Economical Active Feature-value Acquisition through Expected Utility Estimation

Prem Melville
Dept. of Computer Sciences
Univ. of Texas at Austin
melville@cs.utexas.edu

Maytal Saar-Tsechansky
Red McCombs School of Business
Univ. of Texas at Austin
maytal@mail.utexas.edu

Foster Provost
Stern School of Business
New York University
fprovost@stern.nyu.edu

Raymond Mooney
Dept. of Computer Sciences
Univ. of Texas at Austin
mooney@cs.utexas.edu

Abstract

In many classification tasks training data have missing feature values that can be acquired at a cost. For building accurate predictive models, acquiring all missing values is often prohibitively expensive or unnecessary, while acquiring a random subset of feature values may not be most effective. The goal of active feature-value acquisition *is to incrementally select feature values that are most cost-effective for improving the model's accuracy. We present two policies,* Sampled Expected Utility *and* Expected Utility-ES, *that acquire feature values for inducing a classification model based on an estimation of the expected improvement in model accuracy per unit cost. A comparison of the two policies to each other and to alternative policies demonstrate that* Sampled Expected Utility *is preferable as it effectively reduces the cost of producing a model of a desired accuracy and exhibits a consistent performance across domains.*

1 Introduction

In many predictive modeling problems, feature values for training data are missing, but can be acquired at a cost. Often the cost of acquiring the missing information varies according to the nature of the information or of the particular instance for which information is missing. Consider, for example, patient data used to induce a model to predict whether or not a treatment will be effective for a given patient. Some patient data may have missing demographic information that can be obtained at a low cost. In contrast, acquiring diagnostic test results from different health-care providers can be significantly more expensive and time-consuming. Various solutions are available for learning models from incomplete data, such as imputation methods [8], and learners that ignore missing feature values such as the Naive Bayes classifier. However, these solutions almost always undermine model performance as compared to that of a model induced from complete information. Since obtaining all missing values may be prohibitively expensive, it is desirable to identify what information would be most cost-effective to acquire. In this paper we address this generalized version of the *active feature-value acquisition* (AFA) task for classifier induction [10]: given a model built on incomplete training data, select feature values that would be most cost-effective to acquire for improving the model's accuracy. The problem of feature-value acquisition is different from traditional active learning [2] in which class labels rather than feature values are missing and are costly to acquire.

Unlike prior work [9], we study AFA in a setting where the total cost to be spent on acquisitions is not determined *a priori*, but rather can be determined on-line based on the model's performance as learning progresses. This setting is motivated by the inherent uncertainty regarding the tradeoff between costs and improvement in model accuracy. An incremental spending strategy enables a decision maker to re-evaluate the desirability of further expenditures by incrementally exploring the performance curve resulting from a series of acquisition decisions. For example, one may choose not to acquire additional information if the current model accuracy is satisfactory, or if additional information is unlikely to provide a significant improvement in the model. We propose a general setting for AFA that specifies an incremental acquisition schedule. Given the current model, an AFA strategy identifies feature-value acquisitions that are estimated to be most cost-effective with respect to model accuracy.

We present a solution to the AFA task that ranks alternative feature-value acquisitions based on an estimation of the expected improvement in model performance per unit cost. Our approach is general, i.e., it can be applied to select acquisitions for any learner, and to attempt to improve any performance metric. Experimental results on decision tree induction to improve classification accuracy demonstrate that our method does consistently result in significantly improved model accuracy per unit cost compared to random feature-value acquisition. The method is particularly advantageous in challenging tasks for which there is a significant variance across potential acquisitions with respect to their contribution to learning per unit cost.

