-out cases (active-learning style) and then relabeling them; we see that most of MU's advantage disappears [35].

### 1.5.3 Using multiple labels and the dangers of majority voting

Once we have decided to obtain multiple labels for some or all data points, we need to consider how we are going to use multiple labels for training. The most straightforward method is to integrate the multiple labels into a single label by taking an average (mode, mean, median) of the values provided by the labelers. Almost all research in machine learning and data mining uses this strategy, specifically taking the majority (plurality) vote from multiple classification labelers.

While being straightforward and quite easy to implement, the majority vote integration strategy is not necessarily the best. Soft labeling can improve the performance of the resultant classifiers, for example by creating an example for each class weighted by the proportion of the votes [82]. Indeed, soft-labeling can be made "quality aware" if knowledge of the labeler qualities is available (cf., a quality-aware label uncertainty calculation [35]). This leads to a caution for researchers studying strategies for acquiring data and learning models with noisy labelers: showing that our new learning strategy improves modestly over majority voting may not be saying as much as we think, since simply using a better integration strategy (e.g., soft labeling) also shows modest improvements over majority voting (in many cases).

A different, important danger of majority voting comes when labelers can have varying quality: a low-quality labeler will "pull down" the majority quality when voting with higher-quality labelers. With labelers who make independent errors, there is a rather narrow range of quality under which we would want to use majority voting [43, 82]. An alternative is to use a quality-aware technique for integrating the labels.

### 1.5.4 Estimating the quality of acquisition sources

If we want to eliminate low-quality sources or to take their quality into account when coalescing the acquired data, we will have to know or estimate the quality of the sources—the labelers. The easiest method for estimating the quality of the labelers is to give them a reasonably large quantity of "gold standard" data, for which we know the truth, so that we can estimate error statistics. Even ignoring changes in quality over time, the obvious drawback is that this is an expensive undertaking. We would prefer not to waste our labeling budget getting labels on cases for which we already know the answer.

Fortunately, once we have acquired multiple labels on multiple data points, even without knowing any true labels we can estimate labeler quality using a maximum likelihood expectation maximization (EM) framework [19]. Specifically, given as input a set of $N$ objects, $o_1, \ldots, o_N$, we associate with each a *latent* true class label $T(o_n)$, picked from one of the $L$ different possible labels. Each object is annotated by one or more of the $K$ labelers. To each labeler $(k)$ we assign a *latent* "confusion matrix" $\pi_{ij}^{(k)}$, which gives the probabilty that worker $(k)$, when presented with an object of true class $i$, will classify the object into cat-

egory $j$. The EM-based technique simultaneously estimates the latent true classes and the labeler qualities. Alternatively, we could estimate the labeler confusion matrices using a Bayesian framework [16], and can extend the frameworks to the situation where some data instances are harder to label correctly than others [96], and to the situation where labelers differ systematically in their labels (and this bias can then be corrected to improve the integrated labels) [36]. Such situations are not just figments of researchers' imaginations. For example, a problem of contemporary interest in on-line advertising is classifying web content as to the level of objectionability to advertisers, so that they can make informed decisions about whether to serve an ad for a particular brand on a particular page. Some pages may be classified as objectionable immediately; other pages may be much more difficult, either because they require more work or because of definition nuances. In addition, labelers do systematically differ in their opinions on objectionability: one person's R-rating may be another's PG-rating.

If we are willing to estimate quality in tandem with learning models, we could use an EM procedure to iterate the estimation of the quality with the learning of models, with estimated-quality-aware supervision [67, 68].

### 1.5.5 Learning to choose labelers

We possibly can be even more selective. As illustrated in Figure 1.9, we can select from among labelers as their quality becomes apparent, choosing particularly good labelers [22, 99] or eliminating low-quality labelers [23, 20]. Even without repeated labeling, we can estimate quality by comparing the labelers' labels with the predictions from the model learned from all the labeled examples—effectively treating the model predictions as a noisy version of the truth [20]. If we do want to engage in repeated labeling, we can compare labelers with each other and keep track of confidence intervals on their qualities; if the upper limit of the confidence interval falls too low, we can avoid using that expert in the future [23].

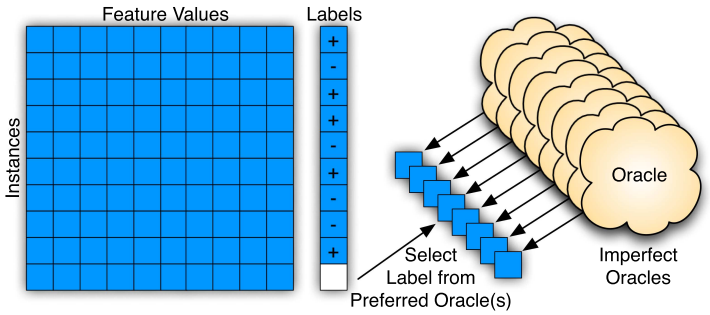


FIGURE 1.9: Selecting Oracle(s)

What's more, we can begin to consider that different labelers may have different sorts of expertise. If so, it may be worthwhile to model labeler quality conditioned on the data instance, working toward selecting the labeler best suited to each specific instance [22, 99].

### 1.5.6 Where to go from here?

Research on selective data acquisition with noisy labelers is still in its infancy. Little has been done to compare labeler selection strategies with selective repeated labeling strategies and with sophisticated label integration strategies. Moreover, little work has addressed integrating all three. It would be helpful to have procedures that simultaneously learn models from multiple labelers, estimate the quality of the labelers, combine the labels accordingly, all the while selecting which examples to re-label, and which labelers to apply to which examples. Furthermore, learning with multiple, noisy labelers is a strict generalization of traditional active learning, which may be amenable to general selective data acquisition methods such as expected utility [22] (cf., Section 1.2). Little has been done to unify theoretically or empirically the notions of repeated labeling, quality-aware selection, and active learning.

Most of these same ideas apply beyond the acquisition of training labels, to all the other data acquisition scenarios considered in this chapter: selectively acquired data may be noisy, but the noise may be mitigated by repeated labeling and/or estimating the quality of labelers. For example, we may want to get repeated feature values when their acquisition is noisy, for example because they are the result of error-prone human reporting or experimental procedures.

---

## 1.6 Prediction Time Information Acquisition

Traditional active information acquisition is focused on the gathering of instance labels and feature values at training time. However, in many realistic settings, we may also have the need or opportunity to acquire information when the learned models are used (call this "prediction time"). For example, if features are costly at training time, why would these features not incur a similar cost at prediction time? Extending the techniques of active feature-value acquisition, we can address the problem of procuring feature values at prediction time. This additional information, can, in turn, potentially offer greater clarity into the state of the instance being examined, and thereby increase predictive performance, all at some cost. In an analogous setting to prediction time active feature value acquisition, if an oracle is available to provide class labels for instances at training time, the same oracle may be available at test time, providing a supplement to error-prone statistical models, with the intent of reducing the total misclassification cost experienced by the system consuming the class predictions. This objective of this section is to develop motivating settings suitable for predictive time information acquisition, and to explore effective techniques for gathering information cost-effectively.

### 1.6.1 Active Inference

At first blush, it might seem that prediction-time acquisition of training labels would not make sense: if labels are available for the instances being processed, then why perform potentially error prone statistical inference in the first place? While "ground truth" labels are likely to be preferable to statistical prediction, all things being equal, such an equal comparison ignores the cost-sensitive context in which the problem is likely to exist. Acquisition of ground truth labels may be more costly than simply making a model-based prediction; however, it may nonetheless be more *cost-effective* to acquire particular labels

at this heightened cost depending on the costs incurred from making certain errors in prediction. We refer to this prediction-time acquisition of label information as *Active Inference*.

