Computational Intelligence, Volume 20, Number 1, 2004

A MULTIPLE RESAMPLING METHOD FOR LEARNING FROM IMBALANCED DATA SETS

ANDREW ESTABROOKS
IBM Toronto Lab, Canada

TAEHO JO AND NATHALIE JAPKOWICZ
University of Ottawa, Canada

Resampling methods are commonly used for dealing with the class-imbalance problem. Their advantage over other methods is that they are external and thus, easily transportable. Although such approaches can be very simple to implement, tuning them most effectively is not an easy task. In particular, it is unclear whether oversampling is more effective than undersampling and which oversampling or undersampling rate should be used. This paper presents an experimental study of these questions and concludes that combining different expressions of the resampling approach is an effective solution to the tuning problem. The proposed combination scheme is evaluated on imbalanced subsets of the Reuters-21578 text collection and is shown to be quite effective for these problems.

Key words: inductive learning, decision trees, class imbalance problem, multiple resampling, text classification.

1. INTRODUCTION

In a concept-learning problem, the data set is said to present a class imbalance if it contains many more examples of one class than the other. Such a situation poses challenges for typical classifiers such as decision tree induction systems or multilayer perceptrons that are designed to optimize overall accuracy without taking into account the relative distribution of each class (Estabrooks 2000; Japkowicz and Stephen 2002). As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately. Unfortunately, this problem is quite pervasive as many domains are cursed with a class imbalance. This is the case, for example, with text classification tasks whose training sets typically contain much fewer documents of interest to the reader than on irrelevant topics. Other domains suffering from class imbalances include target detection, fault detection, or fraud detection problems, which, again, typically contain much fewer instances of the event of interest than of irrelevant events.

A large number of approaches have previously been proposed to deal with the class-imbalance problem.[1] These approaches can be categorized into two groups: the *internal* approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration (Pazzani et al. 1994; Riddle, Secal, and Etzioni 1994; Japkowicz, Myers, and Gluck 1995; Kubat, Holte, and Matwin 1998) and *external* approaches that use unmodified existing algorithms, but resample the data presented to these algorithms so as diminish the effect caused by their class imbalance (Lewis and Gale 1994; Kubat and Matwin 1997; Ling and Li 1998). The internal approaches just mentioned may, in certain cases, be quite effective, but they have the disadvantage of being algorithm specific. This is a problem because data sets presenting different characteristics are better classified by different algorithms (see, for example, Weiss and Kapouleas 1990), and it might be quite difficult—if not, sometimes, impossible—to transport the modification proposed for the class-imbalance problem from one classifier to the other. External approaches, on the other hand,

Address correspondence to Nathalie Japkowicz at SITE, University of Ottawa, 800 King Edward, P.O. Box 450 Station A, Ottawa, Ontario, Canada, K1N 6N5; e-mail: aestabro@ca.ibm.com; nat@siteuottawa.ca; tjo018@siteuottawa.ca

[1] See Estabrooks (2000) for a full review of these works.

are independent of the classifier used and are, thus, more versatile. This is why we chose to focus on these approaches rather than internal ones in this study.

External approaches may, themselves, be divided into two types of categories. First, there are approaches that focus on studying what the best *data* for inclusion in the training set are (Lewis and Gale 1994; Kubat and Matwin 1997) and, second, there are approaches that focus on studying what the best *proportion* of positive and negative examples to include in a training set is (Ling and Li 1998). We decided to focus on the second question with the idea that once a good framework for dealing with the proportion question is chosen, this framework can be refined by making "smarter" resampling choices as per the first category of external approaches.

In more detail, our study considers the two different categories of *resampling* approaches: methods that *oversample* the small class to make it reach a size close to that of the larger class and methods that *undersample* the large class to make it reach a size close to that of the smaller class. The purpose of this paper is to find the best way to tune the resampling paradigm. In particular, we ask the following three questions:

- Should we *oversample* or *undersample*?
- At what *rate* should this oversampling or undersampling take place?
- Can a *combination* of different expressions of the resampling paradigm help improve classification accuracy?

These questions are answered in the context of a decision tree induction system: C4.5, and all resampling is done randomly.

The paper is organized as follows: Section 2 establishes the problems caused by the class-imbalance problem by studying its effect on different artificial and real-world domains. In Section 3, we conduct an experimental study on some of these data sets to explore the problems of oversampling versus undersampling and of finding optimal resampling rates (the first two questions asked above). This study suggests an answer to the third question in the form of a combination scheme that is described in Section 4. In Section 5, the combination scheme is tested, first, on the artificial and real-world data sets used in Sections 2 and 3 and, second, on the top 10 categories of the Reuters-21578 text collection. In the first series of experiments, the combination scheme is pitted against C4.5 applied to the oversampled and undersampled data on domains presenting a very large imbalance. It is shown that the combination scheme is generally more successful than the other methods on these domains. In the second series of experiments, the class imbalances are less drastic, and the combination scheme is pitted against another, very robust, general-purpose combination scheme: Adaboost (using C4.5 as its basic learner). There again, our specialized combination scheme is shown to prevail.

2. THE EFFECTS OF CLASS IMBALANCES

In this section, we study the effect of class imbalances on three categories of domains. The first category consists of data sets representing target concepts of various complexities. In this particular series of domains, the size of the training set is held constant, which means that, as the target concept (represented by the positive class) becomes more complex, the positive class becomes sparser relative to the target concept.[2] This study is relevant because, in real-world data sets, we often encounter situations where the target concept is quite complex, but there are not enough data available to describe it. The second category of domains was taken from the UCI Repository while the third one belongs to the Reuters-21578 data set.

[2] A similar but more thorough study relating different degrees of imbalance ratios, training set sizes, and concept difficulty was conducted by Japkowicz and Stephen (2002). However, that study falls beyond the scope of this paper.

In the first category of domains, seven sets of training and testing data of increasing complexities were created over the domain of DNF expressions. DNF expressions were specifically chosen because of their simplicity as well as their similarity to text data whose classification accuracy we are ultimately interested in improving. In particular, like in the case of text classification, DNF concepts of interest are, generally, represented by much fewer examples than there are counter-examples of these concepts, especially when (1) the concept at hand is fairly specific; (2) the number of disjuncts and literals per disjunct (in the concept to be learnt) is large; and (3) the values assumed by the literals are drawn from a large alphabet. Furthermore, an important aspect of concept complexity can be expressed in similar ways in DNF and textual concepts because adding a new subtopic to a textual concept corresponds to adding a new disjunct to a DNF concept.

The target concepts in the data sets were made to vary in concept complexity by increasing the number of disjunctions in the expression to be learned, while keeping the number of literals in each disjunct constant. In particular, expressions of complexity $c = 4 \times 4$, 4×5, 4×6, 4×7, 4×8, 4×9, and 4×10 were created where the first number represents the number of literals present in each disjunct and the second represents the number of disjuncts in each concept. We used an alphabet of size 50. For each concept, we first created a training set containing 6,000 positive and 6,000 negative examples. We then (1) randomly removed 4,800 positive examples from the training set, thus creating a 1:5 class imbalance in favor of the negative class and (2) randomly removed 960 extra examples from the training set, thus creating a 1:25 class imbalance in favor of the negative class. We then repeated the process, creating class imbalances in favor of the positive class. In all five cases (no class imbalance, a 1:5 class imbalance and a 1:25 class imbalance, both in favor of each class), we tested the classifier on 6,000 positive and 6,000 negative examples. For each expression, the results obtained by C4.5 were averaged over 10 runs on different domains of the same complexity.

