Confidence-Based Feature Acquisition to Minimize Training and Test Costs

Marie desJardins* James MacGlashan† Kiri L. Wagstaff‡

Abstract

We present Confidence-based Feature Acquisition (CFA), a novel supervised learning method for acquiring missing feature values when there is missing data at both training and test time. Previous work has considered the cases of missing data at training time (e.g., Active Feature Acquisition, AFA [8]), or at test time (e.g., Cost-Sensitive Naive Bayes, CSNB [2]), but not both. At training time, CFA constructs a cascaded ensemble of classifiers, starting with the zero-cost features and adding a single feature for each successive model. For each model, CFA selects a subset of training instances for which the added feature should be acquired. At test time, the set of models is applied sequentially (as a cascade), stopping when a user-supplied confidence threshold is met. We compare CFA to AFA, CSNB, and several other baselines, and find that CFA's accuracy is at least as high as the other methods, while incurring significantly lower feature acquisition costs.

1 Introduction

In classification problems, there are often missing data (feature values) in the training set, the test set, or both. In some situations, the missing features may be available at a cost. Examples range from medical diagnosis tasks, where features are test results that have varying costs, to Mars rover exploration, where features come from instruments with varying power and bandwidth costs. Determining which feature values to acquire is sometimes referred to as the *feature acquisition* problem. Previous work in this area has focused on acquiring either missing training data, or missing test data, but not both (see Section 2). We address the more general problem of feature acquisition in the case where there is missing data in both the training and the test set. Generally speaking, one can achieve higher classification performance by paying the cost of acquiring additional features. We do not assume that we know in advance how many instances will be classified by the model (i.e., its "live time"), so we cannot explicitly trade off total cost at training time against total cost at test time. Rather, our goal is to minimize the total acquisition cost (at training and test time) needed to achieve a desired level of expected per-instance predictive performance. Note that this scenario is applicable in a setting in which total training and test costs are expected to be reasonably well balanced. If they are not, alternative strategies may be preferred: for a model with an extremely short live time (few test items), full acquisition of the test features may be desirable, while for a model with an extremely long live time (many test items), full acquisition of the training features, and selective acquisition of test features, may be most effective.

We present Confidence-based Feature Acquisition (CFA), a confidence-based ensemble approach to minimizing the total acquisition cost for problems in which instances have missing features. A user-supplied confidence threshold parameter is used to determine the target performance level. The method can be applied using any classifier that provides a confidence value on its predictions. As is typical in machine learning research [11], we employ the posterior probability as a confidence value. The width of a confidence interval, or another metric, could also be used as a measure of confidence without modifying the underlying CFA algorithm.

In CFA, we train a succession of models, $M_0 \ldots M_f$, such that M_0 uses only the "free" (zero-cost) features, and M_i additionally incorporates costly features F_1 through F_i. Model M_i is trained only on instances that cannot be classified with sufficient confidence by model M_{i-1}. Therefore, values for feature F_i are acquired only for the instances that require it. At test time, each test instance is successively classified by $M_0, M_1, M_2 \ldots$ until its classification is sufficiently confident (i.e., until the confidence of the prediction reaches the confidence threshold). Again, features are acquired for the new instance only as required.

In this paper, we report on the observed cost-accuracy tradeoff for several data sets. We compare our Confidence-based Feature Acquisition (CFA) approach to existing methods that can only acquire missing data during training (Active Feature Acquisition [8]) or during testing (Cost-Sensitive Naive Bayes [2]). We show that CFA achieves predictive accuracy equal to or greater than both methods and several other baselines, while incurring significantly less cost. In addition, CFA is not dependent on the kind of

*University of Maryland, Baltimore County, mariedj@cs.umbc.edu

†University of Maryland, Baltimore County, jmac1@cs.umbc.edu

‡Jet Propulsion Laboratory, California Institute of Technology, kiri.wagstaff@jpl.nasa.gov

base classifier used; we show results using CFA with Naive Bayes, J48 decision trees, and support vector machines as the base classifiers. CFA performs well across a range of base classifiers, but no single base classifier dominates performance (i.e., provides the highest accuracy or the lowest cost). This suggests that the flexibility to use CFA with different base classifiers is a benefit of the algorithm, since different classifiers will provide the best performance in different domains.

2 Related Work

Existing methods for minimizing both misclassification cost and the cost of acquiring feature values can be characterized by whether missing instance values can be acquired during training, testing, or both phases. An *incomplete* instance has one or more missing feature values; a *complete* instance has no missing values.

Feature Acquisition During Training. In some cases, the training instances are incomplete, and the goal is to minimize the cost of acquiring feature values for some of these instances in order to build a classifier that operates on complete data at test time. Zheng and Padmanabhan [19] present two approaches for solving this problem: AVID (Acquisition based on Variance of Imputed Data) and GODA (Goal-Oriented Data Acquisition). AVID imputes the missing values and then acquires the values about which it is least certain. GODA decides which missing values to acquire by imputing the missing values, training a classifier using the original and imputed data, and then acquiring all missing feature values for misclassified instances. After acquiring missing values, AVID and GODA are trained only on the complete training instances.

In contrast with the single-pass acquisition methods of AVID and GODA, Active Feature Acquisition (AFA) incrementally acquires missing values to improve performance, at increasing cost. The initial classifier is trained on a fully imputed version of the original data set. AFA may request all feature values for a batch of m misclassified training instances [8] (like GODA) or may define a utility function and select m individual missing feature values to query in order to maximize the resulting utility [9, 10]. The latter approach is significantly more expensive computationally, since it requires training one classifier per feature to predict the missing values and then estimating the accuracy improvement that would be obtained by re-training with a single missing value filled in. A more recent version of AFA [13] uses subsampling to reduce the computational cost.

Each of these methods was evaluated in terms of the tradeoff between acquisition cost during training and misclassification cost during testing.

