Learning and Making Decisions When Costs and Probabilities are Both Unknown

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

Draft: Please do not distribute further.

December 31, 2000

Abstract

In many machine learning 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. This paper first discusses how to make optimal decisions given cost and probability estimates, and then presents decision tree learning methods for obtaining well-calibrated probability estimates. The paper then explains 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 are successful in a comprehensive comparison with MetaCost that uses the well-known and difficult 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 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 information retrieval, the cost of not displaying a relevant document may be lower or higher than the cost of displaying an irrelevant document.
- 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 the class of allergic patients), 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 [Tur00] 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 [Dom99]. 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 three techniques that allow accurate probability estimates to be obtained from a decision tree: binning, smoothing and early stopping. 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. Then in Section 6 we describe a heuristic method to compensate for possible biases in estimating probabilities and costs. Finally, experimental results using the KDD'98 dataset are presented in Section 7, and in Section 8 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 example 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 of 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. The basic MetaCost idea can be implemented in many ways; our experimental results investigate 24 different implementations in total. Our implementations differ from those described by Domingos [Dom99] in two important ways. First, we do not use bagging to estimate probabilities. Instead, we use simpler methods based on single decision trees, following the lead of Domingos in a recent paper [DP00].

Second, 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 [PF99] 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, and we compare variants of MetaCost that use different methods for estimating example-dependent costs.

Because MetaCost uses Equation 1, it 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 classifier that can provide conditional probability estimates for training examples can provide conditional probability estimates for test examples also. 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.

3 A testbed: The KDD'98 charitable donations dataset

The dataset used in the experimental work described in this paper is a well-studied, difficult dataset that was first used in the data mining contest associated with the 1998 KDD conference. This dataset and associated documentation are available at the UCI KDD repository [Bay]. 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 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, i.e. that achieved profits substantially lower than \$10560. This fact indicates that the individuals in the KDD'98 dataset have already been filtered to be a reasonable set of targets. The task for any cost-sensitive learning and decision-making method is to improve upon the already good performance of the unknown method that has already been 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 cashflows relative to a baseline wealth of \$0, while some costs are counterfactual opportunity costs. Accordingly, our formulation of the problem is in terms of benefits instead of costs. The optimal predicted label for example x, i.e. the optimal decision whether or not to solicit 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 reasonable to replace $B(1,1,x)$ by c. 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) to obtain an optimal labeling, we need to estimate conditional class membership probabilities accurately. Costs 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 discuss methods for estimating costs and probabilities that can be applied in a wide variety of domains.

4 Estimating class membership probabilities

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.

We use the C4.5 decision tree learning method [Qui93] with pruning disabled to obtain scores that are usefully correlated with true class membership probabilities. In the KDD'98 domain, all examples in the training set are used as training data for C4.5, with the following seven fields as attributes:

- `income`: household income code (range 1–8)
- `firstdate`: date of first gift
- `lastdate`: date of most recent gift
- `pgift` = `ngiftall/numprom`: number of gifts/number of promotions received
- `RFA_2F`: frequency code (range 1–4)
- `RFA_2A`: amount of last gift code (range A–G)
- `PEPSTRFL`: RFA (recency, frequency, amount) star status (X or blank).

The derived attribute `pgift` is well-defined because `numprom` is never zero, since all records in the dataset concern people who have donated at least once in the past. Since our research for this paper is not concerned with feature selection, our choice of attributes is fixed and based informally on a KDD'99 winning submission [GM99].

When classifying test examples, by default C4.5 assigns the raw training frequency $p = k/n$ as the score of any example that is assigned to a decision tree 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: The C4.5 algorithm tries to make leaves homogeneous, so observed frequencies are systematically shifted towards zero and one. This problem has been noted by Walker [Wal92] and others.
2. High variance: When the number of training examples associated with a leaf is small, observed frequencies are not statistically reliable.

Pruning methods [EMS97] can in principle alleviate problem (2) by removing leaves that contain too few examples. However, the current C4.5 pruning method is not suitable for unbalanced datasets, because it is based on error rate minimization, not cost minimization. On the KDD'98 dataset this method generates a pruned tree that is a single leaf. Since the base rate of positive examples $P(j = 1)$ is about 5%, the error rate of the single leaf tree is only 5%, but this tree is useless for estimating example-specific conditional probabilities $P(j = 1|x)$. In general trees pruned by C4.5 are not useful for decision-making when the cost of misclassifying a rare true positive example is much higher than the cost of misclassifying a common true negative example.

The standard C4.5 pruning method is not alone in being inappropriate for cost-sensitive tasks. Quinlan's latest decision tree learning method, C5.0, and CART [BFOS84] also do pruning based on error minimization. Both C4.5 and C5.0 have rule set generators that are a commonly used alternative to pruning [Qui93]. Given a decision tree, these methods produce a set of rules that is typically simpler and more accurate than the original tree. However, like pruning, these methods are based on error minimization, so they are not suitable for highly cost-sensitive applications.

Given the unsuitability of standard methods for pruning or otherwise restructuring decision trees, we choose instead to attempt to improve the accuracy of decision tree probability estimates directly. The choice to do no pruning is supported by the results of Bradford *et al.* [BKK$^+$98], 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 (Section 7.3, [BK99]).

4.1 Improving probability estimates by binning

The binning or histogram method is a simple non-parametric approach to probability density estimation [Bis95]. Given a set of examples for which an attribute z is measured, we obtain a histogram by dividing the z axis into a number of bins. The conditional probability of membership in class c given z is approximated by the fraction of examples in the bin containing z that belong to c.

