Partial Example Acquisition in Cost-Sensitive Learning

Victor S. Sheng, Charles X. Ling
Department of Computer Science
The University of Western Ontario
London, Ontario N6A 5B7, Canada
{ssheng, cling}@csd.uwo.ca

ABSTRACT

It is often expensive to acquire data in real-world data mining applications. Most previous data mining and machine learning research, however, assumes that a fixed set of training examples is given. In this paper, we propose an online cost-sensitive framework that allows a learner to dynamically acquire examples as it learns, and to decide the ideal number of examples needed to minimize the total cost. We also propose a new strategy for Partial Example Acquisition (*PAS*), in which the learner can acquire examples with a subset of attribute values to reduce the data acquisition cost. Experiments on UCI datasets show that the new *PAS* strategy is an effective method in reducing the total cost for data acquisition.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – *induction*.

General Terms

Algorithms, Measurement, Performance, Economics.

Keywords

Data acquisition, induction, cost-sensitive learning, data mining, machine learning, active learning, active cost-sensitive learning, interactive and online data mining.

1. INTRODUCTION

It is often expensive to acquire data in real-world data mining and machine learning applications. Most of previous works on data mining and machine learning applications, however, assume that a fixed set of training examples is given to build learning models. However, in reality, there may not be enough data to begin with. Data acquisition is the first, and one of the most important steps in machine learning and data mining. It is well recognized that data acquisition is time consuming and costly. Thus, it is important for the learner to be able to acquire a proper set of training examples during learning.

As data acquisition incurs cost, it is natural to study it as a part of the cost-sensitive learning process. We assume that acquiring an additional example incurs a certain cost. This cost is the sum of attribute costs (given) if their values are needed, plus the label cost (also given). We can further assume that the learner can decide if an example to be acquired is “complete” or “partial”. A complete example contains values of all attributes, thus it costs more to acquire than acquiring a partial example which contains values of a subset of attributes. The learner must decide what attribute values are needed in the partial example acquisition. Furthermore, partial example acquisition can take several steps to acquire attribute values. It allows acquiring other unknown attribute values for the acquired partial examples.

For example, assume that a problem domain has only three attributes *(a_1, a_2, a_3)* with a class label *c*, and that the costs of acquiring the values of these attributes and the class label are C_1, C_2, C_3, and C_L respectively. If the learner decides that a complete example is needed, a complete example (such as $a_1=1$, $a_2=0$, $a_3=1$, *c=true*) is drawn randomly according to a certain distribution governing the generation of training examples, with the cost of $C_1+C_2+C_3+C_L$. If the learner decides to acquire an example with only the first attribute value and the class label first, and later to acquire the second attribute value, the cost is reduced to be $C_1+C_2+C_L$. This is partial example acquisition. Thus, the cost C_3 is saved.

On one hand, the learner should try to avoid acquiring more examples (or more attribute values) than necessary. On the other hand, the learned model is usually more accurate, and thus has lower misclassification cost, when it is built with more examples and examples with fewer missing values. The goal of the learner is thus to minimize the total cost of data acquisition and misclassification on test examples.

We propose and study a novel partial example acquisition strategy (called *PAS*) in this paper. The rest of paper is organized as follows. We first review the related work in Section 2. In Section 3, we describe our data acquisition framework and the evaluation method. Section 4 discusses the complete example acquisition strategy (called *COM*) and partial example acquisition strategy (called *PAS*). We conduct experiments to compare *COM* and *PAS*, and analyze the results in Section 5. Finally we conclude the work in Section 6.

2. RELATED WORK

Cost-sensitive learning is an active research topic in recent years. Turney [20] gives an excellent survey on a variety of costs that may be considered in learning, such as misclassification costs, data acquisition cost (including example costs and attribute costs), active learning costs, computation cost, human-computer interaction cost, and so on. Most previous works on cost-sensitive learning only consider minimizing misclassification costs (e.g., [5, 20, 6, 18]). A few works do attempt to minimize the sum of

KDD ’07, August 12–15, 2007, San Jose, California, USA.

misclassification costs and attribute costs [21, 4, 24, 11]. However, these works do not consider data acquisition cost. That is, they assume that a fixed set of training examples is given, and the leaner cannot acquire additional information during learning.

Some previous works study data acquisition cost, such as [10, 9, 23, 13, 14]. Among them, some [10, 9] study how to acquire attribute values to build an optimal classifier with a certain budget. The others [23, 13, 14] study how to achieve a desired accuracy of a classifier by acquiring missing values in training examples with minimum cost. However, they do not minimize the total cost, which combined the acquisition cost and the misclassification cost. In our work, the cost of attribute acquisition and misclassification is unified as the same currency of cost, and the task is to minimize the total cost. We can also put a "budget" on the total cost, thus our proposed algorithms can be easily adapted to meet such total budget.

Active learning can be viewed as a specific form of data acquisition. The most popular type of active learning is called "pool-based" active learning. Many works have been published in recent years on pool-based active learning, including, for instance, [19, 22, 16, 17, 12, 1] (See [1] for a good review of active learning approaches). All these works assume that a pool of unlabeled examples is given, and the learner can choose which ones to acquire their labels during learning. In our work, we do not have this assumption. We do not assume that there is a pool of examples given. In addition, almost all of these works on active learning are evaluated by accuracy (or learning curves with accuracy) and "deficiency" of active learning [1], as well as the number of examples acquired. However, we propose a new cost-sensitive acquisition framework (based on the actual cost reduction) to integrate the acquisition cost and the misclassification cost. In addition, our learning framework can also acquire partial examples.

