Active Feature 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 estimations of the value of acquisition. Active feature 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 these approaches can induce accurate models using substantially fewer feature acquisitions, and suggest promising directions for improvements.

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 through an intermediary. For building accurate models, ignoring instances with missing values leads to inferior model performance [20, 11], 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.

Consider for example an on-line retailer learning a predictive model to estimate customers’ propensities to buy. The retailer may use private information on its customers’ buying behavior over time, as captured from web logfiles. 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 [9]. Acquiring complete data for all customers may be prohibitively expensive [18]. The retailer should have a cost-efficient feature acquisition strategy that indicates the customers for which it should acquire complete data.

In this paper we address this problem of active feature 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 \( a_1, \ldots, a_n \). 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 [10, 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 4 (including work on the related problem of feature acquisition for model use rather than model building).

In sum, the main advantages of the approaches we study here are that they provide generic principles for active acquisitions that are also simple and computationally efficient. Prior techniques either are tailored to particular inductive algorithms [30] or are very expensive computationally [29]. The techniques we present apply to most classifier induction methods; they are very fast computationally, and are effective for a variety of domains.

These proposed policies for active feature acquisition are based on two straightforward 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 acquisition—during training—the correct classifications are known. Therefore, unlike with traditional active learning, it is possible to employ direct measures of a model’s accuracy for estimating the value of potential acquisitions.
These two observations define a space of possible measures that can be used to define feature acquisition policies: prioritizations of training examples for feature acquisition. The main claim of this paper is that these simple and generic measures (accuracy and confidence of predicted classifications) result in feature acquisition policies that perform remarkably well. We present and evaluate two intuitively appealing policies, one that only uses (lack of) prediction confidence and another that considers accuracy as well. We compare the policies experimentally with each other and with a baseline policy, across a variety of domains. Compared to the baseline, they both require significantly fewer feature acquisitions to induce a model with a given accuracy. We do not find strong evidence that either policy dominates; however, when one policy is substantially better, it tends to be the one that considers both confidence and accuracy.

2. TASK DEFINITION AND ALGORITHM

2.1 Pool-based Active Feature Acquisition

We follow a general setting described by prior work [29]. 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 a subset of the feature values \( a_0, \ldots, a_t \) are known. The values of the remaining features \( a_{t+1}, \ldots, a_n \) 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 [29], 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 [11]. Some induction algorithms include an internal mechanism for handling instances with missing feature values [20, 11]; other induction algorithms require that missing values be imputed first before induction is performed. For the latter learners, many imputation mechanisms are available to fill in missing values (e.g., multiple imputation, nearest neighbor) [12, 1]). Henceforth we assume that the inducer include some treatment for instances with missing values and do not discuss imputation separately from induction.

We study active feature 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 feature values of a subset \( S \subseteq I \) 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 policies correspond to different measures of utility employed to evaluate the informativeness of acquiring features for an instance. Our baseline policy, random selection, is equivalent to using a random utility function—corresponding to random selection from a uniform utility function, which implicitly tends to prefer examples from dense areas of the example space [23].

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

2.2 Uncertainty Sampling

Acquiring missing feature values is effective when the acquired information enables a learner to capture additional predictive patterns and improve the model’s ability to discriminate between instances of different classes. When a model trained on incomplete instances does not capture effective discriminative patterns, it is unable to confidently classify some instances. For such instances, it is useful to acquire additional feature values to enable the model to capture predictive patterns that may exist for these features. Our first approach, Uncertainty Sampling, is based on this observation. The notion of using model uncertainty for active data acquisition originated in work on optimum experimental design [10, 5] and has been extensively explored in the active learning literature for classification, regression, and class probability estimation models [4, 3, 24].

The Uncertainty utility measure captures the model’s ability to distinguish between cases of different classes, and assigns a higher preference to instances where the current model’s discriminative power is low. For a probabilistic model, absence of predictive 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 estimated class probabilities 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 Incorrect Certainty Sampling

While prediction uncertainty is an effective notion underlying active learning methods, it is merely an indirect measure of a model’s performance and potential for improvements through feature acquisition. Acquiring information from instances for which the model’s prediction is merely uncertain, even when it is correct (as is done by Uncertainty), may only reinforce the model’s current hypothesis, or worse, lead to overfitting. A more direct measure of the informative value of missing features for a particular instance is whether the instance has been misclassified by the current model. More information regarding misclassified examples is likely to improve the model’s classification accuracy. Our second approach, Incorrect Certainty is motivated by this reasoning. Incorrect Certainty assigns a high preference to information on examples for which strong patterns in the data lead to a confident prediction that is incorrect. Information about such instances is likely to differ largely from evidence currently in the data and hence may be a valuable input for updating the current model.

