Technical Report ML-TR-44, Department of Computer Science, Rutgers University
August 2, 2001

The Effect of Class Distribution on Classifier Learning: An Empirical Study

Gary M. Weiss GMWEISS@ATT.COM
AT&T Labs, 30 Knightsbridge Road
Piscataway, NJ 08854 USA

Foster Provost FPROVOST@STERN.NYU.EDU
New York University, Stern School of Business
44 W. 4th St., New York, NY 10012 USA

Abstract

In this article we analyze the effect of class distribution on classifier learning. We begin by describing the different ways in which class distribution affects learning and how it affects the evaluation of learned classifiers. We then present the results of two comprehensive experimental studies. The first study compares the performance of classifiers generated from unbalanced data sets with the performance of classifiers generated from balanced versions of the same data sets. This comparison allows us to isolate and quantify the effect that the training set's class distribution has on learning and contrast the performance of the classifiers on the minority and majority classes. The second study assesses what distribution is "best" for training, with respect to two performance measures: classification accuracy and the area under the ROC curve (AUC). A tacit assumption behind much research on classifier induction is that the class distribution of the training data should match the "natural" distribution of the data. This study shows that the naturally occurring class distribution often is not best for learning, and often substantially better performance can be obtained by using a different class distribution. Understanding how classifier performance is affected by class distribution can help practitioners to choose training data—in real-world situations the number of training examples often must be limited due to computational costs or the costs associated with procuring and preparing the data.

1. Introduction

Machine learning and data mining methods, and the acceptance of these methods, have advanced to the point where they are commonly being applied to very large, real-world problems. Addressing these real-world problems has focused attention, and research, on problems that were once only rarely considered. For example, many research papers and several workshops have recently been directed at the problems of learning from data sets with unbalanced class distributions and where the costs of misclassifying examples is non-uniform. In order for the acceptance and the use of these methods to grow, research must continue to address the practical concerns that arise when dealing with real-world data sets.

Our research is motivated by the fact that obtaining data in a form suitable for learning is often costly and that learning from large data sets may also be costly. The costs associated with creating a useful data set include the cost of obtaining the data, cleaning the data, transporting/storing the data, labeling the data, and transforming the raw data into a form suitable for learning. The costs associated with learning from the data involve the cost of computer hardware, the "cost" associated with the time it takes to learn from the data, and the "opportunity cost" associated with not being able to learn from extremely large data sets due to limited com-

putational resources. A more comprehensive list of the costs associated with inductive learning is provided by Turney (2000).

Given these costs, it is often necessary to limit the size of the training set. As a consequence, a common question asked at the start of a data mining project is: how many data records should I use and what is the best class distribution? To minimize the impact that the limited training-set size has on classifier performance it is essential that the training data be chosen carefully. One important choice is the proper class distribution of the training set. It has been a tacit assumption in much machine learning research that the naturally occurring class distribution is best for learning. However, this assumption has been coming under increased scrutiny, partly because many of the data sets now being learned from have a high degree of class imbalance. In these cases, learning from the naturally occurring class distribution produces classifiers that perform poorly on minority-class examples.

A simple example illustrates why choosing the class distribution carefully is important. Imagine that there exist 30 million examples that could be procured for training, of which 10 million belong to class A and 20 million belong to class B. If we limit the training set to only 300,000 examples, then we eliminate the need to clean and store 99% of the records, and we will reduce the learning time dramatically. Should the limited training set contain 100,000 class A examples and 200,000 class B examples, so that it matches the natural distribution? While the answer depends on the particular domain, our results show that the answer is often "no".

In this article we describe and analyze the results from a comprehensive set of experiments designed to investigate the effect that the class distribution of the training set has on classifier performance. We evaluate classifier performance using two performance measures and show that in many cases the naturally occurring class distribution is *not* best for learning and, consequently, when the training-set size needs to be restricted, a class distribution other than the natural class distribution should be chosen. We characterize how the optimal class distribution relates to the naturally occurring class distribution and answer a number of basic questions about how the class distribution of the training set affects learning.

2. Background

In this section we present important background information. Section 2.1 describes related work and the relationship of our research to this work. Section 2.2 answers several fundamental questions concerning the effect of class distribution on learning. Section 2.3 then describes how to adjust a classifier to compensate for training using a distribution other than the naturally occurring class distribution.

2.1 Related Work

Our research investigates the effect that class distribution has on learning when the size of the training set must be limited due to cost concerns. That is, we investigate the question: if only *n* training examples are permitted, what is the best class distribution to use for training? There is little existing research that addresses this specific question. A few studies (Catlett, 1991; Chan & Stolfo, 1998) have examined some aspects of the relationship between training class distribution and classifier performance when the training-set size is held fixed, but have been quite limited in scope. These studies analyzed only a few data sets, which made it difficult to draw general conclusions and focused much of their attention on other issues, rather than attempting to provide a thorough study of the effect of class distribution on classifier learning. Furthermore, these studies did not adjust the induced classifiers to take into account the changes made to the training

set's class distribution. This improperly biases the classifier, and, as we show later, can significantly reduce the accuracy of a classifier. In our research we analyze 25 data sets and adjust the induced classifiers to account for the changes made to the class distribution.

While there is little research that investigates the effect of class distribution on classifier performance when the training set size must be limited, there is considerable research on a related topic: how to build "good" classifiers when the class distribution of the data is highly unbalanced and it is costly to misclassify minority-class examples (Japkowicz et al. 2000). Under these conditions classifiers that optimize for accuracy will tend to generate trivial models that almost always predict the majority class. Research for handling highly unbalanced data sets has in part focused on modifying the class distribution of the training set. In this case, the question is what is the best class distribution to use to form a classifier that performs well for some performance measure other than accuracy. Clearly, our research has some overlap with the research for learning from unbalanced data sets since both deal with changing the class distribution of the training set with the goal of improving classifier performance. However, we are concerned with the situation where there are costs associated with procuring the training data and/or learning from the training data, and therefore it is important to limit the size of the training set. For most research on dealing with unbalanced data sets the motivation for changing the class distribution is to build a more effective classifier from a fixed set of examples, not to minimize costs associated with acquiring or learning from the data, nor to maximize performance with respect to some reduced number of examples.

There are two basic methods for dealing with class imbalance by altering the class distribution of a data set: under-sampling, which eliminates examples in the majority class, and over-sampling, which replicates examples in the minority class (Breiman, et al. 1984; Kubat & Matwin, 1997). These methods cause the class distribution to become less unbalanced. However, both methods have known drawbacks. Under-sampling throws out potentially useful data while over-sampling *increases the size of the training set* and hence the time to build a classifier. Furthermore, since over-sampling typically makes exact copies of minority class examples, overfitting is more likely to occur—leaves/rules that appear to perform well may be induced to cover one (replicated) example.

Recent research has focused on improving these basic methods. Kubat and Matwin (1997) developed an under-sampling strategy that intelligently removes majority examples by removing only those majority examples that are "redundant" or that "border" the minority examples—figuring they may be the result of noise. Chawla et al. (2000) combine under-sampling and over-sampling methods, and, instead of over-sampling by replicating minority-class examples, they form new minority class examples by interpolating between several minority-class examples that lie close together. Thus, they avoid the overfitting problem and cause the decision boundaries for the minority class to spread further into the majority-class space. Clearly some of these techniques, especially the under-sampling techniques, could be used to reduce the training set size while minimizing the impact on classifier performance. However, even the existing research that reduces the training set size via under-sampling is generally not motivated by a desire to reduce the training set size. For example, the research by Kubat and Matwin (1997) motivates the use of under-sampling because "… adding examples of the majority class to the training set can have a detrimental effect on the learner's behavior."

Chan and Stolfo (1998) take a slightly different approach when learning from an unbalanced data set. They first run preliminary experiments to determine the best class distribution for learning (with respect to a specific cost function) and then generate multiple training sets with this class distribution. This is accomplished, in most cases, by including all of the minority-class

examples and some of the majority-class examples in each training set. They then run a learning algorithm on each of the data sets and combine the generated classifiers to form a composite learner. This method ensures that all of the available training data are used, since each majority-class example will be found in at least one of the training sets.

Changing the class distribution is not the only way to improve classifier performance when learning from unbalanced data sets. For data sets with a highly unbalanced class distribution, the cost of misclassifying a minority-class (positive) example is typically much greater than the cost of misclassifying a majority-class (negative) example. Consequently, cost-sensitive learning methods, which form classifiers that minimize cost instead of error rate, can be used to handle unbalanced data sets by factoring in these costs. Cost-sensitive learning methods appear to us to be a more direct and appropriate method for dealing with class imbalance than artificially modifying the class distribution via under-sampling or over-sampling, *if the cost of acquiring and learning from the data is not an issue*. To quote one of the cost-sensitive learning papers that considers this question, "All of the data available can be used to produce the tree, thus throwing away no information, and learning speed is not degraded due to duplicate instances" (Drummond & Holte 2000, page 239).

In this article we analyze the effect of class distribution on learning generally—not just for unbalanced data sets. We use this analysis to focus on the situation where the training set size must be limited. In our experiments we choose training set sizes that are small enough that we can achieve the desired class distributions solely by removing examples belonging to the minority class and/or the majority class. In no instance do we duplicate examples or select examples based on any criteria other than the class value.

2.2 Understanding the Role of Class Distribution on Learning

We now provide a qualitative description of how class distribution influences learning by answering two very basic questions: why do classifiers perform worse on the minority class than on the majority class and why is one training distribution better for learning than another. In addressing the first question we also answer the more general question of why classifiers perform *differently* for the minority and majority classes. Note that throughout this article we consider only two-class problems where the minority class is considered to be the positive class.

2.2.1 Why Do Classifiers Perform Worse on the Minority Class?

It is conventional wisdom that classifiers tend to perform worse on the minority class than on the majority class. As our experimental results in Section 4 show, this conventional wisdom is justified by two observations. The first observation is that the classification "rules" (decision-tree leaves, etc.) that predict the minority class tend to have a much higher error rate than those that predict the majority class. The second observation is that test examples belonging to the minority class are misclassified more often than test examples belonging to the majority class. Note that these observations are very different (and in fact "opposite"): a poor-performing rule that predicts the *minority class* (i.e., frequently classifies majority-class test examples as belonging to the minority class) degrades the classification performance of the test examples belonging to the *majority class*.