More formally, given a set of $n$ discrete classes, $c_j \in \mathcal{C}$, $j = 1, \ldots, n$, and some cost function, cost$(c_k|c_j)$, yielding the penalty for predicting a label $c_k$ for an example whose true label is $c_j$, the optimal model-based prediction for a given $x$ is then: $c = \arg\min_c \sum_j \hat{P}(c_j|x)\text{cost}(c|c_j)$, where $\hat{P}(c_j|x)$ represents a model's estimated posterior probability of belonging in class $c_j$ given an instance $x$. The total expected cost on a given set $\mathbb{T}$ of to-be-classified data is then:

$$\mathcal{L}_{\mathbb{T}} = \sum_{x \in \mathbb{T}} \phi(x) \min_c \sum_j \hat{P}(c_j|x)\text{cost}(c|c_j) \tag{1.8}$$

Where $\phi(x)$ is the number of times a given example $x$ appears during test time. Note that unless stated otherwise, $\phi(x) = 1$, and can therefore simply be ignored. This is the typical case for pool-based test sets where each instance to be labeled is unique.

Given a budget, $B$, and a cost structure for gathering labels for examples at prediction time, $C(x)$, the objective of an active inference strategy is to then select a set of examples for which to acquire labels, $\mathcal{A}$, such that the expected cost incurred is minimized, while adhering to the budget constraints:

$$\begin{aligned} \mathcal{A} = \arg\min_{\mathcal{A}' \subset \mathbb{T}} & \sum_{x \in \mathbb{T} \setminus \mathcal{A}'} \phi(x) \min_c \sum_j \hat{P}(c_j|x)\text{cost}(c|c_j) \\ & + \sum_{x \in \mathcal{A}'} C(x) \\ & \text{s.t. } B \geq \sum_{x \in \mathcal{A}'} C(x) \end{aligned} \tag{1.9}$$

Given a typical setting of evaluating a classifier utilizing only local feature values on a fixed set of test instances drawn without replacement from $P(x)$, choosing the optimal inference set, $\mathcal{A}$ is straight forward. Since the labels of each instance are considered to be i.i.d., the utility for acquiring a label on each instance given in the right side of Equation 1.9 can be calculated independently, and since the predicted class labels are uncorrelated, greedy selection can be performed until either the budget is exhausted or further selection is no longer beneficial. However, there are settings where active inference is both particularly useful and particularly interesting: while performing *collective inference*, where network structure and similarity amongst neighbors is specifically included while labeling, and *online classification*, where instances are drawn with replacement from some hidden distribution. The remainder of this section is dedicated to the details of these two special cases.

#### 1.6.1.1 Active Collective Inference

By leveraging the correlation amongst the labels of connected instances in a network, collective inference can often achieve predictive performance beyond that which is possible through classification using only local features. However, when performing collective inference in settings with noisy local labels (e.g. due to imperfect local label predictions, limitations of approximate inference, or other noise), the blessing of collective inference may become a curse; incorrect labels are propagated, effectively multiplying the number of mistakes that are made. Given a trained classifier and a test network, intent of *Active Collective Inference* is to carefully select those nodes in a network for which to query an oracle for a "gold standard" label that will be hard set when performing collective inference, such that the collective generalization performance is maximally improved [66].

Unlike the traditional content-only classification setting, in collective classification the label, $y$, of a given to a particular example, $x$, depends not only on the features used to represent $x$, but on the labels and attributes of $x$'s neighbors in the network being considered. This makes estimation of the benefits of acquiring a single example's label for active inference challenging; the addition may alter the benefits of all other nodes in the graph, and approximation is generally required in order to make inference tractable. Furthermore, as in other data acquisition tasks (Section 1.2) finding the optimal set is known to be an NP-hard problem as it necessitates the investigation of all possible candidate active inference sets [9, 10, 11]. Because of the computational difficulties associated with finding the optimal set of instances to acquire for active inference, several approximation techniques have been devised that enable a substantial reduction in misclassification cost while operating on a limited annotation budget.

First amongst the approximation techniques for active collective inference are so-called *connectivity metrics*. These metrics rely solely on the graphical structure of the network in order to select those instances with the greatest level of connectivity within the graph. Measures such as closeness centrality (the average distance from node $x$ to all other nodes), and betweenness centrality (the proportion of shortest paths passing through a node $x$) yield information about how central nodes are in a graph. By using graph k-means and utilizing cluster centers, or simply using measures such as degree (the number of connections), locally well-connected examples can be selected. The intent of connectivity-based active inference is to select those nodes with the greatest *potential* influence, without considering any of the variables or labels used in collective inference [66].

*Approximate Inference and Greedy Acquisition* attempts to optimize the first order Markov relaxation of the active inference utility given in Equation 1.9. Here, the expected utility of each instance is assessed and the instance with the most promise is used to supplement $\mathcal{A}$. The utility for an acquiring a particular $x$ then involves an expected value computation over all possible label assignments for that $x$, where the value given to each label assignment is the expected network misclassification cost given in Equation 1.8. The enormous computational load exerted here is eased somewhat through the use of approximate inference [9, 10, 11].

An alternative approach to active collective inference operates by building a collective model in order to predict whether or not the predictions at each node are correct, $P(c|x)$. The effectiveness of acquiring the label for a particular $x_i$ is assumed to be a function of the gradient of $P(c|x_j)$ with respect to a change in $P(c|x_i)$ for all $x_j \in \mathbb{T}$, and the change in prediction correctness for the $x_i$ being considered. It is important to consider both whether a change in the label of a $x_i$ can change many (in)correct labels, as well as how likely it is that $x_i$ is already correctly classified. Appropriate choice of the functional form of $P(c|x)$ leads to efficient analytic computation at each step. This particular strategy is known as *viral marketing acquisition*, due to it's similarity with assumptions used in models used for performing marketing in network settings. Here the effectiveness of making a particular promotion is a function of how that promotion influences the target's buying habits, and how that customer's buying habits influence others [9, 10, 11].

Often, the mistakes in prediction caused by collective inference tend to take the form of closely linked "islands". Focusing on eliminating these islands, *reflect and correct* attempts to find centers of these incorrect clusters and acquire their labels. In order to do so, reflect and correct relies on the construction of a secondary model utilizing specialized features believed to indicate label effectiveness. Local, neighbor-centric, and global features are derived to build a predictive model used to estimate the probability that a given $x_i$'s label is incorrect. Built using the labeled network available at test time, this "correctness model" is then applied to the collective inference results at test time. Rather than directly incorporating the utility of annotating a $x_i$ to reduce the overall misclassification cost given in

Equation 1.8, reflect and correct leverages an uncertainty-like measurement, seeking the example likely to be misclassified with the greatest number of misclassified neighbors. This is akin to choosing the center of the largest island of incorrectness [9, 10, 11].

#### 1.6.1.2 Online Active Inference

Many realistic settings involve online prediction; performing classification on instances drawn sequentially and with replacement from some hidden distribution. Example applications include query classification, traffic analysis at web servers, classification of web pages seen in ad server logs and web logs, and marketing decisions faced in online advertising. Because instances may be seen repeatedly as time progresses, the expected frequency of occurrence for each instance $x$ may be non-zero. As a result, the expected misclassification cost given in Equation 1.8 may be heavily influenced by the frequency of certain examples and the cumulative cost their repetition imparts.

For many of the problems faced on the web, this problem imparted by repeated encounters with certain $x$'s becomes particularly acute; heavy-tailed distributions imply that even though a great many unique instances may be encountered, a handful of instances may impart a majority of the total possible misclassification cost. Fortunately, it is often possible to defer the predictive model's predictions to an oracle, sacrificing a one-time labeling cost in order minimize future misclassification costs for particular examples. The task on this *online active inference* is then to select those instances from the example stream for "gold standard" labeling offering the greatest reduction in "impression-sensitive" expected loss given in Equation 1.8 while factoring in label acquisition costs and adhering to a limited budget [3].

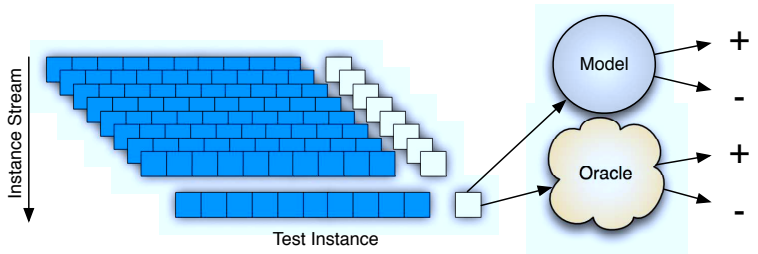


**FIGURE 1.10**: Online Active Inference: presented with an incoming example, a system must decide if it is to sent a model's predicted label to an end system, or pass the example to an oracle for more reliable, albeit more expensive annotation.