2 Task Definition and Algorithm

2.1 Active Feature-value Acquisition

Assume a classifier induction problem where each instance is represented with n feature values and a class label. A training set of m instances can be represented by the matrix F, where $F_{i,j}$ corresponds to the value of the j-th feature of the i-th instance. Initially, the class label, y_i, of each instance is known, and the matrix F is incomplete, i.e., it contains missing values. The learner may acquire the value of $F_{i,j}$ at the cost $C_{i,j}$. We use $q_{i,j}$ to refer to the query for the value of $F_{i,j}$. The general task of active feature-value acquisition is the selection of these instance-feature queries that will result in building the most accurate model (classifier) at the lowest cost. The framework for the generalized AFA task is presented in Algorithm **??**ach step the learner builds a classifier trained on the current data, and scores the available queries based on this classifier. The query with the highest score is selected and the feature value corresponding to this query is acquired. The training data is appropriately updated and this process is repeated until some stopping criterion is met, e.g. a desirable model accuracy has been obtained. To reduce computation costs in our experiments, we acquire queries in fixed-size batches at each iteration.

Alternate problem settings of feature-value acquisition have been explored in the literature. In particular, Melville et al. [10] studied a specialized version of the AFA problem addressed here, where *all* the missing feature values for an instance are acquired at once and an acquisition policy selects the instances for which acquiring all missing values would result in the most accurate classifier. Lizotte et al. [9] studied the *budgeted learning* scenario, in which the total cost (budget) to be spent on feature-value acquisitions is determined *a priori*. We discuss these and other related research in more detail in the related work section.

Algorithm 1 General Active Feature-value Acquisition Framework

```
Given:
$F$ – initial (incomplete) instance-feature matrix
$Y = \{y_i : i = 1, ..., m\}$ – class labels for all instances
$T$ – training set = $< F, Y >$
$\mathcal{L}$ – base learning algorithm
$b$ – size of query batch
$C$ – cost matrix for all instance-feature pairs

Initialize $TotalCost$ to cost of $F$
Initialize set of possible queries $Q$ to $\{q_{i,j} : i = 1, ..., m; j = 1, ..., n;$ such that $F_{i,j}$ is missing$\}$
Repeat until stopping criterion is met
    Generate a classifier, $M = \mathcal{L}(T)$
    $\forall q_{i,j} \in Q$ compute $score(M, q_{i,j}, \mathcal{L}, T)$
    Select a subset $S$ of $b$ queries with the highest $score$
    $\forall q_{i,j} \in S$,
        Acquire values for $F_{i,j}$
        $TotalCost = TotalCost + C_{i,j}$
    Remove $S$ from $Q$
Return $M = \mathcal{L}(T)$
```

2.2 Expected Utility Estimation

Specific solutions to the AFA problem differ based on the method used to score and rank queries. In our approach we provide scores based on the *expected utility* of each query (defined below). For now we assume all features are nominal, i.e., they can take on values from a finite set of values. Assume feature j has K distinct values $V_1, ..., V_K$. The expected utility of the query $q_{i,j}$ can be computed as:

$$E(q_{i,j}) = \sum_{k=1}^{K} P(F_{i,j} = V_k)\mathcal{U}(F_{i,j} = V_k) \qquad (1)$$

where $P(F_{i,j} = V_k)$ is the probability that $F_{i,j}$ has the value V_k, and $\mathcal{U}(F_{i,j} = V_k)$ is the utility of knowing that the feature value $F_{i,j}$ is V_k, given by:

$$\mathcal{U}(F_{i,j} = V_k) = \frac{\mathcal{A}(F, F_{i,j} = V_k) - \mathcal{A}(F)}{C_{i,j}} \qquad (2)$$

where $\mathcal{A}(F)$ is the accuracy of the current classifier; $\mathcal{A}(F, F_{i,j} = V_k)$ is the accuracy of the classifier trained on F assuming $F_{i,j} = V_k$; and $C_{i,j}$ is the cost of acquiring $F_{i,j}$. For this paper, we define the utility of an acquisition in terms of improvement in model accuracy per unit cost. Depending on the objective of learning a classifier, alternate utility functions could be used.

