Learning and Making Decisions When Costs and Probabilities are Both Unknown

Bianca Zadrozny and Charles Elkan
Department of Computer Science and Engineering 0114
University of California, San Diego
La Jolla, California 92093-0114
{zadrozny,elkan}@cs.ucsd.edu

ABSTRACT

In many data mining domains, misclassification costs are different for different examples, in the same way that class membership probabilities are example-dependent. In these domains, both costs and probabilities are unknown for test examples, so both cost estimators and probability estimators must be learned. After discussing how to make optimal decisions given cost and probability estimates, we present decision tree and naive Bayesian learning methods for obtaining well-calibrated probability estimates. We then explain how to obtain unbiased estimators for example-dependent costs, taking into account the difficulty that in general, probabilities and costs are not independent random variables, and the training examples for which costs are known are not representative of all examples. The latter problem is called sample selection bias in econometrics. Our solution to it is based on Nobel prize-winning work due to the economist James Heckman. We show that the methods we propose perform better than MetaCost and all other known methods, in a comprehensive experimental comparison that uses the well-known, large, and challenging dataset from the KDD'98 data mining contest.

1. INTRODUCTION

The design of most supervised learning algorithms is based on the assumption that all errors, that is all incorrect predictions, are equally costly. However, this assumption is not true in many application areas. For example:

- In one-to-one marketing, the cost of making an offer to a person who does not respond is typically small compared to the cost of not contacting a person who would respond.
- In medicine, the cost of prescribing a drug to an allergic patient can be much higher than the cost of not prescribing the drug to a nonallergic patient, if alternative treatments are available.
- In image or text retrieval, the cost of not displaying a relevant item may be lower or higher than the cost of displaying an irrelevant item.
- For most animals, failing to recognize a predator and hence not fleeing is far more costly than fleeing from a non-predator.

In many domains where cost-sensitive learning and decision-making is needed, including the four cases above, each example falls into one of two alternative classes. One class is rare (for example responders, allergic patients, or predators), but the cost of not recognizing that an example belongs to this class is high. In these domains, learning methods that fail to take costs into account do not perform well. In extreme cases, a learning method that is not cost-sensitive may produce a model that is useless because it classifies every example as belonging to the most frequent class.

In recent years, the realization that cost-sensitive learning methods are required in many real-world applications has led to a substantial amount of research. Turney [22] provides a bibliography of this research. Nonetheless, the only general method for cost-sensitive learning published so far is a method named MetaCost due to Domingos [8]. In this paper we present an alternative method that we call direct cost-sensitive decision-making. Our analysis shows that the new method is more general than MetaCost as originally published, and our experimental results show that the new method is preferable to MetaCost.

This paper is organized as follows. In Section 2 we explain MetaCost and direct cost-sensitive decision-making. Then in Section 3 we show how to apply these methods to the difficult real-world dataset used in the KDD'98 data mining contest. Both MetaCost and direct cost-sensitive decision-making require accurate estimates of class membership probabilities. In Section 4 we present two techniques that allow accurate probability estimates to be obtained from a decision tree: smoothing and curtailment. We also present binning as a technique for making naive Bayes probability estimates accurate. Previous research has been based on the assumption that misclassification costs are the same for all examples and known in advance, but in general these costs are example-dependent and unknown, in the same way that class membership probabilities are example-specific and not known in advance. In Section 5 we discuss this issue and the issue of how sample selection bias affects cost estimation. Finally, experimental results using the KDD'98 dataset are presented in Section 6 and in Section 7 we summarize the main contributions of this paper. Related work is discussed as necessary throughout the paper.

2. METACOST VERSUS DIRECT COST-SENSITIVE DECISION-MAKING

In any domain where a cost-sensitive learning method is to be applied, each training or test example x is associated with a cost $C(i,j,x)$ of predicting class i for x when the true class of x is j. If these costs are known for each x and for all i and j then it is straightforward to compute an optimal policy for decision-making. The optimal prediction for x, i.e. the optimal decision concerning x or label to assign to x, is the class i that leads to the lowest expected cost

$$\sum_j P(j|x)C(i,j,x). \tag{1}$$

Given x, for each alternative i the expected cost is a weighted average where the weight of $C(i,j,x)$ is the conditional probability of the class j given x.

The central idea behind the MetaCost method is to change the label of each training example to be its optimal label according to Equation (1), and then to learn a classifier that predicts these new labels. Applying MetaCost requires knowledge of the conditional probability $P(j|x)$ for each training example x and each possible true class j for x. Almost always, these probabilities are not given as part of the training data. Instead, the training data must be used to learn a classifier that estimates $P(j|x)$ for each training example x and each j.

Any learned classifier that can provide conditional probability estimates for training examples can also provide conditional probability estimates for test examples. Using these probability estimates we can directly compute the optimal label for each test example using Equation (1). This process is the method that we call direct cost-sensitive decision-making. Experimental results comparing MetaCost and direct cost-sensitive decision-making are given in Section 6.

The basic MetaCost idea can be implemented in many ways. Our implementation differs from that described by Domingos [8] in two important ways. First, the original description of MetaCost is based on the assumption that costs are known in advance and are the same for all examples, i.e. that $C(i,j,x) = C(i,j)$ with no dependence on x. Provost and Fawcett [17] have pointed out that this assumption is not always true: "For some problems, different errors of the same type have different costs." We generalize MetaCost by relaxing this assumption.

