A Multiple Resampling Method for Learning from Imbalanced Data Sets

Andrew Estabrooks
IBM Toronto Lab,
Office 1B28B (1B/813/1150/TOR)
1150 Eglinton Avenue East,
North York, Ontario,
Canada, M3C 1H7
aestabro@ca.ibm.com

Taeho Jo and **Nathalie Japkowicz**[*]
SITE, University of Ottawa
800 King Edward,
P.O. Box 450 Stn. A
Ottawa, Ontario
Canada, K1N 6N5
{tjo018, nat}@site.uottawa.ca

abstract

Re-Sampling 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 re-sampling 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.

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 Multi-Layer Perceptrons that are designed to optimize overall accuracy without taking into account the relative distribution of each class (Japkowicz & Stephen 2002; Estabrooks 2000). 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.

[*] Corresponding author

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 et al. 1994; Japkowicz et al. 1995;Kubat et al. 1998) and *external* approaches that use un-modified existing algorithms, but re-sample the data presented to these algorithms so as diminish the effect caused by their class imbalance (Lewis & Gale 1994; Kubat & Matwin 1997;Ling & 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 since data sets presenting different characteristics are better classified by different algorithms (see, for example, (Weiss & 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, are independant 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 & Gale 1994; Kubat & 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 & 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 marter” re-sampling 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 in order to make it reach a size close to that of the larger class and methods that *undersample* the large class in order 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 re-sampling 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 re-sampling paradigm help improve classification accuracy?

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

The paper is divided into four parts. The first part establishes the problems caused by the class imbalance problem by studying its effect on different artificial and real-world domains. In the second part, we conduct an experimental study on some of these data sets in order to explore the problems of oversampling versus undersampling and of finding optimal re-sampling 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

[1] For a full review of these works, please consult (Estabrooks 2000).

described in the third part of the paper. In the fourth part, the combination scheme is tested, first, on the artificial and real-world data sets used in Parts I and II of the paper and, second, on the top ten categories of the Reuters-21578 text collection. In the first series of experiments, the combination scheme is pitted against the oversampled and undersampled scheme on data sets 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, but the combination scheme is pitted against another, very robust, general-purpose combination scheme: Adaboost. There again, our specialized combination scheme is shown to prevail.

Part I: The Effects of Class Imbalances

In this part of the paper, 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 since, 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.

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 grows larger; 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 since 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 conjunctions in each disjunct constant. In particular, expressions of complexity c= 4x4, 4x5, 4x6, 4x7, 4x8, 4x9 and 4x10 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,

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

thus creating a 1:5 class imbalance in favour of the negative class and 2) randomly removed 960 extra examples from the training set, thus creating a 1:25 class imbalance in favour of the negative class.[3] In all three cases (no class imbalance, a 1:5 class imbalance and a 1:25 class imbalance), we tested the classifier on 6,000 positive and 6,000 negative examples. For each expression, the results of 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 and particularly imbalanced.

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

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.

Table 1. The Number of Training Examples to each Ratio and each Domain

		|-----------	-------Training-----		----------------------|		Testing
Domains		Balance	Negative Dominant		Positive Dominant		Balance
		1:1	1:5	1:25	5:1	25:1	
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

[3] Imbalanced ratios greater than 1:25 were not tried on this particular problem since we did not want to confuse the imbalance problem for the small sample problem.

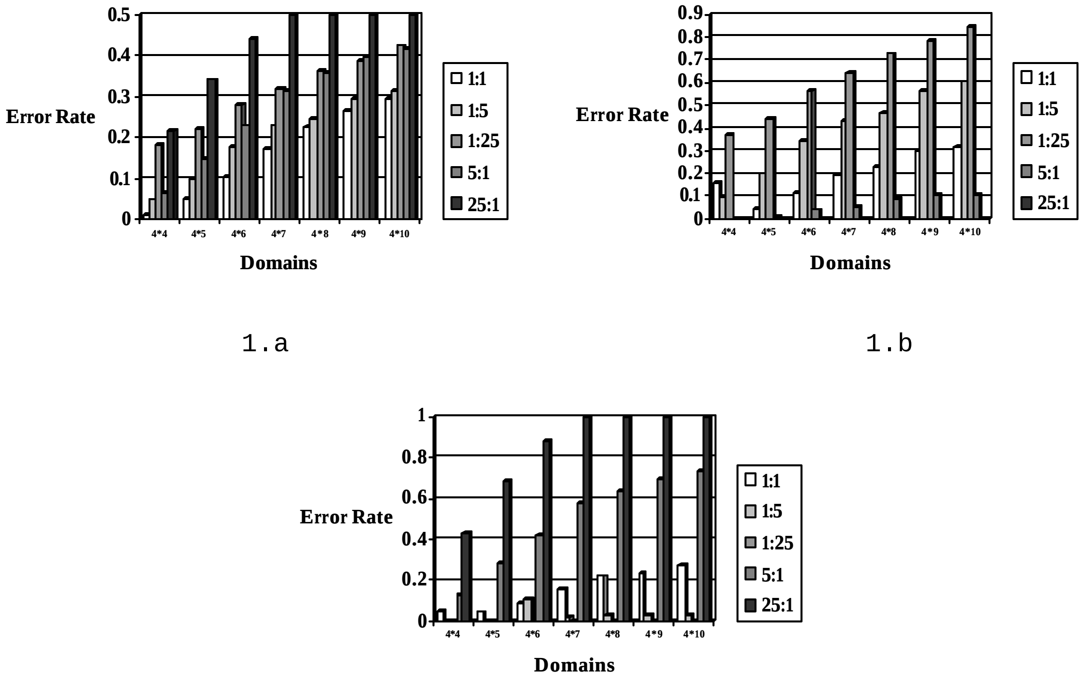
The results of our experiments are presented in Figures 1, 2 and 3.