In the second category of domains, three standard data sets were chosen: Wisconsin Breast Cancer, Pima Indian Diabetes, and Classification of Grass versus Path Images. These three domains are highly challenging binary classification domains and presented the advantage of being a lot smaller than the domains of the first and third categories. This allowed us to observe the behavior of C4.5 on very small class-imbalanced domains.

In the third category of domains, we chose the two top categories of Reuters-21578. Because our study is ultimately geared at text classification, we wanted to keep track of the problem at hand in our preliminary investigation.

As mentioned previously, in all the domains tested, we considered both negative-dominant imbalances (the cases where there are more negative examples than positive ones) and positive-dominant imbalances (the opposite case).

Table 1 presents the distribution of the number of training and testing examples used in the series of experiments we ran in this part of the paper to test the influence of class imbalances on classification performance. In each table cell, the number on the left of the colon represents the number of positive examples in the data set while the right number represents the number of negative examples.[3]

[3]It is common for researchers working with class-imbalanced data sets to evaluate their results using the same distribution in the training and the testing sets. In this paper, we chose not to do so and, thus, to place more emphasis on the underrepresented class than standard accuracy would. (In other words, we are assuming that the cost of misclassifying the underrepresented class is higher than that of misclassifying the dominant class.) We follow this nonstandard approach because our goal is not to measure predictive accuracy, but rather to measure the ability of the learner to learn each class. We believe that our assumption regarding the cost of misclassifying data from the underrepresented class is usually valid since research on class imbalanced data sets is often undertaken for that very reason. Note, however, that in the case where predictive accuracy is really the goal, it can easily be deduced from our graphs by combining the results obtained on the positive and the negative class and combining them, using the desired ratio.

TABLE 1. The Number of Training Examples with Respect to Each Ratio and each Domain

Domains		Training: Balance 1:1	Training: Negative-dominant 1:5	Training: Negative-dominant 1:25	Training: Positive-dominant 5:1	Training: Positive-dominant 25:1	Testing Balance
DNF expression	4 × 4	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 5	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 6	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 7	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 8	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 9	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
	4 × 10	6000:6000	1200:6000	240:6000	6000:1200	6000:240	6000:6000
UCI repository	Breast	150:150	30:150	6:150	150:30	150:6	50:50
	Pima	200:200	40:200	8:200	200:40	200:8	50:50
	Image	250:250	50:250	10:250	250:50	250:10	50:50
Reuter 21578	Earn	2500:2500	500:2500	100:2500	2500:500	2500:100	1000:1000
	ACQ	1500:1500	300:1500	60:1500	1500:300	1500:60	800:800

The results of our experiments are presented in Figures 1–3.

Figure 1 illustrates the influence of class imbalances on the classification of domains from the first category, DNF expressions. In Figure 1(a–c) we report the error on the balanced test set; the error obtained on the positive test set alone (i.e., we show the percent of false positives); and the error obtained on the negative test set alone (i.e., we show the percent

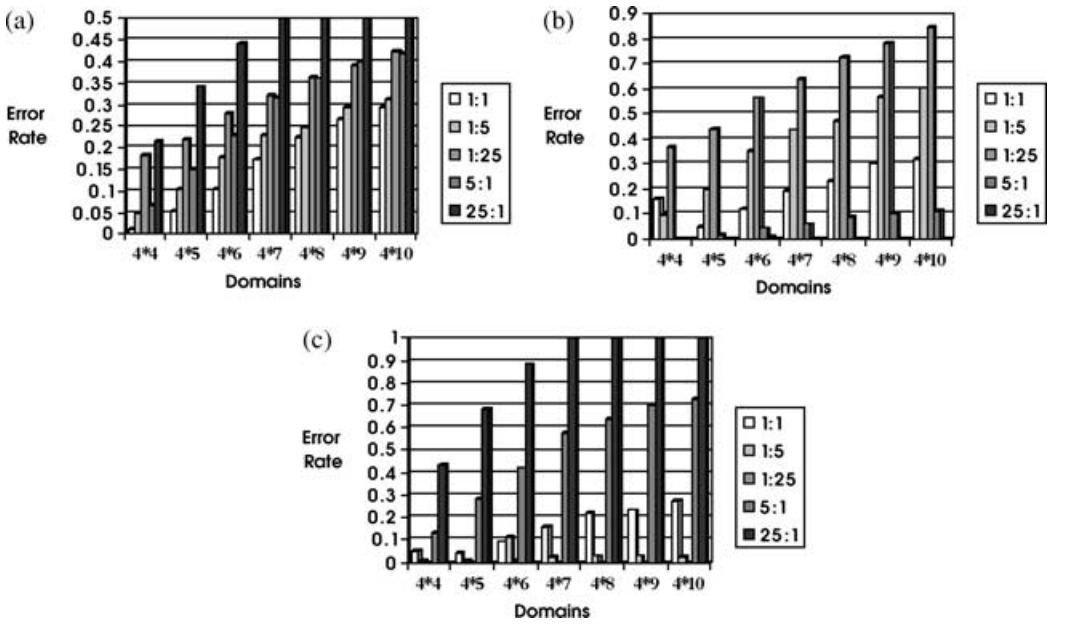

FIGURE 1. The effect of class imbalance on test data in dnf expression (a) shows the effect on the overall balanced set, (b) does so for the positive test set only, and (c) does so for the negative test data only.

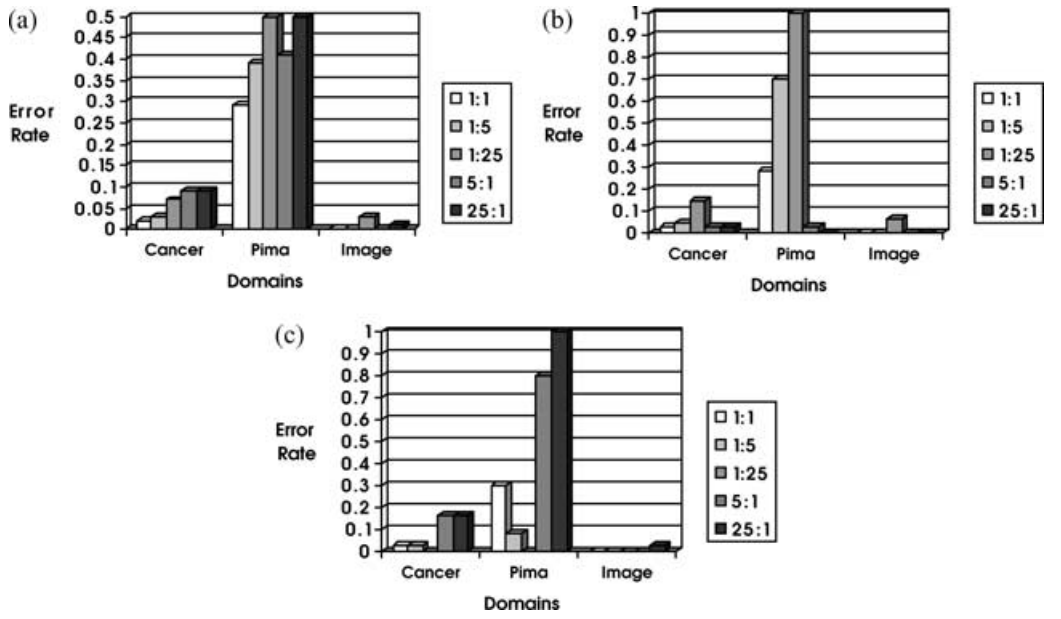

FIGURE 2. The effect of class imbalance on test data in the UCI domains (a) shows the effect on the overall balanced set, (b) does so for the positive test set only, and (c) does so for the negative test data only.

of false negatives). Our results show that the more complex the DNF expression, the higher the error rate. Note that in all the graphs, 1:5 and 1:25 correspond to negative-dominant class imbalances, while 5:1 and 25:1 are positive-dominant class imbalances. Taking this into consideration, it is clear that the classifier is always biased in favor of the dominant class. There is a difference, however, between positive- and negative-dominant imbalances: we see in Figure 1(a) that the positive-dominant class imbalances yield higher error rates than the negative-dominant ones. In DNF expressions, 4×7, 4×8, 4×9, and 4×10, the error rate is 0.5 (or 50%) in the positive-dominant 25:1 class imbalances. As shown in Figure 1(c), this is due to the fact that all the negative examples are misclassified as positive ones. This can be explained as follows: the positive class is more concise than the negative one because it represents a given concept while the negative class represents everything but that concept. When the imbalance is in favor of the positive class, the classifier will naturally by-pass any negative examples that are difficult to describe concisely, given their sparseness and their low degree of representation.