Feature Acquisition During Testing. In other cases, the goal is to train a model from complete data and minimize the cost of classifying new, incomplete instances. Greiner et al. [5] described classifiers that can request feature values at test time as "active" classifiers and provided a PAC-style analysis of learning an optimal active classifier, given a fixed budget. Cost-sensitive decision trees [7] and naive Bayes models (CSNB [2]) have been proposed that minimize acquisition and misclassification costs incurred at test time. The training data may contain incomplete instances, but they are treated as permanently incomplete; no acquisition of these values is considered. CSNB incorporates a misclassification cost matrix to aid in trading off acquisition costs against the cost of classification errors. CSNB was evaluated in terms of total cost (acquisition and misclassification) as a function of the percentage of missing values in the test set. This approach can be extended to also account for delay costs, which can make batch feature acquisition desirable [14].

An alternative approach is to model the feature acquisition process as a sequence of decisions either to acquire a feature value or to output a classification (and terminate), using an HMM [6]. In this work, a corresponding POMDP was trained on randomly generated sequences and their associated costs, then tested on incomplete data.

Feature Acquisition During Training and Testing. To our knowledge, no work has considered the fully generalized problem of deciding which feature values to acquire in both phases. We present Confidence-based Feature Acquisition (CFA), the first method to directly apply to problems in which features can be acquired during training and testing. Note that because it may not be known at training time how many test instances will need to be classified, it is difficult or impossible to explicitly compute a tradeoff between training-time FA cost, test-time FA cost, and test-time misclassification cost. Therefore, the goal in our scenario is to achieve a *target* level of accuracy (specified in terms of classification confidence) at *minimum cost*.

Estimating Confidence Levels. Our approach relies on the use of confidence values at both training and test time to determine which instances have uncertain predictions; additional feature values are acquired for these instances. However, it is well known that most classifiers do not provide reliable confidence values.

As long as the provided confidence values increase monotonically with the "true" confidence values, the performance of CFA should not be affected significantly, since the least confident instances will still be the first ones selected for feature acquisition. However, the decision of when to stop acquiring features (i.e., when the target confidence level has been reached) will be skewed by the inaccurate predictions. This skew would be mitigated by more accurate confidence values.

Niculescu-Mizil and Caruana [11] showed that many classifiers have a characteristic bias to their predicted probabilities: maximum margin methods such as SVMs tend to exhibit a sigmoid distortion in the predictions, whereas naive

Bayes and other methods that make overly strong independence assumptions tend to "push" all probabilities towards 0 and 1. They empirically analyzed two methods for recalibrating the probabilities by passing them through a scaling function: Platt scaling and isotonic regression. They showed that the former method works well when there is relatively little data, whereas the latter works well when there is enough data to prevent overfitting. They also found that some methods (such as SVMs and boosted trees) result in higher-quality calibrated probabilities than other methods (such as naive Bayes and logistic regression). Following on this work, Rüping [12] demonstrated that Platt scaling and isotonic regression are sensitive to outliers, but that preprocessing the training data by "trimming" outliers significantly improves the quality of the recalibration.

In future work, we intend to study whether the use of these calibration techniques does improve CFA's ability to achieve a target confidence value, and, in the case of non-monotonic confidence error, whether calibration improves the overall accuracy of the model.

3 Confidence-Based Feature Acquisition

We assume that there is a non-empty subset of the features that are "free"; that is, every instance in the data set include these features initially, for zero cost. The other features are initially not known for any of the instances in the training or the test set. We also assume that the feature acquisition (FA) cost associated with each feature is known in advance, and that the FA cost for a given feature is the same for all instances. Finally, we assume that the base classifiers produce not only a classification but also a confidence value.

The Confidence-based Feature Acquisition (CFA) approach trains an ensemble of classifiers that use successively larger subsets of the features to classify instances. The training phase is similar to that of boosting ensembles such as AdaBoost [3], in which each new classifier is created to classify instances that are misclassified by the current ensemble. However, AdaBoost was not designed to accommodate feature acquisition; therefore, it trains each new classifier on a reweighted version of the entire data set, which has a fixed dimensionality. In contrast, CFA trains the new classifier, with a higher dimensionality, only on those instances for which the new feature was acquired. Further, boosting permits all classifiers to vote on a new instance, but the classifiers in a CFA ensemble are applied in a cascade fashion, as in Cascade Generalization [4]. A classifier that requires the acquisition of a new feature is applied only if the test instance could not be classified with sufficient confidence by the preceding classifier. Using a cascade approach means that as features are acquired, only the model that uses all of the features acquired to that point is used for predictions. Intuitively, this should increase the prediction quality of the selected model (since it uses more information than earlier models). An alternative approach would be to use a standard voting-based ensemble with all models acquired to date, as discussed in Section 5.

Algorithm 1 CFA-train(D, y, C, F)

1: **Inputs:** training data D, labels y, confidence C, cost-ranked list of features F
2: **Output:** set of trained models $\{M_i\}$
3: $M_0 = \text{train}(D, y)$
4: $(\hat{y}_0, c_0) = M_0(D)$ // Predictions and confidences
5: $D_1 = \text{select-subset}(D, c_0, C)$
6: **if** $D_1 = \{\}$ **then**
7: Return $\{M_0\}$ // Done!
8: **end if**
9: **for** $f = 1$ to $|F|$ **do**
10: Acquire F_f for $d \in D_f$ // F indexed 1 to $|F|$
11: $M_f = \text{train}(D_f, y_f)$
12: $(\hat{y}_f, c_f) = M_f(D_f)$
13: $D_{f+1} = \text{select-subset}(D_f, c_f, C)$
14: **if** $D_{f+1} = \{\}$ **then**
15: Return $\{M_0 \ldots M_{|f|}\}$ // Done!
16: **end if**
17: **end for**
18: Return $\{M_0 \ldots M_{|F|}\}$ // No more features

The CFA approach has some similarities with the "attentional cascade" approach used by Viola and Jones to classify images [15]. Like CFA, the attentional cascade constructs a series of classifiers that are ordered by increasing complexity. However, the cost of feature acquisition is not incorporated into the training process. In addition, instead of training on items that were labeled with low confidence by previous classifiers, the attentional cascade is tailored for positive detection tasks and uses the series of classifiers as a filter. Items that receive a negative label at any point in the cascade are discarded at that point. Only the positively labeled candidates proceed on to the next classifier, and the reliability of the label is not considered. Therefore, instead of selecting features based on acquisition cost and classification confidence, the attentional cascade selects features to achieve the desired detection and false positive rates (as measured on a validation set). We mention it here for its conceptual similarities, but do not include it in our experiments since it was designed to solve a different problem.