Instead of using a raw attribute to separate examples into bins, we use the C4.5 leaf frequency score of each example. Given a test example, we compute its raw C4.5 score and place it in a bin according to this score. The binned conditional probability estimate for the test example is then the fraction of true positive training examples in this bin.

In order to obtain binned estimates that do not overfit the training data, we partition the training set into two subsets. One subset is given to C4.5 for use in learning a decision tree and the other subset is used for validation, i.e. for the binning process. The two subsets are stratified, meaning that the proportion of positive examples in each subset is fixed to be identical. The subset used for training, called `C4.5train`, contains 70% of all training examples. The other subset, `C4.5val`, contains the remaining 30%. More training examples are assigned to `C4.5train` because learning the tree involves making many more choices than setting the binned probabilities.

Concretely, we train C4.5 using `C4.5train` and then apply the resulting decision tree to each example in `C4.5val`. We then sort these examples according to their raw decision tree scores, and divide them into ten equal-sized bins. For each bin, we compute an unbiased estimate of the conditional probability that an example is positive given that its decision tree score places it in this bin, by averaging the true 0/1 labels for the examples from `C4.5val` in that bin.

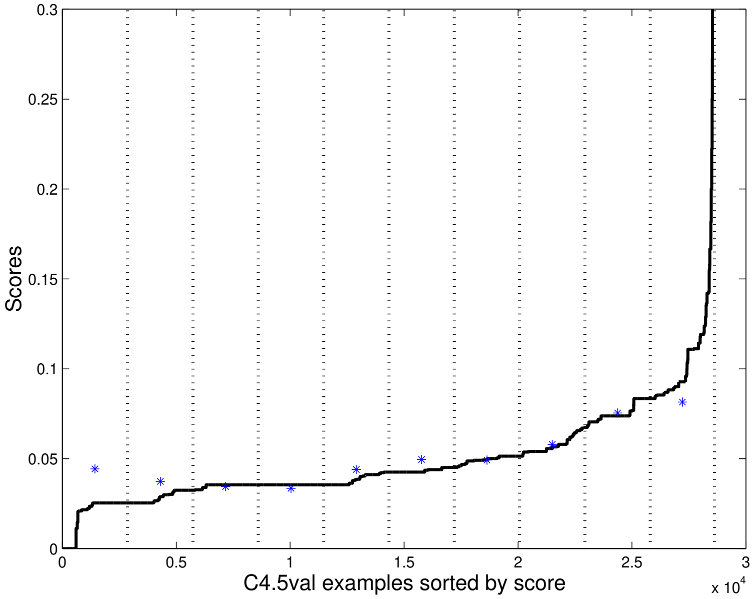

Figure 1: Binning. The solid line is the C4.5 decision tree score for the examples in `C4.5val`. The vertical dashed lines show the separation into ten bins. The stars are the average probability of membership in the positive class for each bin.

Figure 1 shows the results of executing this procedure. The solid line is the output of the decision tree learned by C4.5, for the examples in `C4.5val`. The vertical dashed lines show how these scores are separated into bins. The stars are the average probability of donation for each bin, i.e. the binned score for the examples in that bin. Note that C4.5 considerably overestimates the scores of the examples in the rightmost bin. Similarly, it underestimates the scores of the examples in the leftmost bin. This phenomenon is expected because C4.5 tries to make the leaves of the decision tree homogeneous. A separate validation set is necessary because averaging the 0/1 labels of the examples from `C4.5train` in a bin would not eliminate the overestimation and underestimation bias.

In order to obtain conditional probability estimates for all the examples in the entire dataset, we map the raw C4.5 score of each example into a bin. This mapping is obtained by determining the bin for which the raw score falls between the upper and lower bounds of the bin. The binned score for any example is the average probability of donation in the bin to which the example is mapped.

The number of different probability estimates that binning can yield is limited by the number of alternative bins. This number, ten in our experiments, must be small in order to reduce the variance of the binned probability estimates, by increasing the number of 0/1 values from the validation set that are averaged for each bin. Therefore, 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 the bias.

4.2 Improving probability estimates by smoothing

As discussed by Provost and Domingos [DP00] and others, one way of improving the probability estimates given by decision trees is to make these estimates smoother, i.e. to adjust them to be less extreme. Provost and Domingos suggest using the Laplace correction method. For a two-class problem, this method replaces the conditional probability estimate $p = \frac{k}{n}$ by $p' = \frac{k+1}{n+2}$ where k is the number of positive training examples associated with a leaf and n is the total number of training examples associated with the leaf.

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. In general, one should consider the overall average probability of the positive class, i.e. the base rate, when smoothing probability estimates. 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 and m is a parameter that controls how much scores are shifted towards the base rate. This smoothing method is called m-estimation [Cus93]. For example, if a leaf contains only two training examples, one of which is positive, the raw C4.5 decision tree score of any example assigned to this leaf is 0.5. The smoothed score with $m = 10$ and $b = 0.05$ is

$$p' = \frac{1 + 0.05 \cdot 10}{2 + 10} = \frac{1.5}{12} = 0.1250,$$

while the smoothed score with $m = 100$ and $b = 0.05$ is

$$p' = \frac{1 + 0.05 \cdot 100}{2 + 100} = \frac{6}{102} = 0.0588.$$

As m increases, observed training set frequencies are shifted more towards the base rate. Previous papers have suggested choosing m by cross-validation. Given a base rate b, we suggest using m such that $bm = 5$ approximately. This heuristic is similar to the rule of thumb that a chi-squared goodness of fit test is reliable if the number of examples in each cell of the contingency table is at least five.