Second, we do not estimate probabilities using bagging [5]. Instead of bagging, we use simpler methods based on single decision trees. As pointed out recently by Margineantu [15], bagging gives voting estimates that measure the stability of the base classifier learning method at an example, not the actual class conditional probability of the example. (A classifier learning method is stable at an example if classifiers learned from different resamples predict the same label for the example). For experimental results confirming that bagging is not a good way of improving probability estimates obtained from decision trees, see Zadrozny and Elkan [23].

In general, bagging does not give probability estimates that are unbiased and well-calibrated, whether or not the base learning method is stable. If a learning method is unstable and gives classifiers that make 0/1 predictions, then bagging tends to be useful because voting estimates are numbers between 0 and 1, which are preferable to 0/1 predictions as continuous probability estimates. However, in general these scores are not unbiased estimates. If a learning method gives classifiers that individually yield unbiased probability estimates, then bagging these classifiers is likely to reduce variance beneficially, while maintaining unbiasedness. But then the question remains of how to get individual scores that are unbiased in the first place. Section 4 below answers this question.

3. A TESTBED: THE KDD'98 CHARITABLE DONATIONS DATASET

The dataset used in the experimental work described in this paper is a well-studied, large and challenging dataset that was first used in the data mining contest associated with the 1998 KDD conference. This dataset and associated documentation are available in the UCI KDD repository [2]. The dataset contains information about persons who have made donations in the past to a certain charity. The decision-making task is to choose which donors to request a new donation from. This task is completely analogous to typical one-to-one marketing tasks for many other organizations, both non-profit and for-profit. Mathematically, the task has the same structure as all the two-class cost-sensitive learning and decision-making problems mentioned in the introduction.

The KDD'98 dataset is divided in a fixed, standard way into a training set and a test set. The training set consists of 95412 records for which it is known whether or not the person made a donation (a 0/1 response) and how much the person donated, if a donation was made. The test set consists of 96367 records from the same donation campaign for which similar donation information was not published until after the KDD'98 competition. In order to make our experimental results directly comparable with those of previous work, we use the standard training set/test set division.

Mailing a solicitation to an individual costs the charity \$0.68. The overall percentage of donors among potential recipients is about 5%. The donation amount for persons who respond varies from \$1 to \$200. Given the low response rate and the variation in the value of gifts, it is not easy to achieve a profit that is much higher than that obtained by soliciting all potential donors. The profit obtained by soliciting every individual in the test set is \$10560, while the profit attained by the winner of the KDD'98 competition was \$14712.

Many participants in the KDD'98 competition submitted entries that were worse than useless, because they achieved profits substantially lower than \$10560. One likely reason for low success is that the individuals in the KDD'98 dataset are already filtered to be a reasonable set of prospects. They have been the targets of a real donation campaign, selected using standard techniques in direct marketing such as recency-frequency-amount (RFA) scoring. The task now for any data mining method is to improve upon the already good performance of the unknown method that was applied to create the KDD'98 dataset.

Research on cost-sensitive learning has traditionally been couched in terms of costs, as opposed to benefits or profits. However, in many domains, including the charitable donations domain, it is easier to talk consistently about benefits than about costs. The reason is that all benefits are straightforward cash flows relative to a baseline wealth of \$0, while some costs are counterfactual opportunity costs [11]. Accordingly, our formulation of the problem is in terms of benefits instead of costs. This formulation applies very generally, including to all the scenarios mentioned in the Introduction, because benefits are not necessarily monetary. Benefits are utilities that can be measured in any unit of accounting. We use the word "benefit" here because of the standard phrase "cost/benefit analysis."

The optimal predicted label for example x is the class i that maximizes

$$\sum_j P(j|x)B(i,j,x) \tag{2}$$

where $B(i,j,x)$ is the benefit of predicting class i when the true class is j. Let the label $j=0$ mean the person x does not donate, and let $j=1$ mean the person does donate. If the person donates, the donation is of a variable amount, say $y(x)$. The cost of mailing a solicitation is \$0.68, so we have the following benefit matrix $B(i,j,x)$:

	actual non-donor	actual donor
predict non-donor	0	0
predict donor (mail)	-0.68	$y(x)-0.68$

Notice that $B(1,1,x)$ is example-dependent and unknown for test examples. We shall argue later that no fixed matrix of costs or benefits can lead to good decision-making. There is no constant c such that it would be reasonable to replace $B(1,1,x)$ by the same value c for all x. All approaches to this task, and to other tasks with the same structure, that are based on a fixed cost or benefit matrix will have poor performance. Of course, some approaches can take into account the fact that $y(x)$ is example-dependent without estimating $y(x)$ explicitly.

The expected benefit of not soliciting a person x, i.e. of deciding $i=0$ for x, is

$$P(j=0|x)B(0,0,x)+P(j=1|x)B(0,1,x)=0.$$

The expected benefit of soliciting x is

$$P(j=0|x)B(1,0,x)+P(j=1|x)B(1,1,x)$$

$$=(1-P(j=1|x))(-0.68)+P(j=1|x)(y(x)-0.68)$$

$$=P(j=1|x)y(x)-0.68.$$

The optimal policy is to solicit exactly those people for whom the expected benefit of mailing is greater than the expected benefit of not mailing: individuals for whom

$$P(j=1|x)y(x)-0.68>0.$$

In other words, the optimal policy is to mail to people for whom the expected return $P(j=1|x)y(x)$ is greater than the cost of mailing a solicitation:

$$P(j=1|x)y(x)>0.68. \tag{3}$$

In order to apply this policy, we need to estimate the conditional probability of making a donation $P(j=1|x)$ and the donation amount $y(x)$ for each example x in the training set, in the case of MetaCost. We need to estimate these values for both training and test examples in the case of direct cost-sensitive decision-making.