Figure 1 illustrates the influence of class imbalances on the classification of domains from the first category, DNF expressions. In figure 1.a we report the error on the balanced test set; in figure 1.b, we show the error obtained on the positive test set alone (i.e., we show the percent of false positives); and in figure 1.c, we show the error obtained on the negative test set alone (i.e., we sho w the percent 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 favour 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 figures 1.b and 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 since it represents a given concept while the negative class represents everything but that concept. When the imbalance is in favour of the positive class, the classifier will naturally by-pass any negative examples that are difficult to describe concisely, given their parseness 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 2.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 in order to obtain the desired exagerated imbalance ratios. This was done by removing examples at random. The results show clearly that, once again, class imbalances impair classification performance. Furthermore, as in the first two groups of domains, figures 3.b and 3.c confirm the trend in misclassification already reported.

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

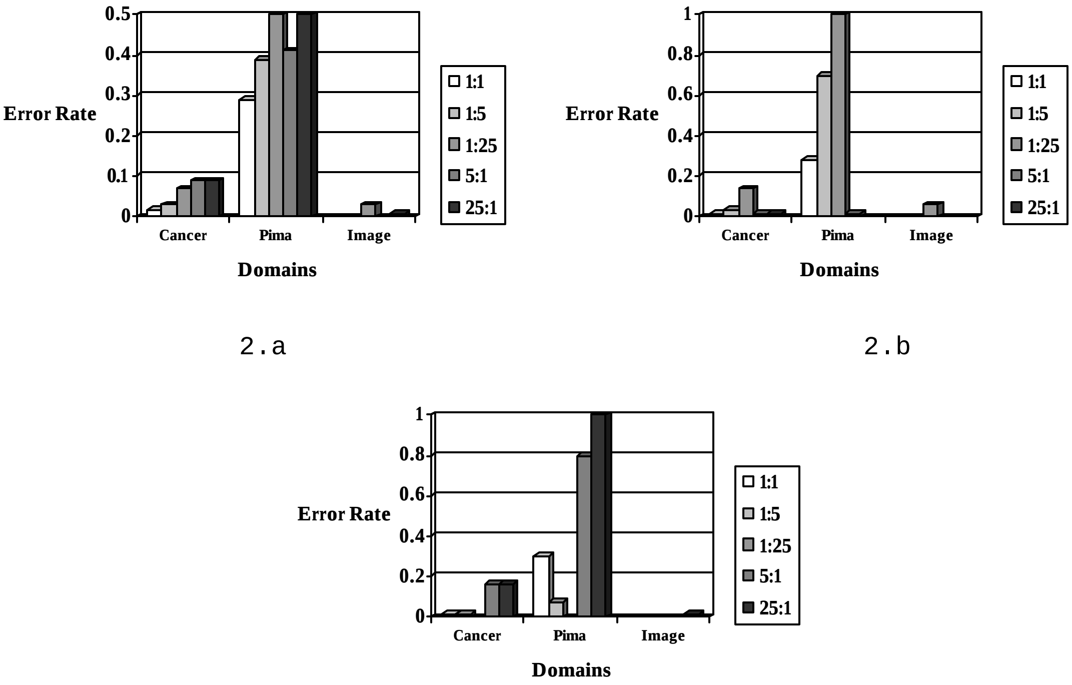


1.a

1.b

1.c

Figure 1: The Effect of Class Imbalance on Test Data in DNF Expression. Figure 1.a shows the effect on the overall balanced set; Figure 1.b does so for the positive test set only and Figure 1.c does so for the negative test data only.