Figure 2 shows the smoothed scores with $m = 100$ 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,

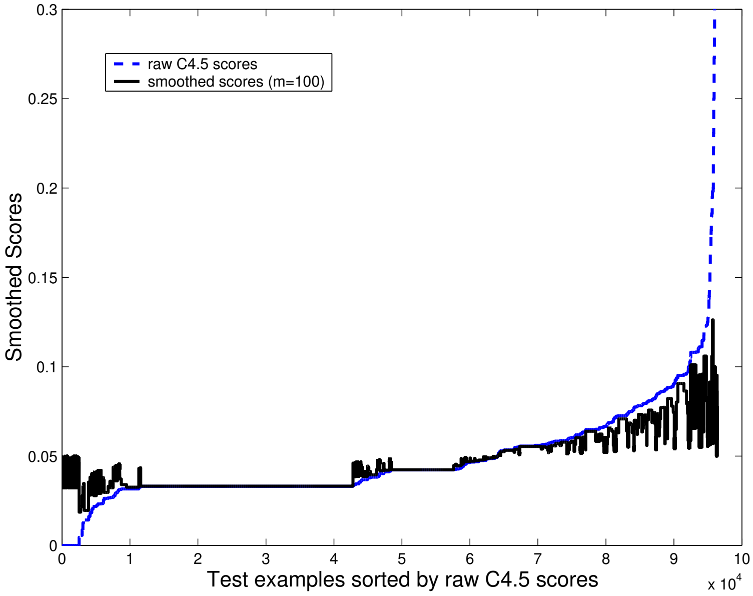

Figure 2: Smoothed scores and raw C4.5 scores for test examples sorted by raw score.

smoothed scores range from 0.0187 to 0.1262. The scores of two sets of examples are essentially unaffected by smoothing. These sets appear as horizontal stretches in the line of C4.5 scores. They correspond to two leaves in the decision tree that contain a number of training examples much larger than m.

4.3 Improving probability estimates by early stopping

As discussed before, C4.5 without pruning tends 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. 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, i.e. a leaf with few training examples, contains enough examples to induce a statistically reliable probability estimate, then assigning this estimate to a test example associated with the small 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.

This method of improving conditional probability estimates is called early stopping because when classifying an example, we stop searching 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 $bv = 5$. We choose $v = 100$ for all our experiments, but the example

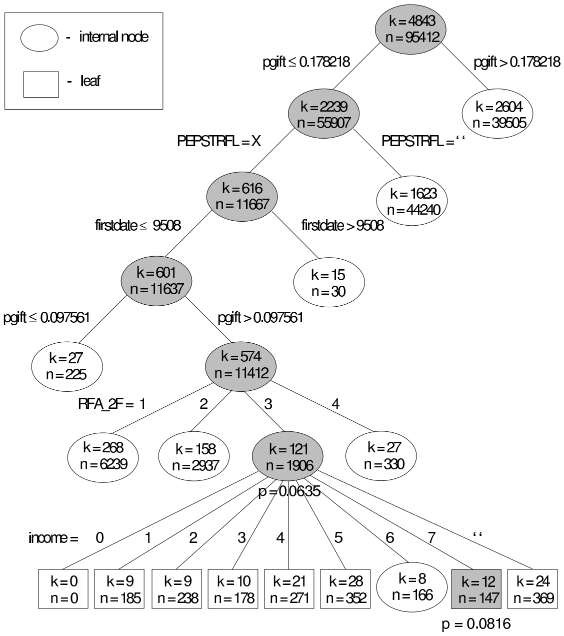

Figure 3: Part of the decision tree generated by C4.5 used to classify a test example with `income`=7, `firstdate`=9202, `lastdate`=9603, `pgift`=0.170213, `RFA_2F`=3, `RFA_2A`=E, `PEPSTRFL`=X. Nodes in grey are on the path that is followed from the root to the leaf when the example is classified.

in the remainder of this section uses $v = 200$ in order to have a smaller decision tree.

Figure 3 shows part of the decision tree generated by C4.5 from the entire KDD'98 training set. Without pruning, this tree has over 1000 leaves. The grey nodes are on the path that is followed from the root to the leaf when the following test example is classified: `income`=7, `firstdate`=9202, `lastdate`=9603, `pgift`=0.170213, `RFA_2F`=3, `RFA_2A`=E, `PEPSTRFL`=X. Note that the leaf contains only 147 examples. If we do early stopping with $v = 200$, we use the score of the parent of the leaf, which contains 1906 examples, providing a more reliable probability estimate. The estimated probability for the test example is changed from 0.0816 to 0.0635.

By eliminating nodes that have few training examples, early stopping effectively creates the decision tree shown in Figure 4. 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. Early stopping is not equivalent to any type of pruning, because pruning eliminates all the children of a node simultaneously, while early stopping may eliminate some children and keep others, depending on the number of training examples associated with each child. Intuitively, early stopping 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.