We give several reasons for this observed behavior. These reasons provide some basic but much needed understanding of how class distribution affects classifier performance. Minority-labeled classification rules perform worse than their majority-labeled counterparts in part for a very simple, but subtle and often overlooked reason: *the test set contains more majority-class*

examples than minority-class examples. Because of this test-distribution effect, all else being equal, the classification rules that predict the minority class will perform worse than those that predict the majority class. To see this, imagine a *randomly generated and randomly labeled* decision tree that is evaluated on a test set with two classes, where the class ratio in the test set is 9:1. In this situation the leaves predicting the minority class have an expected error rate of 90% while the leaves predicting the majority class have an expected error rate of only 10%! A second reason for why minority-labeled rules perform so poorly is provided in Section 4, where we show that these rules are generally formed from fewer training examples than the majority-labeled rules.

Classifiers also tend to perform worse at classifying the minority-class test examples. One reason is due to the fact that the class "priors" (the marginal probabilities of the classes) in the natural training distribution are biased strongly in favor of the majority class. Some learners explicitly factor in these class priors, which biases the learning process by causing the majority class to be predicted more often than it otherwise would have been. This results in improved performance on the majority-class test examples but degraded performance on the minority-class test examples. For example, because decision trees must handle all feature values, even if they are not observed in the training data, it is possible for a leaf in the decision tree to cover no training examples. In this case, if a learner adopts the strategy of labeling the leaf with the majority class, the performance of the classifier on the majority-class examples will improve, but at the expense of the minority-class examples. A second reason that classifiers perform worse on the minority-class test examples is that, all else being equal, a classifier is less likely to fully flesh out the boundaries of the “minority” concept in the concept space because there are fewer examples of that class to learn from. Thus, because of this some minority-class test examples may be classified as belonging to the majority class. If additional examples of the minority class were available for training, one would expect that the minority concept to grow to include additional regions of the concept space—perhaps regions that were not previously sampled at all.

2.2.2 Why Is One Training Distribution Better Than Another?

In this article we determine empirically the best training distributions for a large number of data sets. In this section we provide insight into why one distribution might be better than another. To accomplish this, we extend the notion of a learning curve to the minority and majority classes, so we can show how varying the size of the training set affects the accuracy of a classifier at predicting the class value of test examples that belong to each of these classes.

Figure 1 shows the learning curves for one of the data sets in our study. As is the case for each of the 25 data sets we study, the learning curve for the minority class always remains above the learning curve for the majority class, and, like most, the minority-class curve starts off with a much higher error rate but shows more rapid improvement. This more rapid improvement is not surprising since at any x-value the training set contains fewer minority-class than majority-class examples—and one generally expects more rapid improvement in learning when there are fewer examples (most learning curves are steepest near the beginning).

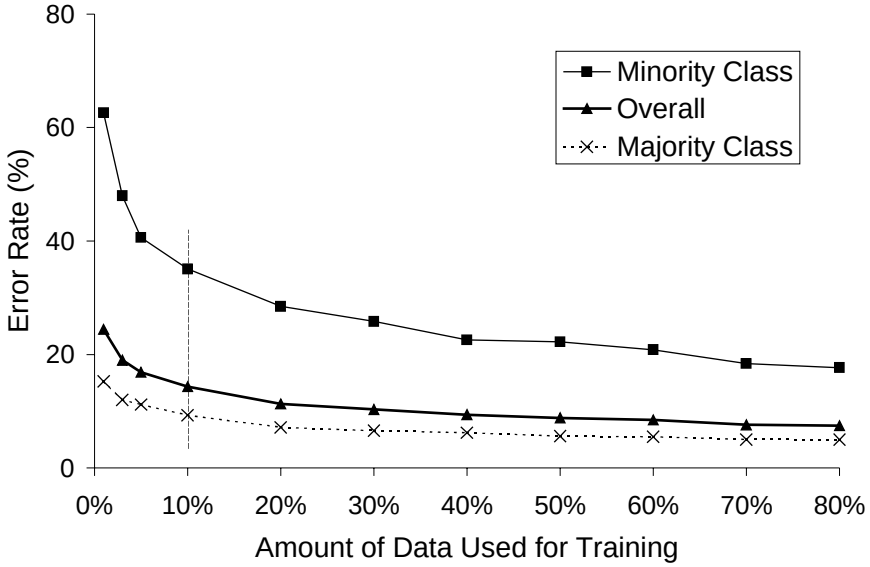

Figure 1: Learning Curves for the Letter-Vowel Data Set

Figure 1 provides some insight into how one can improve classifier performance by modifying the class distribution. These curves were generated using the naturally occurring class distribution, so if the ratio of majority to minority examples (the class ratio) is *n*:1, then as the training-set size increases, *n* majority-class examples are added for each 1 minority-class example. For the letter-vowel data set shown in Figure 1, the class ratio is 4:1 so *n* is 4.

Assume that the current training-set size corresponds to the size at x=10% and one additional example can be added to the training set. If the measure used to evaluate classifier performance equally weights the error rates of the minority-class and majority-class examples, then an example belonging to which class should be chosen? The *relative* improvement in error rate that would occur by adding a majority-class versus a minority-class example can be determined by comparing the slopes of the curves at that point. However, in this comparison the slope of the majority-class curve must be divided by *n* since the curves in Figure 1 are based on adding *n* majority-class examples for every 1 minority-class example. Because the slope of the majority-class curve at x=10% is clearly not 4 times greater than the slope of the minority-class curve at that point, a minority-class example should be added.

If the measure used to evaluate the classifier's performance is undifferentiated classification accuracy, then the decision must reflect the fact that with accuracy the error rate on the majority-class test examples counts for more—in this case *n* (i.e., 4) times as much. Therefore, the class of the example to be added can be determined by simply comparing the slopes—the fact that the majority-class counts *n*-times as much for accuracy is offset by the fact that in Figure 1 *n* majority-class examples are added for each 1 minority-class example. Based on accuracy one would still add a minority-class example for the training set size corresponding to x=10%, based on the slopes of the curves. It should be noted that once the class distribution of the training set deviates significantly from the natural class distribution, the curves in Figure 1 no longer apply. Nonetheless, these curves provide insight into how changing the class distribution of the training set will affect classifier performance.

In Figure 1 the majority-class learning curve plateaus first, which is not surprising since at any x-value on the curve there are more majority-class than minority-class training examples. Thus, even if it were initially more profitable to add majority-class examples, once the training-

set size becomes sufficiently large it may then become more profitable to start adding minority-class examples. We therefore expect the optimal training-set distribution to vary with the training-set size (we do not investigate this).

2.3 Correcting for Changes to the Class Distribution of the Training Set

In the context of this research, the purpose for modifying the class distribution of the training set is to improve the *overall* performance of the classifier. By preferentially sampling one class, we expect the classifier to improve its performance at classifying test examples of that class. We want this improvement to result from the larger number of examples available for learning—not from the bias that would result from the distorted training-set class distribution. That is, classifier induction algorithms assume that the class distribution of the test set will match that of the training set. Thus, for a decision tree, any change to the class distribution of the training set will result in biased posterior class-probability estimates at the leaves. This bias will cause the preferentially sampled class to be predicted "too often." While this bias will improve the performance of the classifier on test examples belonging to the preferentially sampled class, this bias will cause the overall performance of the classifier to suffer (we show this in Appendix A).

For our experiments, we use a decision-tree learner. The leaves of a decision tree are typically labeled by estimating the probability of each class occurring at the leaf and then assigning the most likely class to the leaf. We take the difference between the training-set distribution and test-set distribution into account by developing probability estimates that are sensitive to these differences. Rather than computing new estimates from scratch, we start with existing error estimates and "correct" them to account for the differences in the class distributions.

Two common probability estimates are listed in Table 1. For each, let A (B) represent the number of minority (majority) class examples at $Leaf_i$, so that they estimate the probability of seeing a minority-class example at $Leaf_i$. The uncorrected versions, which are in common use, are based on the assumption that the training and test sets are drawn from the same population. The frequency-based estimate is straightforward and requires no explanation. However, this estimate does not perform well when the sample size (i.e., A+B) is small—and is not even defined when the sample size is 0. For these reasons the Laplace estimate often is used instead. We consider a version based on the Laplace law of succession (Good, 1965). This probability estimate will always be closer to 0.5 than the frequency-based estimate, but the difference between the two estimates will be minimal for large sample sizes.

Estimate Name	Uncorrected	Corrected
Frequency-Based	A/(A+B)	A/(A + *c*B)
Laplace (law of succession)	(A+1)/(A+B+2)	(A+1)/(A+*c*B+2)

Table 1: Probability Estimates of Seeing a Minority-Class Example

The corrected versions of the estimates do not assume that the training and test sets have the same class distribution. These estimates account for the differences in class distribution by factoring in *c*, the ratio of minority-class to majority-class examples in the training set divided by the same ratio in the naturally occurring class distribution. As an example, if the ratio of minority to majority examples is 4:1 in the training set and 2:1 in the test set, then *c* would be 2 (assuming the test set is drawn from the naturally occurring distribution).

Many classifiers, including the decision tree learner C4.5 (Quinlan, 1993) which we use in this article, assign labels to their classification "rules" based on whether the value of the fre-

quency-based estimate associated with the rule is above or below the 0.5 probability threshold. Note that it is not necessary to use the Laplace estimate to assign the class labels because it can never move the probability estimate across the 0.5 threshold. The Laplace estimate is included in Table 1 because, as described later, we use it to generate ROC curves.

To account properly for the differences between the training and test (i.e., natural) class distributions, the probability estimates are corrected and then the leaves of the decision tree are relabeled using the corrected estimates with the 0.5 probability threshold. As an example, suppose the ratio of minority-class examples to majority-class examples in the naturally occurring class distribution for a 2-class problem is 1:5, but the training distribution is modified so that the ratio is 1:1. In this case, the value of c is 1.0/0.2, or 5. For a leaf to be labeled with the minority class the probability must be greater than 0.5, so, using the corrected frequency-based estimate, $A/(A + 5B) > 0.5$, or, $A > 5B$. Thus, a leaf is labeled with the minority class only if it covers 5 times the number of majority-class examples as minority-class examples; in general a leaf will be labeled with the minority class only if it covers c times the number of majority class examples. Note that in calculating c we use the class ratios and not the fraction of examples belonging to the minority class (if we mistakenly used the latter in the above example, then c would be one-half divided by one-sixth, or 3). Using the class ratios substantially simplifies the formulas and leads to more easily understood estimates.[1]

The results in this article are based on the use of the corrected frequency-based estimate to label the leaves of the decision tree produced by C4.5. However, the uncorrected frequency-based estimate was also used to label the decision trees and in Appendix A these results are compared to those using the corrected estimate. This comparison shows that when the class distribution of the training set is modified, the corrected frequency-based estimate yields classifiers that substantially outperform those labeled using the uncorrected version of the estimate (error rate is reduced, on average, 8.3%). Consequently it is critical to take the differences in the class distributions into account. Previous work on modifying the class distribution of the training set (Catlett, 1991; Chan & Stolfo, 1998; Japkowicz, 2000) has not taken these differences into account and this undoubtedly affected the results.

