A Mixture-of-Experts Framework for Text Classification

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

Nathalie Japkowicz
SITE, University of Ottawa,
150 Louis Pasteur, P.O. Box 450 Stn. A,
Ottawa, Ontario, Canada K1N 6N5
nat@site.uottawa.ca

Abstract

One of the particular characteristics of text classification tasks is that they present large class imbalances. Such a problem can easily be tackled using re-sampling methods. However, although these approaches are 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 a method for combining different expressions of the re-sampling approach in a mixture of experts framework. The proposed combination scheme is evaluated on a very imbalanced subset of the REUTERS-21578 text collection and is shown to be very effective on this domain.

1 Introduction

A typical use of Machine Learning methods in the context of Natural Language Processing is in the domain of text classification. Unfortunately, several characteristics specific to text data make its classification a difficult problem to handle. In particular, the data is typically highly dimensional and it presents a large class imbalance, i.e., there, typically, are very few documents on the topic of interest while texts on unrelated subjects abound. Furthermore, although large amounts of texts are available on line, little of them are labeled. Because the class imbalance problem is known to negatively affect typical classifiers and because unlabeled data have no place in conventional supervised learning, using off-the-shelf supervised classifiers is likely not to be very successful in the context of text data. It is, instead, recommended to devise a classification method specifically tuned to the text classification problem.

The purpose of this study is to target some of the characteristics of text data in the hope of improving the effectiveness of the classification process. The topics of finding a good representation for text data and dealing with its high dimensionality have been investigated previously with, for example, the use of Wordnet [e.g., (Scott & Matwin, 1999)] and Support Vector Machines [e.g., (Joachims, 1998)], respectively. We will not be addressing these problems here. The question that we will tackle in this paper, instead, is that of dealing with the class imbalance, and, in the process of doing so, that of finding a way to take advantage of the extra, albeit, unlabeled data that are often left unused in classification studies.[1]

Several approaches have previously been proposed to deal with the class imbalance problem including a simple and yet quite effective method: re-sampling [e.g., (Lewis & Gale, 1994), (Kubat & Matwin, 1997), (Domingos, 1999)]. This paper deals with the two different types of re-sampling 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. Because it is unclear whether oversampling is more effective than undersampling and which oversampling or undersampling rate should be used, we propose a

[1] Note, however, that unlabeled data is not always left unused as in the work on co-learning of (Blum & Mitchell, 1998). As discussed below, however, our approach will make use of the unlabeled data in a different way.

method for combining a number of classifiers that oversample and undersample the data at different rates in a mixture of experts framework. The mixture-of-experts is constructed in the context of a decision tree induction system: C5.0, and all resampling is done randomly. This proposed combination scheme is, subsequently, evaluated on a a subset of the REUTERS-21578 text collection and is shown to be very effective in this case.

The remainder of this paper is divided into four sections. Section 2 describes an experimental study on a series of artificial data sets to explore the effect of oversampling and undersampling and oversampling or undersampling at different rates. This study suggests a mixture-of-experts scheme which is described in Section 3. Section 4 discusses the experiment conducted with that mixture-of-experts scheme on a series of text-classification tasks and discusses their results. Section 5 is the conclusion.

2 Experimental Study

We begin this work by studying the effects of oversampling versus undersampling and oversampling or undersampling at different rates.[2] All the experiments in this part of the paper are conducted over artificial data sets defined over the domain of 4 x 7 DNF expressions, where the first number represents the number of literals present in each disjunct and the second number represents the number of disjuncts in each concept.[3] We used an alphabet of size 50. For each concept, we created a training set containing 240 positive and 6000 negative examples. In other words, we created an imbalance ratio of 1:25 in favor of the negative class.

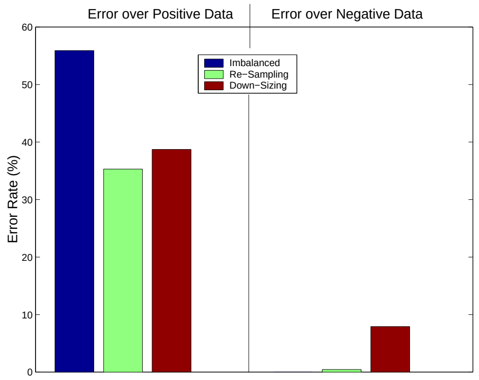

Figure 1: Re-Sampling versus Downsizing

2.1 Re-Sampling versus Downsizing

In this part of our study, three sets of experiments were conducted. First, we trained and tested C5.0 on the 4x7 DNF 1:25 imbalanced data sets just mentioned.[4] Second, we randomly oversampled the positive class, until its size reached the size of the negative class, i.e., 6000 examples. The added examples were straight copies of the data in the original positive class, with no noise added. Finally, we undersampled the negative class by randomly eliminating data points from the negative class until it reached the size of the positive class or, 240 data points. Here again, we used a straightforward random approach for selecting the points to be eliminated. Each experiment was repeated 50 times on different 4x7 DNF concepts and using different oversampled or removed examples. After each training session, C5.0 was tested on separate testing sets containing 1,200 positive and 1,200 negative examples. The average accuracy results are reported in Figure 1. The left side of Figure 1 shows the results obtained on the positive testing set while its right side shows the results obtained on the negative testing set.