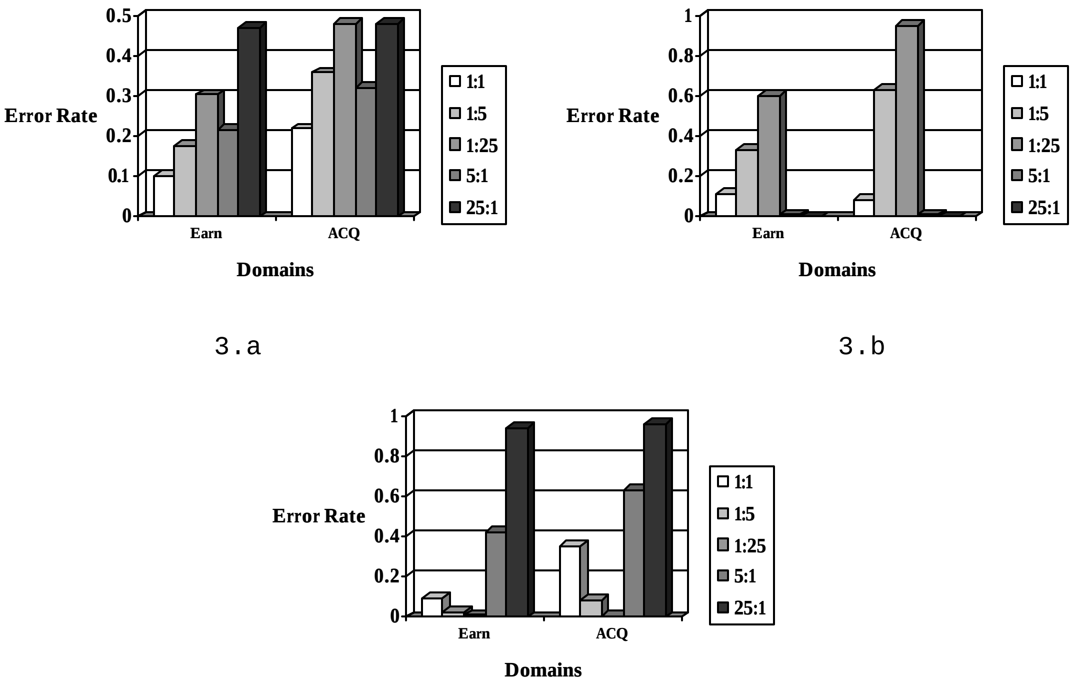


2.a

2.b

2.c

Figure 2: The Effect of Class Imbalance on Test Data in the UCI Domains. Figure 2.a shows the effect on the overall balanced set; Figure 2.b does so for the positive test set only and Figure 2.c does so for the negative test data only.



3.a

3.b

3.c

Figure 3: The Effect of Class Imbalance on Test Data in the Reuters Domains. Figure 3.a shows the effect on the overall balanced set; Figure 3.b does so for the positive test set only and Figure 3.c does so for the negative test data only.

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 which didn seem much affected by the class imbalance.

Part II: Over-Sampling versus Under-Sampling

In this part of the paper, we study the effects of oversampling versus undersampling and oversampling or undersampling at different rates.[5] The part is divided into two sections. In the first section, we study the effect of over-sampling versus under-sampling when both methods keep on re-sampling 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.

[5] Throughout the experiments of this section, we consider a fixed imbalance ratio, a fixed number of training examples and a fixed degree of concept complexity.

2.1 Over-Sampling and Under-Sampling to Full Balance

The purpose of this section is to explain the effects of full oversampling and undersampling on imbalanced domains. In order to illustrate these effects, a subset of the domains of Part I was used: 4x7 DNF concepts, Wisconsin Breast Cancer, Pima Indian Diabetes, Earn and ACQ. Each domain was designed with a 1:25 class imbalance in favour of each class in turn.

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.