The C4.5 code was not modified to utilize these corrected estimates. Instead, the corrected probabilities were calculated and then the leaf labels reassigned as a post-processing step. Although this approach does not permit the differences in class distribution to be factored in during the tree-building process, research has shown that this is not necessary in order to build a good classifier. Drummond and Holte (2000) showed that there are decision tree splitting criteria that are relatively insensitive to a data set's class distribution—or changes to that distribution—and that these splitting criteria perform as well or better than methods that factor in the differences. However, because the differences in class distribution are not factored in by C4.5, if its pruning strategy, which attempts to minimize error rate, were allowed to execute, it would prune based on false assumptions (i.e., that the test distribution matches the training distribution). Since this may negatively affect the generated classifier and affect our study on the effect of class distribution on classifier performance, the results in this article are based on C4.5 with its pruning strategy disabled. Research indicates that when target misclassification costs (or class distributions) are unknown then pruning should be avoided anyway (Zadrozny & Elkan, 2001; Bauer & Ko-

[1] Recently, in independent research, Elkan (2001) used Bayes' rule to derive a formula that estimates class membership probabilities given changes to the class distribution of the training set. Our corrected frequency-based estimate is equivalent to Elkan's formula (which is expressed in terms of fractions).

havi, 1999) and even if the pruning strategy is adapted to take the costs and distributions into account, this does not significantly improve the performance of the classifier (Bradford et al. 1998).

3. Experimental Methodology

The experiments in this article use C4.5, a program for inducing decision trees from labeled examples (Quinlan, 1993). For the reasons just described, C4.5 is run with its pruning strategy disabled. In order to allow us to generalize from our results, all experiments are run on the collection of 25 data sets described in Table 2. These data sets include 20 data sets from the UCI repository (Blake & Merz, 1998) and 5, identified with a "+", from previously published work by researchers at AT&T (Cohen & Singer, 1999). The data sets are listed in order of decreasing class imbalance, a convention used throughout this article. In order to simplify the presentation and the analysis of our results, data sets with more than two classes are mapped into two-class problems. This is accomplished by designating one of the original classes, typically the least frequently occurring class, as the minority class and then mapping all of the remaining classes into a new class—the majority class. Each data set that originally contained more than 2 classes is identified with an asterisk (*) in Table 2. Except for the letter-recognition data set, all of the original data set names are used. The letter-a data set was created from the letter-recognition data set by assigning the examples labeled with the letter "a" to the minority class; the letter-vowel data set was created by assigning the examples labeled with any vowel to the minority class.

#	Dataset	% Minority Examples	Dataset Size	#	Dataset	% Minority Examples	Dataset Size
1	letter-a*	3.9	20000	14	network1+	29.2	3577
2	pendigits*	8.3	13821	15	car*	30.0	1728
3	abalone*	8.7	4177	16	german	30.0	1000
4	sick-euthyroid	9.3	3163	17	breast-wisc	34.5	699
5	connect-4*	9.5	11258	18	blackjack+	35.6	15000
6	optdigits*	9.9	5620	19	weather+	40.1	5597
7	solar-flare*	15.7	1389	20	bands	42.2	538
8	letter-vowel*	19.4	20000	21	market1+	43.0	3181
9	contraceptive*	22.6	1473	22	crx	44.5	690
10	adult	23.6	21281	23	kr-vs-kp	47.8	3196
11	splice-junction*	24.1	3175	24	move+	49.9	3029
12	network2	27.9	3826	25	coding	50.0	20000
13	yeast*	28.9	1484				

Table 2: Description of Data Sets

In our experiments the class distribution of the training set is varied so that the minority class accounts for between 2% and 95% of the training data. For each experimental run the training and test sets are formed as follows. First, the test set is formed by randomly selecting 25% of the minority-class examples and 25% of the majority-class examples from the original data set, without replacement. The resulting test set will therefore adhere to the original class distribution. The remaining data are then available for training. To ensure that the results using different class distributions can be compared fairly, the training-set size is required to be the same for each training class distribution. This is accomplished by setting the training-set size, S, equal to the total number of minority-class examples still available for training (i.e., 75% of the original number). This makes it possible, without replicating any examples, to generate any class distri-

bution for training-set size S. The training set is then formed by stratified random sampling, without replacement, from the remaining data so that the desired class distribution is achieved. Each data set selected for our study was required to contain a minimum of 200 minority-class examples in order to ensure a reasonable amount of training data. All results are based on the averages computed over multiple runs. Ten runs are used for the experiments in Section 4 and thirty runs for those in Section 5 (to improve the power of the statistical significance tests).

Given a fixed amount of training data, different class distributions will cause an induction algorithm to generate different classifiers. What class distribution will yield the best classifier? In order to answer this question a performance measure first must be chosen. We now describe the two performance measures used in our study.

For two-class problems the performance of a classifier can be described using the "confusion matrix" shown below. One of our performance measures is *classification accuracy*, which is defined as (TP+TN)/(TP+FP+FN+TN), or, equivalently, *classification error rate*, which is defined as one minus accuracy. Throughout this article, consistent with previous research, we consider the minority class to be the positive class.

	Actual Positive	**Actual Negative**
Predict Positive	True Positive (TP)	False Positive (FP)
Predict Negative	False Negative (FN)	True Negative (TN)

We consider classification accuracy in part because it is the most common evaluation metric in machine-learning research. However, using accuracy as a performance measure assumes that the target (marginal) class distribution is known and unchanging and, more importantly, that the error costs—the costs of a false positive and false negative—are equal. These assumptions are unrealistic (Provost et al. 1998). Accuracy is particularly suspect as a performance measure when studying the effect of class distribution on learning since it is heavily biased to favor the majority class, as described in Section 2.2.1. However, highly unbalanced problems generally have highly non-uniform error costs that favor the minority class, which is often the class of primary interest (consider medical diagnosis or fraud detection). Classifiers that optimize for accuracy for these problems are of questionable value since they rarely predict the minority class.

An alternative method for evaluating classifier performance is Receiver Operating Characteristic (ROC) analysis (Swets et al. 2000), which represents the false-positive rate on the x-axis of a graph and the true-positive rate on the y-axis. Using the terminology introduced in the confusion matrix, the true-positive rate is defined as TP/(TP+FN) and the false-positive rate as FP/(FP+TN). ROC *curves* are produced by varying the threshold on a classification model's numeric output—in our case by varying the threshold on the class-probability estimate at the leaves of a decision tree. The Laplace estimate is used because it has been shown to yield consistent improvements in ROC curves (Provost & Domingos, 2001).

Thus, one point on an ROC curve may correspond to the case where leaves of the decision tree are labeled with the minority class only if the probability of an example at a leaf belonging to the minority class is .5; another point on the curve may correspond to a probability threshold of .1. The use of ROC analysis for machine learning is described in detail elsewhere (Bradley 1997; Provost & Fawcett, 2001). For our purpose, the primary advantage of ROC curves is that they evaluate the performance of a classifier independent of the naturally occurring class distribution or error cost. One interesting consequence of this is that the method described in Section 2.3 to correct for modifications to the training set class distribution will have no affect on the ROC curves, even though, as shown in Appendix A, it affects accuracy substantially.

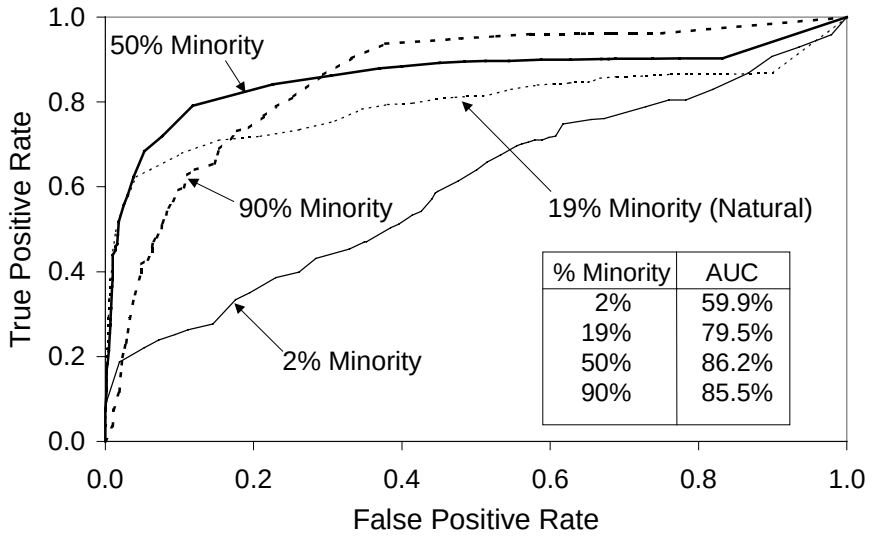

Figure 2: ROC Curves for the Letter-Vowel Data set

Figure 2 shows four ROC curves, each generated from the letter-vowel data set using the same number of training examples, but with a different training class distribution. In ROC analysis, a classifier A is better than a classifier B if it is located to the northwest of B in ROC space. The point (0,0) corresponds to the strategy of never making a positive/minority prediction and the point (1,1) to always predicting the positive/minority class.

Observe that different training distributions perform better in different areas of ROC space. Specifically note that the classifier trained with 90% minority-class examples performs substantially better than the classifier trained with the natural distribution for high true-positive rates. To our knowledge such differences in performance with class distribution never before have been shown convincingly nor analyzed. One important thing to note in Figure 2 is that the curve generated from the balanced training set (i.e., with 50% minority-class examples) often outperforms the curve associated with the natural distribution (for low false-positive rates the natural distribution performs slightly better).