Our cost-sensitive acquisition framework and two data acquisition strategies (to be presented in Sections 3 and 4) utilize the cost-sensitive decision tree (*CSDT* in short) as a base learning algorithm; thus we briefly review it here. *CSDT* [11] is similar to C4.5 [15], but it uses the total cost of attributes and misclassifications, instead of entropy, as the attribute split criterion. At each step, *CSDT* always chooses an attribute from the available attribute set with the maximum cost reduction on training data, similar to maximum entropy reduction, to build decision trees. Cost reduction $(E–E_A–T_C)$ is the difference between the expected cost E before splitting and the sum of the expected cost E_A after splitting with attribute A and the total attribute costs T_C. The attribute with the largest cost reduction, if it is greater than 0, is chosen to split the training data (otherwise it is not worth to build the branch further, and a leaf is formed). The same procedure is applied recursively to build subtrees. However, *CSDT* and C4.5 assume that all examples are available to build the tree. In our work, we assume that it is possible for the learner to acquire more examples in the tree building.

3. COST-SENSITIVE DATA ACQUISITION FRAMEWORK

In this section we will describe our cost-sensitive data acquisition framework and algorithms.

3.1 The Framework

The framework of our cost-sensitive data acquisition algorithm is quite simple. It is like a "wrapper" and can be applied to any cost-sensitive learning algorithm, such as ICET in [21], MetaCost in [5], csNB in [4], and *CSDT* in [11], as the base learner. At a high level, the data acquisition is a simple on-line process. That is, it acquires examples at cost gradually while monitoring an evaluation criterion until it is met. The evaluation computes the sum of the acquisition cost and misclassification cost of test examples to see when it reaches the minimum (details are presented in next subsection). Examples given to the learner are drawn randomly according to a certain fixed distribution. As the effect of one extra example is often too small, the learner always acquires units of examples in each acquisition. In this paper we set a unit to have 10 examples.

More specifically, each time, the learner acquires one unit or a number of units of examples, and includes them in the training set. Then a new cost-sensitive learning model is built by the base learner from the expanded training set, and is evaluated to see if the total cost of the example acquisition and misclassification of test examples is reduced. If it is, then the process repeats; if not, the learner stops acquiring more units of examples, and the current learned model is produced.

We re-implement and use cost-sensitive decision tree (called *CSDT*) as the base learner in the rest of the paper. This is because *CSDT* (reviewed in Section 2) is itself cost sensitive, and it is also very fast in building many decision trees needed in the data acquisition process. The pseudo-code of the cost-sensitive data acquisition algorithm is presented below. We assume that the learner is given an initial set T of examples (T can be empty), and that each time some units of training examples are acquired and added into the training set.

Algorithm

```
Input: an initial training set $T$, and a stopping criterion.
While stopping criterion is not met
    a. Acquire training examples with certain attribute values
       available  at a cost, adding them into $T$
    b. Call CSDT to build a cost-sensitive tree on $T$
    c. Evaluating the tree
Output: a cost-sensitive decision tree
```

Figure 1. The cost-sensitive data acquisition algorithm.

We will first discuss the evaluation method in the next subsection.

3.2 Evaluation Methods

One might think that it would be easy to calculate the total cost of misclassification and example acquisition – just sum them up. There are quite a few intriguing issues to be resolved, as we discuss as follows.

To evaluate the tree (or any learned model) built in the acquisition procedure for future test performance, we must use a part of available training examples as a held-out set. These examples are not used in building the models. However, from the cost-sensitive point of view, holding out some examples for testing excludes them from building the model, making some acquiring examples wasted. To reduce the waste of acquired training examples used for testing, we use leave-one-out cross-validation (*LOO* in short) to evaluate the learned model, so only one example is "wasted". As the decision tree learning algorithm is quite efficient, this would not be a major problem in most real-world applications. Thus all available examples except one are used to train the decision tree to estimate the average misclassification cost of a test example in *LOO*.

The following procedure describes details on how to estimate the misclassification cost of one test example (in *LOO*). For binary classification (used in this paper), we use the following notations: *TP* and *FP* are the cost of true and false positive, *TN* and *FN* are the cost of true and false negative, *tp* and *fn* are the number of true positive and false negative examples, and *tn* and *fp* are the number of true negative and false positive examples. For a leaf in the cost-sensitive decision tree, let C_P *(= tp×TP+fp×FP)* be the total misclassification cost of being a positive leaf, and C_N *(=tn×TN+fn×FN)* be the total misclassification cost of being a negative leaf. Then the probability of a leaf being positive is estimated by the relative cost of C_P and C_N; the smaller the cost, the larger the probability (as minimum cost is sought). Thus, the probability of the leaf being positive is thus: $1-\frac{C_P}{C_P+C_N}=\frac{C_N}{C_P+C_N}$. Similarly, the probability of a leaf being a negative is $\frac{C_P}{C_P+C_N}$. However, these probabilities are not used directly in estimating misclassification costs, because the number of training examples in leaves is usually very small, especially at the beginning of example acquisition. To reduce the effect of extreme probability estimations, we apply the Laplace correction [8, 3] to smooth probability estimates in leaves. We modify the original Laplace based on accuracy for estimation with misclassification cost. The original Laplace correction for accuracy can be expressed as $\frac{n_C+1}{N+n}$, where n_C is the number of examples which belong to class *C, N* is the number of training examples, and *n* is the number of classes. As we consider misclassification costs now, the probability of a leaf being positive is $\frac{C_N+\lambda}{C_N+C_P+k}$, where $\lambda=|FP-FN|$ and $k=FP+FN$. Similarly, the probability of a leaf being negative is $\frac{C_P+\lambda}{C_P+C_N+k}$. Thus, the expected misclassification cost of a true negative example is $\frac{C_N+\lambda}{C_P+C_N+k}\times FP$, and the expected misclassification cost of a true positive example is $\frac{C_P+\lambda}{C_P+C_N+k}\times FN$.