As can be expected, the results show that the number of false negatives (results over the pos-

[2] Throughout this work, we consider a fixed imbalance ratio, a fixed number of training examples and a fixed degree of concept complexity. A thorough study relating different degrees of imbalance ratios, training set sizes and concept difficulty was previously reported in (Japkowicz, 2000).

[3] 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.

[4] (Estabrooks, 2000) reports results on 4 other concept sizes. An imbalanced ratio of 1:5 was also tried in preliminary experiments and caused a loss of accuracy about as large as the 1:25 ratio. Imbalanced ratios greater than 1:25 were not tried on this particular problem since we did not want to confuse the imbalance problem with the small sample problem.

itive class) is a lot higher than the number of false positives (results over the negative class). As well, the results suggest that both naive oversampling and undersampling are helpful for reducing the error caused by the class imbalance on this problem although oversampling appears more accurate than undersampling.[5]

2.2. Re-Sampling and Down-Sizing at various Rates

In order to find out what happens when different sampling rates are used, we continued using the imbalanced data sets of the previous section, but rather than simply oversampling and undersampling them by equalizing the size of the positive and the negative set, we oversampled and undersampled 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 chose to make the 100% oversampling rate correspond to the fully oversampled data sets of the previous section but to make the 90% undersampled rate correspond to the fully undersampled data sets of the previous section.[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 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 the previous section while a 100% undersampling rate corresponds to the case where no negative examples are present in the training set.

Once again, and for each oversampling and undersampling rate, the rules learned by C5.0 on the training sets were tested on testing sets containing 1,200 positive and 1,200 negative examples. The results of our experiments are displayed in Figure 2 for the case of oversampling and undersampling respectively. They represent the averages of 50 trials. Again, the results are reported separately for the positive and the negative testing sets.

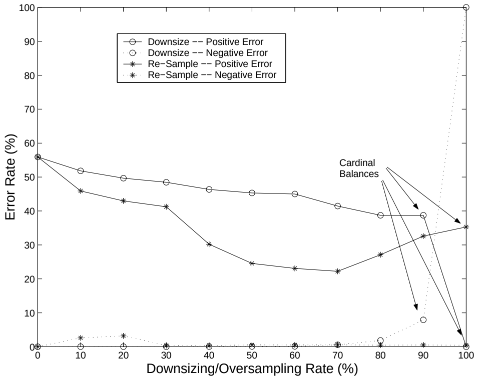

Figure 2: Oversampling and Downsizing at Different Rates

These results suggest that different sampling rates have different effects on the accuracy of C5.0 on imbalanced data sets for both the oversampling and the undersampling method. In particular, the following observation can be made:

> Oversampling or undersampling until a cardinal balance of the two classes is reached is not necessarily the best strategy: best accuracies are reached before the two sets are cardinally balanced.

In more detail, this observation comes from the fact that in both the oversampling and undersampling curves of figure 2 the optimal accuracy is not obtained when the positive and the negative classes have the same size. In the oversampling curves, where class equality is reached at the 100% oversampling rate, the average error rate obtained on the data sets over the positive class at that point is 35.3% (it is of 0.45% over the negative class) whereas the optimal error rate is obtained at a sampling rate of 70% (with an error rate of 22.23% over the positive class and of 0.56% over the negative class). Similarly, although less significantly, in the undersampling curves, where class equality is reached at the 90% undersampling rate[7], the average error rate ob-

[5] Note that the usefulness of oversampling versus undersampling is problem dependent. (Domingos, 1999), for example, finds that in some experiments, oversampling is more effective than undersampling, although in many cases, the opposite can be observed.

[6] This was done so that no classifier was duplicated in our combination scheme. (See Section 3)

[7] The sharp increase in error rate taking place at the 100%

tained at that point is worse than the one obtained at a sampling rate of 80% since although the error rate is the same over the positive class (at 38.72%) it went from 1.84% at 90% oversampling over the negative class to 7.93%.[8]

In general, it is quite likely that the optimal sampling rates can vary in a way that might not be predictable for various approaches and problems.

3 The Mixture-of-Experts Scheme

The results obtained in the previous section suggest that it might be useful to combine oversampling and undersampling versions of C5.0 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 (cf. results of Section 2.1) and may learn a same concept in different ways.[9] On the other hand, the combination of classifiers using different oversampling and undersampling rates may be useful since we may not be able to predict, in advance, which rate is optimal (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 will be tested on a subset of the REUTERS-21578 text classification domain.[10]

3.1 Architecture

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 section.