To assess the *overall* quality of a classifier we measure the fraction of the total area that falls under the ROC curve, which is equivalent to several other statistical measures for evaluating classification and ranking models (Hand, 1997). The area under the ROC curve (AUC) effectively factors in the performance of the classifier over all costs and distributions. Larger AUC values indicate generally better classifier performance and, in particular, indicate a better ability to rank cases by likelihood of class membership. Figure 2 includes the AUC values for the four curves. Based on the AUC values in Figure 2 we see that the balanced class distribution generates the best overall classifier, since it maximizes the AUC. It should be kept in mind that for *specific* cost and class distributions the best model may be not be the one that maximizes AUC. If there is not a single dominating ROC curve, multiple classifiers can be combined to form a classifier that performs optimally for all costs and distributions (Provost & Fawcett, 2001).

Although we measure classifier performance using both AUC and error rate, we prefer the AUC measure because we are interested in drawing conclusions across a variety of data sets for which the true misclassification costs are unknown. We believe that the results based on the AUC measure are more important because we believe AUC provides a more realistic assessment of the performance of a classifier for real-world problems.

4. Results: Learning from Unbalanced vs. Balanced Data Sets

We now assess the performance of the classifiers formed from the 25 data sets using the natural class distributions and then assess the classifiers formed from the balanced versions of these same data sets. In order to make the comparisons fair, the data-set sizes for the balanced and unbalanced versions of each data set are forced to be equal, using the methodology described in Section 3. Because the coding data set begins with a balanced class distribution, one class is arbitrarily designated the minority class. So that our analysis is not affected by this arbitrary decision, the average and median values displayed in Tables 3 and 4 and the data points shown in Figures 3 and 4 exclude the results from this data set.

4.1 Learning from Unbalanced Data Sets

Classifiers were generated and evaluated on the 25 data sets using the naturally occurring class distributions. The results are summarized in Table 3. The first column in Table 3 identifies the data set and the second column specifies the natural class distribution, which was also displayed in Table 2. The third column specifies the percentage of the total test errors that result from misclassifying test examples belonging to the minority class. These results show that for almost all data sets the majority of errors come from (misclassifying) the minority-class examples, even though, as shown by the second column, the minority class typically accounts for far fewer than half of the examples. The fourth column specifies the number of leaves labeled with the minority and majority classes. The results indicate that there are fewer leaves labeled with the minority class than with the majority class.

	% Minority	% Errors	Leaves		Coverage		Leaf ER		Example ER		Recall	
Dataset	Examples	from Min.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.
letter-a	3.9	60.0	11	138	2.2	4.3	32.5	1.7	41.5	1.2	58.5	98.8
pendigits	8.3	31.2	6	8	16.8	109.3	25.8	1.3	14.3	2.7	85.7	97.3
abalone	8.7	69.7	5	8	2.8	35.5	69.8	7.7	84.4	3.6	15.6	96.4
sick-euthyroid	9.3	53.6	4	9	7.1	26.9	22.5	2.5	24.7	2.4	75.3	97.6
connect-4	9.5	51.8	47	128	1.7	5.8	55.8	6.0	57.6	5.7	42.4	94.3
optdigits	9.9	72.2	15	173	2.9	2.4	18.0	3.9	36.7	1.5	63.3	98.5
solar-flare	15.7	64.6	12	48	1.7	3.1	67.8	13.7	78.9	8.1	21.1	91.9
letter-vowel	19.4	61.9	233	2547	2.4	0.9	27.0	8.7	37.5	5.6	62.5	94.4
contraceptive	22.6	48.7	31	70	1.8	2.8	69.8	20.1	68.3	21.1	31.7	79.0
adult	23.6	57.0	314	2084	2.7	1.4	37.0	13.1	43.7	10.2	56.3	89.8
splice-junction	24.1	58.5	26	46	5.5	9.6	15.1	6.3	20.3	4.5	79.7	95.5
network2	27.9	57.2	50	61	4.0	10.3	48.2	20.4	55.5	16.2	44.5	83.9
yeast	28.9	59.2	8	12	14.4	26.1	45.6	20.9	55.0	15.6	45.1	84.4
network1	29.2	57.5	42	49	5.1	12.8	46.2	21.0	53.9	16.7	46.1	83.3
car	30.0	56.8	38	42	3.1	6.6	14.0	7.7	18.6	5.6	81.4	94.4
german	30.0	55.6	34	81	2.0	2.0	57.1	25.4	62.4	21.5	37.6	78.5
breast-wisc	34.5	42.3	5	5	12.6	26.0	11.4	5.1	9.8	6.1	90.2	93.9
blackjack	35.6	81.5	13	19	57.7	188.0	28.9	27.9	64.4	8.1	35.7	91.9
weather	40.1	50.7	134	142	5.0	7.2	41.0	27.7	41.7	27.1	58.3	72.9
bands	42.2	91.2	52	389	1.4	0.3	17.8	34.8	69.8	4.9	30.2	95.1
market1	43.0	50.2	87	227	5.1	2.7	30.9	23.4	31.2	23.3	68.8	76.7
crx	44.5	50.4	28	65	3.9	2.1	23.2	18.9	24.1	18.5	75.9	81.5
kr-vs-kp	47.8	52.5	23	15	24.0	41.2	1.2	1.3	1.4	1.1	98.6	98.9
move	49.9	61.3	235	1025	2.4	0.6	24.4	29.9	33.9	21.2	66.1	78.8
coding	50.0	61.0	1483	2412	2.5	1.6	30.7	35.6	40.9	26.1	59.1	73.9
Average	26.6	58.1	61	308	7.8	22.0	34.6	14.6	42.9	10.5	57.1	89.5
Median	28.4	57.1	29	63	3.5	6.2	29.9	13.4	41.6	7.1	58.4	92.9

Table 3: Comparative Performance of Minority & Majority Classes (Unbalanced Data Sets)

The fifth column, "Coverage", specifies the average number of *training* examples that each minority-labeled or majority-labeled leaf classifies. The results show that the leaves labeled with the minority class are formed from far fewer training examples than those labeled with the majority class. In section 2.2.1 we stated one reason why minority-labeled leaves have a higher error rate than the majority-labeled leaves is because of the "test distribution effect" of having more majority-class than minority-class test examples. Table 3 suggests a second reason, which is related to the fact that minority-labeled leaves tend to be formed from fewer training examples as a consequence of having fewer minority-class examples in the training set. *Small disjuncts*, which are disjuncts (i.e., classification rules, decision-tree leaves, etc.) that cover few training examples, typically have a much higher error rate than large disjuncts (Holte, et al. 1989; Weiss & Hirsh, 2000). The main reason for this behavior is that because small disjuncts are formed from fewer training examples than large disjuncts, one cannot be as confident in the probability estimate. Consequently, we expect the rules/leaves labeled with the minority class to have a higher error rate because they suffer more from this "problem of small disjuncts."

The last three columns of Table 3 provide additional class-specific information about the classifiers. The "Leaf ER" column specifies the error rates for *leaves* labeled with the minority and majority classes, based on the performance of these leaves at classifying the test examples. The "Example ER" column specifies the error rates for the test examples belonging to the minority and majority classes. The last column specifies the recall for the two classes. Recall is a measurement commonly used in information retrieval, which in this case represents the percentage of the total minority-class and majority-class test examples, respectively, that are correctly classified. This information from the last three columns is presented graphically in Figure 3.

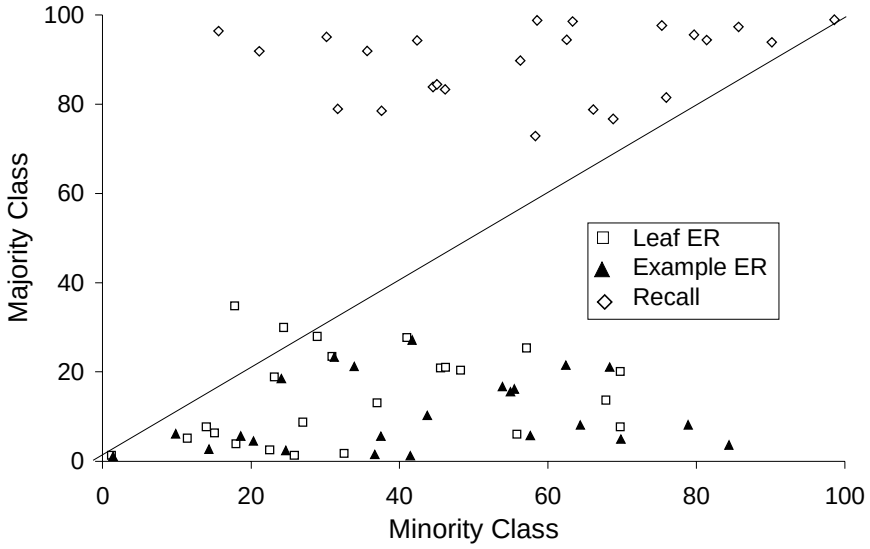

Figure 3: Comparison of Minority and Majority Class Measurements (Unbalanced Data Sets)

According to the results in Table 3 and Figure 3, the classifiers perform much worse on the minority-class test examples than on the majority-class test examples. Over the 24 data sets (coding is excluded from the analysis), the average error rate for the minority-class test examples is 42.9%, whereas for the majority-class test examples it is only 10.5%. We also see that the leaves labeled with the minority class have an average error rate of 34.6%, while the leaves labeled with the majority class have an error rate of 14.6%. On average the classifiers correctly classify 57.1% of the minority-class examples but 89.5% of the majority-class examples.

Some of these results appear contradictory, but can be reconciled. Given that the *leaves* labeled with the minority class have a higher error rate than the leaves labeled with the majority class, all else being equal, the majority-class test examples should have a higher error rate than the minority-class test examples. This is because, as described in Section 2.2.1, a high error rate on the minority-labeled leaves means more *majority-class* test examples are misclassified. The reason we observe a lower error rate on the majority-class test examples is because the majority class is predicted far more often than the minority class (cf. Recall).

One would expect the observed differences in error rate on the test-set examples to diminish if a classifier predicted the minority-class examples as often as it predicted the majority-class examples. The results in the next section address this, since with a balanced class distribution one would expect the frequency of predicting each class to be (more) balanced.

4.2 Learning from Balanced Data Sets

The experimental results described in this section are based on training and test sets that are created by selecting equal numbers of minority-class and majority-class examples. Therefore, these data sets can be considered to be balanced versions of the ones used in Section 4.1. By balancing the classes, we essentially eliminate the factors, described in Section 2.2, which tend to "favor" the majority class. Note that in this case the "minority" class refers to the class that occurs less frequently in the natural (i.e., original) distribution. The classifiers generated for the balanced versions of the data sets are described in Table 4 and Figure 4. The data points in Figure 4 that are black correspond to classifiers built from data sets where the majority class was formed by combining two or more classes (as described in Section 3).