A typical use case of online active inference is seen in Figure 1.10. For each example $x$ seen in the stream, the active inference system computes an expected label acquisition benefit, $EU(x) = \left[\hat{\phi}(x) \min_c \sum_j \hat{P}(c_j|x)\text{cost}(c|c_j)\right] - C(x)$, where $C(x)$ is the label acquisition cost for $x$. As in other active inference settings, an ideal system would seek the $x \in \mathcal{A}$ optimizing $\sum_{x \in \mathcal{A}} EU(x)$ constrained such that $\sum_{x \in \mathcal{A}} C(x) \leq B$.

However, the online, stream-based setting imparts several complications when making the above optimization. First, the estimated probability, $P(x)$, and associated $\phi(x)$, is unknown a priori, and must be estimated on the fly. This estimation relies on the established field of univariate density estimation, or using the feature values of each $x$ in order to compute a conditional density of each $x$. Second, each time a given $x$ is encountered, this instance's expected misclassification cost is reduced. Accurate utility measurements

require that this reduction in expected cost should be extrapolated to the future. A third complication stems from how the budget, $B$ is set. A budget-per-instance a one-time fixed budget, and periodic fixed budgets yield different objective functions, complicating the task faced in online active inference.

A fourth and particularly insidious difficultly stems from the fact that $P(x)$ is unknown. This implies that a system sampling a stream of examples from $P(x)$ may not even know the set of unique $x$'s that can potentially be encountered; as a consequence, the set of utilities under consideration. However, it is possible to build a distribution on the expected utility values conditioned on the probability of at least one occurrence during the time period under consideration. An appropriate lower bound, $\tau$ can then be chosen whereby labeling all examples with a utility greater than $\tau$ gathers as many of the highest utility examples as is possible while exhausting the available budget during that time period [3].

While labeling for the purpose of online active inference, it is also appealing to consider how these labeled examples reduce overall cost via incorporation into the training set and updating the underlaying model. Indeed, it is reasonable to consider how to allocate a single budget in order to achieve the best overall performance, while taking into account both active learning and active inference. While it is an interesting direction to consider how expected utility-optimizing techniques for active learning can be incorporated into the same scale as the proposed technique for online active inference, it can be shown that by those examples with the hightest $EU(x)$ for active inference doubles as a generalized uncertainty sampling strategy for active learning potentially offering reasonable label choices for the task of model improvement while explicitly reducing misclassification cost [3].

### 1.6.2 Prediction Time Active Feature Value Acquisition

In many domains we can acquire additional information for selected test instances that may help improve our classification performance on the test set. When this additional information comes at a cost, or is potentially noisy, it is best to actively select the instances that are most likely to benefit from acquiring additional information. This problem of prediction-time Active Feature-value Acquisition is analogous to AFA during induction as discussed in Section 1.3. This setting has been studied in the context of customer targeting models [37], where, when making a decision for a customer, additional information may be purchased, if necessary, to make a more informed decision.

At the time of classifier induction, class labels are available for all instances including the incomplete instances. This information can be used effectively to estimate the potential value of acquiring more information for the incomplete instances. However, this label information is obviously not present during prediction on test instances. Yet, we must evaluate the benefit of acquiring additional features for an incomplete instance, versus making a prediction using only incomplete feature information.

However, uncertainty-based heuristics are still applicable in this setting. In particular, given a set of incomplete test instances, one can apply a previously-trained classifier to predict the class membership probabilities of these instances, using only the feature-values that are known. The uncertainty of these predictions can be computed using a measure such as unlabeled margin in Equation 1.2. Then all the feature-values for the most uncertain instances can be acquired until the acquisition budget is exhausted.

The above Uncertainty Sampling strategy aims to obtain more information about an uncertain prediction, with the hope that it will allow a more certain prediction, which in turn is assumed to result in a more accurate prediction. Even if more certain predictions are likely to be more accurate, acquiring additional information on the most uncertain instance may not result in the highest payoff. For example, if an instance is inherently ambiguous. Alternatively, we can select additional features-values only if they are *expected* to reduce

the uncertainty in classification after acquisition. Again, this can be viewed as a special case of the Expected Utility framework, where utility $\mathcal{U}$ in Equation 1.4 can be measured by the log of the unlabeled margin [37].

Another factor to be taken into consideration at prediction-time is the cost of misclassifying an instance. Often misclassification costs are non-symmetric, such as the cost of misclassifying a malignant tumor as being benign. In such cases, one needs to weigh the cost of acquisitions with the cost of misclassification for each test instance. When acquisition and misclassification costs can be represented in the same units, we can selectively acquire feature values, so as to minimize the sum of the acquisition cost and expected misclassification cost [81].

Just as misclassification costs associated with different types of errors may not be uniform, it is also often possible to acquire arbitrary subsets of feature values, each at a different cost, a setting that has not been extensively explored in the research literature. It may be natural to extend the Expected Utility framework to consider sets of categorical features. However, estimating the joint probability distribution of all possible sets of values may render such a policy hopelessly inefficient. To overcome the constraints of this computational complexity, one may consider only potentially relevant feature subsets for acquisition, by combining innovative data structures and dynamic programming for incrementally updating the search space of informative subsets as new evidence is acquired—exploiting dependencies between missing features so as to share information value computations between different feature subsets, making the computation of the information value of different feature subsets tractable [8].

Finally, if we know that feature values can be acquired at test time, it would make sense to account for this at the time of classifier induction. For example, by avoiding expensive features at induction that may result in a more accurate model, but are not worth the cost at prediction time. Such a setting can be addressed in a budgeted learning framework where a learner can spend a fixed learning budget $b_l$ to acquire training data, so as to produce a classifier that spends at most $b_c$ per test instance[38].

## 1.7 Alternative acquisition settings

There are other sorts of data that can be acquired at a cost to improve machine learning, besides the values for individual variables for the construction of predictive models. This section covers some alternative settings for applying a modeling budget. This section begins with a treatment of the active collection of feature values for the purpose of performing unsupervised learning. Here, additional information may be acquired in order to improve the quality of tasks such as clustering.

This section then continues to investigate the selection of entire examples by class, in order that the training data comprise a "better" proportion of the constituent classes. We then discuss the related scenario where a human can be deployed to *search* for relevant information, for example, instances belonging to a given class, as well as feature values and feature labels. Finally, we present the important and often-ignored fact that in many practical situations, learned models are applied in a decision-making context, where maximizing the accuracy (or some other decision-agnostic measure) of one model is not the ultimate goal. Rather, we want to optimize decision making. Decision-focused strategies may acquire different data from model-focused strategies.

### 1.7.1 Information Acquisition for Unsupervised Learning

Almost all work in information acquisition has addressed supervised settings. However information may also be acquired for unstructured tasks, such as clustering. Interestingly, different clustering approaches may employ different types of information which may be potentially acquired at a cost, such as constraints as well as feature values.

Most clustering policies assume that each instance is represented by a vector of feature values. As is the case in supervised learning, clustering is undermined when the instance representation is incomplete and can therefore benefit from effective policies to improve clustering through cost-effective acquisition of information. Consider for example data on consumers' preferences. Clustering can be applied to this data in order to derive natural groupings of products based on customer preferences, such as for data-driven discovery of movie genres [6]. Clustering has also been used to help produce product/service recommendations to consumers [30]. In all these cases, performance can be improved with cost-effective acquisition of information.

Let us first consider the Expected Utility framework for clustering. On the outset it may appear that, given clustering algorithms typically aim to optimize some objective function, changes in the value of this function can be used to reflect the utility $\mathcal{U}$ from prospective feature value acquisitions. However, recall that typically only different clustering assignments yield changes in the corresponding clustering algorithm's objective function. Yet, as noted in [91], a costly feature value acquisition may alter the value of algorithm's objective function, without changing the assignment of even a single instance into a different cluster. For the popular K-means clustering algorithm, for example, such an approach may select feature values that alter cluster centroid locations (decreasing the total distances between instances and their respective centroids); however, these changes may not change the cluster assignments significantly or at all. The effect of such wasteful acquisitions can be significant. Alternatively, it is useful to consider utilities which capture the impact on clustering *configuration* caused by an acquisition. For example, utility can be measured by the number of instances for which cluster membership changes as the result of an acquisition [91].

