Class Imbalances versus Small Disjuncts

Taeho Jo
SITE, University of Ottawa
800 King Edward Ave., P.O. Box 450 Stn.A
Ottawa, Ontario, Canada K1N 6N5
tjo018@site.uottawa.ca

Nathalie Japkowicz
SITE, University of Ottawa
800 King Edward Ave., P.O. Box 450 Stn.A
Ottawa, Ontario, Canada K1N 6N5
nat@site.uottawa.ca

ABSTRACT

It is often assumed that class imbalances are responsible for significant losses of performance in standard classifiers. The purpose of this paper is to the question whether class imbalances are truly responsible for this degradation or whether it can be explained in some other way. Our experiments suggest that the problem is not directly caused by class imbalances, but rather, that class imbalances may yield small disjuncts which, in turn, will cause degradation. We argue that, in order to improve classifier performance, it may, then, be more useful to focus on the small disjuncts problem than it is to focus on the class imbalance problem. We experiment with a method that takes the small disjunct problem into consideration, and show that, indeed, it yields a performance superior to the performance obtained using standard or advanced solutions to the class imbalance problem.

Keywords

class imbalance, small disjuncts, rare cases, resampling, within-class imbalance, between-class imbalance.

1. INTRODUCTION

Although class imbalances have been reported to hinder the performance of standard classifiers on many different types of problems[1], no study has made a point of linking the class imbalance problem directly to this loss. As a matter of fact, although the performance of standard classifiers may decrease on many class imbalanced domains, that does not necessarily demonstrate that it is the imbalance, per se, that causes this decrease. Rather, it is quite possible that class imbalances yield certain conditions that hamper classification, which would suggest 1) that class imbalances are not necessarily always a problem and, perhaps even more importantly, 2) that dealing with class imbalances alone will not always help improve performance.

The purpose of this paper is to question whether class imbalances are truly to blame for the reported losses of performance or whether these deficiencies can be explained in some other way. We show that class imbalances are, actually, often not a problem by themselves, but that, in small and complex data sets, they come accompanied with the problem of small disjuncts [5], which in turn causes a degradation in standard classifiers' performance.[2] We conclude the paper by summarizing and testing an approach that we have previously designed (and described in greater length, elsewhere [7],[8]) that counters the effect of small disjuncts.

Though, the results presented in this paper may not be fully surprising, it is important to note that the paper's main contribution rather lies in the shift of focus it proposes, which leads to new insights and solutions.

The remainder of the paper is divided into seven sections. Section 2 discusses the class imbalance, rare case and small disjunct problems. Section 3 describes the artificial and real domains on which our study is based. Section 4 shows the effect of class imbalances on these domains. The results suggest that class imbalances are often problematic, but not always. Section 5 shows the result of further experiments that contrast the class imbalance problem to the problem of small disjunct. These results show that it is the small disjunct problem rather than the class imbalance problem that is to blame for the loss of performance. Section 6 describes four standard methods designed for the class imbalance problem only (methods that ignore small disjuncts), a standard method for pruning small disjuncts (a method that completely eradicates the small disjuncts), as well as our newer approach, cluster-based oversampling [7], [8], that oversamples while addressing both the class imbalance and the small disjuncts problems (a method that inflates the small disjuncts). Section 7 shows the effects of cluster-based oversampling on artificial and real domains and contrasts them to those of the other methods described in Section 6. Section 8 concludes the paper.

2. CLASS IMBALANCES, RARE CASES AND SMALL DISJUNCTS

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. As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately. Such a problem occurs in a large number of practical domains and is often dealt with by using re-sampling or cost-based methods. While cost-based methods were previously reported to perform better than random re-sampling approaches (e.g., [10]), they do not have the flexibility offered by re-sampling approaches. In

[1] For example, the problem has been reported on cases as diverse as: the detection of oil spills in satellite radar images [1], the detection of fraudulent telephone calls [2], information retrieval and filtering [2], diagnoses of rare medical conditions such as thyroid diseases [4]

[2] Please note that this conclusion was reached within the settings we used in this paper. In another setting—namely, in the domain of microarray time-series analysis—[6] suggests that the class imbalance problem is more significant than the problem of small disjuncts. Our results support this observation in only one of the domains we tested [See footnote 6]. This suggests further research into the conditions that make a domain more or less sensitive to class imbalances than to small disjuncts.

particular, re-sampling approaches can generate novel examples or re-sample different parts of the space differently. Cost-based approaches, instead, take a monolithic approach to the re-balancing of data sets. This paper will explore whether the flexibility of re-sampling approaches can be exploited to the advantage of the class imbalanced problem. Its particular emphasis is on whether isolating rare cases and inflating them at different rates can be beneficial.