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

Domain	Train			Test	
	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

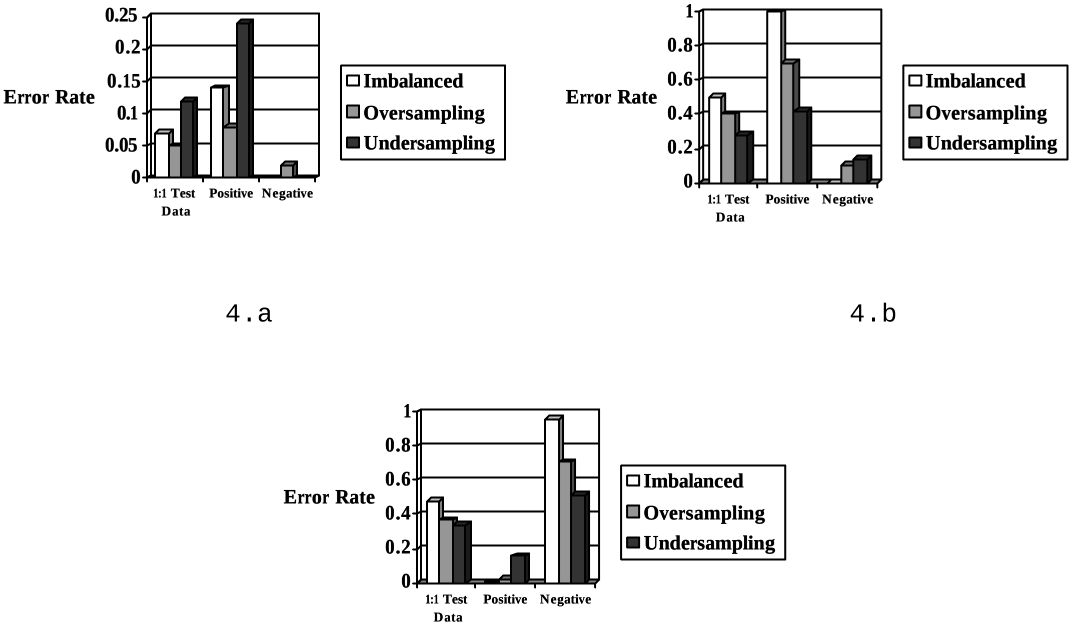
Re-sampling was conducted as follows: oversampling consisted of copying existing training examples at random and adding them to the training set until a full balance was reached. Undersampling consisted of removing existing examples at random until a full balance was reached. Since each run of this experiment 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 when a negative-dominant. The negative-dominant 4*7 DNF expression case is related to the Wisconsin Breast Cancer case since oversampling was also more useful than undersampling. However, undersampling did not hurt C4.5 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 oversampling and undersampling help but undersampling helps more than oversampling. Finally, Figure 4.c represents the other rare case where oversampling and undersampling 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.

Oversampling surpasses Undersampling	Undersampling surpasses Oversampling	Undersampling is equivalent To Oversampling
• Negative-dominant Wisconsin Breast Cancer • Negative-dominant 4*7DNF Expressions	• Positive-dominant WisconsinBreast Cancer • Negative-dominant ACQ • Pima(both dominances) • Earn (both dominances)	• Positive-dominant ACQ • Positive-dominant 4*7 DNF Expressions

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 employed by each strategy is of a different nature. Figure 4.b, which represents the most common case, suggests that the biases of oversampling and undersampling methods are, indeed, different since undersampling reduces the error on the positive examples and increases the error on the negative ones relatively more than oversampling. Our experiments, thus, motivate the search for a combination of the two re-sampling methods rather than the selection of either of them.



4.a

4.b

4.c

Figure 4. The Comparison of Resampling Methods in the case of Class Imbalances. All three characteristic cases are displayed. Figure 4.a shows the case where oversampling is better than undersampling (Wisconsin Breast Cancer with negative-dominant imbalance). Figure 4.b shows the case where undersampling is better than oversampling (Pima Indian Diabetes with negative-dominant imbalance). Figure 4.c shows the case where the two resampling methods are almost equivqlent (ACQ with positive-dominant imbalance).

2.2. 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. In order to illustrate our answer to these questions, we considered the same domains as in Section 2.1 However, this time, rather than simply oversampling and undersampling our domains by equalizing the size of the positive and the negative training 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 2.1 but to make the 90% undersampled rate correspond to its fully undersampled data sets.[6] 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

[6] This was done so that no classifier was duplicated in our combination scheme. (See Section 3.1)

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 2.1 while 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 always considered characteristic of most data sets in Section 2.1. Figure 5 shows the results obtained on the negative-dominant version of the problem while Figure 6 shows the results obtained on the positive-dominant version of the problem.

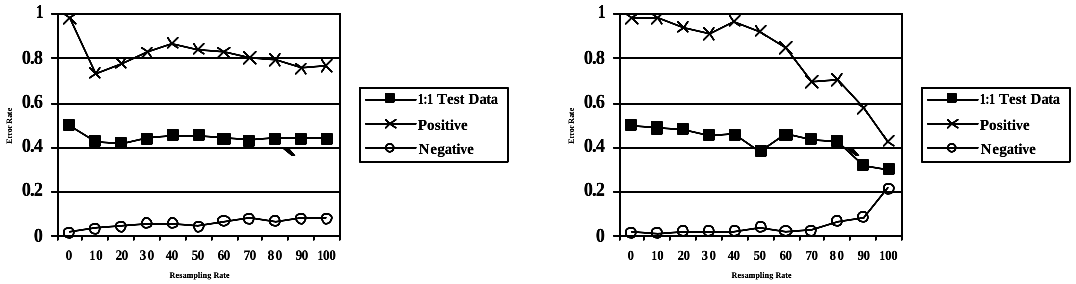


Figure 5: The Effect of Oversampling and Undersampling at different rates on the Pima Indian Diabetes dataset with a negative-dominant imbalance. 5.a: Oversampling; 5.b: Undersampling.