	% Minority	% Errors	Leaves		Coverage		Leaf ER		Example ER		Recall	
Dataset	Examples	from Min.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.
letter-a	50.0	63.0	40	359	7.7	0.9	5.9	8.5	8.9	5.8	91.1	94.2
pendigits	50.0	39.1	7	8	66.3	53.8	3.7	2.5	2.5	3.7	97.5	96.3
abalone	50.0	45.5	18	18	8.2	7.8	31.3	28.3	28.0	32.0	72.0	68.0
sick-euthyroid	50.0	48.2	7	11	17.5	11.1	7.6	7.5	8.1	7.6	91.9	92.4
connect-4	50.0	53.6	125	219	3.2	1.9	25.7	27.5	28.7	24.8	71.3	75.2
optdigits	50.0	69.1	18	148	12.9	1.5	3.5	6.9	7.3	3.5	92.8	96.5
solar-flare	50.0	55.7	28	62	2.7	1.5	35.4	37.2	41.1	32.2	58.9	67.8
letter-vowel	50.0	64.5	375	3177	3.9	0.5	13.1	19.6	21.5	11.8	78.5	88.2
contraceptive	50.0	53.1	52	74	2.5	1.7	39.0	38.6	42.4	35.7	57.6	64.3
adult	50.0	53.0	483	2432	4.1	0.8	22.1	23.6	24.3	21.5	75.8	78.5
splice-junction	50.0	53.2	31	46	9.3	6.3	10.2	11.5	12.0	10.0	88.0	90.0
network2	50.0	56.0	62	61	6.3	7.2	31.7	34.1	37.1	29.0	62.9	71.0
yeast	50.0	42.1	9	10	22.9	22.5	32.9	28.2	26.8	35.9	73.2	64.1
network1	50.0	58.3	54	55	6.9	7.8	30.3	33.9	37.9	27.0	62.2	73.1
car	50.0	62.7	41	37	4.8	5.4	8.0	11.8	12.8	7.7	87.2	92.3
german	50.0	55.4	42	84	2.7	1.4	38.5	39.0	43.8	34.0	56.2	66.1
breast-wisc	50.0	56.7	6	5	15.6	19.4	7.1	8.2	8.3	7.0	91.7	93.0
blackjack	50.0	56.3	20	21	101.7	112.1	33.1	35.8	39.1	30.3	60.9	69.7
weather	50.0	50.6	141	138	6.0	6.1	33.8	34.0	34.3	33.6	65.7	66.4
bands	50.0	96.0	57	399	1.5	0.2	10.2	42.6	72.1	3.2	27.9	96.8
market1	50.0	50.8	89	212	5.8	2.5	27.2	27.6	28.0	27.0	72.1	73.0
crx	50.0	48.5	32	64	3.7	1.9	22.0	21.5	21.8	22.1	78.2	77.9
kr-vs-kp	50.0	61.2	23	15	25.5	38.6	1.2	1.2	1.2	1.2	98.8	98.8
move	50.0	60.1	241	1010	2.4	0.6	25.9	30.7	34.4	22.9	65.6	77.1
coding	50.0	61.1	1481	2409	2.5	1.6	30.9	35.9	41.2	26.3	58.8	73.7
Average	50.0	56.4	83	361	14.3	13.1	20.8	23.3	25.9	19.6	74.1	80.4
Median	50.0	55.4	41	62	6.0	5.4	25.7	27.6	28.0	22.9	72.1	77.1

Table 4: Comparative Performance of Minority & Majority Classes (Balanced Data Sets)

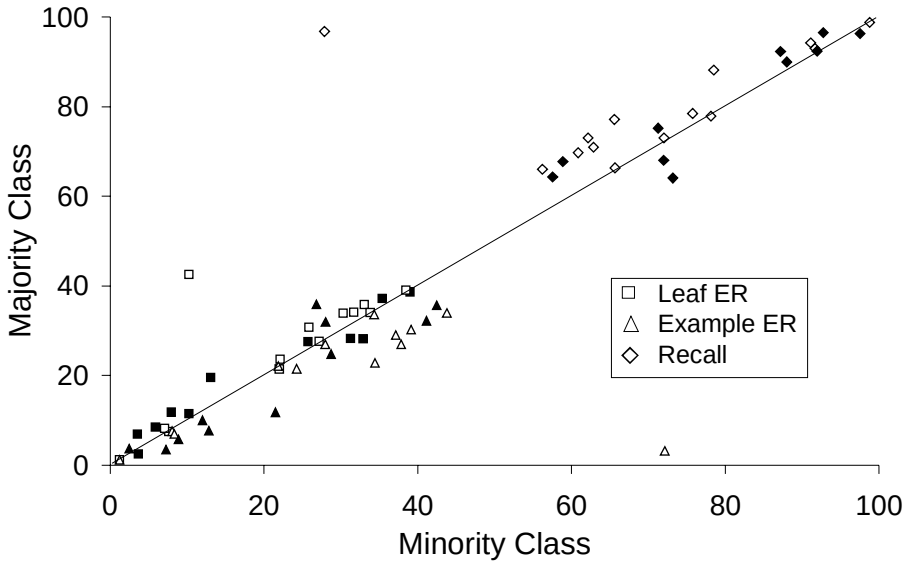

Figure 4: Comparison of Minority and Majority Class Error Rates for Balanced Data Sets

By comparing Table 4 with Table 3 and Figure 4 with Figure 3, we see that, as one would expect, the minority and majority classes behave much more similarly when the classifiers are built from the balanced data sets. However, the classifiers built from the balanced data sets do *not* show identical, or even symmetric behavior for the two classes and there are some consistent, and surprising, differences. For 19 of 24 balanced data sets the minority-class test examples have the higher error rate and for 19 of 24 data sets the majority class has the higher recall. This is the same trend we observed for the unbalanced data sets. However, for 17 of 24 data sets the leaves predicting the minority class now have a *lower* error rate (consistent with the previous discussion). Table 4 also shows us that there are generally fewer leaves labeled with the minority class than with the majority class—even though the examples occur in equal numbers. We can summarize these results as follows: *when learning from the balanced versions of the unbalanced data sets, the induction algorithm generally produces fewer but more accurate classification rules for the minority class than for the majority class.*

We believe these differences exist because the balanced distributions are derived from naturally unbalanced distributions—in which the minority and majority classes exhibit fundamental differences. Specifically, we believe the minority classes tend to be a more homogeneous set of entities, while the majority class often corresponds to "everything else." For example, for the "letter-a" data set, the minority class corresponds to the letter "a" and the majority class corresponds to all other letters. Given equal numbers of examples for each class, one might expect the more homogeneous class to be better learned. Note that the differences cannot be attributed to the fact that for 11 of the 24 data sets the majority class was formed from multiple distinct classes. By comparing the black versus the white data points in Figure 4, we see that the classifiers built from the "naturally" occurring 2-class problems (the white points) also exhibit consistent differences between the majority and minority classes.

5. Results: The Effect of Training-Set Class Distribution on Classifier Performance

The goal of this work is to assess, for a fixed training-set size, if and to what extent different training-set class distributions affect classifier performance, and how classifiers generated from

the natural class distribution compare to those built from other class distributions. In order to complete this assessment we now vary the training-set class distributions for all 25 data sets as described in Section 3, so that between 2% and 95% of the distribution is made up of minority-class examples. In particular, we evaluate the following twelve minority-class distributions for each of the data sets: 2%, 5%, 10%, 20%, 30%, … , 80%, 90%, 95%. For each data set we additionally evaluate the performance using the naturally occurring class distribution.

5.1 Methodology for Determining the Optimum Training Class Distribution(s)

In Sections 5.2 and 5.3 we present experimental results that measure classifier performance in terms of error rate and AUC, respectively. In each of these sections we must address several issues that affect our ability to determine the "optimal" training-set class distribution. First, because we do not evaluate every possible class distribution, we cannot determine *the* optimal class distribution—we are limited to determining the best among the 13 distributions sampled. Beyond this concern, however, are issues of statistical significance and, because we generate classifiers for 13 training distributions, the issue of multiple comparisons (Jensen & Cohen, 2000). Because of these issues we cannot necessarily conclude that the training distribution that yields the best performing classifiers is truly the best one for training.

We take several steps to address the issues of statistical significance and multiple comparisons. To enhance our ability to identify true differences in classifier performance with respect to changes in class distribution, all experimental results presented in this section are based on 30 runs rather than the 10 runs used in Section 4. We then perform statistical significance testing, as will be described shortly. Rather than trying to determine *the* optimal class distribution, we adopt a more conservative approach, and instead try to identify an "optimal range" of class distributions. We will be confident that this range includes the optimal class distribution. We expect classifier performance to be unimodal with respect to changing class distribution. That is, for a specific induction algorithm, data set, and training-set size, we believe that an optimal class distribution exists and that as we move further away from this optimal distribution, in either direction, classifier performance will degrade progressively. The results will show that this expectation is almost always met when error rate is the performance measure and is generally true for AUC. A unimodal distribution allows us, more confidently, to compare other points with the "best" point; the chances are slim that a unimodal distribution would be observed if the differences between points weren't truly significant.

In the remainder of this section we describe how we determine the "optimal range" of class distributions. We first identify, for each data set, the class distribution that yields the classifiers that perform best over the 30 runs (i.e., have the lowest average error rate or greatest average AUC). We then perform t-tests to compare the performance of these 30 classifiers with the 30 classifiers generated using each of the other twelve class distributions (i.e., 12 t-tests each with n=30 data points). If a t-test yields a probability $\leq .10$ then we conclude that the "best" distribution *is* statistically different from the "other" distribution (i.e., we are >90% confident of this); otherwise we cannot conclude that they truly perform differently and therefore "group" the distributions together. These grouped distributions collectively form an "optimum range"—a range of class distribution values within which we are confident that the true, optimum class distribution falls (with the caveat that this is with respect to the 13 *evaluated* distributions). One of the things we are interested in is whether the optimum range includes the natural distribution.

5.2 The Effect of Class Distribution on Error Rate

For each of the 25 data sets, the 13 class distributions previously mentioned were formed and classifiers were built from these distributions. The results are displayed in Table 5. This table warrants some explanation; we use the first data set listed, letter-a, as an example.