If we were to plot a graph of accuracy versus model cost after every iteration of AFA, our *Expected Utility* approach would correspond to selecting the query that is expected to result in the largest slope for the next iteration. If all feature costs are equal, this corresponds to selecting the query that would result in the classifier with the highest expected accuracy.

Since the true distribution of each missing feature value is unknown, we estimate $P(F_{i,j} = V_k)$ in Eq. 1 using a learner that produces class probability estimates. For each feature j, we train a classifier M_j, using this feature as the target variable and all other features along with the class as the predictors. When evaluating the query $q_{i,j}$, the classifier M_j is applied to instance i to produce the estimate $\hat{P}(F_{i,j} = V_k)$.

In Eq. 2, the true values of $\mathcal{A}(.)$ are also unknown. However, since the class labels for the training data are available at selection time we can estimate $\mathcal{A}(F)$ and $\mathcal{A}(F, F_{i,j} = V_k)$ based on the training set accuracy. In our experiments, we used 0-1 loss to measure the accuracy of the classifiers. However, other measures such as class entropy or GINI index could also be used [9]. In our preliminary studies we did not observe a consistent advantage to using entropy.

When the *Expected Utility* method described here is applied to learn a Naive Bayes classifier and feature costs are assumed to be equal, it is similar to the *greedy loss reduction* approach presented in [9]. Similar approaches to expected utility estimation have also been used in the related task of traditional active learning [12, 7, 14].

Computing the estimated expectation $\hat{E}(.)$ for query $q_{i,j}$ requires training one classifier for each possible value of feature j. Selecting the best from *all* available queries would require exploring, in the worst case, mn queries. So exhaustively selecting a query that maximizes the expected utility is computationally very intensive and is infeasible for most interesting problems. We make this exploration tractable by reducing the search space to a random subsample of the available queries. We refer to this approach as *Sampled Expected Utility*. This method takes a parameter α ($1 \leq \alpha \leq \frac{mn}{b}$) which controls the complexity of the search. To select a batch of b queries, first a random sub-sample of αb queries is selected from the available pool, and then the expected utility of each query in this sub-sample is evaluated. The value of α can be set depending on the amount of time the user is willing to spend on this process. One can expect a tradeoff between the amount of time spent and the effectiveness of the selection scheme.

2.3 Instance-based Active Feature-value Acquisition

In *Sampled Expected Utility* we use a random sample of the pool of available queries to make the *Expected Utility* estimation feasible. However, it may be possible to improve performance by applying *Expected Utility* estimation to a sample of queries that is better than a random sample. One approach could be to first identify potentially informative *instances*, and then select candidate queries only from these instances. In previous work we studied a specialized version of AFA, where *all* the missing feature values for an instance are acquired at once and an acquisition policy selects the instances for which acquiring all missing values would result in the most accurate classifier[10]. The method proposed in this work, *Error Sampling* (ES), can be readily used to identify informative instances from which we can then choose candidate queries. *Error Sampling* orders incomplete instances in terms of potential informativeness in the following way. It ranks instances that have been misclassified by the current model as the most informative. Next, it ranks correctly classified instances in order of decreasing uncertainty in the model's prediction. *Error Sampling* requires building only one model at each step of AFA, and hence is not too computationally intensive to use in place of random sampling in our *Sampled Expected Utility* approach. We call this new approach *Expected Utility-ES*, in which *Error Sampling* is used to rank instances from which the first αb missing instance-feature pairs are selected as candidate queries. Where b is the desired batch size and α is the exploration parameter.

Though *Error Sampling* was designed for selecting instances, it can also be modified to acquire single feature values in our general AFA setting. The method ranks instances for acquisition, but does not provide a mechanism for selecting the most informative features for a given instance. We therefore examine a version of *Error Sampling* in which instances are ordered using the *Error Sampling* ranking, and the first b missing feature values are selected for acquisition.