Although we use the KDD'98 dataset for concreteness, the methods described in this paper apply to cost-sensitive learning in general. In any cost-sensitive learning application, in order to use Equation (1) or (2) to obtain an optimal labeling, we need to estimate conditional class membership probabilities accurately. Costs or benefits must also be estimated whenever they are unknown for some examples.

In general, if x is a test example then $C(i,j,x)$ will be unknown for all i and j. If x is a training example then $C(i,j,x)$ will be known for some i and j pairs, but unknown for other pairs. Of course, if costs are not example-dependent, that is, if $C(i,j,x)=C(i,j,y)$ for all examples x and y, then costs do not need to be estimated for any training or test examples. This special case is the only case considered in previous general research on cost-sensitive learning. In the remainder of this paper, we present new methods for estimating costs and probabilities. All these methods can applied without change in a wide variety of domains.

4. PROBABILITY ESTIMATION METHODS

An estimate of the conditional probability of membership in each class is required for each training example if MetaCost is used, and for each test example if direct cost-sensitive decision-making is used.

This section explains our methods for obtaining calibrated probability estimates from decision tree and naive Bayesian classifiers. We first explain the deficiencies that cause standard decision tree methods not to give accurate probability estimates, and we then explain methods to overcome these limitations. A final subsection presents a simple method for obtaining calibrated probabilities from a naive Bayesian classifier.

4.1 Deficiencies of decision tree methods

Throughout this paper, C4.5 [18] is the representative decision tree learning method used, but all our analyses and suggestions apply equally to other decision tree methods such as CART [6].

When classifying a test example, C4.5 and other decision tree methods assign by default the raw training frequency $p=k/n$ as the score of any example that is assigned to a leaf that contains k positive training examples and n total training examples. These training frequencies are not accurate conditional probability estimates for at least two reasons:

1. High bias: Decision tree growing methods try to make leaves homogeneous, so observed frequencies are systematically shifted towards zero and one.

2. High variance: When the number of training examples associated with a leaf is small, observed frequencies are not statistically reliable.

Pruning methods as surveyed by Esposito *et al.* [12] can in principle alleviate problem (2) by removing leaves that contain too few examples. However, standard pruning methods are not suitable for unbalanced datasets, because they are based on accuracy maximization. On the KDD'98 dataset C4.5 produces a pruned tree that is a single leaf. Since the base rate of positive examples, that is the overall probability $P(j = 1)$, is about 5%, this tree has accuracy 95%, but it is useless for estimating example-specific conditional probabilities $P(j = 1|x)$.

In general, trees pruned with the objective of maximizing accuracy are not useful for ranking test examples, or for estimating class membership probabilities. The standard C4.5 pruning method is not alone in being incompatible with accurate probability estimation. Quinlan's recent decision tree learning method, C5.0, and CART also do pruning based on accuracy maximization. Both C4.5 and C5.0 have rule set generators that are a commonly used alternative to pruning [18]. These methods are also based on accuracy maximization, so they are also unsuitable for probability estimation.

We show how to improve directly the accuracy of decision tree probability estimates. Our experiments use C4.5 without pruning and without collapsing to obtain raw scores that can be transformed into accurate class membership probabilities. The choice to do no pruning is supported by the results of Bradford *et al.* [4], who find that performing no pruning and variants of pruning adapted to loss minimization both lead to similar performance. Not using pruning is also suggested by Bauer and Kohavi [1] in their Section 7.3.

The methods we propose transform the leaf scores of a standard decision tree. Completely different methods have been suggested, but they have major drawbacks. Smyth *et al.* [19] use kernel density estimators at the leaves of a decision tree. However their algorithms are based on C4.5 and CART with pruning, so they are unsuitable for highly unbalanced datasets. Their experiments use only synthetic, reasonably balanced datasets. Our experiments use an unbalanced real-world dataset where the less probable class has a base rate of only about 5%. Estimating probabilities using bagging has been suggested by Breiman [5] and by Domingos [8], but as explained above in Section 2, bagging does not give unbiased probability estimates in general.

4.2 Smoothing

One way of improving the probability estimates given by a decision tree is to make these estimates smoother, i.e. to adjust them to be less extreme. Provost and Domingos [16] suggest using the Laplace correction method. For a two-class problem, this method replaces the conditional probability estimate $p = \frac{k}{n}$ by $p' = \frac{k+1}{n+2}$.

The Laplace correction method adjusts probability estimates to be closer to 1/2, which is not reasonable when the two classes are far from equiprobable, as is the case in many real-world applications. From a Bayesian perspective, a conditional probability estimate should be smoothed towards the corresponding unconditional probability.