The next issue is how to integrate the misclassification cost of *one* test example (obtained above) with the cost of training examples acquired. A simple sum of the two, as in [12], would not be reasonable, as it depends on *how many* future test examples (or how often) the model will be used to predict. Clearly, if the model built will be seldom used (only once or twice), we can reduce the total cost through building a rough model with acquiring only a few examples. On the other hand, if the model built will be used very frequently (for instance, millions of times), it would be worthwhile to acquire more examples to build a highly accurate model with very low misclassification cost. As we may not know the number of future test examples during the model building process, we introduce a variable *t* to represent the number of future test examples. As the cost of acquiring training examples is shared by all the test examples, each test example has to burden the cost $\frac{1}{t}\sum_{i=1}^{Tr}E_{C_i}$, where E_{Ci} is the cost of acquiring the *i*-th example, and *Tr* is the number of acquiring examples. The total cost, which is the sum of the share of the cost of acquiring training examples and the misclassification cost, is thus:

$$\text{Total cost} = \frac{1}{t}\sum_{i=1}^{Tr}E_{C_i} + MisCost, \qquad (1)$$

where *MisCost* is the average misclassification cost of one example. Section 5.5 studies the effect of different *t* values for the cost-sensitive data acquisition algorithm.

The next issue is how to use the total cost as a stopping criterion for the incremental cost-sensitive active learners. Ideally, we can expect the total cost to decrease initially, and it would then reach a minimum before it goes up. This is because the gain in reduced misclassification cost would be deducted as more and more examples or labels are acquired (with a constant cost for each complete example, or with a variant cost for each partial example, see Section 4). Thus, we can obtain a learning curve in terms of the total cost and the number of examples required (see Figure 3). If the curve is smooth, then indeed the learning algorithm has found the optimal number of training examples needed. However, as we will see in Section 5, the curve of the total cost may not be smooth, and the local minimum may not be the global one. Thus it is necessary to "look ahead" and acquire a few more (units of) examples to ensure that the local minimum is global. The extra examples in *LOO* and look-ahead will be extra (or wasted) on top of the minimum total cost found. We will describe the look-ahead strategy obtained empirically in Section 5.1.

4. DATA ACQUISITION

There are different strategies to acquire training data. In this section, we first describe the simplest data acquisition strategy, that is, the complete data acquisition strategy (*COM*). *COM* acquires training examples with all attribute values and their labels. It will be used as the baseline to compare with the *PAS* (partial data acquisition strategy) in Section 4.2.

4.1 Complete Attribute Strategy (*COM*)

The complete attribute strategy, which acquires additional training examples with complete attribute values, is quite simple. In the pseudo-code presented in Figure 1, the step 1(a) simply stipulates that all attributes should be acquired during example acquisition. That is, this strategy does not consider omitting some attribute values for reduced cost; in the next section, we consider the partial attribute strategy which does.

4.2 Partial Attribute Strategy (*PAS*)

If the learner is able to acquire partial examples with a subset of attribute values at reduced cost, it may help to reduce the total cost in the end.

Intuitively, the attributes which do not appear in the final decision tree would not help in the tree construction. A simple idea is to simply acquire partial examples with attributes appearing in the cost-sensitive decision tree. For the example described in Section 1, if only attributes a_1 and a_2 appear in the final decision tree, we only need acquire the values of these two attributes with the class label for each example.

However, we do not know the final cost-sensitive decision tree at the beginning; thus, we do not know which attributes will appear in the final cost-sensitive decision tree. *PAS* "guesses" the attributes in the final decision tree based on the current tree, by "extending" the current tree with more potentially useful attributes, and acquire examples with those attributes. Hopefully the attributes in the current tree plus the extended attributes can "cover" attributes in the final decision tree.

More specifically, *PAS* consists of the following four major steps (as the Steps 1(a) and 1(b) in the framework in Figure 1). Details of these steps will be explained later.

1. Build a cost-sensitive decision tree based on the current training set.
2. Acquire a unit of *m* partial examples based on the current tree.
3. For each leaf node, determine potentially useful attributes (that would further extend the tree).
4. For all examples in each leaf, acquire the values of those extended attributes.

We will provide details of each step below.

In the first step, it is possible that when the tree is re-constructed, a different attribute is chosen as the root (or internal node), and some examples may not have values on this attribute. In this case, *PAS* will acquire unknown values on the path of classifying them to a leaf in the tree built. Though this may be cumbersome to acquire attribute values of some previously acquired examples in real-world applications, it does save the overall acquisition cost.

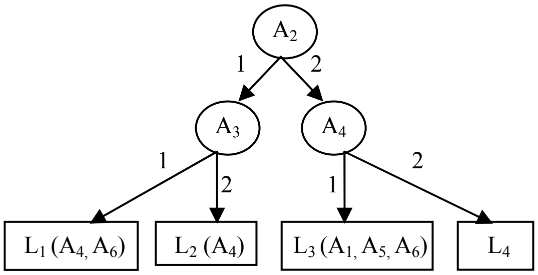

Figure 2. **An example of an intermediate cost-sensitive decision tree.**

In the second step, to save the acquisition cost, *PAS* uses a conservative "sequential strategy", explained using the example shown in Figure 2. Supposed there is an intermediate tree built by *CSDT* shown in Figure 2, *PAS* utilizes the intermediate tree structure to guide the sequence of attributes of the example to be acquired in the following way: *PAS* first acquires the value of the attribute in the root of the tree. That is, the value of attribute A_2 in Figure 2. According to the value of attribute A_2, *PAS* further acquires the value of attribute A_3 or A_4. For example, if the attribute A_2 has a value of 1, *PAS* will acquire the value of attribute A_3. This way, *PAS* acquires all the values of the attributes on the path one by one until it classifies this example into a leaf.

