Active Feature-Value Acquisition for Classifier Induction

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

Many induction problems, such as on-line customer profiling, include missing data that can be acquired at a cost, such as incomplete customer information that can be filled in by an intermediary. For building accurate predictive models, acquiring complete information for all instances is often prohibitively expensive or unnecessary. Randomly selecting instances for feature acquisition allows a representative sampling, but does not incorporate other value estimations of acquisition. Active feature-value acquisition *aims at reducing the cost of achieving a desired model accuracy by identifying instances for which complete information is most informative to obtain. We present approaches in which instances are selected for feature acquisition based on the current model's ability to predict accurately and the model's confidence in its prediction. Experimental results on several real-world data sets demonstrate that our approach can induce accurate models using substantially fewer feature-value acquisitions as compared to a baseline policy and a previously-published approach.*

1 Introduction

Many predictive modeling tasks include missing data that can be acquired at a cost, such as incomplete customer information which can be obtained though an intermediary. For building accurate models, ignoring instances with missing values leads to inferior model performance [15, 10], while acquiring complete information for all instances often is prohibitively expensive or unnecessary. To reduce the cost of acquiring feature information, it is desirable to identify a subset of the instances for which complete information is most informative to acquire.

The setting we explore was first introduced at ICDM 2002 [22] and applies to a variety of business and other domains. Consider an on-line retailer learning a predictive model to estimate customers' propensities to buy. The retailer may use private information on its customers and their buying behavior over time, as captured from the retailer's own web log-files. To improve the model, the retailer may also acquire additional information capturing its customers' buying preferences and lifestyle choices from a third-party information intermediary [8]. Acquiring complete data for all customers may be prohibitively expensive [13]. Hence, the retailer could benefit from having a cost-efficient feature acquisition strategy that can select the customers it should acquire complete information for, so as to most benefit the predictive model. A similar challenge is faced by marketing research firms that, in order to model consumer behavior, often obtain consumer responses to a short survey, and due to the cost of acquiring information, acquire responses to an extended survey from only a small, representative subset of those consumers. An effective acquisition strategy that acquires complete responses from consumers that are particularly informative for the model, can increase the accuracy of the model compared to that induced with the default strategy.

In this paper we address this problem of *active feature-value acquisition* (AFA) for classifier induction: given a feature acquisition budget, identify the instances with missing values for which acquiring complete feature information will result in the most accurate model. Formally, assume m instances, each represented by n features $a_1, ..., a_n$. For all instances, the values of a subset of the features $a_1, \ldots, a_i$ are known, along with the class labels. The values of the remaining features $a_{i+1}, \ldots, a_n$ are unknown and can be acquired at a cost.

The problem of feature-value acquisition is different from active learning [3] and optimum experimental design [9, 5], where the class labels rather than feature values are missing and costly to obtain. There has been relatively little work on acquisition of missing features, and we survey the work in Section 5.

The approaches we present here provide *generic* principles for active acquisitions; they apply to most classifier induction methods. They are also very effective and computationally efficient. These proposed policies for active feature acquisition are based on three observations:

1. In addition to categorical classifications, most classification models provide estimates of the confidence of

classification, such as estimated probabilities of class membership. Therefore principles underlying existing active-learning methods like uncertainty sampling [3] can be applied.

2. For the data items subject to active feature-value acquisition, the correct classifications are known during training. Therefore, unlike with traditional active learning, it is possible to employ direct measures of the current model's accuracy for estimating the value of potential acquisitions.

3. Class labels are available for all complete and incomplete instances. Therefore, we can exploit all instances (including incomplete instances) to induce models, and to guide feature acquisition.

These observations define a space of possible feature acquisition *policies* (prioritizations of training examples for feature acquisition). The main claim of this paper is that these simple-to-implement and computationally efficient policies perform remarkably well. Policies derived from these notions result in statistically significant and substantial improvements compared to random selection. A policy that considers model accuracy is also shown to be superior to a computationally intensive policy proposed earlier for this problem [22].

2 Task Definition and Algorithm

2.1 Pool-based Active Feature Acquisition

Assume a classifier induction problem, where each instance is represented with n feature values and a class label. For a subset G of the training set T, the values of all n features are known. We refer to these instances as complete instances. For all other instances in T, only the values of a subset of the features $a_1, \ldots, a_i$ are known. The values of the remaining features $a_{i+1}, \ldots, a_n$ are missing and the set can be acquired at a fixed cost. We refer to these instances as incomplete instances, and the set of all incomplete instances is denoted as I. The class labels of all instances in T are known.