We replace the probability estimate $p = \frac{k}{n}$ by $p' = \frac{k+b \cdot m}{n+m}$, where b is the base rate of the positive class and m is a parameter that controls how much scores are shifted towards b. This smoothing method is called m-estimation [7]. For example, if a leaf contains four training examples, one of which is positive, the raw C4.5 decision tree score of any example assigned to this leaf is 0.25. The smoothed score with $m = 200$ and $b = 0.05$ is

$$p' = \frac{1 + 0.05 \cdot 200}{4 + 200} = \frac{11}{204} = 0.0539.$$

Previous papers have suggested choosing m by cross-validation. Given a base rate b, we suggest using m such that $bm = 10$ approximately. This heuristic ensures that raw probability estimates that are likely to have high variance, those with $k \leq 10$, are given low credence. Experiments show that the effect of smoothing by m-estimation is qualitatively similar for a wide range of values of m, so, as is highly desirable, the precise value chosen for m is unimportant.

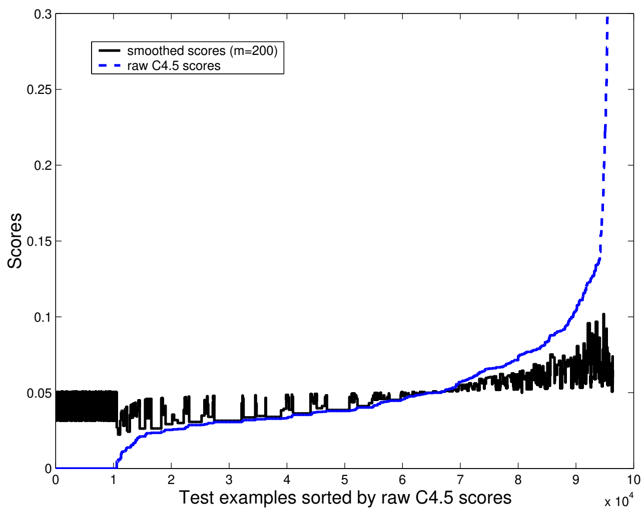

Figure 1: Smoothed scores and raw C4.5 scores for test examples sorted by raw score. The figure shows how the scores change after smoothing is applied. In particular, examples that are assigned a score close to 0 (left-hand side) or 1 (right-hand side) by C4.5 have their scores significantly shifted towards the base rate by smoothing.

Figure 1 shows the smoothed scores with $m = 200$ of the KDD'98 test set examples sorted by their raw C4.5 scores. As expected, smoothing shifts all scores towards the base rate of approximately 0.05, which is desirable given that C4.5 scores tend to be overestimates or underestimates. While raw C4.5 scores range from 0 to 1, smoothed scores range from 0.0224 to 0.1018.

4.3 Curtailment

As discussed above, without pruning decision tree learning methods tend to overfit training data and to create leaves in which the number of examples is too small to induce conditional probability estimates that are statistically reliable (which we call small leaves). Smoothing attempts to correct these estimates by shifting them towards the overall average probability, i.e. the base rate b. However, if the parent of a small leaf contains enough examples to induce a statistically reliable probability estimate, then assigning this estimate to a test example associated with the leaf may be more accurate then assigning it a combination of the base rate and the observed leaf frequency, as done by smoothing. If the parent of a small leaf still contains too few examples, we can use the score of the grandparent of the leaf, and so on until the root of the tree is reached. At the root, of course, the observed frequency is the training set base rate.

We call this method of improving conditional probability estimates curtailment because when classifying an example, we curtail search through the decision tree as soon as we reach a node that has less than v examples, where v is a parameter of the method. The score of the parent of this node is then assigned to the example in question. As for smoothing, v can be chosen by cross-validation, or using a heuristic such as making $bv = 10$. We choose $v = 200$ for all our experiments. Informal experiments show that values of v between 100 and 400 give similar results, so the exact setting of v is not critical.

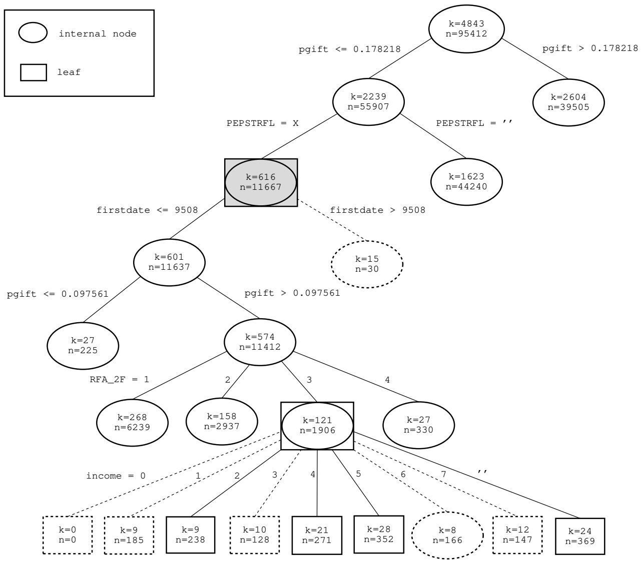

Figure 2: Part of the decision tree obtained by curtailment with $v = 200$. The dotted nodes are present in the original C4.5 tree, but are effectively eliminated from the curtailment tree because $n < v$.

Given the KDD'98 training set, curtailment effectively creates the decision tree shown in part in Figure 2. The distinction between internal nodes and leaves is blurred in this tree, because a node may serve as an internal node for some examples and as a leaf for others, depending on the attribute values of the examples. The node in gray is an example of a node that can serve both as an internal node and as a leaf, because one of its branches has been eliminated from the tree, but not all.