3.1 CFA Training. CFA uses two algorithms: CFA-train and CFA-predict. CFA-train (Algorithm 1) takes in a training data set D initially described by a set of zero-cost features, the data set's labels y, the desired training confidence C, and a list of non-zero-cost features F that is ranked by increasing cost. The cheapest-first heuristic intuitively should build a low-cost ensemble, but is greedy and may not result in an optimal feature set. In particular, if some of the inexpensive

Algorithm 2 CFA-predict(d, M, C)

1: **Inputs:** data instance with d all free features, trained cascade $M = \{M_i\}$, confidence threshold C
2: **Output:** prediction $\hat{y}$ with confidence c
3: $(\hat{y}_0, c_0) = M_0(d)$ // Prediction and confidence
4: **if** $c_o \geq C$ **then**
5: Return $\hat{y}_0, c_0$ // Done!
6: **end if**
7: **for** $f = 1$ to $|M|$ **do**
8: Acquire F_f for d
9: $(\hat{y}_f, c_f) = M_f(d)$
10: **if** $c_f \geq C$ **then**
11: Return $\hat{y}_f, c_f$ // Done!
12: **end if**
13: **end for**
14: Return $\hat{y}_{|M|}, c_{|M|}$ // No more models

features are irrelevant to the class, then effort will be wasted at both training and test time. On the other hand, since these irrelevant features are necessarily cheaper than later features, the wasted effort on average should be small relative to the total cost. In Section 5 we discuss some alternative heuristics for feature selection, using background knowledge or partial acquisition.

CFA's output is a series of trained models $\{M_i\}$. The algorithm first constructs the base classifier M_0 using only the zero-cost features. Next, the select-subset subroutine (line 5) selects the instances, D_1, that are used to construct the next model, M_1.

In standard CFA, D_1 will contain all instances with classification confidences c_0 less than the target confidence C. In effect, we are evaluating the predictive quality of M_0 and selecting those instances about which M_0 is insufficiently certain. (An alternative approach to model evaluation and instance selection within the ensemble, Error-based Feature Acquisition, is discussed in Section 3.3.) If no instances are returned, then training is complete and the current ensemble is returned. Otherwise, CFA loops over the features in F (lines 9-17). Each iteration acquires a new feature value for all instances in the selected subset D_f and trains a new model M_f. The model is applied to the training data, and a new subset is selected for the next iteration. The training phase ends when the next subset is empty (line 15) or when all features have been used (line 18).

3.2 CFA Testing. Costs are also minimized, with respect to confidence threshold C, when making predictions for new instances. Algorithm 2 describes CFA-predict, which successively classifies a new instance d using $M_0, M_1, \ldots$ until either the confidence threshold is achieved (line 11) or there are no more models to apply (line 14). Note that the loop in lines 7–13 depends on the number of trained models, not the number of features, since not all features may have been used during training to create models.

3.3 Error-based Feature Acquisition. The confidence values returned by a model M_f are not necessarily reflective of the model's true ability to make accurate predictions. A model may return confidence values that either underestimate or overestimate its ability to make accurate predictions. At test time, confidence values are the only available estimate of the model's performance. However, at training time, the true error is available. Therefore, it seems intuitive that perhaps using this additional information would improve performance of the learned cascade model. The Error-based Feature Acquisition (EFA) variant of CFA was designed to take advantage of this "true confidence" knowledge at training time. Specifically, in step 5 of Algorithm 1, select-subset ignores the confidences and instead returns the subset of instances that are *misclassified* by the current ensemble.

4 Experimental Results

We show results on the Protein data set used by Xing et al. [18] and two additional data sets from the UCI Machine Learning Repository [1]: the Pima Indian Diabetes and Liver Disorders data sets. These data sets are moderately sized, both in terms of number of instances and number of features. We chose to work with such data sets because they are typical of the kind of scenario for which CFA was designed.

The Protein data set contains 20 numeric features and a class label that can take one of six values. The Protein data set does not include feature acquisition costs, so it permits us to evaluate CFA in a case where costs are unknown. In this case, we arbitrarily assign zero cost to the first feature and unit cost to the rest. (Since CFA obtains feature values in order of cost, ties are resolved by choosing the feature that appears earlier in the data set.) This data set contains 116 instances. Because Protein has more features than do the other data sets, there may be more potential for significant cost savings in reducing the number of feature values that are obtained.

The Pima data set is a medical diagnosis data set that describes female patients who are at least 21 years old and are of Pima Indian heritage. There are eight numeric features to describe each patient and a binary class label (which is 1 if the patient has diabetes, and 0 otherwise). The data set also includes acquisition costs in Canadian dollars for each feature. These costs were determined by the Ontario Health Insurance Program fee schedule; they represent the cost of obtaining each feature value individually. The eight features and their associated costs are listed in Table 1. Note that several of the features in the Pima data set have equal (unit) cost. As with the Protein data set, ties for feature selection order are broken by the order in which the features appear in the data set. Also, since there are no zero-cost features

Table 1: Pima Data Set Feature Acquisition Costs

Times pregnant	1.0
Diastolic blood pressure	1.0
Triceps skin fold thickness	1.0
Body mass index	1.0
Diabetes pedigree function	1.0
Age	1.0
Glucose concentration	17.61
2-hour serum insulin	22.78

Table 2: Liver Disorders Feature Acquisition Costs

No. alcoholic drinks/day	0.0
Mean corpuscular volume	7.27
Alkaline phosphate	7.27
Alamine aminotransferase	7.27
Aspartate aminotransferase	7.27
Gamma-glutamyl transpeptidase	9.86

in this domain, we start by acquiring the first of the cheapest features ("Times pregnant") for all training instances in order to learn M_0. The rest of the learning process then proceeds as usual. This data set was chosen because it has real-world costs associated with the features, and contains a reasonably large number of instances (768), reducing the possibility of overfitting in the later models in the ensemble, which are based on a subset of the training data.