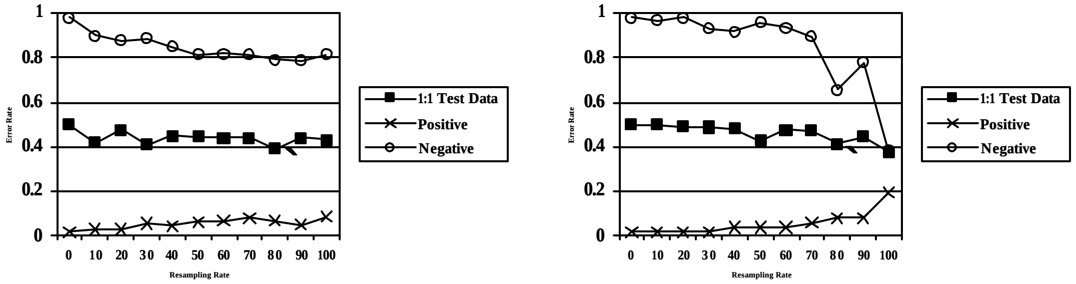


Figure 6: The Effect of Oversampling and Undersampling at different rates on the Pima Indian Diabetes dataset with a positive-dominant imbalance. 6.a: Oversampling; 6.b: Undersampling.

The results of these experiments allow us to make two remarks of interest. First, re-sampling to full balance is not necessarily optimal (e.g., Figure 5.a where optimality is reached at 20% re-sampling) and second, the best re-sampling rate is not always the same (e.g., in Figure 5.a, it occurs at 20% while in Figure 6.a, it occurs at 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. In order to reach additional conclusions, we summarized our results in terms of general trends of the effect of re-sampling in Table 4. Table 4 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 data sets taken 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 undersampling and oversampling behave.

Table 4. The Summary of Resampling Trends in all domains but Wisconsin Breast Cancer

		1:1	Negative	Positive
Negative-basedClass Imbalance	Over	Gradual Reduction	Gradual Increment	Gradual Reduction
	Under	Gradual Reduction	Radical Increment	Radical Reduction
Positive-basedClass Imbalance	Over	Gradual Reduction	Gradual Reduction	Gradual Increment
	Under	Gradual Reduction	Radical Reduction	Radical Increment

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 abnormaly unstable (in the case of undersampling). This probably results from the fact that the 6 examples in the minor class were not sufficient to desribe the concept at hand no matter how often it was duplicated.

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 re-sampling rate varies from domain to domain and re-sampling strategy to re-sampling 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 stably with different rates, while undersampling does so radically and in an unstable manner.

Part III: Multiple Resampling Methods

The results obtained in the previous part of the paper suggest that it might be useful to combine oversampling and undersampling versions of C4.5 sampled 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 Section 2.1 and 2.2). On the other hand, the combination of classifiers using different oversampling and undersampling rates may be useful since optimal sampling 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 2.2). We will now describe the combination scheme we designed to deal with the class imbalance problem. This combination scheme is first tested on some artificial domains 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 mixture of experts 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.

3.1 Architecture

In order for a combination method to be effective, it is necessary for the various classifiers that constitute the combination to make different decisions (Hansen 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 3-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 trained on data sets sampled at different rates. In particular, the 10 oversampling classifiers oversample the data at rates 10%, 20%, ... 100% (the positive class is oversampled until the two classes are of the same size) and the 10 undersampling classifiers undersample the negative class at rate 0% (no re-sampling), 10%, ..., 90% (the negative 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 (1998)'s Integrated Classification Machine.[7]

[7] However, (Shimshoni 1998) presents a general architecture. It was not tuned to the imbalance problem, nor 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 1998) uses ensemble methods to combine his various classifiers

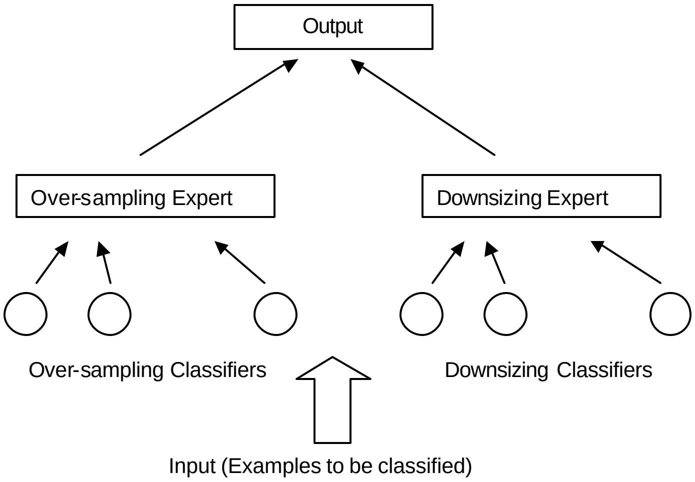


Figure 7: The Architecture of Multiple Resampling Methods

3.2. Detailed Combination Scheme

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

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

Observation #2: In class imbalanced domains, classifiers tend to make many Classification errors on the non dominant class. (See Part I)

In order 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 non-overlapping areas, each best classified by a different expert. 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 has a good chance of being unreliable as well. In order 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 ten fold cross validation fashion to the original imbalanced training data. The individual classifiers of the combination scheme (trained with various re-balanced versions of the training set) displaying with lower error rates than the average of ten fold cross validation error of all classifiers are selected and trained again with all of training examples. 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 non-eliminated 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 under-represented 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 favour of the under-represented class, however, is mitigated by our elimination scheme which strenuously eliminates any classifier believed to be too biased towards that class.