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 experiments in Section 2 of this paper suggest that undersampling and oversampling at different rates will produce classifiers able to make different decisions, including some corresponding to the "optimal" undersampling or oversampling rates that could not have been 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%, 10%, ..., 90% (the negative class is undersampled until the two classes are of the same size). Figure 3 illustrates the architecture of this combination scheme that was motivated by (Shimshoni & Intrator, 1998)'s Integrated Classification Machine.[11]

3.2 Detailed Combination Scheme

Our combination scheme is based on two different facts:

Fact #1: Within a single testing set, different testing points could be best classified by different single classifiers. (This is a general fact that can be true for any problem and any set of classifiers).

Fact #2: In class imbalanced domains for which the positive training set is small and the negative training set is large, classifiers tend to make many false-negative errors. (This is

undersampling point is caused by the fact that at this point, no negative examples are present in the training set.

[8] Further results illustrating this point over different concept sizes can also be found in (Estabrooks, 2000).

[9] In fact, further results comparing C5.0's rule sizes in each case suggest that the two methods, indeed, do tackle the problem differently [see, (Estabrooks, 2000)].

[10] This combination scheme was first tested on DNF artificial domains and improved classification accuracy by 52 to 62% over the positive data and decreased the classification accuracy by only 7.5 to 13.1% over the negative class as compared to the accuracy of a single C5.0 classifier. See (Estabrooks, 2000) for more detail.

[11] However, (Shimshoni & Intrator, 1998) is 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.

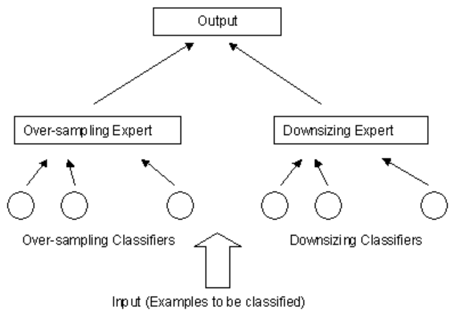

Figure 3: Re-Sampling versus Downsizing

a well-known fact often reported in the literature on the class-imbalance problem and which was illustrated in Figure 1, above).

In order to deal with the first fact, 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 our second fact in that it invalidates any classifier labeling too many examples as positive. Since the classifiers of the combination scheme have a tendency of being naturally biased towards classifying the examples as negative, we assume that a classifier making too many positive decision is probably doing so unreliably.

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.

The elimination scheme used at the classifier level uses the following heuristic: the first (most imbalanced) and the last (most balanced) classifiers of each expert are tested on an unlabeled data set. The number of positive classifications each classifier makes on the unlabeled data set is recorded and averaged and this average is taken as the threshold that none of the expert's classifiers must cross. In other words, any classifier that classifies more unlabeled data points as positive than the threshold established for the expert to which this classifier belongs needs to be discarded.[12]

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

[12] Because no labels are present, this technique constitutes an educated guess of what an appropriate threshold should be. This heuristic was tested in (Estabrooks, 2000) on the text classification task discussed below and was shown to improve the system (over the combination scheme not using this heuristic) by 3.2% when measured according to the F_1 measure, 0.36% when measured according to the F_2 measure, and 5.73% when measured according to the $F_{0.5}$ measure. See the next section, for a definition of the F_B measures, but note that the higher the F_B value, the better.

scheme which strenuously eliminates any classifier believed to be too biased towards the positive class.

4 Experiments on a Text Classification Task

Our combination scheme was tested on a subset of the 10 top categories of the REUTERS-21578 Data Set. We first present an overview of the data, followed by the results obtained by our scheme on these data.

4.1 The Reuters-21578 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 1:

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

Table 1: The top 10 Reuters-21578 categories

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[13]. The words in each document were then stemmed using the Lovins stemmer[14] and the 500 most frequently occurring features were used as the dictionary for the bag-of-word vectors representing each documents.[15] 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. Furthermore, although the Reuters Data set contains more than 100 examples in each of its top 10 categories (see Table 1), we found it more realistic to use a restricted number of positive examples.[16] Having restricted the number of positive examples in each problem, it is interesting to note that the class imbalances in these problems is very high since it ranges from an imbalance ratio of 1:60 to one of 1:100 in favour of the negative class.

4.2 Results

The results obtained by our scheme on these data were pitted against those of C5.0 ran with the Ada-boost option.[17] The results of these exper-

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

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

[15] A dictionary of 500 words is smaller than the typical number of words used (see, for example, (Scott & Matwin 1999)), however, it was shown that this restricted size did not affect the results too negatively while it did reduce processing time quite significantly (see (Estabrooks 2000)).