Some clustering approaches do not employ a vectorial data representation; instead, the proximity of pairs of instances is provided. Hence, for a data set with $N$ instances, the required information scales with $O(N^2)$. It has been shown [34, 15] that identifying the most informative proximity measures to acquire can be critical for overcoming the inherent data sparseness of proximity data, making such methods feasible in practice. For example, using the expected value of information to quantify the gain from additional information, [34] proposed an algorithm which identifies proximity values so as to minimize the risk from estimating clusters from the existing data. Thus, selected acquisitions aim to minimize the loss from deciding only based on the incomplete information instead of the optimal decision, when all proximity values are known.

Semi supervised clustering also offers interesting opportunities to acquire limited supervision to improve clustering cost-effectively. For example, a subset of instances, must-link and cannot-link constraints can specify instance pairs that must and must-not belong to the same cluster, respectively. Similarly, cluster-level constraints indicate whether or not instances in two different clusters ought to be assigned to the same cluster. To improve clustering, an information acquisition policy may suggest the constraints for which pairs would be most informative to acquire. Similar to active learning, the feature values, from which distances are derived, are known. Acquiring constraints for clustering may

not bear much similarity to instance-level information acquisition for supervised learning. However, the uncertainly reduction principle has been effectively adapted to this setting as well. Specifically, for cluster-level constraint acquisition [39] proposes to acquire constraints when it is most difficult to determine whether two clusters should be merged. In contrast, for acquiring instance-level constraints, [7] proposes a policy which does not follow principles used in supervised settings. In particular, given the impact of initial centroids on clustering outcomes of the K-means clustering algorithm, they propose a constraint acquisition policy which improves the identification of cluster centroids and their neighborhoods with fewer queries. To accomplish this, they employ the farthest first traversal scheme to identify points from *different* clusters using fewer queries. A subsequent consolidation phase, acquires additional pair-wise constraints to explore the structure of each cluster's neighborhood.

In future work it would be interesting to consider policies that can consider different types of information, such as constraints information and feature values, which may be acquired at different costs. However, as reflected in the work we discuss above, and quite differently from the supervised setting, employing the Expected Utility framework for this task may not be the obvious choice in this case.

### 1.7.2 Class-Conditional Example Acquisition

In many model induction tasks, the amount of training data is constrained not by the cost of instance labels, but by the cost of gathering the independent covariates that are used to predict these labels. For example, in certain settings examples can be acquired by class, but gathering the feature values requires costly physical tests and experiments, large amounts of time or computation, or some other source of information that requires some budget outlay. Consider, for example, the problem of building predictive models based on data collected through an "artificial nose" with the intent of "sniffing out" explosive or hazardous chemical compounds [48, 50, 49]. In this setting, the reactivity of a large number of chemicals are already known, representing label-conditioned pools of available instances. However, producing these chemicals in a laboratory setting and running the resultant compound through the artificial nose may be an expensive, time-consuming process. This problem appears to face the inverse of the difficulties faced by active learning—labels essentially come for free, while the independent feature values are *completely* unknown and must be gathered at a cost (let's say, all at once). In this setting, it becomes important to consider the question: "in what proportion should classes be represented in a training set of a certain size?" [95]

Let's call the problem of proportioning class labels in a selection of $n$ additional training instances, "Active Class Selection" (ACS) [48, 50, 49, 95]. This process is exemplified in Figure 1.11. In this setting, there is assumed to be large, class-conditioned (virtual) pools of available instances with completely hidden feature values. At each epoch, $t$, of the ACS process, the task is to leverage the current model when selecting examples from these pools in a proportion believed to have the greatest effectiveness for improving the generalization performance of this model. The feature values for each instance are then collected and the complete instances are added to the training set. The model is re-constructed and the processes is repeated until $n$ examples are obtained (because the budget is exhausted or some other stopping criterion is met, such as a computational limit). Note that this situation can be considered to be a special case of the instance completion setting of Active Feature Acquisition (discussed above). It is a degenerate special case because, prior to selection, there is no information at all about the instances other than their classes.

For ACS, the extreme lack of information to guide selection leads to the development of unique uncertainty and utility estimators, which, in the absence of predictive covariates,

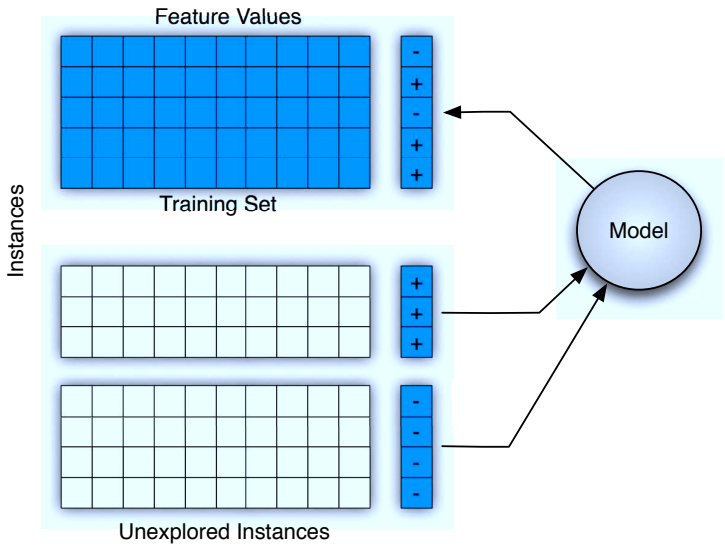


**FIGURE 1.11**: Active Class Selection: Gathering instances from random class-conditioned fonts in a proportion believed to offer greatest improvement in generalization performance.

require unique approximations. While alternative approaches to active class selection have emerged, for thematic clarity, uncertainty-based and expected utility-based approaches will be presented first. Note that because effective classification requires that both sides of a prediction boundary be represented, unlike typical active learning techniques, active class selection typically *samples* classes from their respective score distributions [72, 73].

**Uncertainty-based approaches**
This family of techniques for performing active class selection is based on the volatility in the predictions made about certain classes—those classes whose cross-validated predictions are subject to the most change between successive epochs of instance selection are likely to be based upon an uncertain predictor, and amenable to refinement by the incorporation of additional training data [48, 49]. Analogous to the case of more traditional uncertainty-based data acquisition, several heuristics have been devised to capture the notion of variability.

One measure of the uncertainty of a learned model is how volatile its predictive performance is in the face of new training data. For example, in Figure 1.8 we see various, typical learning curves. With reasonably accurate training data, the modeling is much more volatile at the left side of the figure, showing large changes in generalization performance for the same amount of new training data. We can think that as the predictor gains knowledge of the problem space, it tends to solidify in the face of data, exhibiting less change and greater certainty. For ACS, we might wonder, if the learning curves will be equally steep regardless of the class of the training data [48, 50, 49]. With this in mind, we can select instances at epoch $t$ from the classes in proportion to their improvements in accuracy at $t-1$ and $t-2$. For example, we could use cross-validation to estimate the generalization performance of the classifier with respect to each class, $\mathcal{A}(c)$; class $c$ can then be sampled according to:

$$p^t_{\mathcal{A}}(c) \propto \frac{\max\left\{0, \mathcal{A}^{t-1}(c) - \mathcal{A}^{t-2}(c)\right\}}{\sum_{c'} \max\left\{0, \mathcal{A}^{t-1}(c') - \mathcal{A}^{t-2}(c')\right\}}$$

Alternatively, we could consider general volatility in class members' predicted labels, beyond improvement in the model's ability to predict the class. Again, by using cross-validated predictions at successive epochs, it is possible to isolate members of each class, and observe changes in the predicted class for each instance. For example, when the predicted label of a given instance changes between successive epochs, we can deem the instance to have been *redistricted* [48, 50, 49]. Again considering the level of volatility in a model's predictions to be a measurement of uncertainty, we can sample classes at epoch $t$ according to each classes' proportional measure of redistricting:

$$p^t_{\mathcal{R}}(c) \propto \frac{\frac{1}{|c|}\sum_{x \in c} \mathbb{I}(f^{t-1}(x) \neq f^{t-2}(x))}{\sum_{c'} \frac{1}{|c'|}\sum_{x \in c'} \mathbb{I}(f^{t-1}(x) \neq f^{t-2}(x))}$$

Where $\mathbb{I}(\cdot)$ is an indicator function taking the value of 1 if its argument is true and 0 otherwise. $f^{t-1}(x)$ and $f^{t-2}(x)$ are the predicted labels for instance $x$ from the models trained at epoch $t-1$ and $t-2$ respectively [48, 50, 49].