Part IV: 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 Part II of this paper 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 ten categories of the Reuter 21578 collection will be useed and the proposed approach will be compared to C4.5 and AdaBoost.

4.1 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 experiment, 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 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 ROC (Receiver Operating Characteristic) 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. 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 another one.

ROC curve is the plotted curve, where the x-axis is the false positive rate and y-axis the true positive rate. This measure is from the signal detection to characterize the trade off between hit rate and false alarm rate. The false positive rate means the rate of the number of examples classified into positive ones among all negative examples, and the true positive rate does that of the number of examples classified into such ones among all positive examples.

We show the results obtained on two domains and, in order to save space, we summarize the others in Table 4. Figure 8(a) and 8(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 Figure 8(b) focuses on the ROC curves. Figures 9(a) and 9(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 while Figure 9 shows an example where the combined approach is clearly superior.

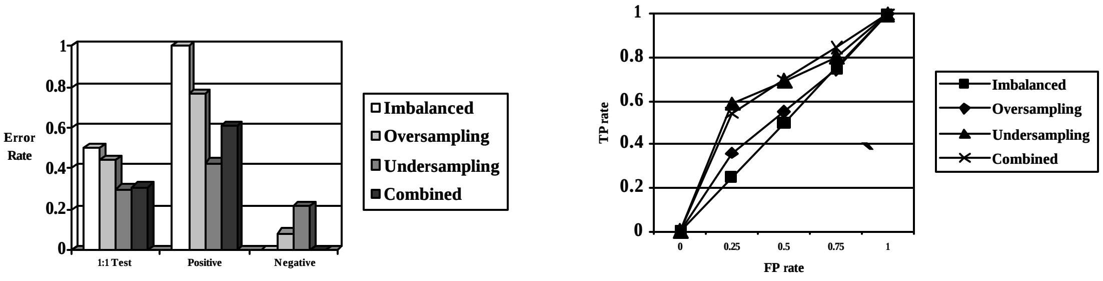


Figure 8: Results on the Pima Indian Diabetes domain with negative-dominant imbalance. 8.a: Error rates; 8.b: ROC curves

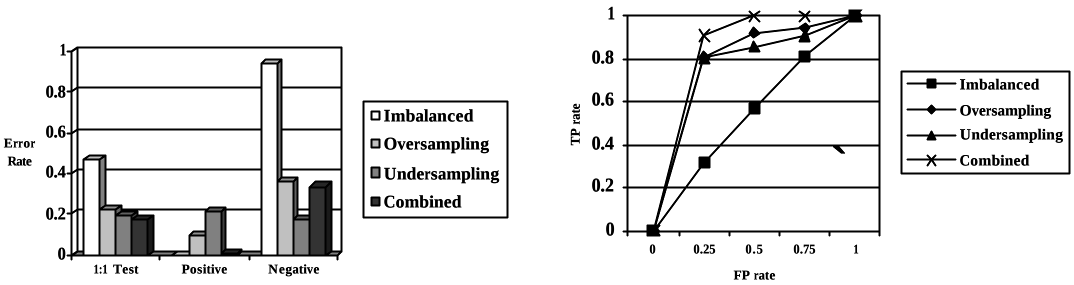


Figure 9: Results on the Pima Indian Diabetes domain with positive-dominant imbalance. 9.a: Error rates; 9.b: ROC curves

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 rows of the table corresponding to a win for the proposed approach are boldened. 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.

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**

4.2 Text Classification

Since our combination scheme was shown to help increase classification accuracy on several classes of domains, we also decided to test it systematically on a real-world domain. In particular, we chose to test it on a subset of the ten largest categories of the 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 these data.

The ten largest categories of the Reuters-21578 data set consist of the documents included in the classes of financial topics listed in table 6:

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

Several typical pre-processing steps were taken to prepare the data for classification. First, the data was divided according to the ModApte split which consists of considering all labelled documents published before 04/07/87 as training data (9603 documents, altogether) and all labelled documents published on or after 04/07/87 as testing data (3299 documents altogether). The unlabelled documents represent 8676 documents and were used during the classifier elimination step.