Curtailment is not equivalent to any type of pruning, nor to traditional early stopping during the growing of a tree, because those methods eliminate all the children of a node simultaneously. In contrast, curtailment may eliminate some children and keep others, depending on the number of training examples associated with each child. Intuitively, curtailment is preferable to pruning for probability estimation because nodes are removed from a decision tree only if they are likely to give unreliable probability estimates.

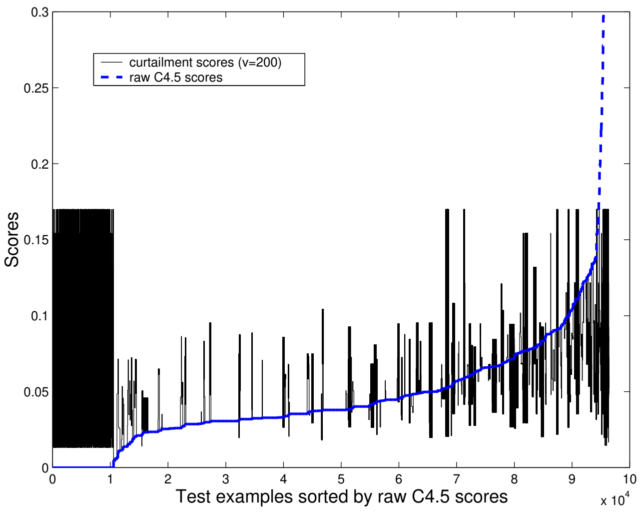

Figure 3: Curtailment scores and raw C4.5 scores for test examples. Examples are sorted by raw C4.5 score. The figure shows that scores change significantly after curtailment is applied, in particular for examples that are assigned a score close to 0 (left-hand side) or 1 (right-hand side) by C4.5.

Figure 3 shows the curtailment scores with $v = 200$ of the KDD'98 test set examples sorted by their raw C4.5 scores. The jagged lines in the chart show that many scores are changed significantly by curtailment. Overall, the range of scores is reduced as with smoothing, but not as much. The minimum curtailment score is 0.0045 while the maximum is 0.1699.

4.4 Calibrating naive Bayes classifier scores

Naive Bayesian classifiers are based on the assumption that within each class, the values of the attributes of examples are independent. It is well-known that these classifiers tend to give inaccurate probability estimates [9]. Given an example x, suppose that a naive Bayesian classifier computes the score $n(x)$. Because attributes tend to be positively correlated, these scores are typically too extreme: for most x, either $n(x)$ is near 0 and then $n(x) < P(j = 1|x)$ or $n(x)$ is near 1 and then $n(x) > P(j = 1|x)$. However, naive Bayesian classifiers tend to rank examples well: if $n(x) < n(y)$ then $P(j = 1|x) < P(j = 1|y)$.

We use a histogram method to obtain calibrated probability estimates from a naive Bayesian classifier. We sort the training examples according to their scores and divide the sorted set into b subsets of equal size, called bins. For each bin we compute lower and upper boundary $n(\cdot)$ scores. Given a test example x, we place it in a bin according to its score $n(x)$. We then estimate the corrected probability that x belongs to class j as the fraction of training examples in the bin that actually belong to j.

The number of different probability estimates that binning can yield is limited by the number of alternative bins. This number, $b = 10$ in our experiments, must be small in order to reduce the variance of the binned probability estimates, by increasing the number of examples whose 0/1 memberships are averaged inside each bin. Binning reduces the resolution, i.e. the degree of detail, of conditional probability estimates, while improving the accuracy of these estimates by reducing both variance and bias compared to uncalibrated estimates.

Binning is a discrete non-parametric method for calibrating probability estimates. In future work, we should consider using continuous methods such as the super-smoother or loess to obtain calibrated probability estimates with a greater degree of detail. Sobehart *et al.* [20] use Gaussian kernel regression method in a similar context. Applying parametric methods to calibrate naive Bayes scores is not straightforward. For example, Bennett [3] reports that sigmoid functions cannot transform naive Bayes scores into well-calibrated probability estimates.

With most learning methods, in order to obtain binned estimates that do not overfit the training data, we should partition the training set into two subsets. One subset would be used to learn the classifier that yields uncalibrated scores, while the other subset would be used for the binning process. More training examples would be assigned to the first subset because learning a classifier involves setting many more parameters than setting the binned probabilities. For naive Bayesian classifiers, however, separate subsets are not necessary because this learning method does not overfit the training data much. So we use the entire training set both for learning the classifier and for binning.

4.5 Averaging probability estimates

If different methods provide noisy probability estimates that are partially uncorrelated, it is intuitive that averaging the probability estimates given by these methods reduces the noise, thereby improving the probability estimates.

This intuition is formalized by Tumer and Ghosh [21]. They show that by combining the probability estimates given by different classifiers through averaging we can reduce the variance of the probability estimates. The reduction in the variance depends on the degree of correlation of the noise in the probability estimates produced by each classifier and on how many classifiers are used.

Assuming that the variance of the probability estimates given by each classifier is approximately the same, the variance of the averaging combiner is given by

$$\bar{\sigma}^2 = \frac{1 + \rho(N-1)}{N}\sigma^2$$

where σ^2 is the variance of each original classifier, N is the number of classifiers and ρ is the correlation factor among all classifiers. If the classifiers are independent ($\rho = 0$), the combined variance is reduced by N. On the other hand, if the classifiers are completely correlated ($\rho = 1$), the variance is unchanged.