Rare or exceptional cases correspond to small numbers of training examples in particular areas of the feature space. When learning a concept, the presence of rare cases in the domain is an important consideration. The reason why rare cases are of interest is that they cause small disjuncts to occur, which are known to be more error prone than large disjuncts [5]. In more detail, learning systems usually create concept definitions that consist of several disjuncts. Each disjunct, in turn, is a conjunctive definition of a subconcept of the original concept. The coverage of a disjunct corresponds to the number of training examples it correctly classifies, and a disjunct is considered to be a "small disjunct" if that coverage is low. In fact, small disjuncts are not inherently more error prone than large disjuncts. What makes them more error prone are the bias of the classifiers [5] as well as the effect of attribute noise, missing attributes, class noise and training set size on the rare cases which cause them [13],[14].

Table 1, in the next section, illustrates very well how, in the two families of artificial domains, as the class imbalance and the concept complexity increases while the size of the training set decreases, the number of rare cases increases. For example, at concept complexity 3, imbalance level 1:3 and large training size, there are no rare cases since the smallest cases contain 100 examples. However, in the small setting, there are 8 rare/cases represented by 5 or 15 examples. The situation is even worse at class imbalance 1:9 where the smallest cases are represented by only 2 examples.

In the real world domains, rare cases are unknown since high dimensional data cannot be visualized to reveal areas of low coverage. In the remainder of this paper, when the rare cases of a real-world domain are necessary to consider, we will approximate them, using an unsupervised method (e.g., k-means). It is important to note that, in doing so, we make two assumptions:

1) the small disjuncts constructed by our unsupervised algorithm do correspond fully to the (unknown) rare cases of the domain;
2) there is a correspondence between the small disjuncts learned by the unsupervised method (the supposed rare cases of the domain) and those subsequently learned by the supervised method.

3. DATA DOMAINS

This section describes the artificial and real domains on which our experiments are based. We generated two different families of artificial data. The first one uses a simple uni-dimensional backbone model while the second one uses a more complex, multi-dimensional model. The purpose of using artificial domains is to understand the nature of the observed degradation in domains that present a class imbalance. Using real-world domains only could have yielded results difficult to decipher. Nonetheless, artificial domains are not sufficient to fully test a hypothesis. We also need to see if this hypothesis applies to real data. In order to do so, we selected two data sets from the UCI Repository for Machine Learning: Wisconsin Breast Cancer and Pima Indian Diabetes, as well as one larger domain: Customer foreign exchange data for currency risk management. These three domains are highly challenging and quite imbalanced, especially the currency risk management domain whose minority class accounts for less than 3% of the data.

3.1 Artificial Domain with Single Dimension

For our first artificial family of domains, we created 27 data sets with various combinations of three parameters which we deemed significant for out study: concept complexity, training set size, and degreee of imbalance. This family of domains was already used in our previous work and explained at length there [10]. These 27 data sets were generated in the following way: each of the domains is one dimensional with input in the [0, 1] range associated with one of the two classes (1 or 0). The input range is divided into a number of regular intervals (i.e, intervals of the same size), each associated with a different class value. Contiguous intervals have opposite class values and the degree of concept complexity corresponds to the number of alternating intervals present in the domain. Actual training sets are generated from these backbone models by sampling points at random (using a uniform distribution), from each of the intervals. The number of points sampled from each interval depends of the size of the domain as well as on its degree of imbalance. An example of a backbone model is shown in Figure 1.

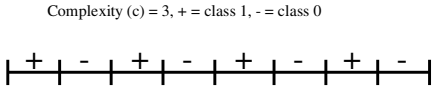

Figure 1. A Backbone Model of Complexity 3

Three different complexity levels were considered (c=1, 2, and 3) where each level, c, corresponds to a backbone model composed of 2^c regular intervals. For example, the domains generated at complexity level c=1 are such that every point whose input is in range [0, 0.5) is associated with a class value of 1, while every point whose input is in range (0.5, 1] is associated with a class value of 0; At complexity level c=2, points in intervals [0, 0.25) and (0.5, 0.75) are associated with class value 1 while those in intervals (0.25, 0.5) and (0.75, 1] are associated with class value 0; etc., regardless of the size of the training set and its degree of imbalance.

Table 1 shows the distribution of training examples in each interval of the 27 domains. The top sub-table corresponds to c=1, the middle one corresponds to c=2, and the bottom one corresponds to c=3, where c is the level of complexity. In Table 1, each row indicates the combination of class imbalance and training set size and each column indicates each interval and its label. In the definition of class imbalance, 1:9 is defined as the high class imbalance, 1:3 corresponds to the middle class imbalance, and 1:1 corresponds to the low class imbalance (actually, it corresponds to a complete class balance). In the definition of the training set sizes, 80 is defined as the small training set size, 400 corresponds to the middle one, and 1600 corresponds to the large one. Each entry of Table 1 indicates the

number of training examples in each interval. This table shows that the training examples are distributed uniformly into intervals.