3 Experimental Evaluation

3.1 Methodology

We begin by evaluating our proposed approaches on four datasets from the UCI repository [1], the details of which are presented in Table 1. For the sake of simplicity, we selected datasets that have only nominal features. In the future work section, we describe how we can extend our approach to handle numeric features. None of the UCI datasets provide feature acquisition costs – in our initial experiments we simply assume all costs are equal. Later, we present additional experiments with different cost structures.

We compare all the proposed methods to *random feature acquisition*, which selects queries uniformly at random to provide a representative sample of missing values. For the *Sampled Expected Utility* and *Expected Utility-ES* we

Table 1. Summary of Data Sets

Name	Instances	Features	Classes
vote	435	16	2
car	1727	6	4
lymph	148	18	4
audio	226	69	24

set the exploration parameter α to 10. Given the computational complexity of *Expected Utility* it is not feasible to run the exhaustive *Expected Utility* approach on all datasets. However, we did run *Expected Utility* on the *vote* dataset. For all methods, as a base learner we used J48 decision-tree induction, which is the Weka [16] implementation of C4.5 [11]. Laplace smoothing was used with J48 to improve class probability estimates.

The performance of each acquisition scheme was averaged over 10 runs of 10-fold cross-validation. In each fold of cross-validation, we generated learning curves in the following fashion. Initially, the learner is given a random sample of feature values, i.e. the instance-feature matrix is partially filled. The remaining instance-feature pairs are used to initialize the pool of available queries. At each iteration, the system selects a batch of queries, and the values for these features are acquired. This process is repeated until a desired number of feature values is acquired. Classification accuracy is measured after each batch acquisition in order to generate a learning curve. One system (A) is considered to be *significantly* better than another system (B) if the average accuracy across the points on the learning curve of A is higher than that of B according to a paired t-test ($p < 0.05$). As in [10], the test data contains only complete instances, since we want to approximate the true generalization accuracy of the constructed model given complete data for a test instance. For each dataset, we selected the initial random sample size to be such that the induced model performed at least better than majority class prediction. The batch size for the queries was selected based on the difficulty of the dataset. For problems that were harder to learn, we acquired a larger number of feature-values and consequently used larger batch sizes.