Formally, given that the true class for instance \( x \), is \( t \), the Incorrect Certainty score is computed as \( P_t(x) - \max\{P_y(x) | y \neq t\} \). This measure is often referred as the classification mar-
gin of an instance [25]. At each iteration Incorrect Certainty sampling acquires complete feature values for the \( m \) instances with the lowest scores. So, for feature acquisition, this approach first prefers instances that have been misclassified — ranking them from most to least confidently misclassified. It then prefers instances that have been correctly classified with the highest uncertainty.

Algorithm 1: Active Feature 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
- \( k \) - number of selective sampling iterations
- \( m \) - size of each sample

1. Repeat \( k \) times
2. Generate a classifier, \( C = \mathcal{L}(T) \)
3. \( \forall x_j \in I \), compute Score\( (C, x_j) \)
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 \( \mathcal{L}(T) \)

3. EXPERIMENTAL EVALUATION

3.1 Methodology

We compared the two proposed strategies to random feature acquisition. The performance of each system was averaged over two runs of 10-fold cross-validation. In each fold of cross-validation, we generated learning curves in the following fashion. Initially, the learner is only given incomplete 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 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 [29] 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. The resulting learning curves evaluate how well an active feature acquisition method orders the set of incomplete instances in terms of the utility of acquiring their features. 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 20.

The performance of an AFA scheme can be evaluated by how much it improves accuracy over random sampling given that we are limited to acquiring a fixed number of complete instances. To measure this, we compute the percentage reduction in error over random sampling and report the average over all points on the learning curve. 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 top 20% of points on the learning curve where the largest improvements are produced. This is similar to a measure reported in [24].

All the experiments were run on 5 web-usage datasets (used in [19]) and 5 datasets from the UCI machine learning repository [2]. The web-usage data we used contains information from top on-line retailers about customer behavior and purchases. This data exhibits a natural dichotomy with a subset of features owned by a particular retailer and additional 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, whether the session occurred on a weekday or a weekend and whether the customer purchased an item by the end of the session. These are referred to as site-centric features. In addition, the data contain information about consumers that is not owned by any individual retailer, capturing customers’ 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. All the datasets used in this study are summarized in Table 1. We selected UCI dataset that had more than 25 features. The web usage data has a clear division of features—the first 15 are site-centric and the rest are user-centric. So the pool of incomplete instances was initialized with only the first 15 features. For the UCI datasets, we create an artificial split of the feature set, using only the first 30% of the features in the incomplete instances.

<table>
<thead>
<tr>
<th>Name</th>
<th>Instances</th>
<th>Classes</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmng</td>
<td>2417</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>expedia</td>
<td>3125</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>qvc</td>
<td>2152</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>eToys</td>
<td>270</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>priceline</td>
<td>447</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>anneal</td>
<td>898</td>
<td>6</td>
<td>38</td>
</tr>
<tr>
<td>soybean</td>
<td>683</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>1000</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>hypo</td>
<td>1000</td>
<td>4</td>
<td>29</td>
</tr>
<tr>
<td>autos</td>
<td>205</td>
<td>6</td>
<td>25</td>
</tr>
</tbody>
</table>

The active framework and specific policies we have proposed can be implemented using an arbitrary probabilistic classi-
3.2 Results using Tree Induction

We first applied our methods to J48, which is the Weka [28] implementation of C4.5 decision tree induction [21]. The results comparing the different active feature acquisition policies to random selection for J48 are summarized in Table 2. In general, using Incorrect Certainty or Uncertainty significantly improves on the accuracy of J48 compared to random selection. Although random selection obtains features from a representative set of instances, features acquired by the active policies apparently are more informative for induction. The learning curves (Figure 1) demonstrate that Incorrect Certainty and Uncertainty both produce effective rankings of the instances by their informativeness.

Neither active policy is consistently superior to the other across domains. Let us consider that one policy substantially outperforms the other (somewhat arbitrarily) if it has a higher average error reduction and it exhibits at least a 5 percentage-point improvement in top-20% error reduction. Incorrect Certainty substantially outperforms Uncertainty for three of the datasets, whereas Uncertainty does not substantially outperform Incorrect Certainty for any dataset. The learning curves provide a qualitative view (see Figures 1(a)-1(c)).