Grafting is a variant of pruning used by C4.5 that can eliminate internal nodes as well as leaves [EMS97]. Although early stopping is reminiscent of grafting, the

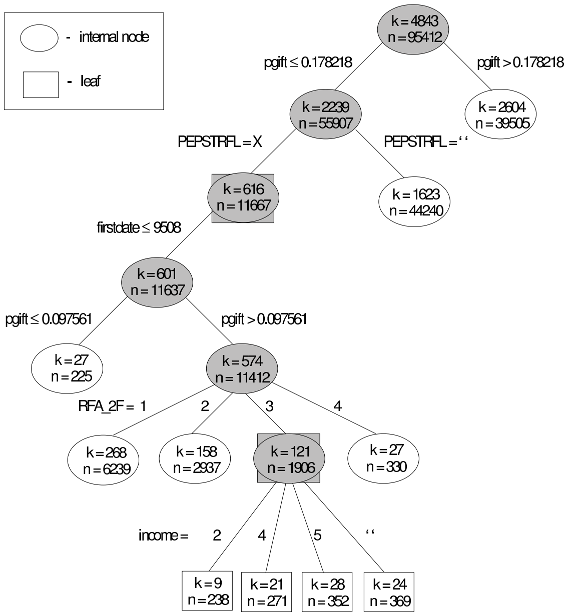

Figure 4: Decision tree obtained by early stopping with $v = 200$ from the tree in Figure 3.

two methods have a fundamental difference. Grafting replaces a subtree by one of its branches. This operation completely eliminates switching based on the attribute tested at the root of the subtree. Early stopping, on the other hand, eliminates switching based on some values of an attribute, those values for which too few training examples are available, but keeps switching based on the other values of the same attribute. In other words, both pruning and grafting either split based on all the different values of an attribute, or based on none. Early stopping allows splitting for some but not all of the values of an attribute.

Figure 5 shows the early stopping scores with $v = 100$ of the test set examples sorted by their raw C4.5 scores. The jagged lines in the chart show that many scores are changed significantly by early stopping. As with smoothing, the scores of two subsets of examples, which appear as horizontal lines in the chart, are not affected by early stopping. These subsets correspond to two leaves in the decision tree that contain a number of training examples much larger than v. Overall, the range of scores is reduced as with smoothing, but not as much. The minimum early stopping score is 0.0045 while the maximum is 0.2414.

4.4 Combining smoothing and early stopping

Early stopping effectively eliminates from a decision tree nodes that yield probability estimates that are not statistically reliable because they are based on a small sample of training examples. But since C4.5 tries to make decision tree nodes homogeneous, even probability estimates that are based on many training examples tend to be too high or too low. As explained in Section 4.2, smoothing can compensate for

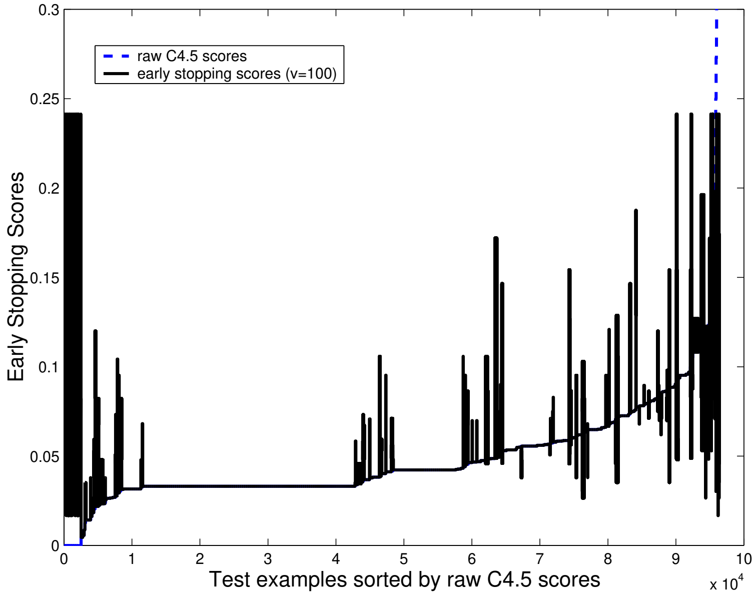

Figure 5: Early stopping scores and raw C4.5 scores for test examples. Examples are sorted by raw C4.5 score.

this bias by shifting estimates towards the overall average probability. Therefore we investigate the combination of smoothing and early stopping.

Figure 6 shows smoothed early stopping scores with m=100 and v=100 for the KDD'98 test set examples sorted by their raw C4.5 scores. Comparing this chart with the one in Figure 5 shows that the smoothed early stopping scores are less extreme, as expected. They range from 0.0187 to 0.1837.

5 Estimating donation amounts

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

When costs and probabilities are both unknown, 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, early stopping with smoothing, gives conditional probabilities in the narrow range from 0.0187 to 0.1837. Estimating donation amounts is easier because past amounts are excellent predictors

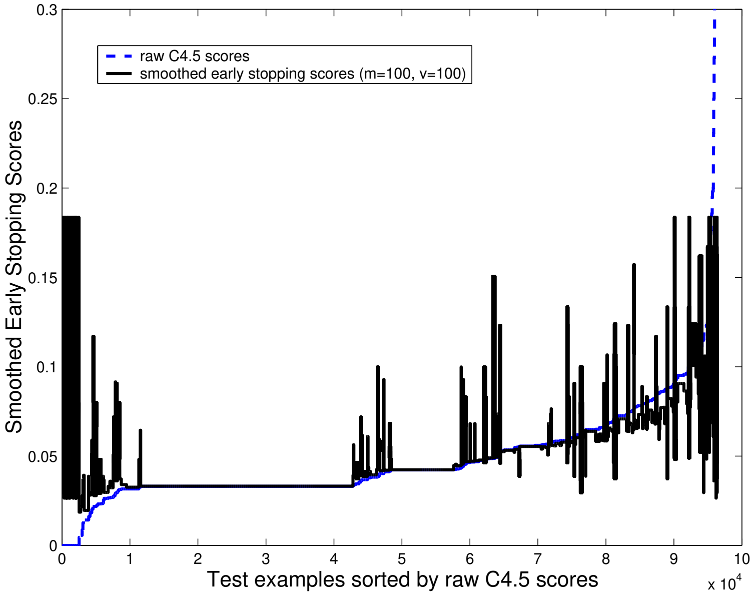

Figure 6: Smoothed early stopping scores and raw C4.5 scores for test examples. The examples are sorted by raw C4.5 score.