The Liver Disorders medical diagnosis data set was provided by BUPA Medical Research, Ltd. This data set contains six numeric features and one binary class label. The features and their associated costs are listed in Table 2. As with the Pima data set, the costs are determined by the Ontario Health Insurance Program fee schedule, and ties for feature selection order are broken by the order in which the features appear in the data set. The data set contains 345 instances.

4.1 Methodology. In all of our CFA experiments, we began with a training set that contained only the zero-cost (free) features and built parsimoniously from there. Training and test feature values were acquired only when needed to achieve the specified confidence threshold.

We measured performance in terms of total feature acquisition cost and classification accuracy on held-out test sets, with 10-fold cross-validation. The code was implemented in Weka [16] and relies on the provided implementations of Naive Bayes, decision trees (J48), and support vector machines (SVMs). For Naive Bayes, unless otherwise indicated, we did not discretize numeric features for CFA. Instead, we used Weka's default procedure, which models the observed values for each feature with a Gaussian distribution.

We compared the empirical performance of CFA and EFA (the error-based variant of CFA presented in Subsection 3.3) to feature acquisition methods that can acquire features only at training (Cost-Sensitive Naive Bayes, CSNB [2]) or only at testing (Active Feature Acquisition, AFA [8]). CSNB can accommodate individual missing values in the training data by computing probabilities over only the existing values. However, if a feature is missing values for *all* training instances, it cannot be used at all by CSNB. Since there is no information about the utility of such a feature in the training data, it will never be acquired during testing. Therefore, CSNB in our setting first acquires all of the training feature values and operates on complete data to build its model. It then selectively acquires values at test time according to the regular CSNB process. Because this causes CSNB to incur a large initial acquisition cost, we also report the test-time costs separately. CSNB requires that numeric features be discretized, so we also provide results for a discretized version of CFA to help determine whether performance differences arise from avoiding discretization or from the CFA cascade approach itself.

AFA was designed to acquire features for batches of m instances at a time. AFA uses the new feature values to re-train a single classifier, rather than generating a new classifier for a cascade as CFA does. We used a batch size of $m = 10$, as was used in the original AFA experiments [8]. This version of AFA requests feature values for misclassified items during training (similar in motivation to EFA, except that all missing values are acquired for each misclassified item, rather than incrementally obtaining one feature value at each iteration). Later versions of AFA refined this selection strategy to instead estimate the utility of acquiring a given missing value, to avoid acquiring values for misclassified items that are not expected to be useful in improving overall performance [13]. We intend to evaluate CFA against this variant of AFA in future work.

Since the true labels are not known on the test data, AFA cannot determine when to acquire feature values when making new predictions and therefore must incur the full acquisition cost for each test item. Similarly to our test-cost comparison with CSNB, we could compare AFA to CFA in terms of training-time costs only. However, with 10-fold cross-validation, we train on 90% of the data and test on only 10% of it, so the total cost exhibits the same trends as the training costs, and we do not show those results separately.

Neither CSNB nor AFA have stopping criteria based on confidence. CSNB trades off acquisition cost against misclassification cost to decide when to stop, and AFA does not specify its stopping criterion (an accuracy threshold is one suggestion). To enable a comparison between CFA and

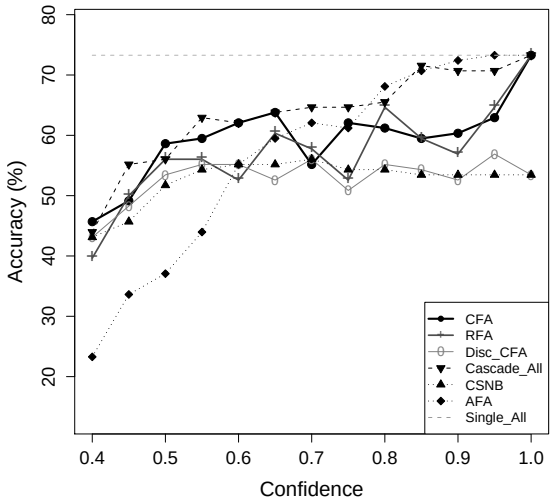


Figure 1: Test accuracy for Protein, as a function of confidence threshold (10-fold cross validation).

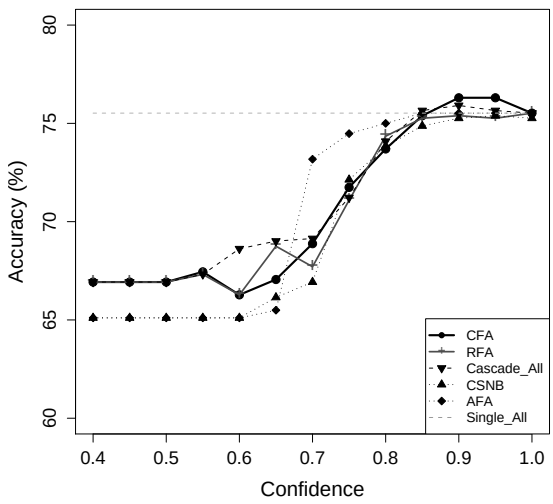


Figure 3: Test accuracy for Pima, as a function of confidence threshold (10-fold cross validation).

these methods, we modified them to use the same confidence threshold, stopping when the posterior probability of the current example's classification meets or exceeds that threshold. Because the number of features acquired under this threshold method monotonically increases with the threshold parameter, it enables a direct comparison of CSNB to CFA as they acquire more features.