In the third step, for each leaf, *PAS* determines the potentially useful attributes, which are possibly used to extend the current intermediate cost-sensitive decision tree. Naturally, all the remaining attributes that do not appear on the path of the leaf are candidates. For example, the remaining attributes of the left-most leaf L_1 in Figure 3 are attributes A_1, A_4, A_5 and A_6, assuming the complete attribute set is (A_1, A_2, A_3, A_4, A_5, and A_6). For each leaf, *PAS* computes the cost reduction of its remaining attributes and chooses the attributes whose cost reduction is not less than a lower bound value L_B as the potentially useful attributes. The lower bound value corresponds to the expected total misclassification cost E before splitting, the expected total misclassification cost E_A after splitting on attribute A and the total cost T_C of acquiring the values of attribute A for all examples in the leaf. It is defined as

$$L_B = -f\times(E+E_A+T_C), \qquad (2)$$

where f is a variation coefficient with a small given value, controlling the size of potential attributes of the leaves. The larger the value of f, the more attributes are chosen as potentially useful attributes; however, the more attribute values acquired could be wasted. In this paper, we let f be 0.05 unless otherwise stated. We investigate the effect of the value of f in Section 5.4. In all, *PAS* chooses an attribute as a potentially useful attribute that has the property: $(1+f)\times E-(1-f)\times(E_A+T_C)\geq 0$. *PAS* uses the same formula used in building *CSDT* to compute the expected total misclassification costs before and after splitting for each attribute not on the path to the leaf.

PAS can be regarded as a beam-search strategy with an adaptive beam size. The beam size is automatically determined by *PAS* according to the examples in each leaf. *PAS* localizes the attribute selection in each leaf. It is possible that *PAS* chooses zero or several attributes as the potentially useful attributes for a leaf. As the example in Figure 2, A_4 and A_6 are the 2 potentially useful attributes from the left-most leaf L_1. However, there are no extended attributes for the right-most leaf L_4.

In the fourth step, *PAS* further acquires values of those extended attributes for all examples in each leaf. For example, *PAS* further acquires the values of the two extended attributes (A_4, A_6) for all examples in leaf L_1. Note that some unknown attribute values of some previously acquired examples may be acquired again in this step. Again, though this may be cumbersome in real-world applications, it does save the overall acquisition cost.

In sum, *PAS* uses the heuristics of guessing the attributes in the final tree using the intermediate trees, and only acquires the values of these attributes (part of attributes) for each example. Thus, the acquisition cost can be saved significantly, comparing with *COM*. If the misclassification cost does not increase too much, the total cost of *PAS* would be less than that of *COM*. We investigate this in the next section (Section 5).

5. EXPERIMENTS

We conduct experiments with *PAS* and *COM* on 10 real-world datasets downloaded from the UCI Machine Learning Repository [2]. These datasets are chosen because they have at least some discrete attributes, binary class, and a good number of examples.

The numerical attributes in datasets are discretized first using minimal entropy method [7] as *CSDT* can currently only deal with discrete attributes. We also try to choose datasets with very different number of attributes. The number of attributes and other features of these datasets are listed in Table 1. If misclassification costs of these datasets are not available, we assign them with values in a reasonable range, following [5, 11, 24, 4]. This is fair and reasonable as all experimental comparisons are conducted with the same cost assignments. We assign random values from 0 to 100 to test costs. We assign the labeling cost to be 100, and the misclassification costs as *FP/FN = 1000/3000* (*TP=TN=0*). We also assume that the learned model will be tested on 1,000 test examples (i.e., *t=1,000*) for now.

Table 1. Features of the 10 Datasets used in the experiments.

	No. of Attributes	No. of Examples	Class dist. (N/P)
Ecoli	6	332	230/102
Breast	9	683	444/239
Heart	8	161	98/163
Thyroid	24	2000	1762/238
Australia	15	653	296/357
Tic-tac-toe	9	958	332/626
Mushroom	21	8124	4208/3916
Kr-vs-kp	36	3196	1527/1669
Voting	16	232	108/124
Cars	6	446	328/118

To compare the performance of *COM* and *PAS*, we set the initial training set with 30 examples. The original datasets listed in the Table 1 are the natural pool of examples to be acquired. *COM* and *PAS* randomly acquire units of extra examples from these original datasets to make their own training sets. For *COM*, all attribute values of each fresh example are available. However, for *PAS*, the attribute values of each fresh example will be given only when it is requested. In addition, we repeat all our experiments 10 times (randomly selecting the initial training set), and only average values are displayed in following figures and tables.

5.1 Experiments with *COM*

We first conduct experiments for the complete attribute strategy *COM* (Section 4.1). The experiment results are shown in Figure 3, which displays the learning curve for a typical dataset (Mushroom). Again, the unit *m* is chosen to be 10 in all experiments. The horizon axis is the number of acquired examples, and the vertical axis is the total cost, which is sum of the share of the cost of acquiring training examples and the average misclassification cost of *t* future test examples, assuming *t* is 1,000 here (other values of *t* have been tried and resulted in similar curves).

From Figure 3, we can see clearly that the *LOO* curve produced by *COM* has a desirable trend: it goes down at the beginning (after a few units of training examples are acquired), and then it goes up after a certain point, forming a global minimum (the minimal total cost). All datasets produce similar curves as shown in Figure 3.

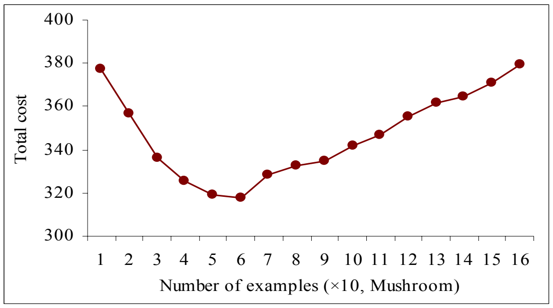

Figure 3. **The ideal total cost curves with *COM.* The vertical axis is the total cost, and the horizontal axis is the number of example units acquired. This is actually the experimental results on the dataset Mushroom.**

However, the *LOO* curve is not always smooth, thus look-ahead is needed to find the global minimum. We have calculated the minimum length of look-ahead needed for each dataset in Table 2, and have found that the number of look-ahead can be set to be 2. This is relatively small for the total cost wasted in searching for the global minima. Note that the total costs shown in the paper have included the 2 units of look-ahead.