**Expected Class Utility**

The previously described active class selection heuristics are reliant on the assumption that adding examples belonging to a particular class will improve the predictive accuracy with respect to that class. This does not directly estimate the utility of adding members of a particular class to a model's overall performance. Instead, it may be preferable to select classes whose instances' presence in the training set will reduce a model's misclassification cost by the greatest amount in expectation.

Let cost$(c_i|c_j)$ be the cost of predicting $c_i$ on an instance $x$ whose true label is $c_j$. Then the expected empirical misclassification cost over a sample data set, $\mathbb{D}$ is:

$$\hat{R} = \frac{1}{|\mathbb{D}|}\sum_{x \in \mathbb{D}}\sum_{i} \hat{P}(c_i|x)\text{cost}(c_i|y)$$

Where $y$ is the correct class for a given $x$. Typically in the active class selection setting, this expectation would be taken over the training set (e.g. $\mathbb{D} = T$), preferably using cross-validation. In order to reduce this risk, we would like to select examples from class $c$ leading to the greatest reduction in this expected risk [50].

Consider a predictive model $\hat{P}^{T \cup c}(\cdot|x)$, a model built on the training set, $T$, supplemented with an arbitrary example belonging to class $c$. Given the opportunity to choose an additional class-representative example to the training pool, we would like to select the class that reduces expected risk by the greatest amount:

$$\bar{c} = \arg\max_{c} U(c)$$

Where

$$U(c) = \frac{1}{|\mathbb{D}|}\sum_{x \in \mathbb{D}}\sum_{i} \hat{P}^{T}(c_i|x)\text{cost}(c_i|y) - \frac{1}{|\mathbb{D}|}\sum_{x \in \mathbb{D}}\sum_{i} \hat{P}^{T \cup c}(c_i|x)\text{cost}(c_i|y)$$

Of course the benefit of adding additional examples on a test data set is unknown. Furthermore, the impact of a particular class's examples may vary depending on the feature values of particular instances. In order to cope with these issues, we can estimate via cross-validation on the training set. Using sampling, we can try various class-conditional

additions and compute the expected benefit of a class across that class's representatives in $T$, assessed on the testing folds. The above utility then becomes:

$$\hat{U}(c) = E_{x \in c}\left[\frac{1}{|\mathbb{D}|}\sum_{x \in \mathbb{D}}\sum_{i}\hat{P}^{T}(c_i|x)\text{cost}(c_i|y) - \frac{1}{|\mathbb{D}|}\sum_{x \in \mathbb{D}}\sum_{i}\hat{P}^{T \cup c}(c_i|x)\text{cost}(c_i|y)\right]$$

Note that it is often preferred to add examples in batch. In this case, we may wish to sample from the classes in proportion to their respective utilities:

$$p^{t}_{\hat{U}}(c) \propto \frac{\hat{U}(c)}{\sum_{c'}\hat{U}(c')}$$

Note that diverse class-conditional acquisition costs can be incorporated as in Section 1.2, utilizing $\frac{\hat{U}(c)}{\omega_c}$ where $\omega_c$ is the (expected) cost of acquiring the feature vector of an example in class $c$.

**Alternative approaches to ACS**

In addition to uncertainty-based and utility-based techniques, there are several alternative techniques for performing active class selection. Motivated by empirical results showing that barring any domain-specific information, when collecting examples for a training set of size $n$, a balanced class distribution tends to offer reasonable AUC on test data [92, 95], a reasonable baseline approach to active class selection is simply to select classes in balanced proportion.

Search strategies may alternately be employed in order to reveal the most effective class ratio at each epoch. Utilizing a nested cross-validation on the training set, the space of class ratios can be explored, with the most favorable ratio being utilized at each epoch. Note that it is not possible to explore all possible class ratios in all epochs, without eventually spending too much on one class or another. Thus, as we approach $n$ we can narrow the range of class ratios, assuming that there is a problem-optimal class ratio that will become more apparent as we obtain more data [95].

It should be noted that many techniques employed for building classification models assume an identical or similar training and test distribution. Violating this assumption may lead to biased predictions on test data where classes preferentially represented in the training data are predicted more frequently. In particular "increasing the prior probability of a class increases the posterior probability of the class, moving the classification boundary for that class so that more cases are classified into that class" [76, 61]. Thus in settings where instances are selected specifically in proportions different from those seen in the wild, posterior probability estimates should be properly calibrated to be aligned with the test data, if possible [61, 26, 95].

### 1.7.3 Guided Selection

A much more general issue in selective data acquisition is the amount of control ceded to the "oracle" doing the acquisition. The work discussed so far assumes that an oracle will be queried for some specific value, and the oracle simply returns that value (or a noisy realization). However, if the oracle is actually a person, he or she may be able to apply considerable intelligence and other resources to "guide" the selection. Such guidance is especially helpful in situations where some aspect of the data is rare—where purely data-driven strategies are particularly challenged.

Let's continue thinking about the active class selection setting as a concrete case. In many practical settings, one class is quite rare. As a motivational example, consider building a predictive model from scratch designed to classify web pages containing a particular topic of interest. While large absolute numbers of such web pages may be present on the web, they may be outnumbered by uninteresting pages by a million to one or worse (take, for instance, the task of detecting and removing hate speech from the web [5]).

Unfortunately, when the class distribution is so skewed, active learning strategies can fail completely—and the failure is not simply due to the hurdles faced when trying to learn models in settings with skewed class distribution, a problem that has received a fair bit of attention [93, 4]. Rather, the problem faced by the active learner is far more treacherous: learning techniques cannot even concentrate on the rare instances as the techniques are unaware which instances to focus on.

Perhaps even more insidious is the difficulty posed by classes consisting of rare, disjunctive sub-concepts. These disjuncts can emerge even in problems spaces without such an extreme class skew: when members of an important class do not manifest themselves as a simple, continuously dense region of the input space, but rather as many small disjoint clusters embedded throughout the input space [94]. For an active learner, these "small disjuncts" act like rare classes: when an active learner has not been exposed to instances of a sub-concept, how can it best choose instances to label in order to properly distinguish that subconcept from its containing space?

While a plethora of techniques have been proposed for performing active learning specifically in the high-skew setting [89, 13, 104, 27] as well as techniques where the geometry and feature-density of the problem space are explicitly included when making instance selections [105, 33, 21, 59, 98, 53], these techniques, as initially appealing as they may seem, may fail just as badly as traditional active learning techniques. Class skew and sub-concept rarity may be sufficient to thwart them completely [5, 4].

However, in these extremely difficult settings, we can task humans to search the problem space for rare cases, using tools (like search engines) and possibly interacting with the base learner. Consider the motivating example of hate speech classification on the web (from above). While an active learner may experience difficulty exploring the details of this rare class, a human oracle armed with a search interface is likely to expose examples of hate speech easily. In fact, given the coverage of modern web search engines, a human can produce interesting examples from a much larger sample of the problem space, far beyond that which is likely to be contained in a sample pool for active learning. This is critical due to hardware-imposed constraints on the size of the pool an active learner is able to choose from—e.g., a random draw of several hundred thousand examples from the problem space may not even contain any members of the minority class or of rare disjuncts!

Guided Learning is the general process of utilizing oracles to search the problem space, using their domain expertise to *seek* instances representing the interesting regions of the problem space. Figure 1.12 presents the general guided learning setting. Here, given some interface enabling search over the domain in question, an oracle searches for interesting examples, which are either supplemented with an implicit label by the oracle, or sent for explicit labeling as a second step. These examples are then added to the training set and a model is re-trained. Oracles can leverage their background knowledge of the problem being faced. By incorporating the techniques of active class selection oracles can be directed to gather instances in a class-proportion believed to most strongly help train a model. Further, by allowing the oracle to interact with the base learner, confusing instances, those that "fool" the model can be sought out from the problem space and used for subsequent training in a form of human-guided uncertainty sampling.