[16] Indeed, very often in practical situations, we only have access to a small number of articles labeled “of interest” whereas huge number of documents “of no interest” are available

[17] Our scheme was compared to C5.0 ran with the Adaboost option combining 20 classifiers. This was done in order to present a fair comparison to our approach which also uses 20 classifiers. It turns out, however, that the Adaboost option provided only a marginal improvement over using a single version of C5.0 (which itself compares favorably to state-of-the-art approaches for this problem) (Estabrooks, 2000). Please, note that other experiments using C5.0 with the Ada-boost option combining fewer or more classifiers should be attempted as well since 20 classifiers might not be

iments are reported in Figure 4 as a function of the micro-averaged (over the 10 different classification problems) F_1, F_2 and $F_{0.5}$ measures. In more detail, the F_B-measure is defined as:

$$F_B = \frac{(B^2+1)\times P\times R}{B^2\times P+R}$$

where P represents precision, and R, recall, which are respectively defined as follows:

$$P = \frac{TruePositives}{TruePositives+FalsePositives}$$

$$R = \frac{TruePositives}{TruePositives+FalseNegatives}$$

In other words, precision corresponds to the proportion of examples classified as positive that are truly positive; recall corresponds to the proportion of truly positive examples that are classified as positive; the F_B-measure combines the precision and recall by a ratio specified by B. If $B = 1$, then precision and recall are considered as being of equal importance. If $B = 2$, then recall is considered to be twice as important as precision. If $B = 0.5$, then precision is considered to be twice as important as recall.

Because 10 different results are obtained for each value of B and each combination system (1 result per classification problem), these results had to be averaged in order to be presented in the graph of Figure 4. We used the Micro-averaging technique which consists of a straight average of the F-Measures obtained in all the problems, by each combination system, and for each value of B. Using Micro-averaging has the advantage of giving each problem the same weight, independently of the number of positive examples they contain.

The results in Figure 4 show that our combination scheme is much more effective than Adaboost on *both* recall and precision. Indeed, Adaboost gets an F_1 measure of 52.3% on the data set while our combination scheme gets an F_1 measure of 72.25%. If recall is considered as twice more important than precision, the results are even better. Indeed, the mixture-of-experts scheme gets an F_2-measure of 75.9% while Adaboost obtains an F_2-measure of 48.5%. On the other hand, if precision is considered as twice more important than recall, then the combination scheme is still effective, but not as effective with respect to Ada-boost since it brings the $F_{0.5}$-measure on the reduced data set to only 73.61%, whereas Ada-Boost's performance amounts to 64.9%.

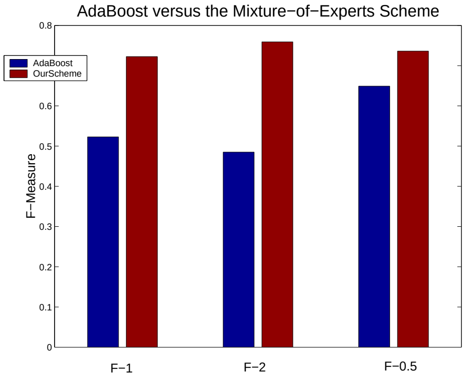

Figure 4: Average results obtained by Ada-Boost and the Mixture-of-Experts scheme on 10 text classification problems

The generally better performance displayed by our proposed system when evaluated using the F_2-measure and its generally worse performance when evaluated using the $F_{0.5}$-measure are not surprising, since we biased our system so that it classifies more data points as positive. In other words, it is expected that our system will correctly discover new positive examples that were not discovered by Ada-Boost, but will incorrectly label as positive examples that are not positive. Overall, however, the results of our approach are quite positive with respect to both precision and recall. Furthermore, it is important to note that this method is not particularly computationally intensive. In particular, its computation costs are comparable to those of commonly used combination methods, such as AdaBoost.

5 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 so as 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 elimina-

C5.0-Ada-boost's optimal number on our problem.

tion strategy designed to prevent unreliable classifiers to participate in the process. The technique was shown to be very effective on a drastically imbalanced version of a subset of the REUTERS text classification task.

There are different ways in which this study could be expanded in the future. First, our technique 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 (Lewis & Gale, 1994) and (Kubat & Matwin, 1997). Second, 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

We would like to thank Rob Holte and Chris Drummond for their valuable comments. This research was funded in part by an NSERC grant. The work described in this paper was conducted at Dalhousie University.

References

Blum, A. and Mitchell, T. (1998): Combining Labeled and Unlabeled Data with Co-Training *Proceedings of the 1998 Conference on Computational Learning Theory.*

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 (2000): The Class Imbalance Problem: Significance and Strategies, *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, 111–117.

Joachims, T. (1998): Text Categorization with Support Vector Machines: Learning with many relevant features, *Proceedings of the 1998 European Conference on Machine Learning.*

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.

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*.

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 NN.*