We also tested four baseline approaches:

- *RFA (Random Feature Acquisition)*: This randomized baseline allows us to evaluate the benefits of CFA's method for selecting the instances with which to train the next classifier. It is identical to CFA except that select-subset chooses random instances from the previous model's training data, instead of using the confidence threshold. It creates a cascade containing the same number of classifiers, each trained with the same number of instances, as those used by CFA, and at prediction time it uses the same confidence threshold.

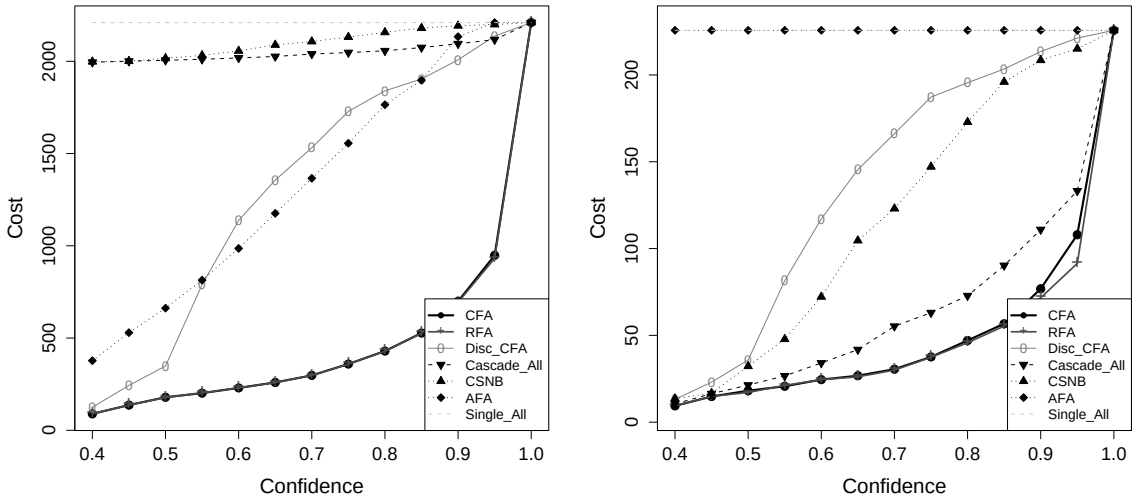
- *Disc_CFA (Discretized CFA)*: This method exists only to provide a direct comparison to CSNB, which must discretize numeric input features. Disc_CFA therefore always uses Naive Bayes as its base classifier.

- *Single_All*: This is a single classifier that uses all of the training instances with all possible features acquired (at both training and test time). It therefore provides an upper limit on the accuracy achievable with a single classifier, at maximal FA cost.

- *Cascade_All*: This baseline differs from CFA in that each successive classifier uses the *entire* training set, rather than a subset of the previous classifier's training set. It provides an upper limit on the accuracy achievable with a cascade ensemble, at maximal FA cost.

4.2 Results and Discussion. Figures 1 through 6 show the experimental results for applying the seven feature acquisition methods described in Section 4.1 (CFA, RFA, CSNB, AFA, Single_All, and Cascade_All) to the Protein, Pima, and Liver Disorders data sets, respectively. (EFA is discussed separately, in Section 4.3.) Each of these experiments used Naive Bayes as the base classifier. In Section 4.4, we give results for CFA using other base classifiers.

Protein. Figure 1 shows how accuracy varied as the confidence threshold parameter was increased. Single_All provides an upper bound on single-classifier accuracy, after acquiring all features, and therefore is unaffected by the confidence threshold. CFA generally outperformed the other methods at low confidence thresholds, except Cascade_All, which incurred the maximum (training) FA cost by acquiring all possible values. This difference in accuracy may be in part because the confidence estimates are not always reliable, leading to errors both in model construction at training time and in model application at test time. At confidence levels greater than 0.7, AFA outperformed CFA (but at higher cost, as we will see), and RFA performed similarly to CFA. CFA consistently outperformed CSNB, but Disc_CFA was almost identical to CSNB, indicating that most of the difference in accuracy was due to the ability to discretize the features. For this data set, the accuracy differences between CFA and RFA were not statistically significant, but the differences between CFA and all other methods *were* statistically significant.



(a) Total cost (training + testing)

(b) Test cost only

Figure 2: Feature acquisition costs (lower y-axis values are better) for Protein (10-fold cross-validation).

This indicates that CFA was effective at determining *how many* instances to acquire new features for—but that *which* instances are acquired was less important (especially since all features had the same cost).

However, the other critical factor in evaluating these methods was the FA cost they incurred (see Figure 2). CFA (and RFA) consistently incurred the lowest cost due to their selective (and parsimonious) acquisition of feature values. In contrast, Single_All acquired all feature values during training and testing and therefore had the highest total cost. Cascade_All and CSNB acquired all values for all items during training but could stop early during testing if the item being classified was sufficiently confident. Therefore, their total costs were also very high, but a separation was seen in the test costs (Figure 2(b)); Cascade_All was significantly cheaper than CSNB in test cost, while achieving higher accuracy. AFA instead acquired all values at test time (maximum cost in Figure 2(b)) but was able to reduce its training costs and ultimately its total cost was below that of CSNB (Figure 2(a)). However, it was still far more expensive than CFA. As above, Disc_CFA and CSNB had similar accuracy results, but as shown in Figure 2(a), Disc_CFA incurred far less total cost than CSNB. In general, the fact that CFA does not require discretization (and can use base classifiers other than Naive Bayes) means that it can be applied to a wider range of data sets more naturally than can CSNB. CFA and RFA were an order of magnitude cheaper (in total cost) than CSNB, except at confidence thresholds greater than 0.9.