Since the probability estimates obtained from the decision tree and naive Bayesian classifiers are partially uncorrelated, averaging them should yield estimates that are more accurate than those given by each individual method. In Section 6 we show experimental results that confirm this hypothesis.

5. ESTIMATING DONATION AMOUNTS

In general, in cost-sensitive learning we need to estimate example-specific misclassification costs, in addition to example-specific class conditional probabilities. We need to estimate misclassification costs for training examples when using MetaCost, and for test examples when using direct cost-sensitive decision-making.

If costs and probabilities are both unknown, then estimating costs well can be more important for making good decisions than estimating probabilities well. Cost estimates are more important if the relative variation of costs across different examples is greater than the relative variation of probabilities. The dynamic range of costs may be greater than the dynamic range of probabilities either because the dynamic range of true costs is greater, or because estimating costs accurately is easier than estimating probabilities accurately.

In the KDD'98 domain for example, estimating donation probabilities is difficult. Our best method for this task, the averaging of smoothing, curtailment, and binned naive Bayes, gives conditional probabilities in the narrow range from 0.0172 to 0.1189. Estimating donation amounts is easier because past amounts are excellent predictors of future amounts.

It may appear that for non-donors in the training set we should impute a donation amount of zero, since their actual donation amount is zero. But this imputation would be analogous to imputing a donation probability of zero for the non-donors based on the fact that they have not donated, which is clearly wrong. When responding to a solicitation a person has to make two decisions. The first is whether to donate or not, while the second is how much to donate. Conceptually, these decisions are governed by two different random processes, not necessarily sequential or independent of course. For donors in the training set, the outcome of the random process that sets the donation amount is known, while for non-donors, this outcome is unknown. For individuals in the test set, the outcome of both random processes is unknown. Whenever the outcome of one or both processes is unknown, the learning task is to estimate its outcome. For non-donors in the training set, the task is to estimate the amounts that they would have donated, if they had made donations.

It is also wrong to impute any fixed quantity as a donation estimate for test examples. Using the same donation estimate for all test examples means that the decision whether or not to solicit a person is based exclusively on the probability that they will donate. This method is equivalent to using a fixed cost matrix for test examples. In general, whenever misclassification costs are assumed to be fixed, different decisions for different examples can only be based on different conditional probability estimates for those examples.

For clarity, the arguments in the previous paragraphs are expressed in language that is specific to the donations domain. However, similar arguments apply to any scenario where costs or benefits are different for different examples. These costs or benefits must be estimated for each example, whenever they are unknown. Assuming that unknown costs or benefits are zero or constant is incorrect.

The method we use for estimating donation amounts is least-squares multiple linear regression (MLR). The donors in the training set that have donated at most \$50 are used as input for the regression, which is based on one original attribute and one derived attribute:

- `lastgift`: dollar amount of most recent gift,
- `ampergift`: average gift amount in responses to the last 22 promotions.

Since the topic of this paper is not variable selection, we somewhat arbitrarily choose these two attributes based on previous work. We use the linear regression equation to estimate donation amounts for all examples in both the training and test sets.

Donations of more than \$50 are very rare in our domain: 46 of 4843 donations recorded in the training set. We eliminate these examples from the regression training set as a heuristic attempt to reduce the impact of outliers on the regression. If included, these examples have the most influence on the regression equation, because they have the highest y values and the regression equation is chosen to minimize the sum of squared y errors. However, it is less important to estimate y values accurately for these individuals, because the optimal decision is always to solicit them, given that predicted donation probabilities are always over 1.5%. Accurate predicted donation probabilities are never close to zero because of the intrinsic difficulty of predicting whether or not a person will donate. In future work, we shall consider using non-linear regression methods that are able to cope adaptively with outliers.

5.1 The problem of sample selection bias

When estimating donation amounts, a fundamental problem is that any estimator, for example a regression equation, must be learned based on examples of people who actually donate. But this estimator must then be applied to a different population, i.e. both donors and non-donors. This problem is known in general as sample selection bias. It occurs whenever the training examples used to learn a model are drawn from a different probability distribution than the examples to which the model is applied.

In the donations domain, the donation amount and the probability of donation are negatively correlated. People who are more likely to respond to a solicitation tend to make smaller donations, while people who make larger donations are less likely to respond. This relationship is illustrated in Figure 4. Since examples of people who actually donate are the only training examples for the regression, donation amounts estimated by the regression equation tend to be too low for test examples that have a low probability of donation.

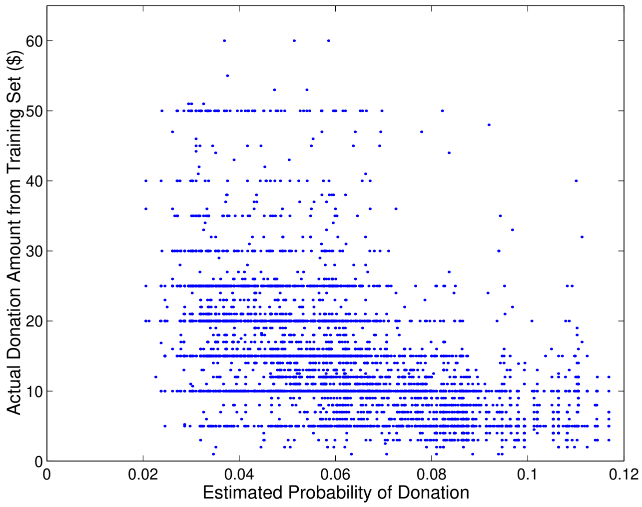

Figure 4: Actual donation amount versus estimated probability of donation, for all donors in the training set. A negative correlation between donation amount and probability of donation is visible.