While guided learning often presents an attractive and cost-effective alternative to ac-

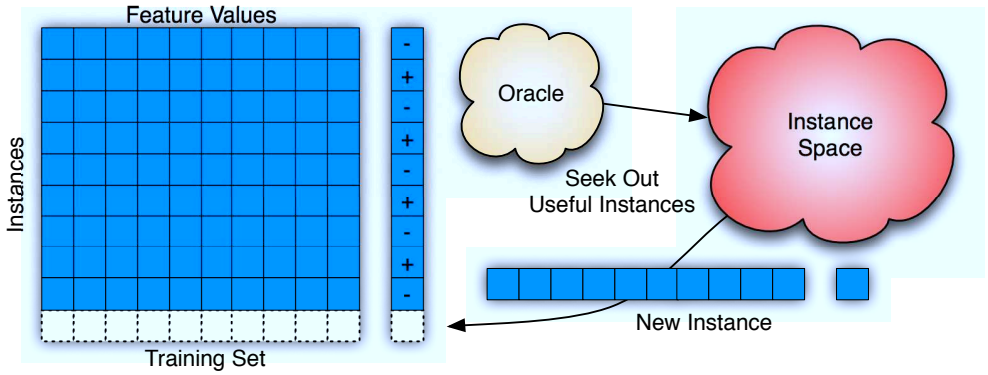


**FIGURE 1.12**: Guided Learning: An oracle selecting useful examples from the instance space.

tive learning, particularly in difficult settings, the overall guided information acquisition paradigm is flexible, human intelligence and background knowledge can be used to *seek* a wide variety of information. As an example of alternative sources of information available through guided information acquisition, consider the problem of finding useful features when building classification models. In many settings, the choice of which features to use as a capable separator of classes may not be initially obvious. Simply including many features also is not an appealing option; with insufficient labeled examples, the underlaying model or feature selection technique may confuse signal with noise, attributing predictive power to certain features simply due to noise and false-signals [65, 64].

However, human are often able to apply their background knowledge, suggesting discriminative features for a given problem or removing useless features currently being utilized by an existing model in order to approve predictive performance [65, 64, 17]. An example of this process can be seen in Figure 1.13. In this *guided feature selection* paradigm, human oracles describe additional features to the predictive system, in effect adding additional columns to the instance information matrix. Note that these features may be functions of existing feature values, or they may require explicit feature value acquisition as in Section 1.3.

Active feature labeling and active dual supervision are demonstrated to be effective techniques for reducing the total annotation effort required for building effective predictive models (See Section 1.4). However, while appropriately chosen feature labels may facilitate the construction of models able to effectively discriminate between classes, as with active learning, particularly difficult situations may stymie active feature selection and active dual supervision, wasting many queries on uninformative queries, simply because the base model has very little knowledge of the problem space [1]. However, just as in guided learning, human oracles may be requested to *seek* polarity labels for those features thought to most effectively discern between the classes. This process may be seen in Figure 1.14. Note that this process may be seamlessly incorporated with guided feature selection, adding new features and their associated label simultaneously. In the case of text classification, this may involve adding a term or phrase not originally considered, and assigning to that phrase a class polarity. This guided feature labeling has been demonstrated to offer much greater effectiveness than active feature labeling in "difficult" classification problems [1].

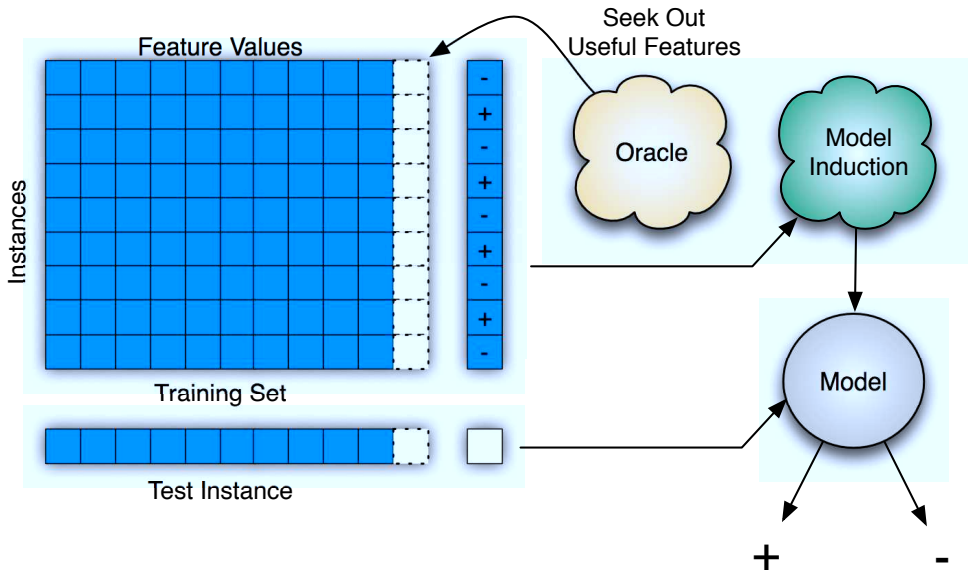


FIGURE 1.13: Guided Feature Selection: tasking oracles with finding useful features.

### 1.7.4 Beyond improving a single model's accuracy

Thus far, we have discussed different types of information that can be acquired. However, in the context of learning predictive models we have not considered settings in which the objective for information acquisitions differ from that of active learning policies, i.e., improving a single model's predictive accuracy (by some measure). As an example of such a setting, consider an application where the labels of training examples can be (actively) acquired in order to augment the training data of multiple classification models.

Combining the aspects of multi-task learning and active learning, selecting examples for Multiple Task Active Learning (MTAL) [69] can be approached in several ways. The first approach would be simply to alternate between acquisitions that benefit each of the models. The second approach would attempt to produce a single ranking of examples based on the benefits across the learning tasks. In this case, each prospective acquisition is first ranked by each of learning tasks, using traditional active learning policies. The overall rank for a prospective acquisition is then computed via the Borda aggregation procedure, i.e., by summing the prospective acquisition's rank numbers across all learning tasks. Both of these methods [69] are general, in that they can be used to learn different types of models. In fact, in principle, these policies can also apply when a different active learning policy is used to rank prospective acquisitions for different types of models (e.g., aggregation and classification models). However, their effectiveness has not been evaluated for these settings related to multiple active learning techniques. When multiple prediction tasks are related, such as when the labels of instances corresponding to each satisfy certain constraints, it is also possible to exploit these dependencies to improve the selection of informative instances [101].

An interesting scenario which implies yet another objective for information acquisition in practice arises when information acquisition aims to improve repetitive decisions, which are, in turn, informed by predictive models [74, 40] . Furthermore, many decisions

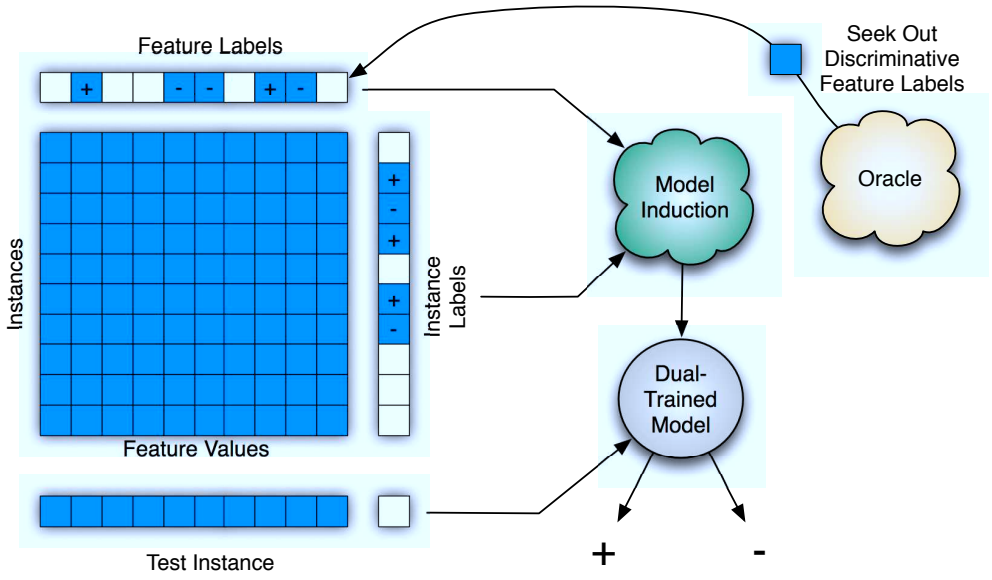


**FIGURE 1.14**: Guided Feature Labeling: tasking oracles with finding features and their associated class polarity believed to be best able to discriminate between classes.

are informed by multiple predictive models of different types. For instance, firms' sales tax audit decisions are typically informed by the predicted probability of tax fraud, but also by the predicted revenue that might be recovered. Note that prospective audit acquisitions will affect both predictive models. Interestingly, for each model, a different audit acquisition might be more desirable to improve the corresponding estimation. Hence, it is beneficial to understand how should an acquisition budget be allocated among them so as to benefit future audit decisions the most?