The most striking gains of active feature acquisition are seen on anneal; where, with only 200 complete instances, Incorrect Certainty achieves an accuracy of 99%, compared to random selection which only achieves 90%. Furthermore, it takes random selection twice as many (approximately 400) acquisitions of complete instances to achieve the same accuracy. The learning curves for anneal are quite steep, hence the advantage from informative feature values is captured well by the error reduction metric. For other datasets, such as QVC, learning in general requires more feature acquisitions to improve accuracy (the learning curves are not as steep). For such domains the advantage of active acquisitions is better captured by the number of examples required by either policy to obtain a given accuracy level. For example, on QVC once Incorrect Certainty acquires approximately 400 complete instances, it induces a model with an accuracy of 87%. In contrast, random selection requires approximately 1200 acquisitions to obtain the same accuracy level.

On one dataset, hypo, using the active strategies seem to perform worse than random selection, according to the percentage error reduction. It appears that the first 9 features of this dataset (those present in the incomplete instances) are not very discriminative. As such, it takes several complete instances before the learner can start discriminating between the different classes. The majority class accounts for 91.3% of the instances, which results in seemingly high accuracy even without many discriminative features. As can be seen from the Figure 1(f), Incorrect Certainty is the first algorithm to pick instances that start providing more discriminative information. However, random selection’s advantage in selecting a representative data sample is clear.

Table 2: Comparing active policies for J48: Percent error reduction with respect to random selection.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>%Error Reduction</th>
<th>Top 20% %Err. Red.</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmg</td>
<td>10.92 8.19</td>
<td>21.40 14.28</td>
</tr>
<tr>
<td>expedia</td>
<td>12.11 9.01</td>
<td>22.47 18.53</td>
</tr>
<tr>
<td>qvc</td>
<td>16.64 13.87</td>
<td>26.65 24.65</td>
</tr>
<tr>
<td>etoys</td>
<td>14.30 16.74</td>
<td>31.67 31.00</td>
</tr>
<tr>
<td>priceline</td>
<td>21.01 24.38</td>
<td>35.86 38.79</td>
</tr>
<tr>
<td>anneal</td>
<td>19.58 11.07</td>
<td>86.71 62.36</td>
</tr>
<tr>
<td>soybean</td>
<td>2.61 5.55</td>
<td>9.93 14.31</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>8.39 -2.80</td>
<td>26.64 6.44</td>
</tr>
<tr>
<td>hypo</td>
<td>-18.84 -10.46</td>
<td>14.17 0.00</td>
</tr>
<tr>
<td>autos</td>
<td>7.02 7.01</td>
<td>14.74 13.16</td>
</tr>
<tr>
<td>Mean</td>
<td>7.53 7.20</td>
<td>27.20 27.25</td>
</tr>
</tbody>
</table>

3.3 Results on DECORATE

In addition to J48, we also tested the active feature acquisition strategies with an ensemble learner (DECORATE) [15]. DECORATE constructs a diverse set of hypotheses using additional artificially constructed training examples. The technique is a simple, general meta-learner that can use any strong learner as a base classifier to build diverse committees. We used DECORATE for the following three reasons: (1) DECORATE ensembles of decision trees produce higher accuracies than single trees [14]; (2) DECORATE has been successfully used for active learning using the Uncertainty measure described here [16]; and (3) DECORATE is more resilient to missing features than single decision trees, Bagging, and AdaBoost [17].

In our experiments, we built DECORATE ensembles of 15 classifiers, using J48 as our base learner and generated learning curves as described in Section 3.1. However, to make the experiments tractable when using DECORATE, we used only the first 1000 instances of larger datasets. The results comparing the different active feature acquisition policies to random selection for DECORATE are summarized in Table 3. As with single trees, using either active strategy gives considerable improvements in accuracy over DECORATE using random selection. However, the difference between Incorrect Certainty and Uncertainty is far less pronounced than it was for trees. In particular, there are no substantial differences (as defined above) between the two policies. This is primarily because DECORATE tends to fit the training data fairly well. If for all the training data, the correct class is the class with the highest probability, then Incorrect Certainty is equivalent to Uncertainty as there is no difference in the scores they employ.

DECORATE is more accurate; hence, improving on it is a more challenging task. But as can be seen from the result for the top 20% error-reduction metric, actively selecting features for DECORATE results in substantial improvements over random sampling. As discussed above, this metric gives an indication of the magnitude of error reduction that can be expected where active feature acquisition has the most impact. Depending on the dataset, Incorrect Certainty produces a wide range of improvements in this metric, from moderate (13.89% on etoys) to high (85.39% on anneal).

Figure 2 presents datasets which clearly demonstrate the advantage of using active feature acquisition over random.