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.

We compare three different methods for obtaining donation amount estimates. The first method uses the average donation amount $\bar{y}$ of known donors, for individuals for whom the actual donation amount is unknown. For donors in the training set whose actual donation amount is known, this method uses the actual amount. Note that 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.

The second method of estimating donation amounts uses 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 two derived attributes:

- `lastgift`: dollar amount of most recent gift,
- `pgift` = `ngiftall/numprom`: number of gifts/number of promotions received,
- `ampergift`: average gift amount in responses to the last 22 promotions.

As mentioned above, the topic of this paper is not variable selection, so we somewhat arbitrarily choose these three attributes based on previous work. Also as mentioned above, `pgift` is well-defined because `numprom` is always at least one. 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.

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 7. 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.

As we have explained previously [Elk00], 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 [Hec79]. 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

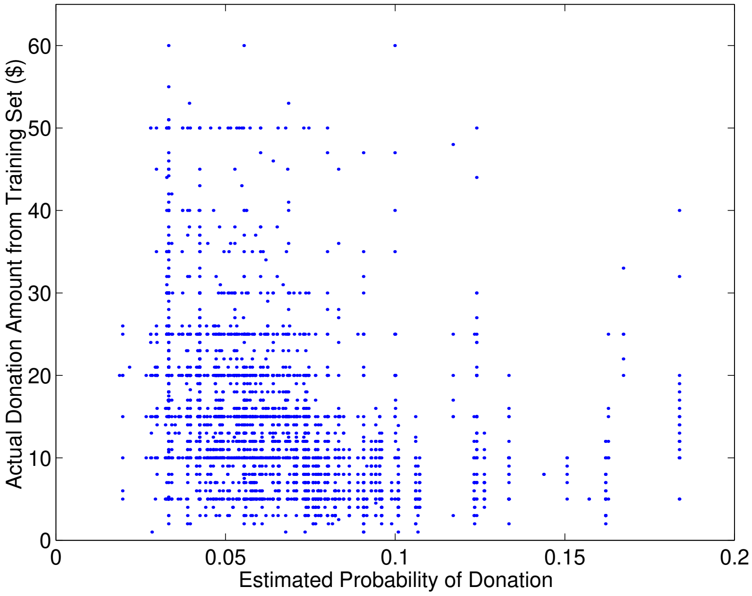

Figure 7: Actual donation amount versus estimated probability of donation, for all donors in the training set. A negative correlation between donation amount and probability of donation is visible.

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 [Hec79].

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

6 Choosing a threshold for decisions

As seen in Section 3, in the charitable donations domain the optimal policy is to assign the predicted label $i = 1$ to an example x if and only if $P(j = 1|x)y(x) > 0.68$. Here $y(x)$ is the estimated amount of the donation that x might contribute, $P(j = 1|x)$ is the probability that x actually donates, and \$0.68 is the cost of mailing a solicitation.

Using \$0.68 as a threshold for making decisions is optimal only if the estimates of $y(x)$ and $P(j = 1|x)$ are unbiased. As an attempt to compensate for errors in estimated donation probabilities and amounts, we can replace the \$0.68 threshold by another threshold c. Heuristically, Equation 3 in the optimal decision-making strategy is changed to

$$P(j = 1|x)y(x) > c. \tag{4}$$

The threshold c is determined empirically by first ranking the examples in the training set according to the product $P(j = 1|x)y(x)$, and then finding the threshold that yields maximum profit. This threshold is then used when test examples are labeled.

The relationship between attained profit and threshold value is illustrated in Figure 6. For each possible value of the threshold c, the chart shows the profit obtained by sending solicitations to all examples x in the training set such that $P(j = 1|x)y(x) > c$.

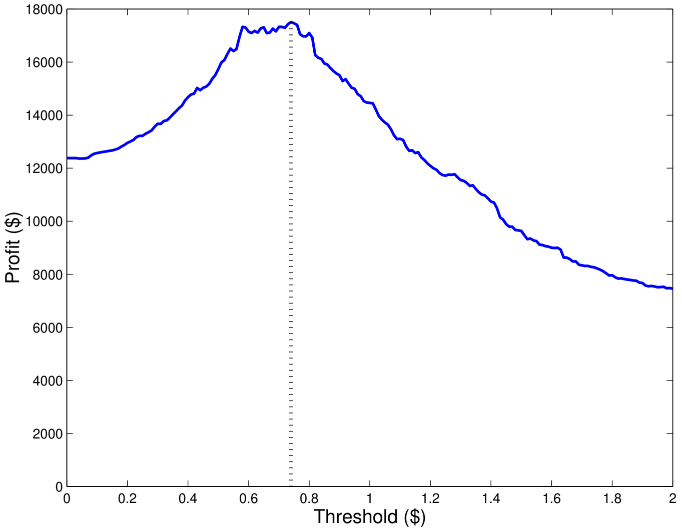

Figure 8: Dependence of the attained profit for the training set on the threshold value. The optimal threshold value, marked by the dashed line, is \$0.74.