Figure 2 shows the influence of class imbalances on three domains from the UCI Repository (Wisconsin Breast Cancer, Pima Indian Diabetes, Image Classification of Path or Grass). The results show that in both the Cancer and Pima domains, both kinds of class imbalances hamper the performance of C4.5. In the Image domain, however, class imbalances have little if any noticeable effect on classification accuracy. Figures 2(b) and (c) show that the trends observed with the DNF data sets with regard to both general dominance and positive- versus negative-dominance, are also the ones followed in the UCI domains.

Figure 3 displays the results obtained on two selected domains from Reuters-21578.[4] In these experiments, the number of positive and negative examples was manipulated to obtain the desired imbalance ratios. This was done by removing examples at random. The

[4]See Section 5 of the paper to gather more detail about the construction of these data sets from the raw Reuters data.

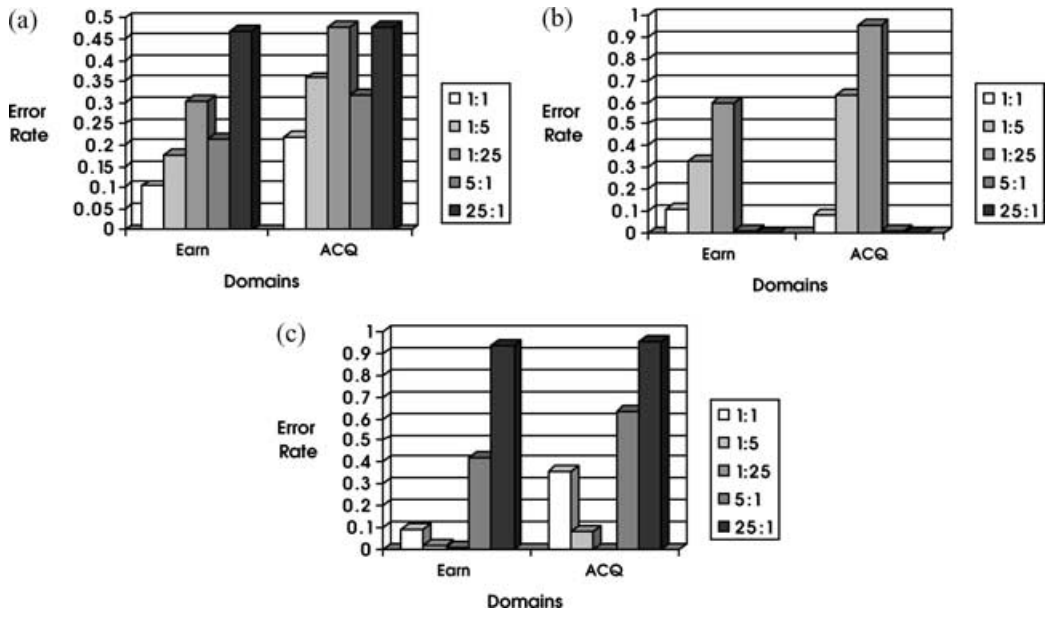

FIGURE 3. The effect of class imbalance on test data in the Reuters domains (a) shows the effect on the overall balanced set, (b) does so for the positive test set only, and (c) does so for the negative test data only.

results show clearly that, once again, class imbalances impair classification performance. Furthermore, as in the first two groups of domains, Figures 3(b) and (c) confirm the trend in misclassification already reported.

The results of this section can be generalized as follows: class imbalances usually tend to hamper the classification performance of C4.5. The data belonging to the dominating class tend to be very well classified while those belonging to the minor class tend to be misclassified. Furthermore, these results get amplified in the case of a positive rather than negative dominance. All these trends were seen in all the domains except for the UCI Image classification domain that did not seem much affected by the class imbalance.

3. OVERSAMPLING VERSUS UNDERSAMPLING

In this section, we study the effects of oversampling versus undersampling and oversampling or undersampling at different rates. The section is divided into two subsections. In the first section, we study the effect of oversampling versus undersampling when both methods keep on resampling until the imbalance has completely vanished. The second section considers the question of resampling at different rates rather than until the two classes get fully balanced.

3.1. Oversampling and Undersampling to Full Balance

The purpose of this section is to explain the effects of full oversampling and undersampling on the classification of imbalanced domains. To illustrate these effects, a subset of the domains in Section 1 was used: 4×7 DNF concepts, Wisconsin Breast Cancer, Pima Indian Diabetes, Earn and ACQ. Each domain was designed with a 1:25 class imbalance in favor of each class in turn.

TABLE 2. The Number of Training and Test Examples in this Experiment

	Train			Test	
Domain	Imbalanced	Over	Under	Positive	Negative
4 × 7	6000:240 240:6000	6000:6000	240:240	6000	6000
Breast	150:6 6:150	150:150	6:6	50	50
Pima	200:8 8:200	200:200	8:8	50	50
Earn	2500:100 100:2500	2500:2500	100:100	1000	1000
ACQ	1500:60 60:1500	1500:1500	60:60	800	800

Table 2 summarizes the number of training and test examples before and after resampling took place. This experiment considers both positive- and negative-dominant class imbalances and resampling was applied to both of them. Five domains participated in this experiment, as illustrated in this table.

Resampling was conducted using the following strategies: oversampling consisted of copying existing training examples at random and adding them to the training set until a full balance was reached. Undersampling consisted of removing existing examples at random until a full balance was reached. Since each run of this experiment consists of different sets of resampled training examples, the error reported is the average of 25 repeated runs.

Figure 4 shows the effect of the oversampling and undersampling strategies on three of the domains that were tested. These domains were selected because they each depict a different situation. In Figure 4(a), we see a case in which oversampling is more useful than undersampling, which, actually hurts the performance of C4.5. This is a rare case that occurred only in the Wisconsin Breast Cancer data set with a negative-dominant imbalance. The negative-dominant 4 × 7 DNF expression case is related to the Wisconsin Breast Cancer case because oversampling was also more useful than undersampling. However, undersampling did not hurt C4.5's performance. The most common case observed in all our domains is the one depicted in Figure 4(b), which displays the results obtained on the Pima Indian Diabetes domain with a negative-dominant imbalance. In this domain, both the oversampling and undersampling strategies help but undersampling helps more than oversampling. Finally, Figure 4(c) represents the other rare case where the oversampling and undersampling strategies are about as helpful in ACQ with a positive-dominant imbalance. The same type of result also occurred in the case of 4 × 7 DNF expression with a positive-dominant imbalance.