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 a stop word list[8]. The words in each document were then stemmed using the Lovins stemmer[9] and the 100 most frequently occurring words were used as the dictionnary for the bag-of-word vectors representing each documents.[10] Finally, the data set was divided into 10 concept learning problems where each problem consisted of a positive class containing 100 examples sampled from 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 multi-class 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, since we decided that it was not fair to compare the effectiveness of a system of 20 classifiers to that of a single classifier, we also ran C4.5 with the Ada-boost option combining 20 classifiers.[11] The results of these experiments are reported in Figure 10 as a

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

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

[10] A dictionary of 100 words is smaller than the typical number of words used (see, for example, (Scott & 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.

[11] 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

function of the micro-averaged (over the 10 different classification problems) Recall, Precision and F_1 measures. Figure 11 shows the same results but for the macro-average.

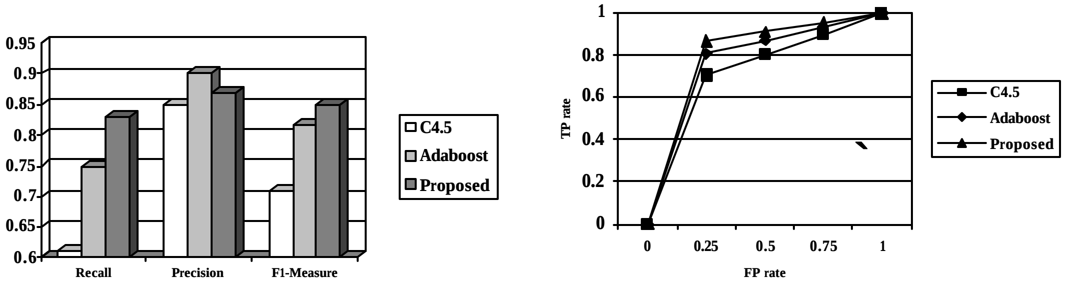
In more detail, Precision, Recall and the F_1 measures are defined as follow:

$$P = TruePositives/(TruePositives + FalsePositives)$$
$$R = TruePositives/(TruePositives + FalseNegatives)$$
$$F_1 = (2 * P * R) / (P + R)$$

where P is the precision, R, the Recall, and $F_{1,}$ the F_1 measure.

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 (1 result per classification problem), these results had to be averaged in order to be presented in a single graph. Micro-averaging 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. Macro-averaging consists of a straight average of the F_1 measure obtained in all the problems, by each combination system. Using Macro-averaging gives each problem the same weight, independently of the number of examples they contain.

Figure 11 shows the micro-averaged 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.



a

b

Figure 10: Micro-averaged Results on Reuters ten top categories. 10.a: Error rates; 10.b: ROC Curves.

than C4.5 given that it is currently considered one of the best general-purpose classification algorithm (Breiman 1998).

Figure 11.a and 11.b show the macro-averaged results of text classification, with the assumption that each category has the same weight. The distribution of macro-averaged performance is similar to that of the micro-averaged one.

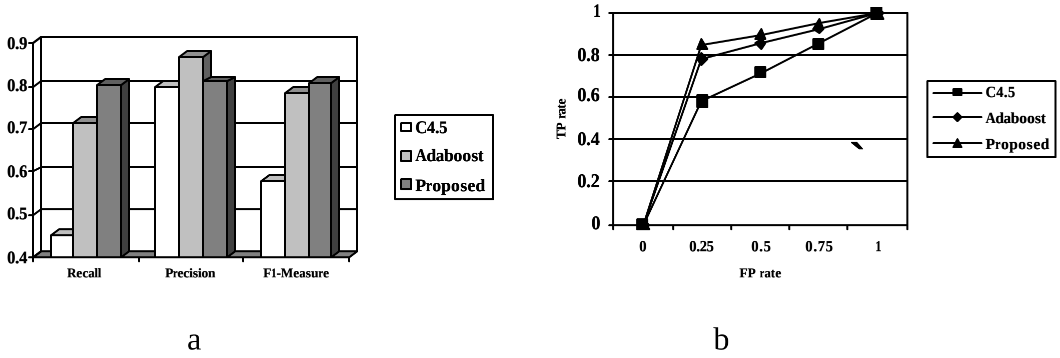


Figure 11: Macro-averaged Results on Reuters ten top categories. 11.a: Error rates; 11.b: ROC Curves.