The first column specifies the data set name. The next 13 columns present the error rate values for the 13 class distributions that are evaluated. The first of these columns presents the error rate values for the natural distribution (the natural distribution is listed in Table 2). The value corresponding to the lowest error rate for each data set is indicated by underlining it and displaying it in boldface. The relative position of the natural distribution within the range of evaluated class distributions is denoted by the use of a vertical bar between columns. For the letter-a data set, the lowest error rate is 2.59%, which occurs when the training set contains 10% minority-class examples. The vertical bar indicates that the natural distribution falls between the 2% and 5% distributions.

The error rate values that are not significantly different, statistically, from the "best" value (i.e., they yield a t-test value > .10) are shaded. Thus, for the letter-a data set, the optimum range goes from 2% to 10% and includes the natural distribution (note that its error rate is also shaded). Following these 13 columns is a column that specifies the t-test probability value computed between the best distribution and the natural distribution. If this value is ≤ .10 then we are confident that these two distributions truly perform differently; in this case the probability (.728) is not displayed in bold, indicating that the natural distribution is not significantly different from the best distribution, and is part of the optimum range. The last column measures the relative improvement in error rate that occurs by learning using the best distribution rather than the natural distribution. While we compute these values in all cases, these differences are only statistically significant when the t-test probability value, specified in the previous column, is < .10 (in these cases the relative improvement is displayed in bold).

	Error Rate when using Specified Training Distribution (expressed as % Minority)													T-test Prob.	Relative Improv.
Dataset	**Natural**	**2**	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**95**	**Best/Nat.**	**Best/Nat.**
letter-a	2.78	2.86	2.75	**2.59**	3.03	3.79	4.53	5.38	6.48	8.51	12.37	18.10	26.14	.728	6.8%
pendigits	3.74	5.77	3.95	3.63	**3.45**	3.70	3.64	4.02	4.48	4.98	5.73	8.83	13.36	.419	7.8%
abalone	10.46	**9.04**	9.61	10.64	13.19	15.33	20.76	22.97	24.09	26.44	27.70	27.73	33.91	.439	13.6%
sick-euthyroid	**4.10**	5.78	4.82	4.69	4.25	5.79	6.54	6.85	9.73	12.89	17.28	28.84	40.34	1.000	0.0%
connect-4	10.56	**7.65**	8.66	10.80	15.09	19.31	23.18	27.57	33.09	39.45	47.24	59.73	72.08	**.000**	**27.6%**
optdigits	4.68	8.91	7.01	4.05	3.05	2.83	**2.79**	3.41	3.87	5.15	5.75	9.72	12.87	**.000**	**40.4%**
solar-flare	19.98	**16.54**	17.52	18.96	21.45	23.03	25.49	29.12	30.73	33.74	38.31	44.72	52.22	**.010**	**17.2%**
letter-vowel	**11.63**	15.87	14.24	12.53	11.67	12.00	12.69	14.16	16.00	18.68	23.47	32.20	41.81	1.000	0.0%
contraceptive	30.47	**24.09**	24.57	25.94	30.03	32.43	35.45	39.65	43.20	47.57	54.44	62.31	67.07	**.000**	**20.9%**
adult	17.95	19.19	18.24	**17.54**	17.72	18.51	19.67	20.86	22.73	25.17	28.97	35.80	42.35	**.088**	**2.3%**
splice-junction	8.37	20.00	13.95	10.72	8.68	8.50	**8.15**	8.74	9.86	9.85	12.08	16.25	21.18	.682	2.6%
network2	26.67	27.37	25.91	25.71	**25.66**	26.94	28.65	29.96	32.27	34.25	37.73	40.76	37.72	.269	3.8%
yeast	26.59	29.08	28.61	27.51	**26.35**	26.93	27.10	28.80	29.82	30.91	35.42	35.79	36.33	.820	0.9%
network1	27.59	27.90	27.43	26.78	**26.58**	27.45	28.61	30.99	32.65	34.26	37.30	39.39	41.09	.229	3.7%
car	8.85	23.22	18.58	14.90	10.94	8.63	8.31	7.92	**7.35**	7.79	8.78	10.18	12.86	**.004**	**16.9%**
german	33.41	**30.17**	30.39	31.01	32.59	33.08	34.15	37.09	40.55	44.04	48.36	55.07	60.99	**.000**	**9.7%**
breast-wisc	6.82	20.65	14.04	11.00	8.12	7.49	6.82	**6.74**	7.30	6.94	7.53	10.02	10.56	.925	1.2%
blackjack	**28.40**	30.74	30.66	29.81	28.67	28.56	28.45	28.71	28.91	29.78	31.02	32.67	33.87	.887	1.2%
weather	33.69	38.41	36.89	35.25	33.68	**33.11**	33.43	34.61	36.69	38.36	41.68	47.23	51.69	**.056**	**1.7%**
bands	32.53	38.72	35.87	35.71	34.76	33.33	**32.16**	32.68	33.91	34.64	39.88	40.98	40.80	.666	1.1%
market1	26.16	34.26	32.50	29.54	26.95	26.13	26.05	**25.77**	26.86	29.53	31.69	36.72	39.90	.535	1.5%
crx	**20.39**	35.99	30.86	27.68	23.61	20.84	20.82	21.48	21.64	22.20	23.98	28.09	32.85	1.000	0.0%
kr-vs-kp	1.39	12.18	6.50	3.20	2.33	1.73	**1.16**	1.22	1.34	1.53	2.55	3.66	6.04	.588	16.5%
move	28.57	46.13	42.10	38.34	33.48	30.80	28.36	**28.24**	29.33	30.21	31.80	36.08	40.95	.477	1.2%
coding	33.68	45.63	42.74	39.79	36.62	34.62	33.90	**33.62**	33.81	34.89	36.62	40.63	43.87	.742	0.2%

Table 5: Effect of Training Set Class Distribution on Error Rate

The results in Table 5 show that for 8 of the 25 data sets we are confident, based on the t-tests, that the natural distribution is not within the range of optimal class distributions. Furthermore, for most of these data sets using the best distribution rather than the natural distribution yields a remarkably large decrease in error rate. We feel that this is sufficient evidence to conclude that for accuracy it is *not* safe to assume that the natural distribution should be used for training. Inspection of the error-rate results in Table 5 also shows that the best distribution does not differ from the natural distribution in any consistent manner—sometimes it includes more minority-class examples (e.g., optdigits, car) and sometimes fewer (e.g., connect-4, solar-flare, contraceptive, adult, german, and weather). However, it is clear that for data sets with a substantial amount of class imbalance (the ones in the top half of the table), a 50:50 class distribution also is not the best class distribution for training, to minimize undifferentiated error rate.

We have examined the error-rate values for the remaining 17 data sets for which the t-test results do not permit us to conclude that the best observed distribution truly outperforms the natural distribution. In these cases we see that the error rate values for the 12 training set class distributions almost always form a perfectly unimodal distribution. This suggests that "adjacent" class distributions may indeed produce classifiers that truly perform differently, but that our statistical testing is not sufficiently powerful.

Figure 5 shows the behavior of the learned classifiers for the adult, car, and optdigits data sets in a more visual form. In this figure the natural distribution is denoted by the enlarged diamond-shaped tick mark. The error rate at this point is noted above the marker; the error rate for the best distribution is specified below the corresponding data point. The range of error-rate values displayed in Figure 5 was limited and consequently a few data points for the adult curve are omitted.

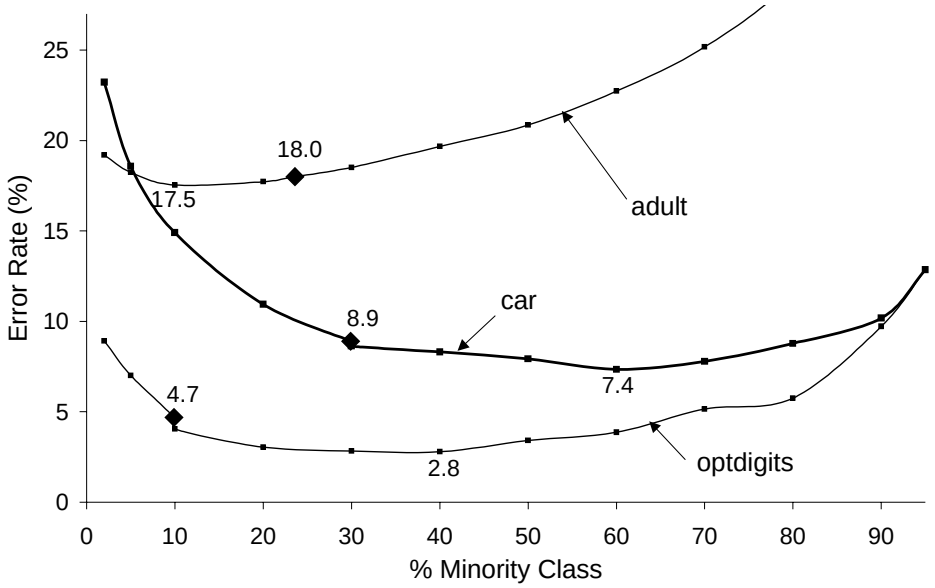

Figure 5: Effect of Class Distribution on Error Rate for Select Data Sets

Note that all three curves are perfectly unimodal. It is also clear that near the distribution that minimizes error rate, changes to the class distribution yield only modest changes in the error rate—far more dramatic changes occur elsewhere. This is also evident for most data sets in Table 5. This is a convenient property given the common goal of minimizing error rate. *This prop-*

erty would be far less evident if the correction described in Section 2.3 were not performed, since then classifiers induced from class distributions deviating from the naturally occurring distribution would be improperly biased.

5.3 The Effect of Class Distribution on AUC

Results analogous to the ones displayed in Table 5 are presented in Table 6 for AUC. Recall, however, that for AUC *larger* values indicate improved performance. The relative improvement in classifier performance is specified in the last column in terms of the area above the ROC curve (i.e., 1 – AUC) and then in terms of the area under the curve (AUC). We include the former because it may more accurately reflect the relative improvement—just as in Table 5 we specify relative improvement in terms of the change in error rate instead of the change in accuracy.[2]

In general, the optimum ranges appear to be centered to the right of the 50:50 class distribution and the class distribution which contains 70% minority-class examples is within the optimum range for all but two data sets. For 13 of the 25 data sets the optimum range does not include the natural distribution and for these data sets the natural distribution contains fewer minority-class examples than the class distributions within the optimum range (with the exception of solar-flare which appears to have an anomalous value). Therefore, we conclude even more strongly that for cost-sensitive classification and for ranking, it is not appropriate simply to choose the natural class distribution for training.