In principle, MTAL can be applied to cost-effectively improve the predictive accuracy of all the models informing the decisions. However, costly improvements in model accuracy does not necessarily yield better decisions [74]. Hence, more efficient policies can be derived if such greedy improvements in accuracy are avoided. Furthermore, exploiting knowledge of the decision's structure can yield significant benefits [74, 40, 41]. For example, when information is acquired for multiple models informing the decisions, it is useful to consider how changes in all of the models simultaneously will affect future decisions. This approach has been applied effectively when the same information can augment multiple models simultaneously [40, 41]. It will also be useful to study the effectiveness of this general principle when the training data of each of the models can be augmented by acquisition from a different source (i.e., the same acquisition cannot be used to train all models).

## 1.8 Conclusion

The vast majority of the information acquisition literature focuses on active learning, reflecting only a single setting— the selection of individual instances for the acquisition of a single, correct label with the objective of improving classification performance cost effectively. However, as we outline in this chapter, a rich set of important practical problems arise once we consider the diverse set of information that is available to a machine learning system at some cost, and when the error-free assumptions typically used in such research are relaxed. In this chapter, we discussed three dimensions that define alternative information acquisition problems. Specifically these facets are: the different types of information that can be acquired to inform learning, the quality of the acquired information, and the different objectives which acquisitions ultimately aim to improve, cost-effectively. Since each of these facets can take a variety of settings in practice, cost-effective information acquisition gives rise to a diverse range of important problems.

In order to cope with the complexity of such a diverse range of problems, we have presented two general strategies for information acquisition. Namely, uncertainty reduction and expected utility approaches. While uncertainty-based approaches are appealing initially in that they are often simple to implement and have been subject to extensive study in the realm of traditional active learning, these techniques tend to fall short when faced with the more complex demands of general information acquisition. Combining effectiveness measures for diverse types of information seamlessly in concert with acquisition costs is often beyond the capability of uncertainty-based techniques. Fortunately, expected utility-based approaches offer a one-size-fits-all framework for integrating the empirical benefits of different acquisition types with the costs these acquisitions may incur. While expected utility approaches offer great promise, this is tempered by the difficulties associated with computing an accurate utility estimator in a way that is computationally tractable.

While we have demonstrated a wide range of use cases and settings for selective data acquisition, discussing applications ranging from marketing to medicine, there are innumerable scenarios yet to be explored; more complex combinations of data available at a cost, both at training and at use time, with relaxed assumptions regarding data cleanliness. Though it's difficult to speculate what new applications and future research into the realm of data acquisition will accomplish, we believe that the material presented here provides a solid foundation upon which this new work can build.

# *Bibliography*


[1] Josh Attenberg, Prem Melville, and Foster Provost. Guided feature labeling for budget-sensitive learning under extreme class imbalance. In *BL-ICML '10: Workshop on Budgeted Learning*, 2010.

[2] Josh Attenberg, Prem Melville, and Foster J. Provost. A unified approach to active dual supervision for labeling features and examples. In *ECML/PKDD*, 2010.

[3] Josh Attenberg and Foster Provost. Active inference and learning for classifying streams. In *BL-ICML '10: Workshop on Budgeted Learning*, 2010.

[4] Josh Attenberg and Foster Provost. Inactive learning? difficulties employing active learning in practice. *SIGKDD Explorations*, 12(2), 2010.

[5] Josh Attenberg and Foster Provost. Why label when you can search? strategies for applying human resources to build classification models under extreme class imbalance. In *KDD*, 2010.

[6] Arindam Banerjee, Chase Krumpelman, Sugato Basu, Raymond J. Mooney, and Joydeep Ghosh. Model-based overlapping clustering. In *Proc. of 11th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-05)*, 2005.

[7] Sugato Basu. Semi-supervised clustering with limited background knowledge. In *AAAI*, pages 979–980, 2004.

[8] Mustafa Bilgic and Lise Getoor. Voila: Efficient feature-value acquisition for classification. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 1225–1230, 2007.

[9] Mustafa Bilgic and Lise Getoor. Effective label acquisition for collective classification. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 43–51. ACM, 2008.

[10] Mustafa Bilgic and Lise Getoor. Reflect and correct: A misclassification prediction approach to active inference. *ACM Trans. Knowl. Discov. Data*, 3, December 2009.

[11] Mustafa Bilgic and Lise Getoor. Active Inference for Collective Classification. In *AAAI*, 2010.

[12] Catherine L. Blake and Christopher John Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.

[13] Michael Bloodgood and K. Vijay Shanker. Taking into account the differences between actively and passively acquired data: the case of active learning with support vector machines for imbalanced datasets. In *NAACL*, 2009.

[14] Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

[15] Joachim M. Buhmann and Thomas Zöller. Active learning for hierarchical pairwise data clustering. In *ICPR*, pages 2186–2189, 2000.

[16] Bob Carpenter. Multilevel bayesian model of categorical data annotation, 2008. Available at: `http://lingpipe-blog.com/lingpipe-white-papers/`.

[17] Bruce Croft and Raj Das. Experiments with query acquisition and use in document retrieval systems. In *In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 349–368, 1990.

[18] B.V. Dasarathy, editor. *Information Fusion:An International Journal on Multi-Sensor, Multi-Source Information Fusion*. Elsevier, 2010.

[19] Alexander Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, September 1979.

[20] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *COLT 2009: Proceedings of the 22nd Annual Conference on Learning Theory*. Citeseer, 2009.

[21] P. Donmez and J. Carbonell. Paired Sampling in Density-Sensitive Active Learning. In *Proc. 10 th International Symposium on Artificial Intelligence and Mathematics*, 2008.

[22] Pinar Donmez and Jaime G. Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pages 619–628, 2008.

[23] Pinar Donmez, Jaime G. Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009)*, pages 259–268, 2009.

[24] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[25] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP '09*, pages 81–90. Association for Computational Linguistics, 2009.

[26] Charles Elkan. The foundations of cost-sensitive learning. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[27] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *CIKM*, 2007.

[28] Jerome H. Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. In *AAAI/IAAI, Vol. 1*, pages 717–724, 1996.

[29] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based support vector machine classifiers. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[30] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *ICDM*, pages 625–628, 2005.

[31] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD*, 2004.

[32] 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.

[33] Jingrui He and Jaime G. Carbonell. Nearest-neighbor-based active learning for rare category detection. In *NIPS*, 2007.

[34] Thomas Hofmann and Joachim M. Buhmann. Active data clustering. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 528–534, Cambridge, MA, USA, 1998. MIT Press.

[35] P. Ipeirotis, F. Provost, V. Sheng, and J. Wang. Repeated labeling using multiple, noisy labelers. Technical Report Working Paper CeDER-10-03, Center for Digital Economy Research, NYU Stern School of Business, 2010.

[36] P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67. ACM, 2010.

[37] Pallika Kanani and Prem Melville. Prediction-time active feature-value acquisition for customer targeting. In *Proceedings of the Workshop on Cost Sensitive Learning, NIPS 2008*, 2008.

[38] Aloak Kapoor and Russell Greiner. Learning and classifying under hard budgets. In *ECML*, pages 170–181, 2005.

[39] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML*, pages 307–314, 2002.