3.2 Results

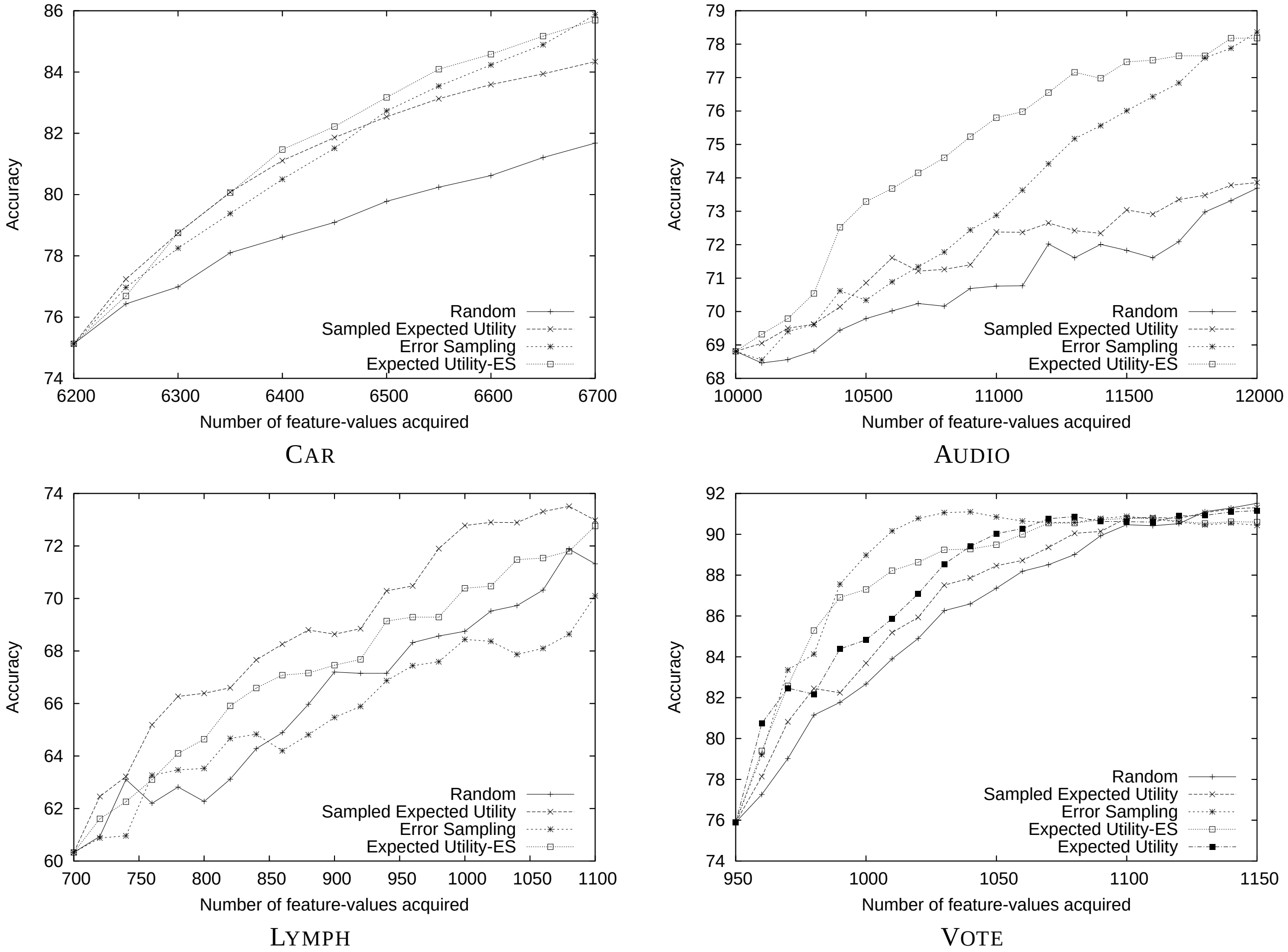
Our results are presented in Figure 1. For all datasets, *Sampled Expected Utility* builds more accurate models than random sampling for any given number of feature acquisitions. These results demonstrate that the estimation of the expected improvement in the current model's accuracy enables effective ranking of potential queries. Consequently, *Sampled Expected Utility* selects queries that on average are more informative for the learner than an average query selected at random. The differences in performance between these two systems on all datasets is significant, as defined above. Since *Sampled Expected Utility* was proposed in order to reduce the computational costs of our original *Expected Utility* approach, we also examined the performance and computational time of the exhaustive *Expected Utility* algorithm for *vote*. We computed the average time it took to select queries in each iteration for each of the methods. These timing results are summarized in Table 2. The results show that constraining the search in *Expected Utility* by random sampling (or *Error Sampling*) can significantly reduce the selection time (by two orders of magnitude in this case) without a significant loss in accuracy.

Table 2. Average selection times on *vote*.

AFA Method	Selection time (msec)
Random	3.8
Expected Utility	3.77×10^5
Sampled Expected Utility	6.64×10^3
Error Sampling	8.04
Expected Utility-ES	7.44×10^3

While *Error Sampling* can rank acquisitions of complete instances effectively, it does not consider the value of individual feature values. Despite this, we observed that *Error Sampling* performs quite well. In particular, it often performs significantly better than random sampling and it sometimes performs better than *Sampled Expected Utility*. However, the performance of *Error Sampling* in this general setting of AFA is inconsistent, as it may perform significantly worse than random selection, as is seen on the *lymph* dataset.