Unlike prior work [22], we assume that models are induced from the entire training set (rather than just from G). This is because both parametric and non-parametric models induced from all available data have been shown to be superior to models induced when instances with missing values are ignored [10]. Beyond improved accuracy, the choice of model induction setting also bears important implications for the active acquisition mechanism, because the estimation of an acquisition's marginal utility is derived with respect to the model. We discuss this issue and its implications in detail in Section 4. Note that some induction algorithms (e.g., C4.5) include an internal mechanism for incorporating instances with missing feature-values [15]; other induction algorithms require that missing values be imputed first before induction is performed [10]. For the latter learners, many imputation mechanisms are available to fill in missing values (e.g., multiple imputation, nearest neighbor) [11, 1]). Henceforth, we assume that the induction algorithm includes some treatment for instances with missing values.

We study active feature-value acquisition policies within a generic iterative framework, shown in Algorithm 1. Each iteration estimates the utility of acquiring complete feature information for each available incomplete example. The missing feature values of a subset $S \in I$ of incomplete instances with the highest utility values are acquired and added to T (these examples move from I to G). A new model is then induced from T, and the process is repeated. Different AFA policies correspond to different measures of utility employed to evaluate the informativeness of acquiring features for an instance. Our baseline policy, random selection, selects acquisitions at random, which implicitly tends to prefer examples from dense areas of the example space [17].

In this study we propose two active feature-value acquisition policies corresponding to the two observations made in Section 1.

2.2 Uncertainty Sampling

The first active feature-value acquisition policy we explore is based on the uncertainty principle that originated in work on optimum experimental design [9, 5] and has been extensively applied in the active learning literature for classification, regression and class probability estimation models [4, 3, 18]. The uncertainty notion had been proposed for the acquisition of class labels and has not been applied previously for feature-value acquisition. For a model trained on incomplete instances, acquiring missing feature-values is effective if it enables a learner to capture additional discriminative patterns that improve the model's prediction. Acquiring feature-values for an example is likely to have an impact, if the model is uncertain of its class membership. In contrast, acquiring feature-values of instances for which the current model already embeds strong discriminative patterns is not likely to impact model accuracy considerably. Our first policy, *Uncertainty Sampling*, is based on this observation. The *Uncertainty* utility measure captures the model's ability to distinguish between cases of different classes. For a probabilistic model, the absence of discriminative patterns in the data results in the model assigning similar likelihoods for class membership of different classes. Hence, the *Uncertainty* score is calculated as the absolute difference between the estimated class prob-

abilities of the two most likely classes. Formally, for an instance x, let $P_y(x)$ be the estimated probability that x belongs to class y as predicted by the model. Then the *Uncertainty* score is given by $P_{y_1}(x) - P_{y_2}(x)$, where $P_{y_1}(x)$ and $P_{y_2}(x)$ are the first-highest and second-highest predicted probability estimates respectively. At each iteration of the feature acquisition algorithm, complete feature information is acquired for the m incomplete instances with the lowest scores, i.e. the highest prediction uncertainties.

2.3 Error Sampling

Prediction uncertainty implies that the likelihood of correctly classifying an example is similar to that of misclassifying it. Hence *uncertainty* provides an indication of a model's performance and potential for improvement through feature acquisition. A more direct measure of the model performance and of the value of acquiring missing features for a particular instance is whether the instance has been misclassified by the current model. Additional feature values of misclassified examples may embed predictive patterns and improve the model's classification accuracy. Our second policy, *Error Sampling* is motivated by this reasoning. *Error Sampling* prefers to acquire feature-values for instances that the current model misclassifies. At each iteration, it randomly selects m incomplete instances that have been misclassified by the model. If there are fewer than m misclassified instances, then *Error Sampling* selects the remaining instances based on the *Uncertainty* score (defined earlier). Formally, the *Error Sampling* score for a potential acquistion is set to -1 for misclassified instances; and for correctly classified instances the *Uncertainty* score is used. At each iteration of the feature acquisition algorithm, complete feature information is acquired for the m incomplete instances with the lowest scores.