Table 3 summarizes our results by showing what type of trend was observed in each domain.

Altogether, our results suggest that neither the oversampling nor the undersampling strategy is *always* the best one to use, and finding a way to combine them could perhaps be useful, especially if the bias resulting from each strategy is of a different nature. Figure 4(b), which represents the most common case, suggests that the biases resulting from the oversampling and undersampling based methods are, indeed, different because C4.5 trained on the undersampled data presents a reduced error on the positive testing examples and an increased one

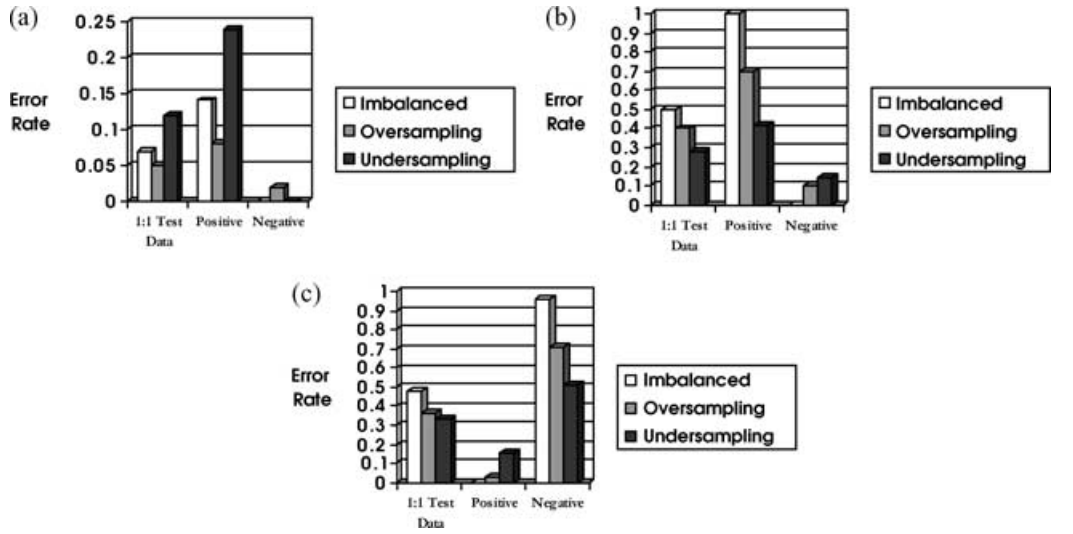

FIGURE 4. The comparison of resampling methods in the case of class imbalances. All three characteristic cases are displayed. (a) Case where oversampling is better than undersampling (Wisconsin Breast Cancer with negative-dominant imbalance). (b) Case where undersampling is better than oversampling (Pima Indian Diabetes with negative-dominant imbalance). (c) Case where the two resampling methods are almost equivalent (ACQ with positive-dominant imbalance). Note: the error rate on the 1:1 (balanced) data set is expressed as the average of the positive and negative error.

on the negative ones that are relatively more significant than those it displays when it is trained on the oversampled data. Our experiments, thus, motivate the search for a combination of the two resampling methods rather than the selection of either of them for an entire data set.

3.2. Oversampling and Undersampling at Various Rates

The purpose of this section is to find out what happens when different oversampling or undersampling rates are used, and whether the effect of using different resampling rates is the same for different domains. To illustrate our answer to these questions, we considered the same domains as in Section 3.1 However, this time, rather than simply oversampling and undersampling our domains by equalizing the size of the positive and the negative training

TABLE 3. Summary of the Resampling Results over Every Domain

Oversampling Surpasses Undersampling	Undersampling Surpasses Oversampling	Undersampling is Equivalent to Oversampling
• Negative-dominant Wisconsin Breast Cancer	• Positive-dominant Wisconsin Breast Cancer	• Positive-dominant ACQ
• Negative-dominant 4×7 DNF expressions	• Negative-dominant ACQ	• Positive-dominant 4×7 DNF expressions
	• Pima (both dominances)	
	• Earn (both dominances)	

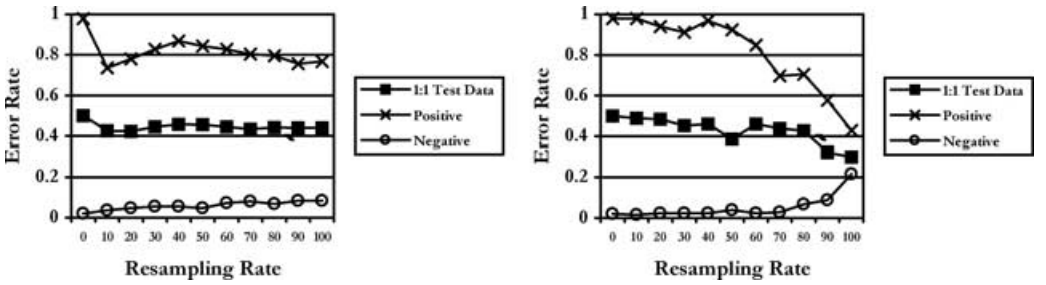

FIGURE 5. The effect of oversampling and undersampling at different rates on the Pima Indian diabetes dataset with a negative-dominant imbalance: (a) oversampling and (b) undersampling.

sets, our experiments consisted of oversampling and undersampling them at different rates. In particular, we divided the difference between the size of the positive and negative training sets by 10 and used this value as an increment in our oversampling and undersampling experiments. We then chose to make the 100% oversampling rate correspond to the fully oversampled data sets of Section 3.1 but to make the 90% undersampled rate correspond to its fully undersampled data sets.[5] For example, data sets with a 10% oversampling rate contain $240 + (6{,}000 - 240)/10 = 816$ positive examples and 6,000 negative examples. Conversely, data sets with a 0% undersampling rate contain 240 positive examples and 6,000 negative ones while data sets with a 10% undersampling rate contain 240 positive examples and $6{,}000 - (6{,}000 - 240)/10 = 5424$ negative examples. A 0% oversampling rate and a 90% undersampling rate correspond to the fully imbalanced data sets designed in Section 3.1, whereas a 100% undersampling rate corresponds to the case where no negative examples are present in the training set.

The results are reported for a single domain, Pima Indian Diabetes, which was already considered characteristic of most data sets in Section 3.1. Figure 5 shows the results obtained on the negative-dominant version of the problem whereas Figure 6 shows the results obtained on the positive-dominant version of the problem.

The results of these experiments allow us to make two remarks of interest. First, resampling to full balance is not necessarily optimal (e.g., in Figure 5(a), optimality is reached at a rate of 20% resampling) and second, the best resampling rate is not always the same (e.g., in Figure 5(a), it occurs at a rate of 20% while in Figure 6(a), it occurs at a rate of 80%). The results on all the other domains but one—Wisconsin Breast Cancer—were similar to those obtained in the Pima Indian Diabetes case and lead to the same observations. Furthermore, to reach additional conclusions, we summarized our results in terms of general trends of the effect of resampling in Table 4. This table shows that the effect of resampling on imbalanced domains is stable and gradual on the full test set. However, its effect is different on the positive and the negative test sets considered separately. Within each class, changes tend to be radical in the case of undersampling and gradual in the case of oversampling. This suggests yet another difference in the way C4.5 applied to undersampled and oversampled data behave.

As mentioned before, the one domain that did not exhibit the types of trends just described is the Wisconsin Breast Cancer data set. In this domain, the results were abnormally stable (in the case of resampling) or abnormally unstable (in the case of undersampling). This probably results from the fact that the six examples in the underrepresented class were not sufficient to describe the concept at hand, no matter how often they were duplicated in the

[5]This was done so that no classifier was duplicated in our combination scheme (see Section 4.1).