For low values of c, although the revenue from donations is high, the cost of mailing solicitations is also high because almost every individual is solicited. On the other hand, for high values of c, too few people are solicited. In this case, although mailing costs are lower, the total profit is low because many donors do not receive a solicitation. In the case shown in the chart, the threshold that is optimal for the training set is \$0.74, but the standard threshold \$0.68 yields almost the same total profit.

Changing a single number, the decision-making threshold, is mathematically sufficient to compensate for biases in estimating $P(j = 1|x)$ and $y(x)$ only if the estimated product $P(j = 1|x)y(x)$ is a monotonically increasing function of the real product. A perfectly monotonic relationship is not likely to be exactly true. In general, adjusting the threshold c cannot compensate completely for errors in estimated donation probabilities and amounts, but may still be useful in practice.

The chart in Figure 6 is similar to a lift curve, also called a gains chart. The major difference is that lift curves are based on probabilities, i.e. $P(j = 1|x)$, instead of on expected revenue, i.e. $P(j = 1|x)y(x)$. One conventional approach to cost-sensitive learning and decision-making is to learn an estimator $s(x)$ of $P(j = 1|x)$, and then to select a threshold d such that an individual x is solicited if and only $s(x) > d$. The choice of threshold d heuristically takes into account the cost matrix and also compensates for the fact that $s(x)$ is typically not well-calibrated as an estimate of $P(j = 1|x)$. A major point of this paper is that any policy of this type is usually suboptimal. In any marketing domain, it is rational to solicit a person whose probability of responding is low, if the expected value of their response, if they do respond, is high. Conversely, it is irrational to solicit someone whose response probability is high, if the expected value of their response is low.

Note that when costs or benefits are different for different individuals, then to make rational decisions we need unbiased estimates of true example-specific class probabilities. Numerical score that are correlated with true probabilities, but not calibrated well, are inadequate. On the other hand, when costs or benefits are the same for all individuals, and there are only two possible classes, then any monotonic transformation of an estimator for $P(j = 1|x)$ is just as useful as a well-calibrated version of the same estimator, because changing the decision threshold can compensate for any calibration error.

7 Experimental results

The previous three sections have discussed alternative methods for each of three subproblems:

(a) estimating example-specific class probabilities,

(b) estimating example-specific costs or benefits, and

(c) setting a threshold for making decisions.

We also have two alternative general methods for cost-sensitive learning: MetaCost and direct cost-sensitive decision-making. We label this last choice (d) and present experimental results for each possible combination of alternatives for (a), (b), (c) and (d). Each combination is one experimental trial.

For each trial, we report the number of people that are solicited, the number of donors that are reached, and the total profit achieved. We give these numbers for the training set and for the test set. The most important number, of course, is the total profit achieved on the test set. Tables 1, 2, 3, 4 and 5 show the results of trials using raw C4.5 scores, binned scores, smoothed scores, early stopping scores and smoothed early stopping scores, respectively.

For the MetaCost experiments, there are two alternative ways of measuring performance on the training set: (i) using the relabeling of training examples directly, or (ii) applying the classifier learned from the relabeling to the examples on the training set to obtain new labels. Since (i) is the same for direct cost-sensitive decision-making and for MetaCost, we report (ii) for the MetaCost experiments. The results of (ii) are better predictors of test set results, because they reflect the behavior of the classifier that is applied to test examples. A significant difference between the results of

			Training Set			Test Set		
Amount	Thresh.	Method	Mailed	Hit	Profit	Mailed	Hit	Profit
average	fixed	MetaCost	37535	2907	$10966	37640	2586	$5229
average	fixed	direct	37124	2526	$25966	37643	2592	$5284
average	adjusted	MetaCost	53435	3563	$12007	53638	3290	$6361
average	adjusted	direct	53371	3526	$28047	53572	3592	$6569
MLR	fixed	MetaCost	48747	2913	$15522	49311	2615	$10764
MLR	fixed	direct	49735	2997	$17261	50319	2648	$12469
MLR	adjusted	MetaCost	36570	2370	$16236	36805	2038	$10941
MLR	adjusted	direct	37554	2457	$17507	38080	2062	$11430
Heckman	fixed	MetaCost	48723	2900	$15455	49279	2617	$10768
Heckman	fixed	direct	49919	3008	$17246	50508	2662	$12493
Heckman	adjusted	MetaCost	36784	2376	$16135	37072	2054	$10940
Heckman	adjusted	direct	37718	2467	$17502	38237	2057	$11510

Table 1: Experimental results using raw C4.5 decision tree scores as probability estimates. The method achieving the highest profit on the test set is highlighted.

(i) and (ii) indicates that C4.5 is not able to find a decision tree that captures the relationship between the attributes and the result of the relabeling.

7.1 Statistical significance

When comparing the results of different trials, it is important to evaluate whether differences in attained profit are statistically significant. We can quantify significance roughly with a simple argument. 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 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 much 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 [Dom99]. 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 main 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.

			Training Set			Test Set		
Amount	Thresh.	Method	Mailed	Hit	Profit	Mailed	Hit	Profit
average	fixed	MetaCost	58061	3601	$9690	58510	3548	$7578
average	fixed	direct	56849	2383	$12612	58537	3553	$7586
average	adjusted	MetaCost	38319	2869	$10616	38447	2613	$4848
average	adjusted	direct	37722	2328	$24918	38505	2626	$5206
MLR	fixed	MetaCost	53517	2816	$14619	54344	2696	$12174
MLR	fixed	direct	54653	2876	$15595	55287	2711	$14068
MLR	adjusted	MetaCost	39261	2124	$14550	39588	1980	$11708
MLR	adjusted	direct	40235	2241	$15844	40678	2006	$13056
Heckman	fixed	MetaCost	55519	2858	$14676	56235	2752	$12543
Heckman	fixed	direct	56139	2883	$15625	56792	2727	$14176
Heckman	adjusted	MetaCost	59058	3048	$14956	59895	2926	$12520
Heckman	adjusted	direct	59625	3090	$15775	60246	2918	$14291

Table 2: Experimental results using binned decision tree scores as probability estimates. The method achieving the highest profit on the test set is highlighted.