3 Experimental Evaluation

3.1 Methodology

We compared the two proposed strategies, *Uncertainty* and *Error Sampling*, to random feature acquisition and to each other. The performance of each system was averaged over five runs of 10-fold cross-validation. In each fold, we generated learning curves in the following fashion. Initially, the learner has access to all incomplete instances, and is given complete feature-values for a randomly selected subset, of size m, of these instances. The learner builds a classifier based on this data. For the active strategies, a sample of instances is then selected from the pool of incomplete instances based on the measure of utility using the current classification model. The missing values for these instances

Algorithm 1 Active Feature-Value Acquisition Framework

Given:
G - set of complete instances
I - set of incomplete instances
T - set of training instances, $G \cup I$
$\mathcal{L}$ - learning algorithm
m - size of each sample

1. Repeat until stopping criterion is met
2. Generate a classifier, $C = \mathcal{L}(T)$
3. $\forall x_j \in I$, compute $Score(C, x_j)$ based on the current classifier
4. Select a subset S of m instances with the highest utility based on the score
5. Acquire values for missing features for each instance in S
6. Remove instances in S from I and add to G
7. Update training set, $T = G \cup I$
8. Return $\mathcal{L}(T)$

are acquired, making them complete instances. A new classifier is then generated based on this updated training set, and the process is repeated until the pool of incomplete instances is exhausted. In the case of random selection, the incomplete instances are selected uniformly at random from the pool. Each system is evaluated on the held-out test set after each iteration of feature acquisition. As in [22], the test data set contains only complete instances, since we want to estimate the true generalization accuracy of the constructed model given complete data. The resulting learning curves evaluate how well an active feature-value acquisition method orders its acquisitions as reflected by model accuracy. Note that, at the end of the learning curve, all algorithms see exactly the same set of complete training instances. To maximize the gains of AFA, it is best to acquire features for a single instance in each iteration; however, to make our experiments computationally feasible, we selected instances in batches of 10 (i.e., sample size m = 10).

We can compare the performance of any two schemes, A and B, by comparing the errors produced by both, given that we are limited to acquiring a fixed number of complete instances. To measure this, we compute the percentage reduction in error of A over B and report the average over all points on the learning curve. The reduction in error is considered to be *significant* if the average errors across

the points on the learning curve of A is lower than that of B according to a paired t-test ($p < 0.05$). To compare two schemes across all domains we report the *Significant Win/Draw/Loss* record, which presents three values — the number of data sets for which algorithm A obtained better, equal, or worse performance than algorithm B with respect to error reduction. A win or loss is only counted if the error reduction is determined to be significant.

All the experiments were run on 5 web-usage datasets (used in [14]) and 5 datasets from the UCI machine learning repository [2]. The web-usage data contain information from popular on-line retailers about customer behavior and purchases. This data exhibit a natural dichotomy with a subset of features owned by a particular retailer and a set of features that the retailer may acquire at a cost. In particular, each retailer privately owns information about its customers' behavior as captured by web logfiles. The retailer's private data contain features such as user demographics, the time of the session or whether the session occurred on a weekday. These are referred to as *site-centric* features. In addition, the data contain information that is not owned by any individual retailer, capturing each customer's aggregated behavior and purchasing patterns across a variety of on-line retailers. These are referred to as *user-centric* features. The learning task is to induce models to predict whether a customer will purchase an item during a visit to the store. The web usage data has a clear division of features—the first 15 are site-centric and the rest are user-centric. Hence the pool of incomplete instances was initialized with only the first 15 features. We selected several UCI datasets that had more than 25 features. For these datasets, 30% of the features were randomly selected to be used in the incomplete instances. A different set of randomly selected features was used for each train-test split of the data. All the datasets used in this study are summarized in Table 1.

Table 1. Summary of Data Sets

Name	Instances	Classes	Features
bmg	2417	2	40
expedia	3125	2	40
qvc	2152	2	40
etoys	270	2	40
priceline	447	2	40
anneal	898	6	38
soybean	683	19	35
kr-vs-kp	3196	2	36
hypo	3772	4	29
autos	205	6	25

The active framework and specific policies we have proposed can be implemented using an arbitrary probabilistic classifier as a learner. For the results in this paper, we used J48, which is the Weka [21] implementation of C4.5 decision-tree induction [16].