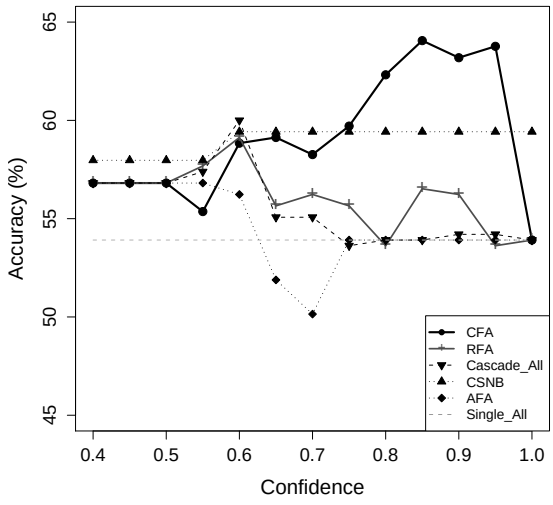
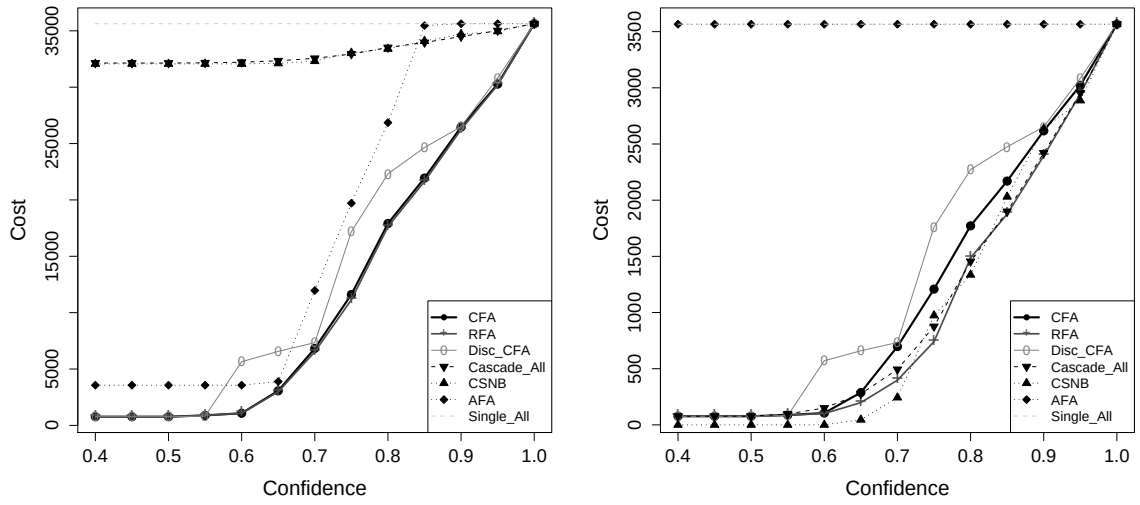


Figure 5: Test accuracy for Liver Disorders, as a function of confidence threshold (10-fold cross validation).

Pima. The results for the Pima domain were similar to those found with Protein, but the accuracy differences between methods were less pronounced (Figure 3). Most of the differences were not statistically significant, except between Cascade_All and Single_All. We again found that CFA and RFA incured significantly less total cost than CSNB or Cascade_All (Figure 4(a)). Here, AFA was competitive with CFA in terms of accuracy and total cost. In contrast, in



(a) Total cost (training + testing)

(b) Test cost only

Figure 4: Feature acquisition costs (lower y-axis values are better) for Pima (10-fold cross-validation).

terms of test cost only (Figure 4(b)), CSNB provided approximately the same degree of accuracy and cost as CFA, while AFA incurred maximal FA cost. CFA therefore provides (as expected) a balance between those two extremes. If the "live time" of the model were expected to be long, then CFA would be a good choice to minimize test cost.

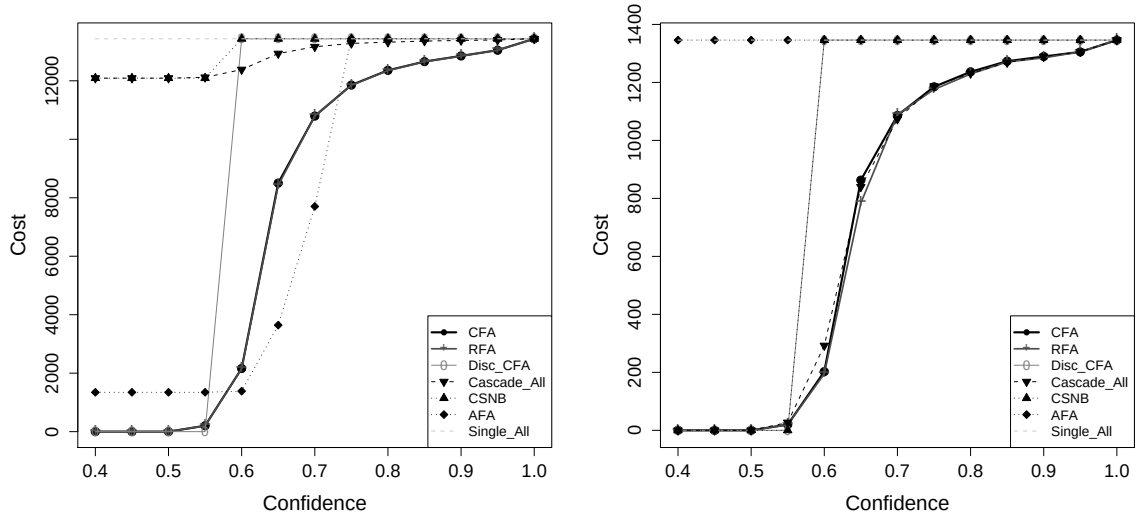
Liver Disorders. The Liver Disorder accuracy results are shown in Figure 5. The first notable difference in these results is that CFA and CSNB both outperformed Single_All at all confidence values except 1.0. This indicates that the data set can benefit from feature selection; it is in fact advantageous *not* to acquire all of the features in order to make decisions. However, making use of this fact requires good selection of the feature values (and for which items) to acquire, and so for the first time we see a clear separation between CFA and RFA. CFA outperformed RFA (and Cascade_All) in the confidence range $[0.80–0.95]$ (the differences were statistically significant, with $p < 0.01$ except at confidence 0.80, where $p = 0.04$). This is a direct measurement of CFA's superior selection of feature values to acquire.