Therefore, our experiments are designed so that each alternative for each of the choices (a), (b), (c), and (d) is tried while holding all other choices fixed. This experimental design allows us to investigate whether a particular alternative for (a) for example systematically yields a higher profit than other alternatives, regardless of what choices are made for (b), (c), and (d).

For (a), (b), (c), and (d) there are respectively five, three, two, and two choices, for a total of 60 choices. Consider for example choice (d). We have results for 30 pairs of trials where one trial uses MetaCost and the other trial uses the same choices for (a), (b), and (c), but uses direct cost-sensitive decision-making instead of MetaCost. Under the null hypothesis that the two methods are equally successful, one would expect MetaCost to appear superior in about $30 \cdot 0.5 = 15$ pairs, with a standard deviation of $\sqrt{30 \cdot 0.5 \cdot 0.5} = 2.7$ approximately if the pairs are independent.

In fact, in all 30 pairs, the test set profit achieved using MetaCost is lower. This result is highly significant statistically, whether or not the magnitude of the difference in individual trials is above or below $1090. We choose not to quantify the level of this statistical significance because doing so would require making assumptions that are certainly false. In particular, because all trials use the same training and test sets, the 30 pairs of trials are not statistically independent.

7.2 Comparing methods for estimating donation amounts

In all trials where the average donation is used as a fixed donation amount estimate, results on the test set are bad. These trials all yield profit on the test set significantly lower than that achievable trivially by classifying all test examples as positive.

Many trials show a huge difference in profit for the training and test sets, which seems surprising, given that the number of people solicited and the number of donors reached are approximately the same for both sets. However, consider the second line of Table 1. In the training set, the average donation of people who are solicited and who then donate is $20.27, but for the test set, the analogous average is only

			Training Set			Test Set		
Amount	Thresh.	Method	Mailed	Hit	Profit	Mailed	Hit	Profit
average	fixed	MetaCost	41057	2970	$10410	41191	2809	$5692
average	fixed	direct	40344	2239	$21975	41322	2828	$5766
average	adjusted	MetaCost	59182	3671	$9936	59574	3634	$7905
average	adjusted	direct	58850	3354	$23870	59592	3642	$7984
MLR	fixed	MetaCost	49155	2728	$15347	49823	2553	$12378
MLR	fixed	direct	49772	2740	$16009	50429	2555	$14100
MLR	adjusted	MetaCost	51404	2853	$15653	52132	2643	$12134
MLR	adjusted	direct	51826	2845	$16031	52571	2650	$13974
Heckman	fixed	MetaCost	50609	2639	$15042	51213	2522	$12610
Heckman	fixed	direct	51107	2678	$16010	51791	2507	$14397
Heckman	adjusted	MetaCost	56964	2985	$15092	57613	2854	$12789
Heckman	adjusted	direct	56949	2997	$16179	57572	2819	$14419

Table 3: Experimental results using smoothed decision tree scores as probability estimates. The method achieving the highest profit on the test set is highlighted.

$11.91. Other experiments using the average training set donation amount as a fixed estimate of unknown donation amounts exhibit similar discrepancies.

The discrepancies occur because the true donation values are used for the positive examples in the training set, but the average donation amount is used for all examples in the test set. On the training set, this process leads to the selection of people who donate larger amounts. But on the test set, because the donation estimate $y(x) = \bar{y}$ is the same for all examples and the probability of donation $P(j = 1|x)$ tends to be lower for large donors, the product $P(j = 1|x)y(x)$ tends to be lower for these people. Therefore, individuals that are likely to donate smaller amounts are selected. Both MetaCost and direct cost-sensitive decision-making are subject to this problem. These results confirm the claim that it is wrong to impute any fixed quantity as a donation estimate for test examples.

When linear regression (MLR) is applied to estimate donation amounts, results are significantly improved for both direct cost-sensitive decision-making and MetaCost. Heckman's procedure, which uses the probability estimates as an additional attribute in the linear regression, improves results further, except in one trial (raw, Heckman, adjusted, MetaCost) where profit is reduced by $1.

Although the improvement due to Heckman's procedure is systematic, it is only $385 on average. The average improvement is small because two of the attributes used in the original linear regression, namely `pgift` and `ampergift`, are highly correlated with the probability of making a donation. Indeed, `pgift` is the historical probability of a person responding, which unsurprisingly is highly correlated with the probability of a future response. Heckman's procedure is effectively already mostly implemented via these attributes. Nonetheless, the fact that improvement is systematic indicates that Heckman's procedure succeeds in correcting sample selection bias. We expect the beneficial impact of Heckman's procedure to be much greater whenever the effect of sample selection bias is not already mostly accounted for in a heuristic way.