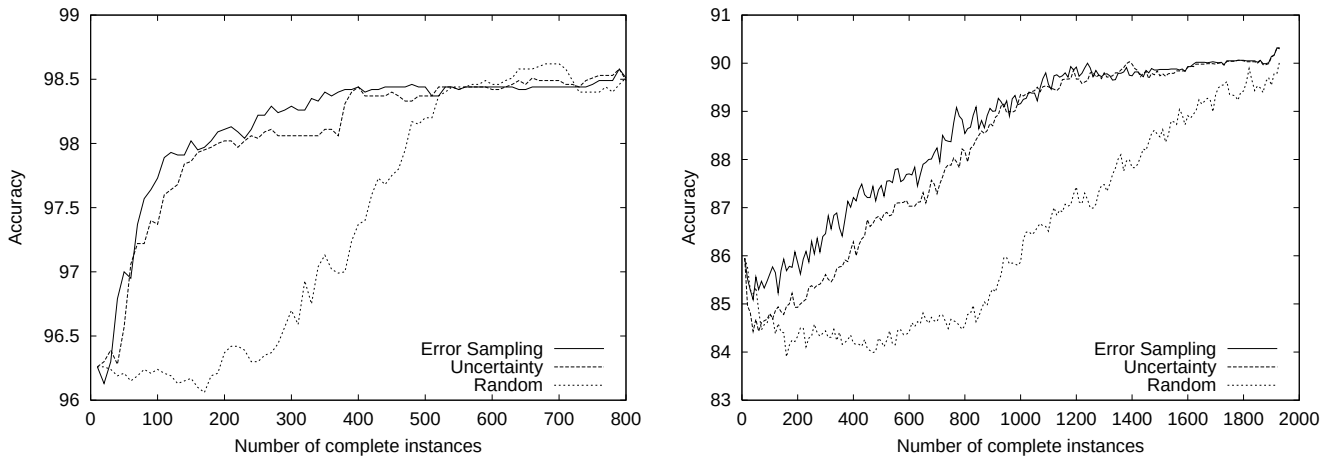
3.2 Results

The results comparing *Uncertainty* and *Error Sampling* to random selection are summarized in Table 2. In this (and subsequent) tables, a significant error reduction is indicated in bold. The significant win/draw/loss record is also summarized at the bottom of the table. As mentioned above, towards the end of the learning curve, all methods will have seen almost all the same training examples. Hence, the main impact of AFA is lower on the learning curve. To capture this, we also report the percentage error reduction averaged over only the 20% of points on the learning curve where the largest improvements are produced. We refer to this as the *top-20% percentage error reduction*, which is similar to a measure reported in [18].

The results show that for all data sets using either *Uncertainty* or *Error Sampling* significantly improves on the model accuracy compared to random selection. The performance of *Uncertainty* demonstrates that although prediction uncertainty was originally proposed for acquiring missing class labels, and employs a noisy signal for model accuracy, it provides effective information for feature acquisitions as well.

Figure 1 presents learning curves that demonstrate the advantage of using either AFA scheme over random acquisition. Apart from average reduction in error, a good indicator of the effectiveness of an active feature-value acquisition scheme is the number of acquisitions required to obtain a desired accuracy. For example, on the *anneal* data set, *Error Sampling* achieves an accuracy of 98% with only 200 acquisitions of complete instances. In contrast, random selection requires more than 400 complete instances to achieve the same accuracy level. Similarly, on *qvc* once *Error Sampling* acquires approximately 400 complete instances, it induces a model with an accuracy of 87%; however, random selection requires approximately 1200 complete instances to achieve the same accuracy.

In order to evaluate the value of information provided by model misclassifications employed by *Error Sampling*, Table 3 summarizes the average reduction in error obtained by *Error Sampling* compared to *Uncertainty*. Acquisitions made by *Error Sampling* lead to significantly superior models on average for 6 of the data sets; and the two policies exhibit comparable performance for the remaining 4 data sets. *Uncertainty* is not superior to *Error Sampling* on any of the domains. For the data sets where *Error Sampling* exhibits statistically significant improvement in accuracy, the top-20% improvements range between 2.02% (*soybean*) and 18.17% (*kr-vs-kr*). The difference in the number of acquisitions required by each policy to achieve a particular accuracy is sometimes quite substantial. For example, for the



(a) anneal

(b) qvc

Figure 1. Comparing active strategies to random feature acquisition.

Table 2. Comparing active policies: Error reduction with respect to random selection.