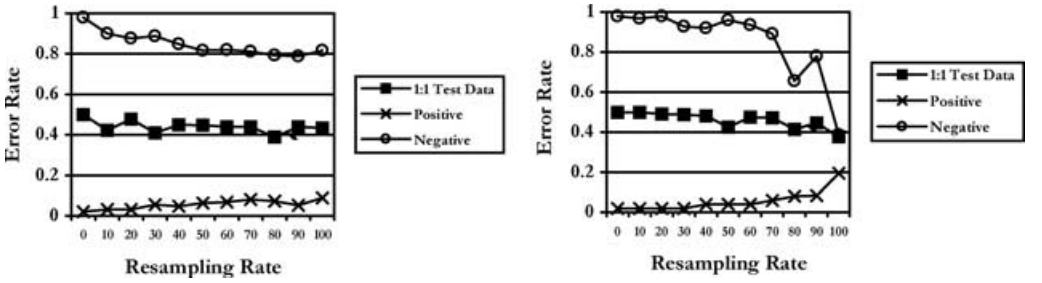

FIGURE 6. The effect of oversampling and undersampling at different rates on the Pima Indian diabetes dataset with a positive-dominant imbalance: (a) oversampling and (b) undersampling.

TABLE 4. The Summary of Resampling Trends in all Domains but Wisconsin Breast Cancer

		1:1	Negative	Positive
Negative-based class imbalance	Over	Gradual reduction	Gradual increment	Gradual reduction
	Under	Gradual reduction	Radical increment	Radical reduction
Positive-based class imbalance	Over	Gradual reduction	Gradual reduction	Gradual increment
	Under	Gradual reduction	Radical reduction	Radical increment

oversampled approach and that a balanced data set of 12 examples is insufficient for this domain, as demonstrated in the case of the undersampled approach.

All in all, the experiments of this section suggest that resampling to full balance is generally not the optimal resampling rate, at least when the test set is balanced. Furthermore, the optimal resampling rate varies from domain to domain and resampling strategy to resampling strategy. Another possible observation is that there, generally, is a trade-off between the two resampling methods with respect to their effect on the positive and negative test data considered separately. In general, oversampling changes its effect gradually and in a stable manner with different rates, while undersampling does so radically and in an unstable manner.

4. MULTIPLE RESAMPLING METHODS

The results obtained in Section 3 suggest that it might be useful to combine C4.5 trained on data that were oversampled and undersampled at different rates. On the one hand, the combination of the oversampling and undersampling strategies may be useful given the fact that the two approaches are both useful in the presence of imbalanced data sets and appear to learn concepts in different ways (cf. results of Sections 3.1 and 3.2). On the other hand, the combination of learning systems using different oversampling and undersampling rates may be useful because optimal resampling rates are different in different domains and we may not be able to predict, in advance, which rate is optimal given a new domain (cf. results of Section 3.2). We will now describe the combination scheme we designed to deal with the class-imbalance problem.[6] This combination scheme is first tested on some artificial domains

[6]Actually, as will be seen below, although we refer to it as to a *combination* scheme, our method implements, in fact, a *selection* scheme. Indeed, the output of the system is not the result of a combination of the output of each classifier present. Rather, the system chooses a classifier for every input data which, *alone*, will determine the outcome of that input data.

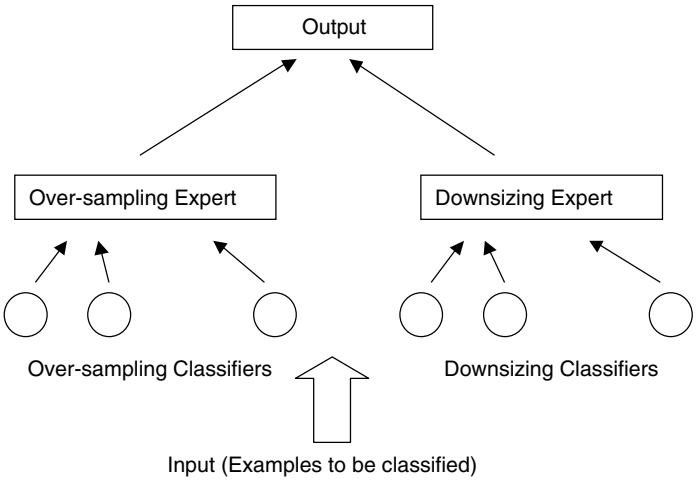

FIGURE 7. The architecture of the multiple resampling method.

and it is then tested on a series of imbalanced subsets of the Reuters-21578 text classification domain.

A combination scheme for inductive learning consists of two parts. On the one hand, we must decide *which* classifiers will be combined and on the other hand, we must decide *how* these classifiers will be combined. We begin our discussion with a description of the architecture of our multiple resampling scheme. This discussion explains which classifiers are combined and gives a general idea of how they are combined. The specifics of our combination scheme are motivated and explained in the subsequent part of the discussion.

4.1. Architecture

For a combination method to be effective, it is necessary for the various classifiers that constitute the combination to make different decisions (Hansen and Salamon 1990). The previous part of our study suggests that undersampling and oversampling will produce classifiers able to make different decisions. Furthermore, different sampling rates will allow us to "hit" an optimal rate, which could not be predicted in advance. This suggests a three-level hierarchical combination approach consisting of the *output level*, which combines the results of the oversampling and undersampling experts located at the *expert level*, which themselves each combine the results of 10 classifiers located at the *classifier level* and that resulted from learners trained on data sets sampled at different rates. In particular, the 10 oversampling learners are trained on data that were oversampled at rates 10%, 20%, . . . , 100% (the underrepresented class is oversampled until the two classes are of the same size) and the 10 undersampling learners are trained on data that had its dominant-class undersampled at rates 0% (no undersampling), 10%, . . . , 90% (the dominant class is undersampled until the two classes are of the same size). Figure 7 illustrates the architecture of this combination scheme that was motivated by Shimshoni and Intrator (1998)'s Integrated Classification Machine.[7]

[7]However, Shimshoni and Inntrator (1998) presents a general architecture. It was not tuned to the imbalance problem, or did it take into consideration the use of oversampling and undersampling to inject principled variance into the different classifiers. Another notable difference is that Shimshoni and Inntrator (1998) uses ensemble methods to combine his various classifiers whereas we do not.

4.2. Detailed Combination Scheme

Our combination scheme is based on two different assumptions/observations.

Assumption 1. Within a single testing set, different testing points could best be classified by different single classifiers.

Observation 2. In class-imbalanced domains, classifiers tend to make many classification errors on the nondominant class (see Section 1).

To deal with the first assumption, we decided not to average the outcome of different classifiers by letting them vote on a given testing point, but rather to let a single "good enough" classifier make a decision on that point. The classifier selected for a single data point needs not be the same as the one selected for a different data point. In general, letting a single, rather than several classifiers decide on a data point is based on the assumption that the instance space may be divided into nonoverlapping areas, each best classified by a different expert.[8] In such a case, averaging the result of different classifiers may not yield the best solution. We, thus, created a combination scheme that allowed single but different classifiers to make a decision for each point.