	AUC when using Specified Training Distribution (expressed as % Minority)													T-test Prob.	Relative Improvement (%)	
Dataset	Natural	2	5	10	20	30	40	50	60	70	80	90	95	best/nat.	1 - AUC	AUC
letter-a	.772	.711	.799	.865	.891	.911	.938	.937	.944	.951	**.954**	.952	.940	**.000**	**79.8**	**23.6**
pendigits	.967	.892	.958	.971	.976	.978	**.979**	**.979**	.978	.977	.976	.966	.957	.167	36.4	1.2
abalone	.711	.572	.667	.710	.751	.771	.775	.776	**.778**	.768	.733	.694	.687	**.000**	**25.8**	**9.4**
sick-euthyroid	.940	.892	.908	.933	.943	.944	.949	.952	.951	**.955**	.945	.942	.921	**.094**	**25.0**	**1.6**
connect-4	.731	.664	.702	.724	.759	.763	.777	.783	**.793**	**.793**	.789	.772	.730	**.000**	**23.1**	**8.5**
optdigits	.803	.599	.653	.833	.900	.924	.943	.948	.959	.967	.965	**.970**	.965	**.000**	**84.8**	**20.8**
solar-flare	.627	.614	.611	.646	.627	.635	.636	.632	.650	**.662**	.652	.653	.623	**.013**	**9.4**	**5.6**
letter-vowel	.793	.635	.673	.744	.799	.819	.842	.849	.861	**.868**	**.868**	.858	.833	**.000**	**36.2**	**9.5**
contraceptive	.611	.567	.613	.617	.616	.622	.640	.635	.635	.640	**.641**	.627	.613	**.007**	**7.7**	**4.9**
adult	.820	.792	.795	.805	.815	.824	.830	.834	.837	**.846**	.841	.841	.832	**.000**	**14.4**	**3.2**
splice-junction	.905	.814	.820	.852	.908	.915	.925	.936	.938	.944	**.950**	.944	.944	**.000**	**47.4**	**16.7**
network2	.708	.634	.696	.703	.708	.705	.704	.705	.702	.706	.710	**.719**	.683	.209	3.8	1.6
yeast	.705	.547	.588	.650	.696	**.727**	.714	.720	.723	.715	.699	.659	.621	.123	10.9	3.1
network1	.705	.626	.676	.697	.709	.709	.706	.702	.704	.708	**.713**	.709	.696	.372	2.7	1.1
car	.879	.754	.757	.787	.851	.884	.892	.916	.932	.931	**.936**	.930	.915	**.000**	**47.1**	**6.5**
german	.646	.573	.600	.632	.615	.635	**.654**	.645	.640	.650	.645	.643	.613	.547	2.3	1.2
breast-wisc	.958	.876	.916	.940	.958	.963	**.968**	.966	.963	.963	.964	.949	.948	.192	23.8	1.0
blackjack	.700	.593	.596	.628	.678	.688	.712	.713	**.715**	.700	.678	.604	.558	**.059**	**5.0**	**2.1**
weather	.736	.694	.715	.728	.737	.738	**.740**	.736	.730	.736	.722	.718	.702	.472	1.5	0.5
bands	**.623**	.522	.559	.564	.575	.599	.620	.618	.604	.601	.530	.526	.536	1.000	0.0	0.0
market1	.811	.724	.767	.785	.801	.810	.808	.816	**.817**	.812	.805	.795	.781	.286	3.2	0.7
crx	.852	.804	.799	.805	.817	.834	.843	.853	.845	.857	.848	.853	**.866**	.279	9.5	1.6
kr-vs-kp	.997	.937	.970	.991	.994	.997	**.998**	**.998**	**.998**	.997	.994	.988	.982	.702	33.3	0.1
move	.734	.574	.606	.632	.671	.698	.726	.735	.738	**.742**	.736	.711	.672	.214	3.0	1.1
coding	.684	.616	.625	.635	.650	.667	.676	.684	.691	**.694**	.692	.676	.657	**.000**	**3.2**	**1.5**

Table 6: Effect of Training Set Class Distribution on AUC

[2] For example, a change in error rate from 4% to 2% represents a 50% reduction in error rate—but is equivalent to an increase in accuracy that is only just slightly greater than 2% (i.e., from 96% to 98%).

Powerful as they are, some of these results are not particularly surprising. Since, unlike error rate, AUC is not affected by the test-set class distribution, and because it factors in classifier performance over *all* class distributions, one might expect the evenly balanced distribution generally to outperform the natural distribution—which it does. However, the results indicate that the optimal range is generally shifted beyond the evenly balanced distribution, and, for 11 of the 25 data sets, it includes the distribution comprising 90% minority-class examples!

The optimum ranges shown in Table 6 are broader than those in Table 5. While changing the class distribution generally improves classifier performance on one class, it generally degrades the performance on the other class. Because AUC factors in the performance over all class distributions and costs, we believe it is less susceptible to changes in class distribution. That is, for some costs improving the performance of one class at the expense of the other will yield better classifier performance, but for others it will not—but these changes in performance will often tend to cancel each other out, yielding the wider optimum ranges.

Figure 6 shows how class distribution affects AUC for three data sets. The natural distribution is indicated by a large diamond tick mark and the corresponding AUC value for that distribution is displayed above it; the maximum AUC value is shown below the corresponding data point.

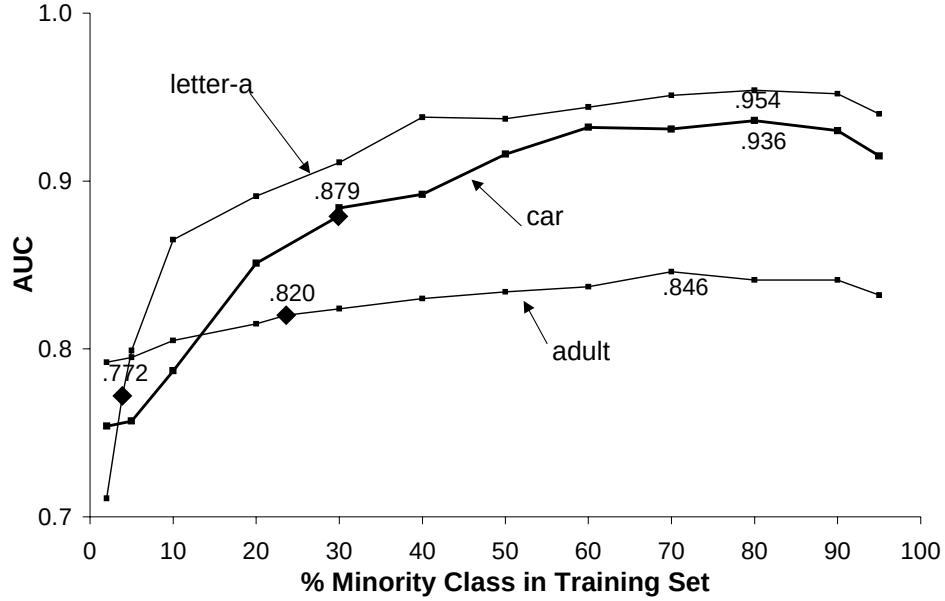

Figure 6: Effect of Class Distribution on AUC for Select Data Sets

5.4 Discussion

In Section 2.2 we provided several reasons why one training distribution may be better than another for learning. We now apply that reasoning to explain some of the results in this section. Because it is impractical to analyze each of the 25 data sets, we restrict our attention to two data sets that exhibit interesting, but very different, behavior. In Tables 5 and 6 the rows for the optdigits data set show that, for both accuracy and AUC, the optimal training distribution contains far more minority-class examples than majority-class examples. In contrast, the rows for the contraceptive data set show that, for accuracy, the optimal distribution contains far fewer minority-class examples (for AUC it also contains more minority-class examples).

The learning curves for these two data sets are shown in Figures 7a and Figure 7b. For these data sets, as for all 25 data sets, the learning curve for the minority class is on top, the overall curve in the middle and the majority-class learning curve on bottom. Thus, the test examples belonging to the minority class always have a higher error rate than those belonging to the majority class. Because, as described in Section 3.1, the training-set sizes for experiments that vary the class distributions are set to equal ¾ of the number of minority class examples, the training-set sizes for the optdigits and contraceptive data sets correspond to a value of x=7.4% (¾ · 9.9%) and 17.0% (¾ · 22.6%), respectively, in Figure 7a and Figure 7b. Note that around these values the minority-class learning curve for the optdigits data set shows dramatic improvement, while for the contraceptive data set it shows only a slight improvement.

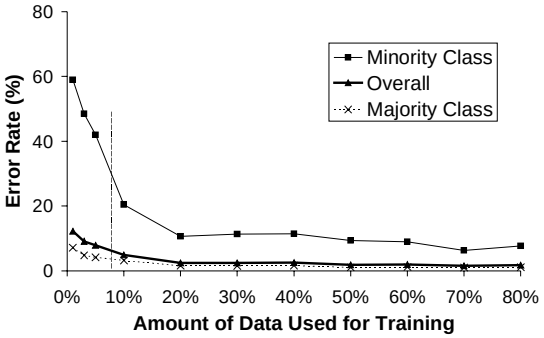

Figure 7a: Learning Curve for Optdigits

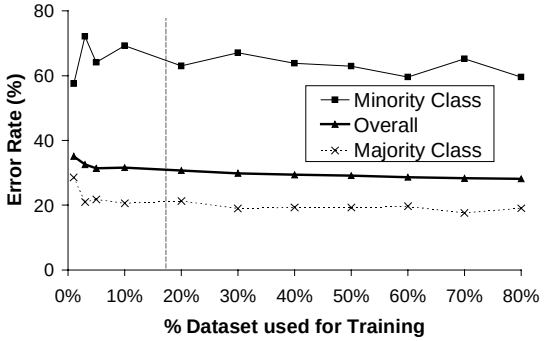

Figure 7b: Learning Curve for Contraceptive

The dramatic improvement in the minority-class learning curve for the optdigits data set (which does not occur for the majority-class learning curve) explains why, for error rate, the optimal training distribution (30% minority) includes far *more* minority examples than its natural distribution (4.9% minority). For the contraceptive data set, the improvement shown around the x-value of 17% is very slight and for smaller values it appears the majority-class learning curve shows more improvement. Consequently, the learning curves in Figure 7b help explain why the optimal distribution (2% minority) includes far *fewer* minority examples than its natural distribution (31.7% minority).