[40] Danxia Kong and Maytal Saar-Tsechansky. Collaborative information acquisition. In *BL-ICML '10: Workshop on Budgeted Learning*, 2010.

[41] Danxia Kong and Maytal Saar-Tsechansky. A framework for collaborative information acquisition. In *Workshop on Information Technology and Systems*, 2010.

[42] Gautam Kunapuli, Kristin P. Bennett, Amina Shabbeer, Richard Maclin, and Jude W. Shavlik. Online knowledge-based support vector machines. In *ECML/PKDD*, pages 145–161, 2010.

[43] Ludmila I. Kuncheva, Christopher J. Whitaker, Catherine A. Shipp, and Robert P.W. Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6(1):22–31, April 2003.

[44] David Lewis and William Gale. A sequential algorithm for training text classifiers. In *Proc. of 17th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-94)*, 1994.

[45] R. Little and D. Rubin. *Statistical Analysis with Missing Data*. John Wiley and Sons,, 1987.

[46] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip Yu. Text classification by labeling words. In *AAAI*, 2004.

[47] Dan Lizotte, Omid Madani, and Russell Greiner. Budgeted learning of naive-Bayes classifiers. In *UAI*, 2003.

[48] R. Lomasky, C. Brodley, M. Aernecke, D. Walt, and M. Friedl. Active Class Selection. In *Machine Learning: ECML*, volume 4701, pages 640–647, 2007.

[49] R. Lomasky, C. E. Brodley, S. Bencic, M. Aernecke, and D Walt. Guiding class selection for an artificial nose. In *NIPS Workshop on Testing of Deployable Learning and Decision Systems*, 2006.

[50] Rachel Lomasky. *Active Acquisition of Informative Training Data*. PhD thesis, Tufts University, 2010.

[51] Gabor Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25(1):79–87, January 1992.

[52] Andrew McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Text Categorization*, 1998.

[53] Andrew K. Mccallum and Kamal Nigam. Employing em in pool-based active learning for text classification. In *ICML*, 1998.

[54] Prem Melville, Wojciech Gryc, and Richard Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *KDD*, 2009.

[55] Prem Melville and Raymond J. Mooney. Diverse ensembles for active learning. In *Proc. of 21st Intl. Conf. on Machine Learning (ICML-2004)*, 2004.

[56] Prem Melville, Foster J. Provost, and Raymond J. Mooney. An expected utility approach to active feature-value acquisition. In *Proc. of 5th IEEE Intl. Conf. on Data Mining (ICDM-05)*, pages 745–748, 2005.

[57] Prem Melville, Maytal Saar-Tsechansky, Foster J. Provost, and Raymond J. Mooney. Active feature-value acquisition for classifier induction. In *Proc. of 4th IEEE Intl. Conf. on Data Mining (ICDM-04)*, pages 483–486, 2004.

[58] Prem Melville and Vikas Sindhwani. Active dual supervision: Reducing the cost of annotating examples and features. In *NAACL HLT 2009*, 2009.

[59] Hieu T. Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *ICML*, 2004.

[60] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86. Association for Computational Linguistics, 2002.

[61] F. Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI2000 Workshop on Imbalanced Data Sets*, 2000.

[62] Foster Provost and Andrea Pohoreckyj Danyluk. Learning from bad data. In *Proceedings of the ML-95 Workshop on Applying Machine Learning in Practice*, 1995.

[63] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1992.

[64] Hema Raghavan, Omid Madani, and Rosie Jones. InterActive feature selection. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 841–846, 2005.

[65] Raghavan, Hema, Madani, Omid, Jones, Rosie, and Kaelbling, Leslie. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7, 2006.

[66] Matthew J. Rattigan, Marc Maier, and David Jensen. Exploiting Network Structure for Active Inference in Collective Classification. Technical Report 07-22, University of Massachusetts Amherst, 2007.

[67] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Labeling images with a computer game. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*, pages 889–896, 2009.

[68] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(7):1297–1322, April 2010.

[69] Roi Reichart, Katrin Tomanek, and Udo Hahn. Multi-task active learning for linguistic annotations. In *ACL*, 2008.

[70] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[71] Maytal Saar-Tsechansky, Prem Melville, and Foster J. Provost. Active feature-value acquisition. *Management Science*, 55(4):664–684, 2009.

[72] Maytal Saar-tsechansky and Foster Provost. Active learning for class probability estimation and ranking. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001*, pages 911–920, 2001.

[73] Maytal Saar-Tsechansky and Foster Provost. Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2):153–178, 2004.

[74] Maytal Saar-Tsechansky and Foster Provost. Decision-centric active learning of binary-outcome models. In *Information Systems Research*, 2007.

[75] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *J. Mach. Learn. Res.*, 8:1623–1657, December 2007.

[76] SAS Institute Inc. *Getting Started with SAS Enterprise Miner*. SAS Institute Inc, Cary, NC, USA, 2001.

[77] Robert E. Schapire, Marie Rochery, Mazin G. Rahim, and Narendra Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[78] Andrew Schein and Lyle Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265–265, October 2007.

[79] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[80] 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, 2008.

[81] Victor S. Sheng and Charles X. Ling. Feature value acquisition in testing: a sequential batch test algorithm. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 809–816, New York, NY, USA, 2006. ACM.

[82] Victor S. Sheng, Foster Provost, and Panagiotis Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, 2008.

[83] Bernard W. Silverman. Some asymptotic properties of the probabilistic teacher. *IEEE Transactions on Information Theory*, 26(2):246–249, March 1980.

[84] Vikas Sindhwani and Prem Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *ICDM*, 2008.

[85] Vikas Sindhwani, Prem Melville, and Richard Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *ICML*, 2009.

[86] Padhraic Smyth. Learning with probabilistic supervision. In Thomas Petsche, editor, *Computational Learning Theory and Natural Learning Systems, Vol. III: Selecting Good Models*. MIT Press, April 1995.

[87] Padhraic Smyth, Michael C. Burl, Usama M. Fayyad, and Pietro Perona. Knowledge discovery in large image databases: Dealing with uncertainties in ground truth. In *kdd94*, pages 109–120, 1994.

[88] Padhraic Smyth, Usama M. Fayyad, Michael C. Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of Venus images. In *nips94*, pages 1085–1092, 1994.

[89] Katrin Tomanek and Udo Hahn. Reducing class imbalance during active learning for named entity annotation. In *K-CAP '09: Intl. Conf. on Knowledge capture*, 2009.

[90] Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70:119–165, 1994.

[91] Duy Vu, Mikhail Bilenko, Maytal Saar-tsechansky, and Prem Melville. Intelligent information acquisition for improved clustering. In *Proceedings of the Ninth International Conference on Electronic Commerce*, 2007.

[92] G. Weiss and F. Provost. The effect of class distribution on classifier learning. Rutgers technical report ml-tr-44, Rutgers, The State University of NJ, 2001.

[93] Gary M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explorations*, 6(1):7–19, 2004.

[94] Gary M. Weiss. The impact of small disjuncts on classifier learning. In Robert Stahlbock, Sven F. Crone, and Stefan Lessmann, editors, *Data Mining*, volume 8 of *Annals of Information Systems*, pages 193–226. Springer US, 2010.

[95] Gary M. Weiss and Foster Provost. Learning when training data are costly: the effect of class distribution on tree induction. *J. Artif. Int. Res.*, 19(1):315–354, 2003.

[96] Jacob Whitehill, Paul Ruvolo, Ting fan Wu, Jacob Bergsma, and Javier Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 2035–2043, 2009.

[97] Xiaoyun Wu and Rohini Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD*, 2004.

[98] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang. Representative sampling for text classification using support vector machines. In *ECIR*, 2003.

[99] Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, J. Dy, and PA Malvern. Modeling annotator expertise: Learning when everybody knows a bit of something. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[100] O. F. Zaidan and J. Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *EMNLP*, 2008.

[101] Yi Zhang. Multi-task active learning with output constraints. In *AAAI 2010*, 2010.

[102] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proc. of IEEE Intl. Conf. on Data Mining*, 2002.

[103] Zhiqiang Zheng and Balaji Padmanabhan. Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Science*, 52(5):697–712, May 2006.

[104] Jingbo Zhu and Eduard Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, 2007.

[105] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *COLING '08*, 2008.