Disc_CFA performed identically to CSNB, while regular (non-discretized) CFA outperformed CSNB in the confidence threshold range $[0.85–0.95]$, with significance test results of $p = 0.05, 0.08, 0.09$ for confidence 0.85, 0.90, and 0.95 respectively. AFA performed poorly in this domain: its accuracy was significantly below any of the CFA variants for confidence thresholds above 0.6.

In terms of FA cost (Figure 6), we found that CFA and RFA again had the lowest costs, except for confidence levels $[0.6–0.7]$, in which AFA had a lower total cost. But as above, its accuracy performance was much lower in this range (the lowest of any method).

4.3 Error-Based Feature Acquisition. EFA, the error-based selection method, performed surprisingly poorly in terms of accuracy. In all domains, EFA had much lower accuracy than any of the other methods except AFA, for any given cost level or confidence threshold. AFA did perform more poorly than EFA for low confidence/cost values, but always outperformed EFA as the confidence threshold (and associated cost). We have omitted the EFA results from the results graphs to improve readability, and because it was always dominated in performance by some other method.

However, because EFA is an intuitively reasonable approach to feature acquisition, we also analyze its behavior in more depth. EFA is much less expensive than the other methods, primarily because it simply chooses fewer instances for which to acquire additional features. The intuition behind EFA is that using all available information at training time should improve performance. However, the result is that the "decision points" in the cascade are created using information that is *not* available at test time. As a result, at test time, the model is unable to make good decisions about which instances to acquire additional feature values for. Furthermore, including only instances that are misclassified in the successive ensembles leads to significant overfitting, so the accuracy actually *decreases* as the user's confidence threshold is



(a) Total cost (training + testing)

(b) Test cost only

Figure 6: Feature acquisition costs (lower y-axis values are better) for Liver Disorders (10-fold cross-validation).

increased.

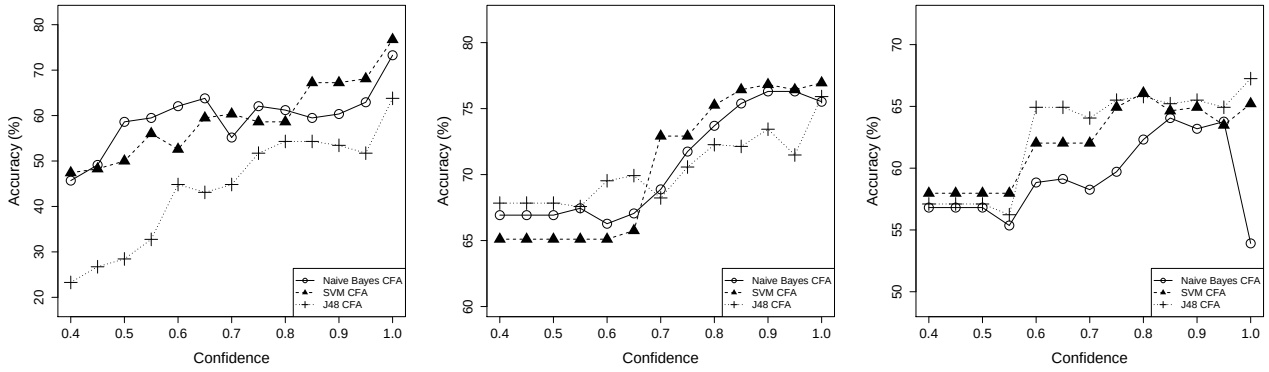
4.4 Alternative Base Classifiers. CFA is not limited to a single type of base classifier, but is instead a general meta-method that can incorporate any base classifier that outputs posterior probabilities. In addition to the Naive Bayes results already presented, we also experimented with using J48 (decision trees) and support vector machines (SVMs) as the base classifiers. Figures 7 and 8 show the accuracy and total FA cost incurred by all three base classifiers when employed by CFA. While all three methods yield good performance, we found that no single base classifier always performed the best.

For Protein and Pima, CFA with Naive Bayes or SVMs achieved about the same accuracy, and both were superior to CFA-J48. For Liver Disorders, CFA-J48 was instead the strongest performer in terms of accuracy; CFA-SVM was second best, and CFA-Naive Bayes was the worst. (Note that in this domain, as shown in Figure 5, even CFA-Naive Bayes provided much better accuracy than the baseline methods.) The costs are similar for all three classifiers, although SVMs are consistently slightly worse (more expensive) than the other methods.

The base classifiers work in different hypothesis spaces, produce different posterior estimates, and therefore acquire features for different items (and different numbers of items), given the same confidence threshold. The difference in performance indicates that, for real applications, testing a variety of base classifiers is important. This result additionally demonstrates that the ability of CFA to incorporate different base classifiers is a useful property of the method.