	%Error Reduction		Top-20% %Err. Red.	
Dataset	*Uncert.*	*Err. Samp.*	*Uncert.*	*Err. Samp.*
bmg	**10.55**	**10.67**	15.53	17.77
etoys	**9.03**	**10.34**	20.95	23.88
expedia	**14.56**	**19.83**	24.99	29.12
priceline	**25.05**	**24.45**	35.66	34.49
qvc	**13.12**	**15.44**	22.64	24.75
anneal	**20.52**	**22.65**	45.93	49.27
soybean	**7.56**	**8.03**	14.16	14.79
autos	**4.10**	**4.24**	9.65	10.50
kr-vs-kr	**33.03**	**36.82**	45.81	53.23
hypo	**10.39**	**16.79**	30.26	40.48
Mean	14.79	16.93	26.56	29.83
Sig. W/D/L	10/0/0	10/0/0		

qvc data set *Error Sampling* achieves an accuracy of 87% with approximately 400 acquisition of complete instances, whereas *Uncertainty* requires approximately 600 complete instances to achieve the same accuracy. The relative performance of the policies demonstrates that *Error Sampling* acquires more informative feature-values, and confirms that feature-values of misclassified examples allow the induction scheme to capture better predictive patterns than are captured from feature-values acquired using *Uncertainty*.

Error Sampling prefers acquisition of feature values for examples that are misclassified by the current model, but treats all misclassified examples equally. We also considered two variants of *Error Sampling* that rank different misclassified examples based on the model's uncertainty of prediction. One variant prefers misclassified examples that the model is most uncertain about, and the other prefers misclassified examples that the model is most confident of. In our experiments, both variants outperformed random acquisitions, but neither produced significant improvement with respect to *Error Sampling*.

Table 3. Comparing *Error Sampling* **to** *Uncertainty***.**

Dataset	%Error Reduction	Top-20% %Err. Red.
bmg	0.14	4.55
etoys	1.28	11.01
expedia	**5.96**	12.88
priceline	-0.90	9.93
qvc	**2.64**	6.88
anneal	**3.58**	12.78
soybean	**0.53**	2.02
autos	0.16	3.99
kr-vs-kr	**6.73**	18.17
hypo	**7.37**	16.38
Mean	2.75	9.86
Sig. W/D/L	6/4/0	

4 Comparison with GODA

The most closely related work to this paper is the study by Zheng and Padmanabhan [22] of the active feature-value acquisition scheme GODA. GODA measures the utility of acquiring feature-values for a particular incomplete instance in the following way. It adds the instance to the training set, imputing the values that are missing. It then induces a new model and measures its performance on the training set. This process is repeated for each incomplete instance, and the instance that leads to the model with the best expected performance is selected for feature-value acquisition.

GODA has an important difference from the methods we have proposed: it induces its models from only the complete instances—ignoring the incomplete instances. Whether one chooses to use or to ignore incomplete instances when inducing a model has a significant bearing on the acquisition scheme. GODA estimates the value of potential acquisitions by the model's improved performance resulting from adding the example to the training set. This confounds the improvement due to acquiring the previously unknown feature values with the improvement due to including the already known feature values. In contrast, the policies we propose estimate the marginal utility of missing feature acquisition with respect to a model induced from all available data. GODA's measure of utility cannot be employed directly when the models are induced from all incomplete instances including imputations of their missing features. Nevertheless, since GODA is (to our knowledge) the only other technique designed for the same acquisition setting, it is informative to compare performance with our approach.

To compare to our approach, we implemented GODA as described in [22], using J48 tree induction as the learner and using accuracy as the *goodness measure* of the model. As in [22], we use *multiple imputation* with Expectation-Maximization to impute missing values for incomplete instances. Experiments comparing *Error Sampling* to GODA were run as in Section 3.1. However, due to GODA's tremendous computational requirements, we only ran one run of 10-fold cross-validation on three of the datasets. The datasets were also reduced in size to make running GODA feasible.

A summary of the results, along with the reduced dataset sizes, is presented in Table 4. The results show that in spite of the high computational complexity of GODA, it results in inferior performance compared to *Error Sampling* for all three domains. All improvements obtained by *Error Sampling* with respect to GODA are statistically significant. Figure 2 presents learning curves for the *priceline* dataset that clearly demonstrate the superior performance of *Error Sampling*. These results suggest that the ability of *Error Sampling* to capitalize on information from incomplete instances, and to utilize this knowledge in feature acquisition, allows it to capture better predictive patterns compared to those captured by GODA.