As we have explained previously [10], the standard method of compensating for sample selection bias in econometrics is a two-step procedure due to James J. Heckman of the University of Chicago [13]. In October 2000 Heckman was awarded the Nobel prize in economics for developing and applying this procedure. Expressed using our notation, Heckman's procedure is applicable when each example x belongs to one of two classes, i.e. $j(x) = 0$ or $j(x) = 1$, and the dependent variable to be estimated $y(x)$ is observed for a training example if and only if $j(x) = 1$. The first step of the procedure is to learn a probit linear model to estimate conditional probabilities $P(j = 1|x)$. A probit model is a variant of logistic regression where the cumulative Gaussian probability density function is the sigmoid function. The second step of Heckman's procedure is to estimate $y(x)$ by linear regression using only the training examples x for which $j(x) = 1$, but including for each x a transformation of the estimated value of $P(j = 1|x)$. Heckman has proved that this procedure yields estimates of $y(x)$ that are unbiased for all x, regardless of whether $j(x) = 0$ or $j(x) = 1$, under certain conditions [13].

Our second method for estimating donation amounts is a nonlinear variant of Heckman's procedure. Instead of using a linear estimator for $P(j = 1|x)$, we use a decision tree or a naive Bayes classifier to obtain probability estimates, as described in Section 4. We then include these probability estimates directly as an additional attribute when applying a learning method to obtain an estimator for $y(x)$. This learning method could be a nonlinear method, for example a neural network method, but in order to investigate carefully

Probability estimation method	Without Heckman		With Heckman	
	Training set	Test set	Training set	Test set
Smoothed C4.5 (sm)	\$19256	\$14093	\$18583	\$14321
C4.5 with curtailment (cur)	\$16722	\$13670	\$17037	\$14161
Binned naive Bayes (binb)	\$14262	\$14208	\$14994	\$15094
Average(sm, cur)	\$18591	\$14518	\$18474	\$14879
Average(sm, cur, binb)	\$18140	\$14877	\$17400	\$15329

Table 1: Profit attained on the training and test sets using each probability estimation method.

the usefulness of Heckman's idea, we hold everything else constant and just provide the estimated $P(j = 1|x)$ values as a third attribute of x to a linear regression that is otherwise the same as in the first method.

6. EXPERIMENTAL RESULTS

In this section, we investigate experimentally how the new probability and cost estimation methods described above affect the profit attained on the KDD'98 data set. We first report our results, and then discuss the issue of statistical significance.

For each of the probability estimation methods described in Section 4, Table 1 shows the profit obtained when we use the multiple linear regression that includes only `lastgift` and `ampergift` as attributes, and when we apply Heckman's procedure by including the probability estimates as an additional attribute to the regression. When we use Heckman's procedure, the profit on the test set increases for all probability estimation methods, on average by \$484. The fact that the improvement is systematic indicates that Heckman's procedure succeeds in correcting sample selection bias.

To implement MetaCost, probability and donation estimates obtained as described in Sections 4 and 5 are used to relabel the training set according to Equation 1. We train C4.5, with pruning and collapsing, on the relabeled training examples and apply the resulting decision tree to the training and test examples. The profit obtained from mailing the people who are labeled positive by the decision tree is given in Table 2.

Probability estimation method	Training set	Test set
Smoothed C4.5 (sm)	\$17359	\$12835
C4.5 with curtailment (cur)	\$15869	\$11283
Binned naive Bayes (binb)	\$13608	\$14113
Average(sm, cur)	\$17547	\$13284
Average(sm, cur, binb)	\$16531	\$13515

Table 2: Profit attained on the training and test sets using MetaCost with each probability estimation method. Donation amount estimates are obtained from the MLR with the Heckman adjustment.

Comparing the results in Table 2 with the results in the second half of Table 1, we see that MetaCost performs consistently less well than direct cost-sensitive decision-making. On average, the profit achieved with MetaCost on the test set is \$1751 lower than the profit achieved with direct cost-sensitive decision-making. The best result with MetaCost is \$14113, while the best result with the direct method is \$15329, which is better than the result obtained by the winner of the KDD'98 contest, \$14712.

We conclude that direct cost-sensitive decision-making is preferable to MetaCost. We attribute the worse performance of MetaCost to the difficulty that any single model must have in estimating costs and probabilities as accurately as two separate models. Learning a single classifier from relabeled training data causes more errors in approximating the ideal decision boundary than learning two estimators.

It is difficult to make definite statements about the statistical significance of the experimental results above. There are 4872 donors in the fixed test set. For these individuals, the average donation is \$15.62. On a different test set drawn randomly from the same probability distribution, one would expect a one standard deviation fluctuation up or down of $\sqrt{4872}$ in the number of donors. This fluctuation would cause a change of about $\$15.62 \cdot \sqrt{4872} = \$1090$ in total profit. Therefore, a profit difference of less than \$1090 between two methods is not statistically significant.