Table 2. The minimum length of look-ahead needed for *COM* to find the true global minimum in the *LOO* curve for each dataset.

	Ecoli	Breast	Heart	Thyroid	Australia	Tic-tac-toe	Mushroom	Kr-vs-kp	Voting	Cars
LOO	2	3	1	1	1	2	1	2	1	2

5.2 Comparison between *COM* and *PAS*

We first conduct experiments to investigate the performance of *PAS*, and then compare it with *COM*. Again, in these experiments, the variation coefficient *f* in *PAS* is set as 0.05. In addition, both *PAS* and *COM* acquire 30 complete examples as the initial training data. The cost of this initialized acquisition is counted into the total acquisition cost. The experimental results are shown in Figure 4, which displays the learning curves for each dataset. For each curve, the horizon axis is the number of acquisition times, and the vertical axis is the average total cost. The smaller the average total cost is, the better. Note that the segments of the curves after the global minima are computed and plotted for demonstrating the increment of the total cost only. In reality, *COM* and *PAS* would stop soon after the minimum (due to look-ahead).

From Figure 4, we can draw several interesting conclusions. First, all curves of *PAS* and *COM* have a similar trend: they go down at the beginning after a few units of training examples are acquired, and then they go up, forming global minima. That is, the performance of *PAS* and *COM* become better at the beginning of acquisition; and they become worse after more acquisitions occur.

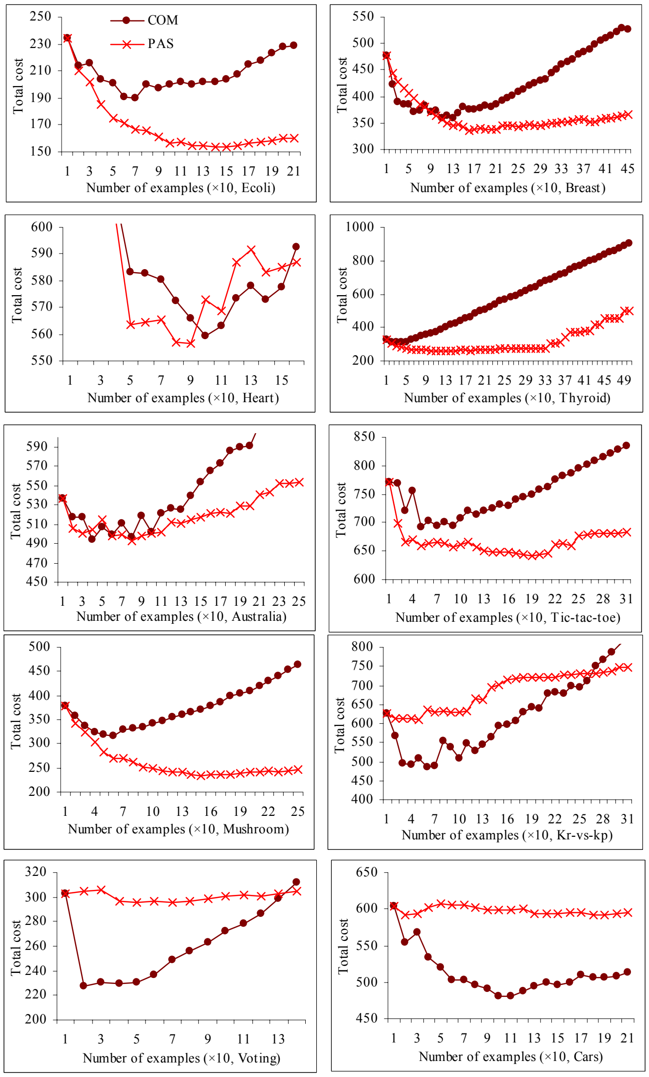

***Figure 4.* Comparing *PAS* with *COM* on the 10 datasets. The smaller the average total cost is, the better.**

Second, *PAS* outperforms *COM* significantly on seven out of the ten datasets (i.e., the first seven datasets in Figure 4). This is because *PAS* indeed reduces the acquisition cost significantly, comparing with *COM*. The most interesting thing is the cost-sensitive decision tree built on the partial examples acquired by *PAS* performs similar to or better than the tree built on the complete examples acquired by *COM* on the seven datasets. We further explain this by analyzing the acquisition cost and misclassification cost separately in Section 5.3.

Third, *PAS* often achieves its global minimum through acquiring more examples. As *PAS* spends less acquisition cost for acquiring partial examples than *COM* does for acquiring complete examples in each acquisition, this allows *PAS* to acquire more partial examples with less total cost.

5.3 Acquisition Cost and Misclassification Cost of *PAS* and *COM*

In the subsection above (Section 5.2), the experimental results show that *PAS* performs better than *COM* on seven out of the ten datasets in terms of the average total cost. In this subsection, we further analyze the details of why *PAS* performs better or worse than *COM*. As the total cost is the sum of the acquisition cost and misclassification cost, we investigate their performances in terms of the acquisition cost and the misclassification cost respectively, to see which cost is the most important contributor. The environment parameters are the same as the subsection above. As the trend of the performance of *PAS* and *COM* is similar in some datasets, particularly the curves of the acquisition cost for each dataset, we only display the curves of the acquisition cost and the misclassification cost of *PAS* and *COM* on three typical datasets (i.e., Ecoli, Mushroom, and Voting). The three typical datasets represent three cases: the misclassification cost of *PAS* is lower than that of *COM*; the classification cost of *PAS* is slightly higher than that of *COM*; the misclassification cost of *PAS* is much higher than that of *COM*. The experimental results of the three datasets are shown in Figure 5. Similar to Figure 4, for each curve, the horizon axis is the number of acquisition times, and the vertical axis is the average acquisition cost and misclassification cost respectively. Again, the smaller the average total cost is, the better.