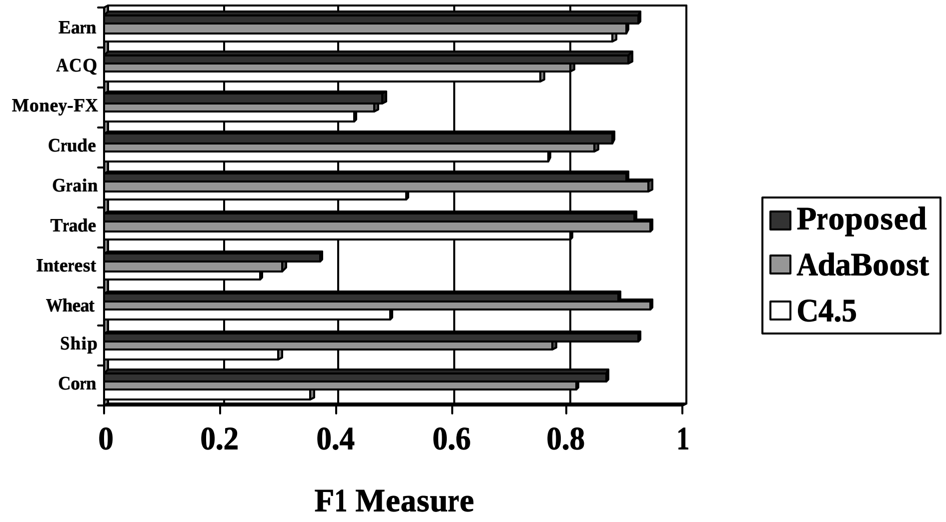


Figure 12: F1-measure on each individual domain for the Proposed combination scheme, Adaboost and C4.5.

Figure 12 shows the results for each individual domain considered. The domains are ordered 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 peformance of the proposed method relative to the other two approaches. However, this figure shows us that our method prevails over C4.5 in all cases and over Adaboost in 7 out of 10 cases.

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

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 re-sampling based classifiers in an informed fashion. In particular, our combination system was built so as to bias the classifiers towards the positive set in order to counteract the negative bias typically developed by classifiers facing a higher proportion of negative than positive examples. The positive 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 classifier or 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 occuring 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 et al. (2002). Like in our work, their study attempts to combine both oversampling and undersampling. Their oversampling method is quite sophisticated, but on the other hand, they do not look at different class distribution ratios as 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 study its various components separately and explain their various 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 than 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. 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 re-sampling approaches such as those of (Kubat & Matwin 1997) or (Chawla et al. 2002). Third, it would be interesting to find out whether our combination approach could also improve on cost-sensitive techniques previously designed. Finally, we would like to test our technique on other domains presenting a large class imbalance.

Acknowledgements

The authors would like to thank Rob Holte and Chris Drummond for their valuable comments. 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., Hall, L. and Kegelmeyer, W. (2002): SMOTE: Synthetic Minority Oversampling Technique, *Journal of Artificial Intelligence Research , 16,* 321—357.

Domingos, Pedro (1999): Metacost: A general method for making classifiers cost sensitive, *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 155--164.

Estabrooks, Andrew (2000):*A Combination Scheme for Inductive Learning from Imbalanced Data Sets*, MCS Thesis, Faculty of Computer Science, Dalhousie University.

Hansen, L. K. and Salamon, P. (1990): Neural Network Ensembles, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993--1001.

Japkowicz, Nathalie and Stephen, Shaju (2003): The Class Imbalance Problem: A Systematic Study, *Intelligent Data Analysis*, *Volume 6,*
Number 5, November 2002, 429—450.

Japkowicz, Nathalie, Myers, Catherine and Gluck, Mark (1995):
A Novelty Detection Approach to Classification, *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, 518--523.

Kubat, Miroslav and Matwin, Stan (1997): Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling, *Proceedings of the Fourteenth International Conference on Machine Learning,* 179--186.

Kubat, Miroslav, Holte, Robert and Matwin, Stan (1997): Machine Learning for the Detection of Oil Spills in Satellite Radar Images,
Machine Learning, Volume 30, 195--215.

Lewis, D. and Gale, W. (1994): Training Text Classifiers by Uncertainty Sampling, *Proceedings of the Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Ling, C. and Li, C. (1998): Data Mining for Direct Marketing: Problems and Solutions, *Proceedings of KDD-98.*

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. and Brunk, C. (1994): Reducing Misclassification Costs, *Proceedings of the Eleventh International Conference on Machine Learning,* 217--225.

Riddle, P., Secal, R. and Etzioni, O. (1991): Representation Design and Brute-Force Induction in a Boeing Manufacturing Domain, *Applied Artificial Intelligence, Volume 8*, 125--147.

Scott, Sam and Matwin, Stan (1999): Feature Engineering for Text Classification, *Proceedings of the Sixteenth International Conference on Machine Learning*, 379--388.

Shimshoni, Y. and Intrator, N. (1998): Classifying Seismic Signals by Integrating Ensembles of Neural Networks, *IEEE Transactions On Signal Processing, Special issue on Neural Networks.*

Weiss, G. and Provost, F (2003): Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction, to appear, *Journal of Artificial Intelligence Research*.

Weiss, S. and Kapouleas, I. (1990): An empirical comparison of pattern recognition, neural nets and machine learning methods, *Readings in Machine Learning*, J.W Shavlik and T.G. Dietterich (editors), Morgan Kauffman.