Recall that when an instance is selected for acquisition, *Error Sampling* adds to the training data only the acquired feature values. GODA, however, adds to the training data the entire instance, i.e., the feature values that are known ex ante (but that are not used for induction by GODA [1]) as well as the acquired feature values and the instance's class membership. Hence, even when the same instance is selected by GODA and by *Error Sampling*, the relative increase in accuracy for GODA is likely to be greater than the increase obtained for a model induced with *Error Sampling*. This difference contributes to the steep learning curve exhibited by the model generated in GODA. Similarly, because the marginal contributions of missing feature values are small, improvement in learning is more substantial over many acquisitions as observed for the large data sets in Figure 1, than for a small number of acquisitions depicted in Figure 3.

In addition to superior accuracy for a given number of acquisitions, the methods we have introduced also have the advantages of being simple-to-implement and having a relatively low computational complexity. GODA, on the other hand, requires inducing a different model for estimating each potential acquisition (i.e., $|I|$ models are induced). Hence for even moderately large data sets this approach is prohibitively expensive, except (perhaps) with an incremental learner such as Naive Bayes. The policies we propose are significantly more efficient because only a single model is induced for estimating the utilities of an arbitrarily large number of potential feature acquisitions.

For the sake of completeness, we are currently running experiments comparing GODA and *Error Sampling* for the remaining datasets.

Table 4. Comparing *Error Sampling* with GODA: Percent error reduction.

Dataset	Size	% Error Reduction
bmg	200	**19.48**
qvc	100	**20.03**
priceline	100	**17.75**

5 Related Work

Recent work on *budgeted learning* [12] also addresses the issue of active feature-value acquisition. However, the

[1] This explains why GODA starts with lower accuracy.

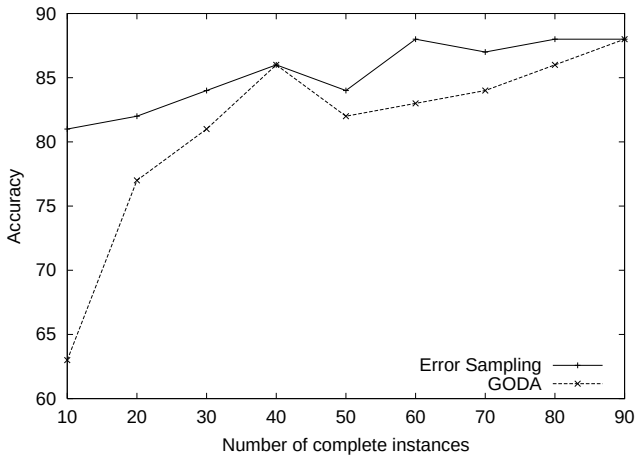


Figure 2. Comparing *Error Sampling* **to** GODA **on** *priceline*

policies developed in [12] assume feature-values are discrete, and consider the acquisition of individual feature-values for instances of a given class (i.e., queries are of the form "acquire value of feature f for some instance in class c."). Therefore, unlike our approach, the policies do not consider requesting additional features for a specific incomplete instance. In addition, the policies cannot be directly applied to estimate the value of acquiring sets of features (as is required in our problem setting). Another important aspect of the policies proposed in [12] is that for each feature and class membership they require estimating the performance of all models induced from each possible value assignment. The induction of most learners is not incremental, hence for each feature class pair, a new model is required to be induced for each value assignment. Although the framework proposed in [12] was not designed to solve the problem discussed here, one may consider an extension to this framework for estimating the utility of acquiring values for a set of features for incomplete instances. However, the number of possible value assignments, and consequently the number of model inductions required will increase considerably. It is unclear whether an algorithm with such a high complexity would be feasible in practice.

Some work on *cost sensitive* learning [20] 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 [19], 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 [7] also addresses the issue of economical feature acquisition during both training and testing; however, it also adopts a very simple 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* [3, 6] 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 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. As described above, our notion of uncertainty is taken directly from prior work on traditional active learning.

6 Conclusions and Future Work

We have presented a general framework for active feature acquisition that can be applied to different learners and can use alternate measures of utility for ranking acquisitions. Within this framework, we show how a fundamental idea from traditional active learning, *uncertainty sampling*, can be applied directly. We also describe an alternative policy that in contrast to traditional active learning, in which class labels are unknown, utilizes the correctness of the current model for each example. We show empirically that both these policies, *Uncertainty* and *Error Sampling*, significantly improves the accuracy of models learned for fixed feature acquisition budgets, when compared with a policy that requests features randomly. The experiments also establish that using the additional information available in this setting—viz., whether or not the predictions of a model learned on the incomplete data are *correct*— yield statistically significant (and sometimes substantial) increases in accuracy over just using the confidence of the predictions as in *Uncertainty*.