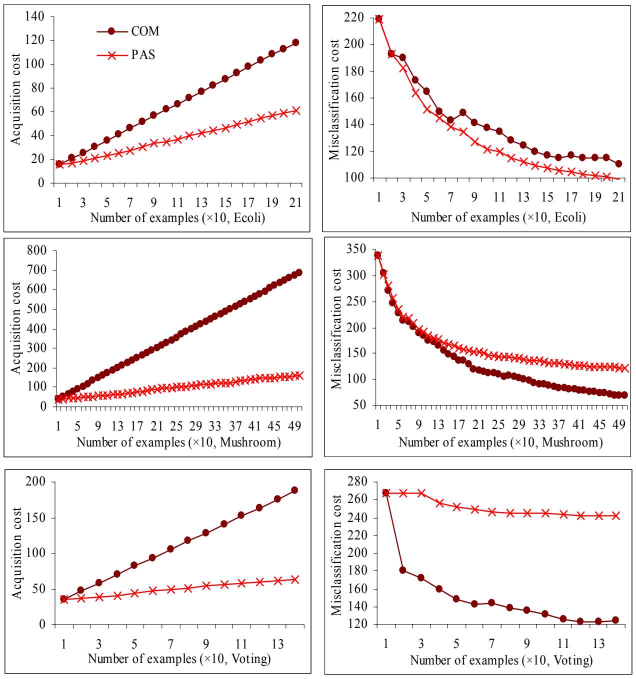

***Figure 5.* Acquisition cost and misclassification cost of three typical datasets (i.e., Ecoli, Mushroom, and Voting). The smaller the average total cost is, the better.**

From Figure 5, we can see that the acquisition cost increases as more information (including partial examples and the extra attribute values for existing examples) is acquired. This is expected. However, the rate of increment of acquisition cost is different between *COM* and *PAS*. We can see that the rate of increment of *COM* is much higher than that of *PAS*.

For the misclassification cost, we can see that the misclassification cost decreases after more information is acquired. However, the extent of the decrement of the misclassification curves under *PAS* and *COM* is also different. For example, the misclassification cost curve of *PAS* goes down faster than that of *COM* on the dataset Ecoli. However, the misclassification cost curve of *PAS* goes down slower than that of *COM* on the dataset Mushroom, and much slower on the dataset Voting. For the first two cases, *PAS* performs better than *COM* in terms of the total cost, since the acquisition cost is the major factor that determines which is better between *PAS* and *COM*. This is the reason why *PAS* performs better than *COM* on the first seven datasets. However, for the last case, the misclassification cost plays the important role in the total cost. This is the reason why *PAS* performs worse than *COM* on the other three datasets (i.e., Kr-vs-kp, Voting, and Cars).

5.4 The Effect of the Variation Coefficient in *PAS*

In the subsections above, we do all experiments with the variation coefficient value *f=0.05* (see equation 2) in *PAS*. Again, the variation coefficient is a small value. In this subsection, we investigate the performance of *PAS* with different variation coefficients, such as, *f=0.01, f=0.05*, and *f=0.1*. The experimental results of the three different variations of *PAS* on the typical dataset (Mushroom) are shown in Figure 6.

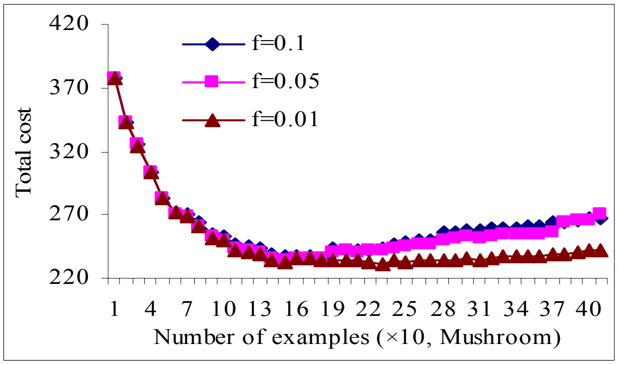

Figure 6. **The total cost of the typical dataset (*Mushroom*) under different variation coefficients in *PAS*.**

From Figure 6, we can see that the performance of *PAS* is slightly different under the different values of the variation coefficient on the dataset Mushroom. Particularly, their global minima are very close. In general, their total costs are almost the same before the global minimum point. After that, the total cost decreases slightly when the value of the variation coefficient decreases, as the acquisition cost decreases when the value of the variation coefficient decreases. This is expected, as the smaller the value of the variation coefficient, *PAS* is more strict to choose potentially useful attributes. Thus, the information acquired by *PAS* is less possible to be wasted.

Figure 6 only shows the performance of *PAS* with different variation coefficients on the typical dataset (Mushroom). As *PAS* performs similar on other datasets, we do not show the learning curves of *PAS* for them. We summarize the performance of *PAS* for all the ten datasets (including Mushroom) in terms of the global minimum total cost in Table 3.

Table 3. Summary of the global minima for each dataset under the three variation coefficients with the assumption the number of future test examples *t*=1,000.

	***f*=0.1**	***f*=0.05**	***f*=0.01**
Ecoli	153.7	153.6	150.6
Breast	345.1	335.5	317.2
Heart	558.0	556.6	548.7
Thyroid	258.3	258.5	252.4
Australia	498.3	492.3	489.7
Tic-tac-toe	657.5	641.5	641.2
Mushroom	236.2	234.5	231.6
Kr-vs-kp	612.0	609.0	607.2
Voting	295.6	295.6	292.9
Car	591.8	591.4	585.9
Average	**420.7**	**416.8**	**411.7**

From Table 3, we can conclude that the value of the variation coefficient does not affect the performance of *PAS* significantly, although the minimum average total cost of *PAS* decreases slightly when the variation coefficient becomes smaller. This is because the acquisition cost decreases slightly when the variation coefficient is smaller. Thus, *PAS* can work well if its variation coefficient is assigned a small value.