Many of the profit differences between methods that we observe are less than \$1090. There are several avenues we could follow to obtain statistically significant differences between methods. One avenue would be to use cross-validation, instead of a single training set and a single test set. However, the training set/test set split we use is standard. If we did not use it, our results would not be comparable with those of previous work using the same dataset.

Another avenue would be to use multiple datasets for comparing different methods, as done for example by Domingos [8]. But, despite the unquestioned importance of differential costs in many learning tasks, the KDD'98 dataset is the only dataset in the UCI repositories for which real-world misclassification cost information is available. Most previous experimental research on cost-sensitive learning has used arbitrary cost matrices. We prefer to use real cost data, especially since we are interested in the situation where costs are different for different examples.

The purpose of the experiments reported here is not so much to identify a single best method for cost-sensitive learning and decision-making, but rather to compare the usefulness of the alternative submethods proposed in previous sections. In all trials, the test set profit achieved using MetaCost is lower and using Heckman's procedure is higher. We choose not to quantify the level of this statistical significance because doing so would require making assumptions that are certainly false, and therefore give misleading conclusions. In particular, because all trials use the same training and test sets, they are not statistically independent. Always using the same training and test set removes one source of variance, so even small differences in performance between data mining methods are in fact likely to be genuine [14].

7. CONCLUSIONS

The main contributions of this paper are the following:

1. We explain a general method of cost-sensitive learning that performs systematically better than MetaCost in our experiments.
2. We provide a solution to the fundamental problem of costs being different for different examples, and unknown in general.
3. As part of (2), we identify and solve the problem of sample selection bias, i.e. the fact that the training set available for learning to estimate costs is not representative of test examples, or indeed of other training examples.

All the methods we propose are evaluated carefully with experiments using a large, difficult and highly cost-sensitive real-world dataset. Previous research has tended to use small datasets with synthetic cost data.

We have used simple methods for both probability estimation and cost estimation in this paper in order to illustrate our general cost-sensitive learning approach and to provide a baseline for future research. Our recommended methods already perform better than the methods of the winners of the KDD'98 and KDD'99 contests. Using a more sophisticated regression method for estimating donation amounts, we already have preliminary results that are a further improvement.

Our experiments are designed so that both MetaCost and the alternative we propose use the same methods for estimating costs and probabilities. Therefore, we expect our conclusion that direct cost-sensitive decision-making is preferable to remain valid with other estimation methods. In particular, both MetaCost and direct cost-sensitive decision-making will be improved by any improvement in techniques for probability estimation.

8. REFERENCES

[1] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999.

[2] S. D. Bay. UCI KDD archive. Department of Information and Computer Sciences, University of California, Irvine, 2000. `http://kdd.ics.uci.edu/`.

[3] P. N. Bennett. Assessing the calibration of naive Bayes' posterior estimates. Technical Report CMU-CS-00-155, School of Computer Science, Carnegie Mellon University, 2000.

[4] J. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. Brodley. Pruning decision trees with misclassification costs. In *Proceedings of the European Conference on Machine Learning*, pages 131–136, 1998.

[5] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[6] L. Breiman, J. H. Friedman, R. A. Olsen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[7] J. Cussens. Bayes and pseudo-Bayes estimates of conditional probabilities and their reliability. In *Proceedings of the European Conference on Machine Learning*, pages 136–152. Springer Verlag, 1993.

[8] P. Domingos. MetaCost: A general method for making classifiers cost sensitive. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 155–164. ACM Press, 1999.

[9] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 105–112. Morgan Kaufmann Publishers, Inc., 1996.

[10] C. Elkan. Cost-sensitive learning and decision-making when costs are unknown. In *Workshop Notes, Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, 2000.

[11] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Aug. 2001.

[12] F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, May 1997.

[13] J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47:153–161, 1979.

[14] E. C. Malthouse. Assessing the performance of direct marketing scoring models. *Journal of Interactive Marketing*, 15(1):49–62, 2001.

[15] D. Margineantu. On class probability estimates and cost-sensitive evaluation of classifiers. In *Workshop Notes, Workshop on Cost-Sensitive Learning, International Conference on Machine Learning*, June 2000.

[16] F. Provost and P. Domingos. Well-trained PETs: Improving probability estimation trees. CDER Working Paper #00-04-IS, Stern School of Business, New York University, NY, NY 10012, 2000.

[17] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

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

[19] P. Smyth, A. Gray, and U. Fayyad. Retrofitting decision tree classifiers using kernel density estimation. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 506–514. Morgan Kaufmann Publishers, Inc., 1995.

[20] J. R. Sobehart, R. M. Stein, V. Mikityanskaya, and L. Li. Moody's public risk model: A hybrid approach to modeling short term default risk. Technical report, Moody's Investors Service, Global Credit Research, 2000. Available at `http://www.moodysqra.com/research/crm/53853.asp`.

[21] K. Tumer and J. Ghosh. Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. Technical Report TR-95-02-98, The Computer and Vision Research Center, The University of Texas at Austin, 1995.

[22] P. Turney. Cost-sensitive learning bibliography. Institute for Information Technology, National Research Council, Ottawa, Canada, 2000. `http://extractor.iit.nrc.ca/bibliographies/cost-sensitive.html`.

[23] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001. To appear.