Finally, in Section 2.3 we described a method for correcting a classifier to compensate for changes to the class distribution of the training set. Although we utilize this correction, previous research has not (Catlett, 1991; Chan & Stolfo, 1998; Japkowicz, 2000). In order to demonstrate and quantify the importance of this correction, the error rates that result when using and not using this correction are presented in Appendix A. These results show that by employing the corrected frequency-based estimate, the error rate is reduced or remains the same for all but 2 of the 25 data sets, and the error rate is reduced, on average, by 8.3%. For the 11 data sets where the class ratio is greater than 3:1, the error rate is reduced even more—by an average of 19.0%. Thus, we conclude that this correction is essential when the class distribution of the training set is altered.

6. Conclusion and Limitations

In this article we demonstrated, for two performance measures (error rate and AUC), that for a fixed number of training examples the naturally occurring class distribution often does not produce to the best-performing classifier. Furthermore, we have shown that when AUC is the per-

formance measure—a measure we believe to be more appropriate than error rate—then the optimal distribution generally contains between 50% and 90% minority-class examples. In this situation, the strategy of always allocating half of the training examples to the minority class, while it will not always yield optimal results, will generally lead to results which are no worse than, and often superior to, those which use the natural class distribution. Thus, if one does not know the true misclassification costs and is unwilling to determine experimentally the optimal training distribution, we then suggest that for maximizing AUC the training set be formed from equal numbers of examples of each class.

This article presents an empirical study of the effect of class distribution on classifier learning. The goal of this article is not how to find the optimal class distribution efficiently. However, if in practice it is cost effective to procure data incrementally, we suggest that a progressive, adaptive, sampling strategy (Provost, Jensen & Oates, 1999) be developed that incrementally requests new examples based on the improvement in classifier performance due to the recently added minority-class and majority-class examples. The best class distribution can then be estimated by using cross-validation.

Our results are based only on C4.5, a decision-tree learner. Nevertheless, we believe our conclusions will hold for other learners as well, since we believe the effect that the training class distribution has on classifier performance is not specific to C4.5, or to decision-tree learners. In particular, we believe that the reasons provided in Section 2.2.1 for why classifiers perform differently on the minority class versus the majority class also apply to other learners.

When faced with complex learning problems that involve highly unbalanced data sets, practitioners often modify the class distribution of the training set. However, these modifications are seldom done in a principled manner and the reasons for changing the distribution are often not fully understood. This article provides a deeper understanding of how class distribution affects learning and discusses at length the issues involved when the class distribution is modified. In particular, we provide several explanations for why the minority class generally has a higher error rate than the majority class, and compare the performances of classifiers generated from balanced and unbalanced data sets. We also demonstrate that by changing the class distribution of the training set one can change the *distribution* of errors between the minority-class and the majority-class.

One of the areas addressed in this article is how a learning algorithm can be modified to account for changes made to the training distribution, so the resulting classifier is not unduly biased. We quantify the performance penalty that occurs if the change in class distribution is not properly accounted for and show that the penalty, for accuracy, can be quite substantial. Nonetheless, the correction we suggest is often ignored, both in research and in practice.

We hope these results help researchers and practitioners to understand better the relationship between the training class distribution and classifier performance, and to learn more effectively from large data sets in situations where the training-set size must be limited.

Acknowledgments

We thank Brian Davison, Chris Mesterharm and Matthew Stone for their comments on this article and Haym Hirsh for the comments and feedback he provided throughout this research. We also thank IBM for a Faculty Partnership Award.

References

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning*, 36, 105-139.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Blake, C., & Merz, C. (1998). UCI Repository of Machine Learning Databases, (http://www.ics.uci.edu/~mlearn/MLRepository.html), Department of Computer Science, University of California.

Bradley, A. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), 1145-1159.

Bradford, J.P., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In *Proceedings of the European Conference on Machine Learning*, pp. 131-136.

Catlett, J. (1991). *Megainduction: machine learning on very large databases.* Ph.D. thesis, Department of Computer Science, University of Sydney.

Chan, P., & Stolfo, S. (1998). Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 164-168, Menlo Park, CA: AAAI Press.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. P. (2000). SMOTE: synthetic minority over-sampling technique. In *International Conference on Knowledge Based Computer Systems*.

Cohen, W., & Singer, Y. (1999). A simple, fast, and effective rule learner. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 335-342, Menlo Park, CA: AAAI Press.

Drummond, C., & Holte, R.C. (2000). Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 239-246.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*. To be published.

Good, I. J. (1965). *The Estimation of Probabilities*. Cambridge, MA: M.I.T. Press.

Hand, D. J. (1997). *Construction and Assessment of Classification Rules.* Chichester, UK: John Wiley and Sons.

Holte, R. C., Acker, L. E., & Porter, B. W. (1989). Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 813-818. San Mateo, CA: Morgan Kaufmann.

Japkowicz, N. (2000). Learning from imbalanced data sets: a comparison of various strategies. In *Papers from the AAAI Workshop on Learning from Imbalanced Data Sets.* Tech. rep. WS-00-05, Menlo Park, CA: AAAI Press.

Japkowicz, N., Holte, R. C., Ling, C. X., & Matwin S. (Eds.) (2000). In *Papers from the AAAI Workshop on Learning from Imbalanced Data Sets*. Tech, rep. WS-00-05, Menlo Park, CA: AAAI Press. Remove or cite it.

Jensen, D. D., & Cohen, P. R. (2000). Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309-338.

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179-186.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing classifiers. In *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*. ACM Press.

Provost, F., & Fawcett, T (2001). Robust classification for imprecise environments. *Machine Learning*, 42, 203-231.

Provost, F., & Domingos, P. (2001). Well-trained PETs: improving probability estimation trees. CeDER Working Paper #IS-00-04, Stern School of Business, New York University, New York, NY.

Quinlan, J.R. (1993). C*4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Swets, J., Dawes, R., & Monahan, J. (2000). Better decisions through science. *Scientific American*, October 2000: 82-87.

Turney P. (2000). Types of cost in inductive learning. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, 15-21, Stanford, CA.

Weiss, G.M., & Hirsh, H. (2000). A quantitative study of small disjuncts, *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 665-670. Menlo Park, CA: AAAI Press.

Zadrozny, B., & Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. Tech. rep. CS2001-0664, Department of Computer Science and Engineering, University of California, San Diego.

Appendix A

The Importance of Adjusting a Classifier to Compensate for Changes to the Class Distribution of the Training Set

Throughout this article we have relied on the corrected frequency-based estimate, described in Section 2.3, to label the leaves of each induced decision tree, so that the tree is not biased by the alterations in the class distribution of the training set. In Table A1 below, we compare the decision trees labeled with the corrected frequency-based estimate (CT-FB) with the uncorrected version (FB). This comparison is based on a training distribution that was modified to contain an *equal number* of minority-class and majority-class examples (the test distribution uses the natural distribution). The data sets are listed in order of decreasing class imbalance.

In Table A1, the second column lists the error rates for the uncorrected and corrected frequency-based estimates (the lowest error rate for each data set is underlined). The third column specifies the relative improvement (i.e., reduction) in error rate that results from using the corrected frequency-based estimate instead of the uncorrected version. The fourth specifies the percentage of the leaves for which use of the corrected estimate leads to a different class value than the uncorrected version. The last column specifies the percentage of the total errors that are contributed by the minority-class test examples.

	Error Rate		% Rel.	% Labels	% Min. Errs	
Dataset	FB	CT-FB	Improv.	Changed	FB	CT-FB
letter-a	9.7	5.3	46.0	41	2.5	6.9
pendigits	4.2	4.1	1.0	5	4.2	6.3
abalone	30.4	20.7	32.0	9	8.5	25.6
sick-euthyroid	10.0	9.1	9.1	4	9.7	10.3
connect-4	29.9	27.4	8.3	14	8.7	10.4
optdigits	6.3	3.3	47.6	46	5.1	18.8
solar-flare	36.0	27.6	23.4	21	19.7	35.5
letter-vowel	19.4	14.4	25.7	44	15.5	29.5
contraceptive	41.3	39.3	4.7	12	21.6	27.5
adult	23.3	20.6	11.4	32	20.1	36.3
splice-junction	8.2	8.2	0.1	14	18.7	27.9
network2	31.0	30.0	3.4	2	32.6	39.1
yeast	31.6	28.3	10.6	4	29.6	46.4
network1	31.4	29.9	4.5	2	33.9	43.1
car	7.5	7.6	-2.4	5	25.1	37.4
german	37.2	35.8	4.0	16	31.5	35.3
breast-wisc	6.8	6.8	0.0	0	44.0	44.0
blackjack	31.5	28.2	10.5	17	47.6	77.9
weather	34.6	34.6	-0.1	0	40.2	40.9
bands	34.2	34.2	0.0	0	92.6	92.6
market1	26.6	26.4	0.9	26	44.0	48.1
crx	19.5	21.2	-9.0	19	47.0	55.3
kr-vs-kp	1.1	1.1	0.0	0	76.9	76.9
move	28.6	28.6	-0.1	21	52.0	60.8
coding	33.1	33.1	0.0	0	61.2	61.2
Average	22.9	21.0	8.3	14	31.7	39.8
Median	28.6	26.4	4.3	13	30.6	38.3

Table A1: Impact of the Training Distribution Correction on Error Rate

Table A1 shows that by employing the corrected frequency-based estimate instead of the uncorrected frequency-based estimate, the error rate of the classifiers is reduced, on average, 8.3%. Furthermore, note that the classifiers labeled with the corrected version of the frequency-based estimate have an equal or lower error rate for all but 2 of the 25 data sets. The correction tends to yield a larger reduction for the most highly unbalanced data sets—where it plays a larger role (i.e., causes a greater percentage of the leaves to change class value). If we restrict ourselves to the 11 data sets where the class ratio is greater than 3:1, then the relative improvement over these data sets is 19.0%. Use of the corrected frequency-based estimate will reduce the number of leaves in the decision tree labeled with the minority class. Consequently, as the last column in the table demonstrates, the corrected version of the estimate will cause more of the errors to come from the minority-class test examples.