A direct comparison of *Error Sampling* with GODA, an alternate active feature-value acquisition approach, demonstrates that in spite of its simplicity, *Error Sampling* exhibits superior performance. *Error Sampling*'s utilization of all known feature-values and of a simple measure of the potential for improvement from an acquisition, results in computational and model accuracy advantages. A more extensive comparative evaluation of the policies with additional data sets is currently underway.

We have only begun to explore the space of utility functions in our framework. There are other factors that could be incorporated into our utility measures. For example, Saar-Tsechansky and Provost [17] show in traditional active learning that estimating the variance of class probability estimates can improve the active learner. The measures of uncertainty we consider here only consider the estimate of the probability itself—in essence, the current methods attempt to reduce a bias in the class probability estimations.[2] However, generally both the bias and the variance contribute to the error of modeling techniques. It may be profitable to consider the bias, the variance, and an estimate of their effect on classification.

Similarly to previous studies on active feature acquisition [22] the test instances in this study are complete, in order to estimate the models performance without confounding effects of incomplete values in test instances. However, it also is important to explore the implications for feature acquisition policies for different missing value patterns in the test instances.

The effectiveness, simplicity, and computational efficiency of *Error Sampling* argues that this policy should be considered by any practitioner or researcher faced with the problem of feature set acquisition. From a research perspective, we suggest that the *Error Sampling* policy be a baseline (in addition to random selection) for future studies of active feature selection.

Acknowledgments

We would like to thank Balaji Padmanabhan and Zhiqiang Zheng for providing us with the web usage datasets. Prem Melville and Raymond Mooney were supported by DARPA grant HR0011-04-1-007.

References

[1] G. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17:519–533, 2003.

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

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

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

[5] V. Federov. *Theory of optimal experiments*. Academic Press, 1972.

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

[7] R. Greiner, A. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174, 2002.

[8] J. Hagel and M. Singerare. *Net Worth: Shaping Markets When Customers Make the Rules*. Harvard Business School Press, 1999.

[9] J. Keifer. Optimal experimental designs. *Journal of the Royal Statistical Society*, 21B:272–304, 1959.

[10] M. Leigh and L. James. Comparison of imputation techniques: Accuracy of imputations, imputed data parameters, imputed model, parameters and quality of marketing decisions implied by the estimated models. Working paper, Department of Marketing, Red McCombs School of Business, University of Texas at Austin, 2004.

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

[12] D. Lizotte, O. Madani, and R. Greiner. Budgeted learning of naive-bayes classifiers. In *Proc. of 19th Conf. on Uncertainty in Artificial Intelligence (UAI-03)*, Acapulco, Mexico, 2003.

[13] New York Times. Doubleclick to buy retailing database keeper. June 5, 1999.

[14] B. Padmanabhan, Z. Zheng, and S. O. Kimbrough. Personalization from incomplete data: what you don't know can hurt. In *Proc. of 7th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-2001)*, pages 154–163, 2001.

[15] J. R. Quinlan. Unknown attribute values in induction. In *Proc. of 6th Intl. Workshop on Machine Learning*, pages 164–168, Ithaca, NY, June 1989.

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

[17] M. Saar-Tsechansky and F. Provost. Active sampling for class probability estimation and ranking. *Machine Learning*, 54:153–178, 2004.

[18] M. Saar-Tsechansky and F. J. Provost. Active learning for class probability estimation and ranking. In *Proc. of 17th Intl. Joint Conf. on Artificial Intelligence (IJCAI-2001)*, pages 911–920, 2001.

[19] M. Tan and J. C. Schlimmer. Two case studies in cost-sensitive concept acquisition. In *Proc. of 8th Natl. Conf. on Artificial Intelligence (AAAI-90)*, pages 854–860, Boston, MA, July 1990.

[20] P. D. Turney. Types of cost in inductive concept learning. In *Proc. of the Workshop on Cost-Sensitive Learning at the 17th Intl. Conf. on Machine Learning*, Palo Alto, CA, 2000.

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

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

[2] In this setting, the bias is due to having incomplete data.