The performance of *Expected Utility-ES* shows that the method can effectively benefit from each of its components. When *Error Sampling* performs better than random sampling, the acquisitions made by *Expected Utility-ES* result in better models than those induced with *Sampled Expected Utility*. The *vote* dataset seems to be an exception, in which *Error Sampling* can at times perform even better than *Expected Utility*, so the combined *Expected Utility-ES* method does not outperform *Error Sampling* here. *Error Sampling*'s inconsistent performance can also undermine the *Expected Utility-ES* acquisition policy, so that when *Error Sampling* fails to improve upon random acquisitions, *Expected Utility-ES* produces inferior models than those induced with *Sampled Expected Utility*. These results suggest that the use of *Error Sampling* in our current AFA setting is a promising direction for future work, but is dependent on improving the *Error Sampling* strategy such that *Error Sampling* consistently performs better than random selection. Note that, in the *instance-completion* setting of AFA for which *Error Sampling* was originally designed, it always



CAR

AUDIO

LYMPH

VOTE

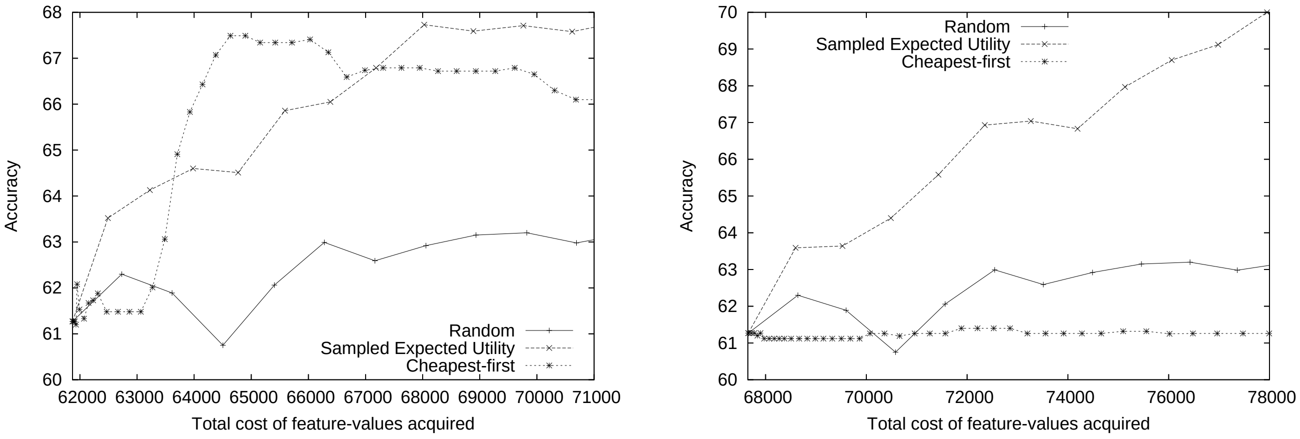
Figure 1. Comparing alternative active feature-value acquisition approaches.

performs better than random [10].

In summary, *Expected Utility-ES* often exhibits superior performance with respect to *Sampled Expected Utility* and random selection. However, it is susceptible to the inconsistent performance of *Error Sampling* and thus may potentially perform worse than random sampling. On the other hand, *Sampled Expected Utility* exhibits consistent improvements over random sampling on all datasets.

3.3 Artificial Data and Feature Costs

As no feature-acquisition costs are provided for the domains we employ here, we initially assumed uniform feature costs. In addition, some of the features in the data are equally discriminative so that there may be little value in selecting between them. In the extreme case, where feature costs are uniform and all features provide equal information about the target concept, random sampling is likely to be a very effective strategy. In order to make the problem setting more challenging, we constructed artificial data in the following way. We took the *lymph* dataset, which is composed of 18 features, and added an equal number of binary features with randomly-selected values, so as to provide no information about the class variable. In addition, we experimented with different cost structures. For the sake of simplicity, instead of having a cost associated with each instance-feature pair, we assume that the cost of acquiring a particular feature is the same irrespective of the instance. With each feature, we associate a cost selected uniformly at random from 1 to 100. Experiments were run as before for 5 different assignments of feature costs. Along with recording the accuracy after each batch acquisition of queries, we also record the current model cost based on the cost of the features acquired. Since random sampling does not take feature costs into account, we also compare *Sampled Expected Utility* with a simple baseline strategy that incorporates feature costs. This approach, which we call *Cheapest-first*, selects feature values for acquisition in or-



(a) Feature cost structure 1

(b) Feature cost structure 2

Figure 2. Comparing different algorithms on artificial data under different cost structures

der of increasing costs. Given the inconsistent performance of *Error Sampling* and *Expected Utility-ES*, we do not apply them to these datasets.