			Training Set			Test Set		
Amount	Thresh.	Method	Mailed	Hit	Profit	Mailed	Hit	Profit
average	fixed	MetaCost	42107	2975	$8515	42353	2938	$6473
average	fixed	direct	41576	2442	$22597	42381	2943	$6489
average	adjusted	MetaCost	58248	3635	$9692	58505	3605	$8034
average	adjusted	direct	58126	3508	$24936	58521	3610	$8069
MLR	fixed	MetaCost	52501	2971	$14081	53370	2897	$12281
MLR	fixed	direct	53372	2988	$14709	54148	2918	$14020
MLR	adjusted	MetaCost	40175	2391	$14247	40658	2303	$12349
MLR	adjusted	direct	40908	2435	$15052	41576	2307	$12982
Heckman	fixed	MetaCost	54488	2952	$14412	55243	2861	$12502
Heckman	fixed	direct	55663	2967	$15093	56583	2908	$14416
Heckman	adjusted	MetaCost	56116	3055	$14459	56719	2967	$13081
Heckman	adjusted	direct	57130	3069	$15362	57997	2988	$14507

Table 4: Experimental results using early stopping scores as probability estimates. The method achieving the highest profit on the test set is highlighted.

7.3 Adjusting the threshold for making decisions

In many trials, adjusting the threshold for making decisions away from $0.68 is not beneficial. When the threshold is changed based on the training data, there is a serious risk of overfitting this data. Results for the test set can then be significantly worse.

If conditional probabilities and donation amounts are estimated in an unbiased way, then adjusting the threshold is unnecessary. For example, in the trials using binned probability estimates described in Table 2 that use Heckman's procedure, the adjusted threshold is $0.66. This number is very close to $0.68, which is the optimal threshold if probability and cost estimates are unbiased. For this reason, the results with fixed and adjusted thresholds are very similar. We conclude that adjusting the threshold is useful only when probability and cost estimates are biased, and in particular when Heckman's procedure is not used.

7.4 Comparing methods for estimating probabilities

The results of trials using binned scores are much better than those of trials using unmodified C4.5 scores, on average by $1250. We attribute this improvement to the fact that binning corrects C4.5 scores that are extreme underestimates or overestimates. Smoothing and early stopping are both systematically slightly better than binning. Compared to binning, smoothing and early stopping improve the accuracy of probability estimates without reducing their resolution so much, i.e. without reducing the number of distinct probability estimates down to just the number of bins. Combining smoothing and early stopping yields slightly better results than either method separately, on average $226 more than early stopping by itself and $313 more than smoothing by itself.

			Training Set			Test Set		
Amount	Thresh.	Method	Mailed	Hit	Profit	Mailed	Hit	Profit
average	fixed	MetaCost	42783	3003	$8727	43118	2952	$6669
average	fixed	direct	42124	2335	$21519	43151	2957	$6671
average	adjusted	MetaCost	59016	3655	$9610	59432	3635	$7863
average	adjusted	direct	58787	3414	$24177	59461	3642	$7913
MLR	fixed	MetaCost	51064	2835	$14680	51960	2743	$12625
MLR	fixed	direct	52039	2834	$14951	52957	2798	$14558
MLR	adjusted	MetaCost	65885	3548	$14462	66687	3530	$12386
MLR	adjusted	direct	66951	3618	$15043	67758	3548	$14462
Heckman	fixed	MetaCost	52296	2831	$14564	53172	2772	$12860
Heckman	fixed	direct	53716	2859	$14951	54601	2803	$14651
Heckman	adjusted	MetaCost	58516	3185	$14703	59286	3091	$12649
Heckman	adjusted	direct	59020	3152	$15233	59910	3048	$14608

Table 5: Experimental results using smoothed early stopping scores as probability estimates. The best result on the test set is highlighted.

7.5 MetaCost versus direct cost-sensitive decision-making

MetaCost performs consistently less well than direct cost-sensitive decision-making. The best result obtained with MetaCost is $13081, while the best result obtained with the direct method is $14651, which is statistically indistinguishable from 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.

8 Conclusions

The main contributions of this paper are the following:

- We explain a general method of cost-sensitive learning that performs systematically better than MetaCost in our experiments.
- We provide a solution to the fundamental problem of costs being different for different examples, and unknown in general. Our solution includes a solution to 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, not small datasets with arbitrary cost data as in previous research.

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. Using a more sophisticated regression method for estimating donation amounts, we already have preliminary results that are better than those of the winners of the KDD'98 and KDD'99 contests.

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. For example, if future work shows that bagging [Bre96] is useful for probability estimation, then MetaCost and our method will both benefit.

References

[Bay] S. D. Bay. UCI KDD archive.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olsen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[Bis95] C. Bishop. *Neural Networks for Pattern Recognition*, chapter 2. Clarendon Press, Oxford, UK, 1995.

[BK99] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999.

[BKK$^+$98] 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.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[Cus93] James 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.

[Dom99] Pedro 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.

[DP00] P. Domingos and F. Provost. Well-trained PETs: Improving probability estimation trees. CDER Working Paper #00-04-IS, Stern School of Business, New York University, NY, NY 10012, 2000.

[Elk00] Charles 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.

[EMS97] 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.

[GM99] Jim Georges and Anne H. Milley. KDD'99 competition: Knowledge discovery contest report. Available at `http://www-cse.ucsd.edu/users/elkan/kdresults.html`, 1999.

[Hec79] J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47:153–161, 1979.

[PF99] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 1999. To appear.

[Qui93] J. Quinlan. *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann, 1993.

[Tur00] Peter Turney. Cost-sensitive learning bibliography. Institute for Information Technology, National Research Council, Ottawa, Canada, 2000. `http://ai.iit.nrc.ca/bibliographies/cost-sensitive.html`.

[Wal92] Michael G. Walker. Probability estimation for classification trees. Technical Report KSL-92-01, Knowledge Systems Laboratory, Stanford University, 1992.