5.5 The Sizes of Future Test Set

In the subsections above, we assume that there are *t=1,000* future test examples. In this subsection, we further investigate the performance of *PAS* (with *f*=0.05) and *COM* under different sizes of future test examples. The experimental results of *PAS* and *COM* with the different sizes (*t=250, 500, and 1,000*) of the future test examples for the typical dataset (Mushroom) are shown in Figure 7.

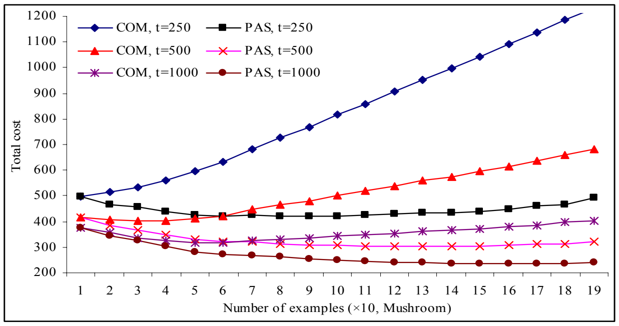

Figure 7. **The total cost of the typical dataset (*Mushroom*) under different sizes of future test examples.**

From Figure 7, we can conclude that the average total cost drops down when the number of future test examples increases. This is because there are more test examples to share the acquisition cost, for the same amount of information acquired. It is most interesting that *PAS* can achieve a much lower average total cost than *COM* does with all the three sizes of future test examples.

When the size of future test examples is smaller, the difference between *PAS* and *COM* is greater.

Figure 7 shows the learning curves of *PAS* and *COM* under different sizes of future test examples on the typical dataset (Mushroom). For other datasets, the learning curves of *PAS* and *COM* show the similar relationship. We do not display them here. Instead, we also summarize the global minimum total costs for the 10 datasets under the three sizes of future test examples in Table 4

Table 4. Summary of the global minima for each dataset under the three different sizes of future test examples.

	t=250		t=500		*t*=1,000	
	COM	***PAS***	***COM***	***PAS***	***COM***	***PAS***
Ecoli	275.3	244.8	232.0	191.7	189.7	153.6
Breast	496.2	482.9	425.1	390.0	358.2	335.5
Heart	702.6	651.9	628.6	592.9	559.4	556.6
Thyroid	458.3	428.7	370.7	322.9	310.7	258.5
Australia	634.3	609.4	559.1	537.5	493.8	492.3
Tic-tac-toe	827.4	740.6	741.2	683.0	691.4	641.5
Mushroom	496.0	419.1	401.8	302.3	317.6	234.5
Kr-vs-kp	791.8	797.4	594.8	673.8	485.7	609.0
Voting	368.4	408.0	274.7	337.7	227.9	295.6
Car	616.8	645.0	542.4	609.5	480.6	591.4
Average	**566.7**	**542.8**	**477.0**	**464.1**	**411.5**	**416.9**

From Table 4, we can conclude that the size of future test examples affects the performance of *PAS* and *COM*. When the size of future test examples increases, both *PAS* and *COM* can achieve lower global minimum on all the 10 datasets. We also can see that *PAS* consistently performs better than *COM* on the seven datasets, and worse than *COM* on the other three datasets.

5.6 The Ratios of Misclassification Costs

In the subsections above, the misclassification costs are assigned as *FP = 1,000* and *FN = 3,000*. (That is, the cost ratio is 1:3.) In this subsection, we investigate the performance of *PAS* under more extreme ratios of misclassification costs. Besides the ratio of the misclassification costs (*FP = 1,000, FN = 3,000*), we set the pair of misclassification costs with more extreme ratios, such as, (*FP = 1,000, FN = 10,000*) and (*FP = 1,000, FN = 20,000*). The experimental results on the typical dataset (Mushroom) under the three different ratios are shown in Figure 8.

From Figure 8, we can see that the total costs of *PAS* and *COM* increase when the sum of the false positive and false negative costs increases. And when the sum of the pair of the misclassification costs increases, both *PAS* and *COM* intend to acquire more information (i.e., taking more iterations) to reach its global minimum points. This is what we expected, as both *PAS* and *COM* intend to acquire more information to build a more accurate model to reduce the probability of misclassification when the misclassification cost increases. In addition, *PAS* always performs better than *COM* under different ratios of misclassification costs. Again, the lower the total cost, the better the performance is.

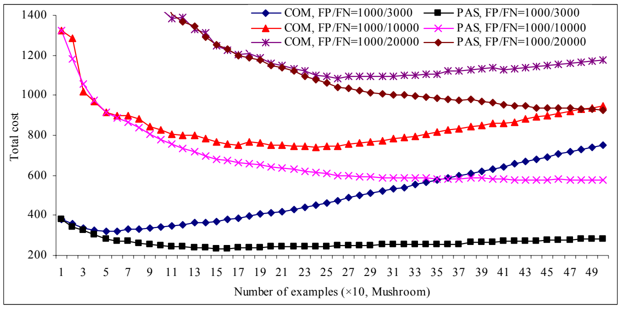

Figure 8. **The total cost of the typical dataset (Mushroom) under different ratios of misclassification costs.**

Figure 8 shows the learning curves of *PAS* and *COM* with different ratios of misclassification costs on the typical dataset (Mushroom). For other datasets, the experimental results show that both *PAS* and *COM* hold the same corresponding relationships as the dataset (Mushroom). Again, we do not show them here. Instead, the global minimum of all the 10 datasets under the three different cost ratios are summarized in Table 5.

Table 5. Summary of the global minima for each dataset under the three different ratios of misclassification costs.