Figure 2 presents plots of accuracy versus model cost for two representative cost structures. The results for all randomly assigned costs structures show that for the same cost, *Sampled Expected Utility* consistently builds more accurate models than random sampling. The differences in performance between these two systems is more substantial than those observed for the UCI datasets with uniform costs. In contrast, the performance for *Cheapest-first* is quite varied for different cost assignments. When highly informative features are assigned low costs, *Cheapest-first* can perform quite well (Figure 2(a)). Since the underlying assumption of the *Cheapest-first* strategy, that the cheapest features are also informative, often holds in this case, it sometime performs better than *Sampled Expected Utility*, which imperfectly estimates the expected improvement in accuracy from each acquisition. However, when many inexpensive features are also uninformative, *Cheapest-first* performs worse than a random acquisition policy (Figure 2(b)). *Sampled Expected Utility*, however, estimates the tradeoff between cost and expected improvement in accuracy, and although the estimation is clearly imperfect, it consistently selects better queries than random acquisitions for all cost structures.

4 Related Work

To the best of our knowledge, the methods we propose here are the first approaches designed for the general problem of incrementally ranking and selecting feature values for inducing any classifier under a general acquisition cost structure. In this section, we discuss alternate settings for the AFA task.

Lizotte et al. [9] study AFA in the *budgeted learning* scenario, in which the total cost to be spent towards acquisitions is determined *a priori* and the task is to identify the best set of acquisitions for this cost. In contrast, our setting aims to enable the user to stop the acquisition process at any time, and as such the *order* in which acquisitions are made is important. Given this criterion, we attempt to select the next acquisition that will result in the most accurate model per unit cost. Lizotte et al. also assume that feature values are independent given the class, and as such consider queries of the form "Give me the value of feature j for any instance in class k." However, our approach evaluates feature-value acquisitions of specific instances, which allows us to 1) incorporate feature-value costs that vary per instance; and 2) to better estimate the expected value of an acquisition by capturing improvements from better modeling of feature interactions. Note that a set of features may exhibit different interactions for different instances, in which case evaluating potential acquisitions for individual instances is critical.

In this paper, we explored the use of the *Error Sampling* policy designed for the *instance-completion* setting, in which all missing feature values are acquired for a selected training instance [17, 10]. *Sampled Expected Utility* selects individual features, and hence can be also employed in the instance-completion setting, e.g., by selecting the instance with the highest sum of utilities of individual feature-value acquisitions.

Some work on *cost sensitive* learning [15] has addressed the issue of inducing economical classifiers when there are costs associated with obtaining feature values. However, most of this work assumes that the *training* data are complete and focuses on learning classifiers that minimize the cost of classifying incomplete *test* instances. An exception, CS-ID3 [13], also attempts to minimize the cost of acquiring features during training; however, it processes examples incrementally and can only request additional information for the current training instance. CS-ID3 uses a simple greedy strategy that requests the value of the cheapest unknown feature when the existing hypothesis is unable to correctly classify the current instance. It does not actively select the most useful information to acquire from a pool of incomplete training examples. The LAC* algorithm [5] also addresses the issue of economical feature acquisition during both training and testing; however, it also adopts a strategy that does not actively select the most informative data to collect during training. Rather, LAC* simply requests complete information on a random sample of instances in repeated *exploration* phases that are intermixed with *exploitation* phases that use the current learned classifier to economically classify instances.