5 Conclusions and Future Work

We have described Confidence-based Feature Acquisition (CFA), the first method that can selectively acquire missing feature values during training and testing. It selects values to acquire so as to meet a user-specified level of performance (confidence), minimizing the cost needed to reach that goal. Our study of CFA on different data sets, in different cost settings, shows that when feature acquisition is possible during both training and testing, it is advantageous to do so. The Active Feature Acquisition (AFA) method [8] acquires values only when training, while the Cost-Sensitive Naive Bayes (CSNB) approach [2] acquires values only at test time. AFA and CSNB in a sense provide "ablated" versions of CFA, since they each acquire values either only during training (AFA) or only during testing (CSNB), while CFA acquires values during both. Consistently, we found that CFA had the lowest overall cost, while achieving accuracy comparable to (or in excess of) the other methods. In some domains, CFA outperformed RFA (random selection of features to acquire), but in other domains, the two methods performed about the same. In all domains, however, CFA performed significantly better than EFA (error-based selection). We conclude that the cascade model, and the confidence values that CFA relies upon to apply the cascade, provide a good decision criterion for acquiring additional information. However, in some domains (namely, those domains where RFA performs well),



(a) Protein (b) Pima (c) Liver Disorders

Figure 7: Test accuracy results for CFA using different base classifiers.

the cascade performance is not sensitive to *which* instances are acquired. As discussed in Section 2, in the case of inaccurate confidence values, recalibration techniques might help to improve CFA's performance at both training and test time [11]. In comparison to baselines that acquire all values, CFA achieved the same accuracy or better, at a fraction of the cost.

CFA is independent of the base classifier used. We experimented with using Naive Bayes, J48 decision trees, and support vector machines as the base classifiers for CFA. We did not find that any single base classifier was always the best (nor did we expect to, given the No Free Lunch theorem [17]), but the results provide evidence that CFA is not restricted to any single base classifier's capabilities.

CFA as described here uses a cascade classifier, in which each model is trained with more features, and predictions are made only with the last applicable classifier. An alternative would be to use a weighted ensemble, in which all applicable models are allowed to "vote," with weights that might be determined by their confidence, by the amount of training data, or in some other way. We have not yet explored these variations, but plan to experiment with alternative ensembles in future work.

CFA currently adopts a greedy approach to feature selection, always choosing the cheapest unused feature to add to the next successive model to be trained. Batch effects, in which acquiring multiple features at once is cheaper than incrementally acquiring them, occur in many domains and could be exploited to further reduce costs, as in Sequential Batch Testing [14]. Furthermore, there may be ways to identify the most informative next feature based on the distribution of instances in the selected subset. One possibility would be to acquire all of the features for a randomly selected subset of the training instances, and use this "complete data" subset to perform feature selection that would be used to guide the feature acquisition process. Background information about the relevance of different features could be used in a similar way. We are also interested in interactive applications, in which the system could train a model and provide feedback to the user when the current set of features is inadequate, in the form of a request for additional features. In scientific investigations, this iterative (active) process could be of great assistance in better understanding the nature of a new domain.

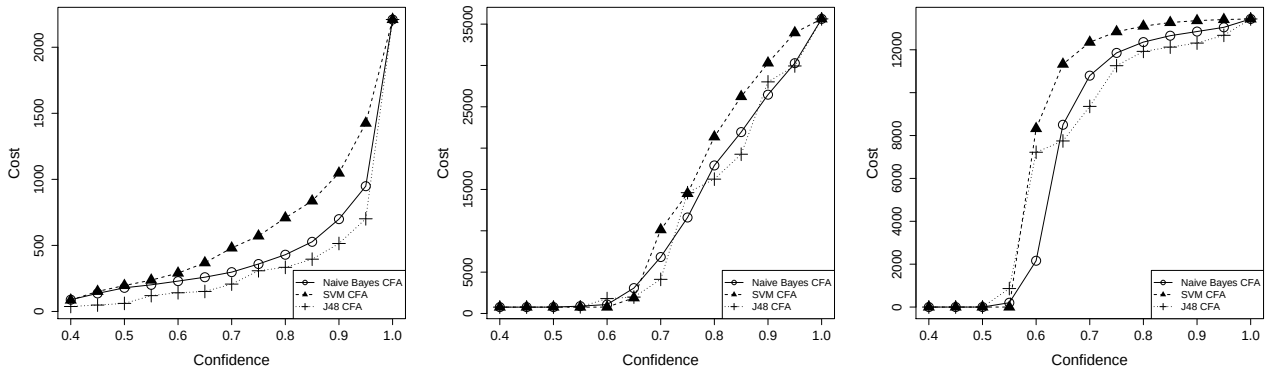
Finally, we are also exploring ways to automatically learn optimal per-model confidence thresholds. Although the user specifies the desired target confidence, individual models in the cascade ensemble may benefit from using different thresholds, since each model is solving a slightly different sub-problem. Initial tests have shown that the number of features used in a classifier can result in increased posterior probabilities for instances that are correctly classified, as well as instances that are misclassified. This suggests that a dynamic threshold per model may behave more consistently than a single threshold for all models.

Acknowledgments

This work was supported by NSF grant #ITR-0325329 and was partly carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] A. Asuncion and D.J. Newman. UCI machine learning



(a) Protein (b) Pima (c) Liver Disorders

Figure 8: Total feature acquisition cost results for CFA using different base classifiers.

repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2007.

[2] Xiaoyong Chai, Lin Deng, Qiang Yang, and Charles X. Ling. Test-cost sensitive Naive Bayes classification. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 51–58, 2004.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[4] J. Gama and P. Brazdil. Cascade generalization. *Machine Learning*, 41:315–343, 2000.

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

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

[7] Charles X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 544–551. ACM Press, 2004.

[8] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 483–486, 2004.

[9] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 745–748, 2005.

[10] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Proceedings of the First International Workshop on Utility-Based Data Mining*, pages 10–16, 2005.

[11] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632. ACM Press, 2005.

[12] Stefan Ruping. Robust probabilistic calibration. In *Proceedings of ECML-2006*, pages 743–750. Springer Berlin, 2006.

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

[14] Victor S. Sheng and Charles X. Ling. Feature value acquisition in testing: A sequential batch test algorithm. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 809–816. ACM Press, 2006.

[15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[16] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques (2/e)*. Morgan Kaufmann, 2005.

[17] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[18] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512, 2003.

[19] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proceedings of the Second IEEE International Conference on Data Mining*, pages 562–569, 2002.