Of course, such an approach is dangerous given that if the single classifier chosen to make a decision on a data point is not reliable, the result for this data point will not be reliable either. To prevent such a problem, we designed an elimination procedure geared at preventing any unfit classifier present at our architecture's classification level from participating in the decision-making process. This elimination program relies on the results of C4.5 applied in a tenfold cross-validation fashion to the original imbalanced training data. The individual classifiers of the combination scheme (resulting from learning systems that were trained with various rebalanced versions of the training set) displaying lower error rates than the average tenfold cross-validation error are selected and the learning systems from which they stemmed are trained again, but this time, not using the cross-validation approach. The others are eliminated from the combination scheme.

In more detail, our combination scheme consists of

- a combination scheme applied to each expert at the expert level;
- a combination scheme applied at the output level;
- an elimination scheme applied to the classifier level.

The expert and output level combination schemes use the same very simple heuristic: if one of the noneliminated classifiers decides that an example is positive, so does the expert to which this classifier belongs. Similarly, if one of the two experts decides (based on its classifiers' decision) that an example is positive, so does the output level, and thus, the example is classified as positive by the overall system.

It is important to note that, at the expert and output level, our combination scheme is heavily biased towards the underrepresented class. This was done as a way to compensate for the natural bias against that class embodied by the individual classifiers trained on the class-imbalanced domain. This heavy bias in favor of the underrepresented class, however,

[8]This way of viewing classification tasks was first proposed in (Jacobs et al. 1991) who designed a combination method that trains (1) several neural networks, each on a different part of the input space and (2) a "gating" neural network whose purpose is to oversee the entire operation by directing each input data towards the "right" classifying neural network.

is mitigated by our elimination scheme, which strenuously eliminates any classifier believed to be too biased towards that class.

5. EXPERIMENTS AND RESULTS

This section will compare the proposed approach for learning in the presence of imbalanced data sets to C4.5, C4.5 resampled, and Adaboost. This will be done through two series of experiments. In the first series, the data from the five domains previously used in Section 2 will be tested and the proposed approach will be compared to resampling methods that resample to full balance. In the second series, the most frequent 10 categories of the Reuter-21578 collection will be used, and the proposed approach will be compared to C4.5 and Adaboost.

5.1. Classification in Artificial, UCI, and two Reuters Domains

The purpose of this series of experiments is to compare the proposed approach to C4.5 in the context of class imbalances, on several domains. In this series of experiments, the ratios of class imbalances are fixed at 1:25 and 25:1. The proposed approach is compared to (1) C4.5 applied to the original imbalanced data, (2) C4.5 applied to the oversampled data, and (3) C4.5 applied to the undersampled data.

The evaluation will be done using two measures: the error rate and receiver operating characteristic (ROC) curves. The first measure will be applied to balanced test examples, negative ones, and positive ones. The second measure is based on the ratio of the true positive rate in positive examples to the false positive rate in negative examples. In more detail, in ROC analysis, a curve is plotted representing the number of false positives on the x-axis and the number of true positives on the y-axis. The number of false positives corresponds to the number of negative examples wrongly classified as positive whereas the number of true positives corresponds to the number of positive examples that were rightly classified as positive. The performance of a classifier is considered better than that of another one if its representative curve is higher than that of the other system. This measure comes from the signal detection literature where it was used to characterize the trade-off between the hit rate and the false alarm rate. It was popularized in the machine learning community by Provost and Fawcett (2001).

Note that the two measures are different from one another and that a particular approach may obtain good results when evaluated by one method and bad results when evaluated by the other one.

We show the results obtained on two domains and, as before, to save space, we summarize the other results in Table 4. Figures 8(a) and (b) show the results obtained on the negative-dominant version of the Pima Indian Diabetes data set. Figure 8(a) reports the results with respect to the accuracy of the method while (b) focuses on the ROC curves. Figures 9(a) and (b) report on similar results for the positive-dominant version of the Earn category of Reuters. Figure 8 is an example where the combined method is about equivalent to (though slightly worse than) the undersampled approach whereas Figure 9 shows an example where the combined approach is clearly superior.

Table 5 summarizes the result of this series of experiment. Its rows correspond to the domain and the class dominance while its columns correspond to the approaches that participated in this experiment and the evaluation method. The entry in each cell indicates the rank of the performance corresponding to the approach, the domain, and the type of class dominance (1 is the best rank and 4 the worst). The values in bold correspond to a win for

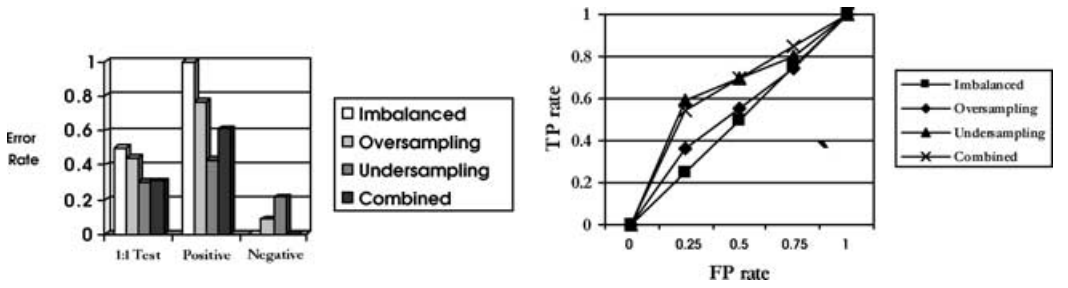

FIGURE 8. Results on the Pima Indian diabetes domain with negative-dominant imbalance: (a) error rates and (b) ROC curves.

the proposed approach. The table shows that such cases are very frequent and this experiment thus allows us to conclude that the proposed method performs generally better than any resampling method that resamples blindly to full balance.

5.2. Text Classification

Because our combination scheme was shown to help increase classification accuracy on several classes of domains, we also decided to test it systematically on a practical domain. In particular, we chose to test it on a subset of the 10 largest categories of the Reuters-21578 data set. Unlike in the previous section, in this case, we do not manipulate the ratio of the training data: we leave the natural imbalance untouched. We first present an overview of the data, followed by the results obtained by our scheme on them.

The 10 largest categories of the Reuters-21578 data set consist of the documents included in the classes of financial topics listed in Table 6.

Several typical preprocessing steps were taken to prepare the data for classification. First, the data was divided according to the ModApte split which consists of considering all labeled documents published before April 7, 1987 as training data (9,603 documents, altogether) and all labeled documents published on or after April 7, 1987 as testing data (3,299 documents altogether).

Second, the documents were transformed into feature vectors in several steps. Specifically, all the punctuation and numbers were removed and the documents were filtered through

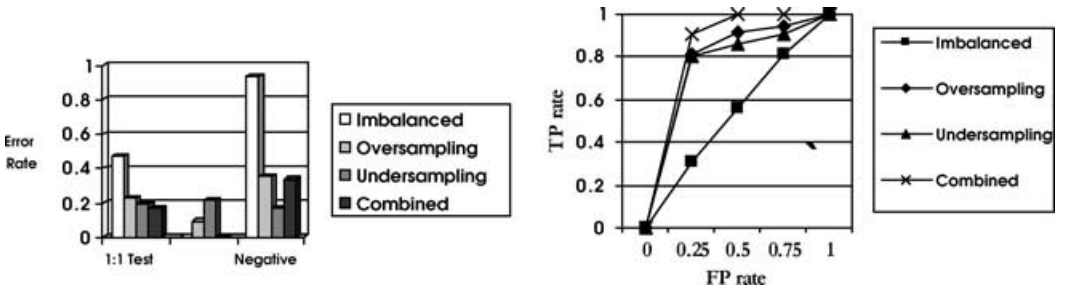

FIGURE 9. Results on the Pima Indian diabetes domain with positive-dominant imbalance: (a) error rates and (b) ROC curves.