Traditional *active learning* [2, 4] assumes access to unlabeled instances with complete feature data and attempts to select the most useful examples for which to acquire class labels. Active feature-value acquisition is a complementary problem that assumes labeled data with incomplete feature data and attempts to select the most useful additional feature values to acquire.

5 Limitations and Future Work

In *Sampled Expected Utility* we used a random sample of the pool of available queries to make the *Expected Utility* estimation feasible; and in *Expected Utility-ES*, we explored the possibility of limiting the set of candidate queries to only potentially informative instances. Alternatively, we can restrict the set of candidate queries to only the most informative features. A subset of such features could be picked using a *feature selection* technique that can capture the interactions among feature values, such as the wrapper approach of John et al. [6].

The performance of *Expected Utility* relies on having good estimates of the feature-value distributions and of the improvement in model accuracy for each potential acquisition. Thus *Expected Utility* is likely to benefit from improving upon the methods we applied to perform these estimations. For example, we could use probability estimation methods that better approximate the feature-value distributions, specifically when there are many missing values.

The *Expected Utility* framework allows us to consider model performance objectives other than accuracy. For example, when the benefits from making different accurate predictions and the error costs are specified, *Expected Utility* can be applied to identify acquisitions that result in the highest growth in benefits per unit cost. Experimenting with such alternate measures of model performance is an avenue for future work.

Our current study was restricted to datasets that are composed of only nominal features. Since many interesting domains include both numeric and nominal features, we would like to extend this study to datasets which also have numeric features. We could apply our current *Expected Utility* method after converting the numeric features to nominal features using a discretization technique, as in [3].

6 Conclusion

In this paper, we propose an expected utility approach to active feature-value acquisition, that obtains feature values based on the estimated expected improvement in model accuracy per unit cost. We demonstrate how this computationally intensive method can be made significantly faster, without much loss in performance, by constraining the search to a sub-sample of potential feature-value acquisitions. Experiments with uniform feature costs show that this *Sampled Expected Utility* approach consistently builds more accurate models than random sampling for the same number of feature-value acquisitions, and exhibits consistent performances across domains as compared to policies employing an instance-based ranking of features. Additional experiments on artificial datasets with different cost structures demonstrate that for the same cost, *Sampled Expected Utility* builds more accurate classifiers than the cost-agnostic random feature acquisition approach. Its performance is also more consistent than that of a simple cost-sensitive method which acquires feature values in order of increasing cost.

Acknowledgments

Prem Melville and Raymond Mooney were supported by DARPA grant HR0011-04-1-007.

References

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

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

[3] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027, Chambéry, France, 1993. Morgan Kaufmann.

[4] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

[5] Russell Greiner, Adam Grove, and Dan Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174, 2002.

[6] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML-94)*, pages 121–129, 1994.

[7] Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. Selective sampling for nearest neighbor classifiers. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 366–371, 1999.

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

[9] Dan Lizotte, Omid Madani, and Russell Greiner. Budgeted learning of naive-Bayes classifiers. In *Proceedings of 19th Conference on Uncertainty in Artificial Intelligence (UAI-2003)*, Acapulco, Mexico, 2003.

[10] Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM-04)*, 2004.

[11] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo,CA, 1993.

[12] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of 18th International Conference on Machine Learning (ICML-2001)*, pages 441–448. Morgan Kaufmann, San Francisco, CA, 2001.

[13] Ming Tan and Jeffery C. Schlimmer. Two case studies in cost-sensitive concept acquisition. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 854–860, Boston, MA, July 1990.

[14] Simon Tong and Daphne Koller. Active learning for parameter estimation in Bayesian networks. In *Advances in Neural Information Processing 13 (NIPS)*, pages 647–653, 2000.

[15] P. D. Turney. Types of cost in inductive concept learning. In *Proceedings of the Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning*, Palo Alto, CA, 2000.

[16] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, 1999.

[17] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proceedings of IEEE International Conference on Data Mining*, 2002.