	FP/FN					
	1000/3000		**1000/10000**		**1000/20000**	
	COM	***PAS***	***COM***	***PAS***	***COM***	***PAS***
Ecoli	189.7	153.6	360.4	267.3	543.8	399.3
Breast	358.2	335.5	1010.5	837.8	1927.0	1504.6
Heart	559.4	556.6	1061.9	830.5	1514.2	920.0
Thyroid	310.7	258.5	855.5	683.1	1433.6	983.7
Australia	493.8	492.3	1218.7	900.9	1850.6	1117.1
Tic-tac-toe	691.4	641.5	848.8	747.5	923.7	814.0
Mushroom	317.6	234.5	740.9	576.1	1085.4	926.9
Kr-vs-kp	485.7	609.0	908.7	942.7	1310.9	1059.2
Voting	227.9	295.7	542.7	572.1	887.4	851.3
Car	480.6	591.4	1206.8	888.7	1856.2	1053.4
Average	**411.5**	**416.9**	**875.5**	**724.7**	**1333.3**	**963.0**

From Table 5, we can conclude that both *PAS* and *COM* have higher global minima on all the 10 datasets when the sum of the misclassification costs increases. *PAS* performs consistently better than *COM* on the first seven datasets under all the cost ratios. The difference between *PAS* and *COM* becomes greater and greater when the sum of misclassification costs increases. It is also interesting that *PAS* performs better than *COM* on the other three datasets (Kr-vs-kp, Voting, and Car), when the ratio of the misclassification costs increases, particularly on the dataset Car.

In all, *PAS* performs much better than *COM* when the ratio of the misclassification costs is extreme.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose two data acquisition strategies: complete attribute strategy (*COM*) and partial attribute strategy (*PAS*) in the cost-sensitive framework. The simple complete attribute strategy (*COM*) acquires examples with all attribute values and labels. However, partial attribute strategy (*PAS*) acquires only a subset of attribute values for each example. It saves the acquisition cost.

Our experimental results and comparative studies from the UCI datasets show that the partial attribute strategy is an effective method. The higher the misclassification costs, the greater the improvement of *PAS* is, comparing with *COM*.

In our future work, we will investigate how *PAS* acquires only once the attribute values of the partial examples acquired.

7. REFERENCES

[1] Baram, Y., El-Yaniv, R., and Luz, K. 2004. Online Choice of Active Learning Algorithms. *Journal of Machine Learning Research*, 5: 255-291,

[2] Blake, C.L., and Merz, C.J. 1998. *UCI Repository of machine learning databases (website)*. Irvine, CA: University of California, Department of Information and Computer Science.

[3] Cestnik, B. 1990. Estimating probabilities: A crucial task in machine learning. *In Proceedings of the 9th European Conference on Artificial Intelligence*, 147-149, Sweden.

[4] Chai, X., Deng, L., Yang, Q., and Ling,C.X.. 2004. Test-Cost Sensitive Naïve Bayesian Classification. *In Proceedings of the Fourth IEEE International Conference on Data Mining*. UK: IEEE Computer Society Press.

[5] Domingos, P. 1999. MetaCost: A General Method for Making Classifiers Cost-Sensitive. *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 155-164. San Diego, CA: ACM Press.

[6] Elkan, C. 2001. The Foundations of Cost-Sensitive Learning. *In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence*, 973-978. Seattle, Washington: Morgan Kaufmann.

[7] Fayyad, U.M., and Irani, K.B. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. *In Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1022-1027. France: Morgan Kaufmann.

[8] Good, I.J. 1965. *The estimation of probabilities: An essay on modern Bayesian methods*. M.I.T. Press, Cambridge, Mass.

[9] Kapoor, A., and Greiner, R. 2005. Learning and Classifying under Hard Budgets. *In Proceedings of the 16th European Conference on Machine Learning* (Porto, Portugal), Springer, 170-181.

[10] Lizotte, D., Madani, O., and Greiner R. 2003. Budgeted Learning of Naive-Bayes Classifiers. *In Proceeding of the Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, August 2003.

[11] Ling, C.X., Yang, Q., Wang, J., and Zhang, S. 2004. Decision Trees with Minimal Costs. *In Proceedings of the Twenty-First International Conference on Machine Learning*. Banff, Alberta. Morgan Kaufmann.

[12] Margineantu, D.D. 2005. Active Cost-Sensitive Learning. *In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland.

[13] Melville, P., Saar-Tsechansky, M., Provost, F., and Mooney, R.J. 2004. Active Feature Acquisition for Classifier Induction. *In Proceedings of the Fourth International Conference on Data Mining*. Brighton, UK.

[14] Melville, P., Saar-Tsechansky, M., Provost, F., and Mooney, R.J. 2005. Economical Active Feature-value Acquisition through Expected Utility Estimation. *UBDM Workshop, KDD 2005*.

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

[16] Roy, N., and McCallum, A. 2001. Toward optimal active learning through sampling estimation of error reduction. *In Proceedings of the 18th International Conference on Machine Learning*, 441–448.

[17] Saar-Tsechansky, M. and Provost, F. 2004. Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2): 153-178.

[18] Ting, K.M. 1998. Inducing Cost-Sensitive Trees via Instance Weighting. *In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, 23-26. Springer-Verlag.

[19] Tong, S., and Koller, D. 2001. Support vector machine active learning with applications to text classification. J*ournal of Machine Learning Research*, 2:45–66.

[20] Turney, P.D. 2000. Types of cost in inductive concept learning. *In Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California.

[21] Turney, P.D. 1995. Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research* 2:369-409.

[22] Zhang, T., and Oles, F. 2000. A probability analysis on the value of unlabeled data for classification problems. *In the International Joint Conference on Machine Learning*, 1191–1198.

[23] Zhu, X. and Wu, X. 2005. Cost-constrained Data Acquisition for Intelligent Data Preparation. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 11. November.

[24] Zubek, V.B. and Dietterich, T. 2002. Pruning improves heuristic search for cost-sensitive learning. *In Proceedings of the Nineteenth International Conference of Machine Learning*, 27-35, Sydney, Australia: Morgan Kaufmann.