TABLE 5. The Summary of the Experiments on the Artificial, UCI, and Reuters Domains

		Performance in 1:1 Test Data				ROC Curve			
		Imbalance	Over	Under	Proposed	Imbalance	Over	Under	Proposed
DNF 4 × 7	**1:25**	**3**	**2**	**4**	**1**	**4**	**3**	**2**	**1**
	25:1	4	2	1	3	4	1	2	3
Cancer	**1:25**	**4**	**2**	**3**	**1**	**4**	**2**	**3**	**1**
	25:1	**3**	**4**	**2**	**1**	**2**	**4**	**3**	**1**
Pima	**1:25**	4	3	1	2	**4**	**3**	**2**	**1**
	25:1	4	3	1	2	4	3	1	2
Reuter Earn	**1:25**	**4**	**3**	**2**	**1**	4	3	1	2
	25:1	**4**	**3**	**2**	**1**	**4**	**2**	**3**	**1**
Reuter ACQ	**1:25**	**4**	**3**	**2**	**1**	**4**	**3**	**2**	**1**
	25:1	**4**	**3**	**2**	**1**	**4**	**2**	**3**	**1**

a stop word list.[9] The words in each document were then stemmed using the Lovins stemmer[10] and the 100 most frequently occurring words were used as the dictionary for the bag-of-word vectors representing each document.[11] Finally, the data set was divided into 10 concept-learning problems where each problem consisted of a positive class containing all the examples in a single top 10 Reuters topic class and a negative class containing the union of all the examples contained in the other 9 top 10 Reuters classes. Dividing the Reuters multiclass data set into a series of two class problems is typically done because considering the problem as a straight multiclass classification problem causes difficulties due to the high class-overlapping rate of the documents, i.e., it is not uncommon for a document to belong to several classes simultaneously.

The results obtained by our scheme on these data were pitted against those of C4.5. However, because we decided that it was not fair to compare the effectiveness of a system of 20 learners to that of a single learner, we also ran C4.5 with the Adaboost option combining 20 learners.[12] The results of these experiments are reported in Figure 10 as a function of the microaveraged (over the 10 different classification problems) Recall, Precision and F_1 measures. Figure 11 shows the same results but for the macroaverage.

In more detail, Precision, Recall, and the F_1 measures are defined as follows:

$$P = \text{TruePositives}/(\text{TruePositives} + \text{FalsePositives})$$
$$R = \text{TruePositives}/(\text{TruePositives} + \text{FalseNegatives})$$
$$F_1 = (2 \times P \times R)/(P + R)$$

where P is the precision, R, the recall, and F_1, the F_1 measure.

[9]The stop word list was obtained at: http://www.dcs.gla.ac.uk/idom/it_resources/linguistic_utils/stop-words.

[10]The Lovins stemmer was obtained from: ftp://n106.isitokushima-u.ac.ip/pub/IR/Iterated-Lovins-stemmer.

[11]A dictionary of 100 words is smaller than the typical number of words used (see, e.g., Scott and Matwin 1999), however, our results show that this restricted size did not affect the results too negatively while it did reduce processing time quite significantly.

[12]C5.0, a cousin of C4.5, was shown in Estabrooks (2000) to obtain results close to those obtained by state-of-the-art classifiers designed for text classification. We expected Adaboost to obtain even better results than C4.5 given that it is currently considered one of the best general-purpose classification algorithm (Breiman 1998). Another generic combination method, Bagging, was not used since it generally performs worse than Adaboost.

TABLE 6. The top 10 Reuters-21578 Categories

Class	Document Count
Earn	3987
ACQ	2448
MoneyFx	801
Grain	628
Crude	634
Trade	551
Interest	513
Wheat	306
Ship	305
Corn	254

More informally, precision corresponds to the proportion of examples classified as positive that are truly positive; recall corresponds to the proportion of truly positive examples that were classified as positive; and the F_1 measure combines precision and recall in a way that considers them as being of equal importance. Because 10 different results are obtained for each combination system (one result per classification problem), these results had to be averaged to be presented in a single graph. Microaveraging consists of the summation of contingency tables of categories. This method considers that each category has different weights based on its number of news articles. Macroaveraging consists of a straight average of the F_1

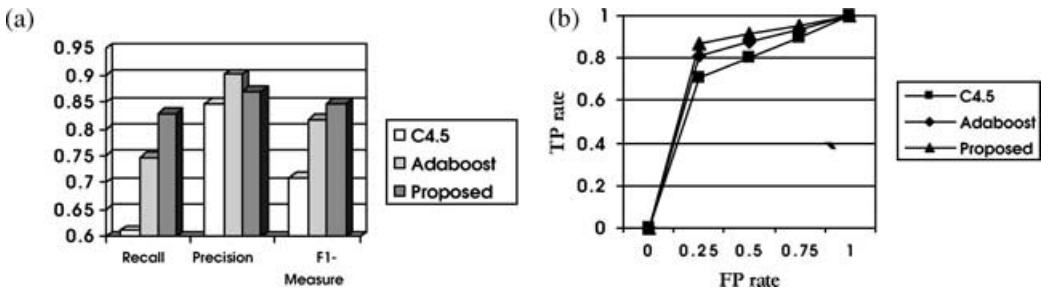

FIGURE 10. Micro-averaged results on Reuters 10 top categories: (a) error rates and (b) ROC curves.

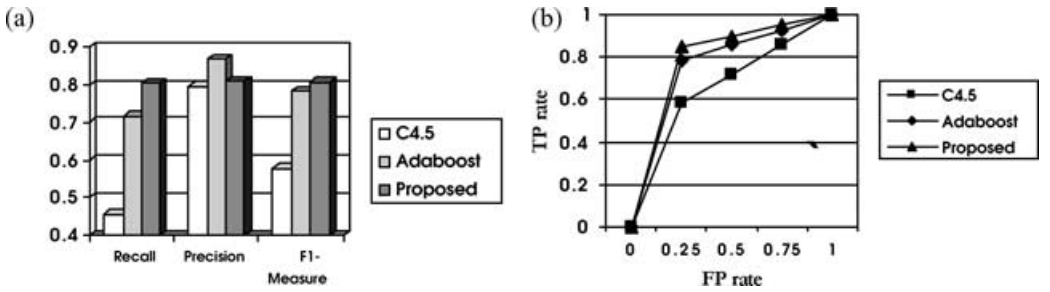

FIGURE 11. Macro-averaged results on Reuters 10 top categories: (a) error rates and (b) ROC curves.

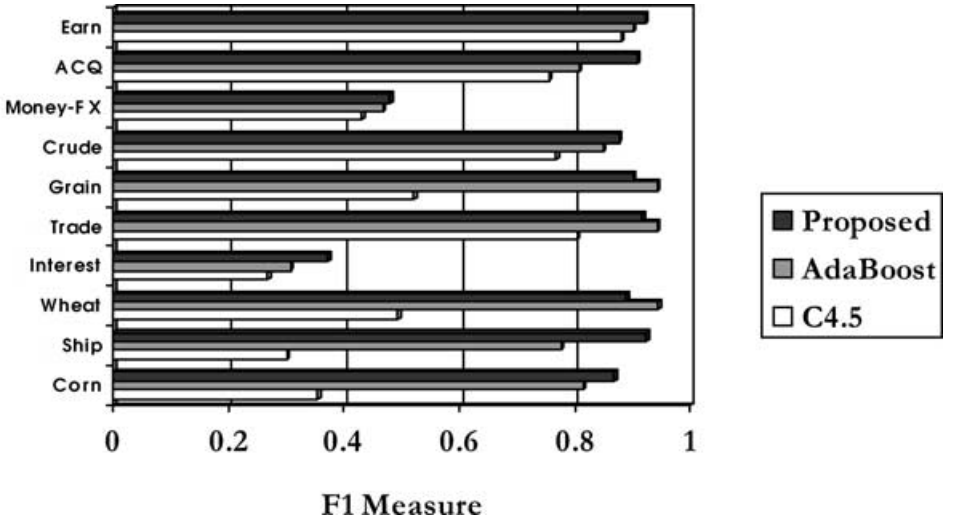

FIGURE 12. F1 measure on each individual domain for the proposed combination scheme, Adaboost, and C4.5.

measure obtained in all the problems, by each combination system. Using macroaveraging gives each problem the same weight, independently of the number of examples they contain.

Figure 11 shows the microaveraged results of text classification, with the assumption that each category has its different weight based on its number of news articles. Although the proposed method is worse than Adaboost in terms of precision, its recall value is excellent, leading to a better general integrated performance in the context of both the F_1 measure and the ROC curves.

Figures 11(a) and (b) show the macroaveraged results of text classification, with the assumption that each category has the same weight. The distribution of macroaveraged performance is similar to that of the microaveraged one.

Finally, Figure 12 shows the results for each individual domain considered. The domains are ordered from bottom to top as a function of increasing class-imbalance ratios. There does not seem to be any correlation between the size of the class-imbalance ratio and the performance of the proposed method relative to the other two approaches. On the other hand, this figure shows us that our method prevails over C4.5 in all cases and over Adaboost in 7 out of 10 cases. This means that our proposed method does as well on various degrees of imbalance and may even be helpful on fully balanced data sets (although, we leave this issue as a future investigation).

Our experiments, thus, confirm that the proposed method performs better than not only a single learner but also a good-performing combination method such as Adaboost, on class-imbalanced problems.

6. RELATED WORK, CONCLUSION AND FUTURE WORK

This paper presented an approach for dealing with the class-imbalance problem that consisted of combining different expressions of resampling based learners in an informed fashion. In particular, our combination system was built so as to bias the learners toward the underrepresented set to counteract the bias typically developed by learners facing a higher

proportion of examples from one class than from the other. The bias we included was carefully regulated by an elimination strategy designed to prevent unreliable classifiers to participate in the process. The technique was shown to be effective on a subset of the Reuters text classification task as compared to a single learner and another general-purpose combination method, Adaboost.

The work presented in this paper is related to two notable studies. The first one is by Weiss and Provost (2003). Their study attempts to find out what data distribution is optimal in a classification problem. Based on results they obtained on a large number of domains, they conclude that the naturally occurring data distribution is not necessarily optimal. Their work is related to our search for and ultimate combination of different class-imbalance ratios. The second study is by Chawla, Hall, and Kegelmeyer (2002). Like in our work, their study attempts to combine both oversampling and undersampling strategies. Their oversampling method is quite sophisticated, but on the other hand, they do not look at different class distribution ratios the way we do.

For the future, there are different ways in which this study could be expanded. First, although experimental results bode well for our method, it would be interesting to conduct an analysis of why the scheme works. In particular, we could study the system's various components separately and explain their specific roles. Such a study, we expect, could lead to a simplification and a strengthening of our framework. For example, we could do an analysis of which classifier gets selected when and eliminate those that are never involved in the classification procedure. Furthermore, we could find ways to eliminate those that are often selected, and often issue an erroneous classification.[13] Second, the technique we presented was used in the context of a very naive oversampling and undersampling scheme. It would be useful to apply our scheme to more sophisticated resampling approaches such as those of Kubat and Matwin (1997) or Chawla et al. (2002). Third, it would be interesting to find out whether our combination approach could also improve on the cost-sensitive techniques previously designed. Fourth, it would be interesting to see how well our scheme generalizes to learning algorithms other than C4.5. Finally, we would like to test our technique on other domains presenting various degrees of class imbalances and compare our results to learning systems other than C4.5 and Adaboost.

ACKNOWLEDGMENTS

The authors would like to thank Chris Drummond and Rob Holte for their valuable comments as well as the three anonymous reviewers who reviewed an early version of this paper. This research was funded in part by an NSERC grant. The initial part of the work described in this paper was conducted at Dalhousie University.

REFERENCES

BREIMAN, L. 1998. Combining Predictors. Technical Report, Statistics Department, 1998.

CHAWLA, N., L. HALL, and W. KEGELMEYER. 2002. SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research, **16**:321–357.

[13] In an early investigation of this issue, it was found that not all classifiers participated in the overall classification of the test data. Instead, a few classifiers were very often used while many were seldom used. This may be due to the fact that the imbalance ratios we selected are not the most appropriate ones, and we leave it to future work to investigate the question of what imbalance ratios to actually include in our scheme.

ESTABROOKS, A. 2000. A Combination Scheme for Inductive Learning from Imbalanced Data Sets. MCS Thesis, Faculty of Computer Science, Dalhousie University.

HANSEN, L. K., and P. SALAMON. 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, **12**(10), 993–1001.

JACOBS, R. A., M. I. JORDAN, S. J. NOWLAN, and G. E. HINTON. 1991. Adaptive mixtures of local experts. Neural Computation, **3**:79–87.

JAPKOWICZ, N., and S. STEPHEN. 2002. The class imbalance problem: a systematic study. Intelligent Data Analysis, **6**(5):429–450.

JAPKOWICZ, N., C. MYERS, and M. GLUCK. 1995, A novelty detection approach to classification. *In* Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, Montreal, Que, pp. 518–523.

KUBAT, M., and S. MATWIN. 1997, Addressing the curse of imbalanced data sets: one-sided sampling. *In* Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, pp. 179–186.

KUBAT, M., R. HOLTE, and S. MATWIN. 1998, Machine learning for the detection of oil spills in satellite radar images. Machine Learning, **30**:195–215.

LEWIS, D., and W. GALE. 1994, Training text classifiers by uncertainty sampling. *In* Proceedings of the Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland.

LING, C., and C. LI. 1998, Data mining for direct marketing: problems and solutions. *In* Proceedings of KDD-98.

PAZZANI, M., C. MERZ, P. MURPHY, K. ALI, T. HUME, and C. BRUNK. 1994, Reducing misclassification costs. *In* Proceedings of the Eleventh International Conference on Machine Learning, New Bronswick, NJ, pp. 217–225.

PROVOST, F., and T. FAWCETT. 2001, Robust classification for imprecise environments. Machine Learning Journal, **42**(3).

RIDDLE, P., R. SECAL, and O. ETZIONI. 1991: Representation design and brute-force induction in a Boeing manufacturing domain. Applied Artificial Intelligence, **8**:125–147.

SCOTT, S., and S. MATWIN. 1999, Feature engineering for text classification. *In* Proceedings of the Sixteenth International Conference on Machine Learning, Bled, Slovenis, pp. 379–388.

SHIMSHONI, Y., and N. INTRATOR. 1998, Classifying seismic signals by integrating ensembles of neural networks. IEEE Transactions On Signal Processing, Special issue on Neural Networks **46**(5):1194–120?.

WEISS, S., and I. KAPOULEAS. 1990, An empirical comparison of pattern recognition, neural nets and machine learning methods. *In* Readings in Machine Learning. *Edited by* J. W Shavlik and T. G. Dietterich. Morgan-Kauffman, San Maeto, CA.

WEISS, G., and F. PROVOST. 2003, Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research, **19**: 315–354.