Table 1. The Distribution of Training Examples with c=1 (Top), c=2 (middle), and c=3 (bottom)

Class Imbalance	Training Set Size	0.0 ~ 0.5 +	0.5 ~ 1.0 -
1:9	80	8	72
	400	40	360
	1600	160	1440
1:3	80	20	60
	400	100	300
	1600	400	1200
1:1	80	40	40
	400	200	200
	1600	800	800

Class Imbalance	Training Set Size	0.0 ~ 0.25 +	0.25 ~0.5 -	0.5~0.75 +	0.75~1.0-
1:9	80	4	36	4	36
	400	20	180	20	180
	1600	80	720	80	720
1:3	80	10	30	10	30
	400	50	150	50	150
	1600	200	600	200	600
1:1	80	20	20	20	20
	400	100	100	100	100
	1600	400	400	400	400

Class Imbalance	Training Set Size	0.0~0.125 +	0.125~0.25 -	0.25~0.375 +	0.375~0.5 -	0.5~0.625 +	0.625~0.75 -	0.75~0.875 +	0.875~1.0 -
1:9	80	2	18	2	18	2	18	2	18
	400	10	90	10	90	10	90	10	90
	1600	40	360	40	360	40	360	40	360
1:3	80	5	15	5	15	5	15	5	15
	400	25	75	25	75	25	75	25	75
	1600	100	300	100	300	100	300	100	300
1:1	80	10	10	10	10	10	10	10	10
	400	50	50	50	50	50	50	50	50
	1600	200	200	200	200	200	200	200	200

Table 2. The Distribution of Test Examples

	0 ~ 0.125	0.125~0.25	0.25~0.375	0.375~0.5	0.5~0.625	0.625~0.75	0.75~0.875	0.875~1.0
c=1	50 +				50 -			
c=2	25 +		25 -		25 +		25 -	
c=3	13 +	13 -	12 +	12 -	13 +	13 -	12 +	12 -

The test set size is fixed at 100 for each complexity level. Table 2 shows the distribution of the test examples per interval. In Table 2, the rows indicate the degree of complexity and the columns represent the actual intervals. Each entry indicates the number of test examples and the class label of each interval. In this domain as well as in all the others used for this study, we purposely chose to test our methods on perfectly balanced testing sets. This decision was made in order to put more emphasis on classifying the minority class examples correctly than accuracy on the "properly" distributed testing set (i.e., with the same class imbalance as in the training set) would. Separating the errors made on the positive and the negative data or reporting ROC curves are alternative approaches that we have used elsewhere [10], [12], but that were not practical here, given the large number of experiments conducted. We believe, however, that the performance measure used here is sufficiently indicative of the trend exhibited by the various methods we tested.

3.2 Artificial Domain with Multiple Dimensions

In the second artificial domains, the dimension of each feature vector is set to five and 27 domains are generated as in the previous subsection. The definitions of the training set size and the class imbalance degree are the same as those used in the previous section and they were summarized in Table 1. The concept complexity of this domain is defined as the number of clusters present in the domain.

Table 3. The Definition of Clusters in this Domain

#Clusters	Cluster Centers	Length
2	[0.0348, 0.9225, 0.1262, 0.2717, 0.7858 +] [0.5348, 0.4225, 0.6262, 0.7717, 0.2858 -]	0.25
4	[0.3241, 0.1006, 0.4903, 0.3863, 0.5487 +] [0.5741, 0.3506, 0.7403, 0.6363, 0.8241 -] [0.8241, 0.6006, 0.9903, 0.8861, 0.0487 +] [0.0741, 0.8506, 0.2403, 0.1363, 0.2987 -]	0.125
8	[0.6819, 0.4511, 0.9610, 0.8550, 0.9932 +] [0.8069, 0.5761, 0.0860, 0.9800, 0.1182 -] [0.9319, 0.7011, 0.2110, 0.1050, 0.2432 +] [0.0569, 0.8261, 0.3360, 0.2300, 0.3682 -] [0.1819, 0.9511, 0.4610, 0.3550, 0.4932 +] [0.3069, 0.0761, 0.5860, 0.4800, 0.6182 -] [0.4319, 0.2011, 0.7110, 0.6050, 0.7432 +] [0.5569, 0.3261, 0.8360, 0.7300, 0.8682 -]	0.0625

Table 3 shows the definition of the clusters we used to generate the data for these domains. These clusters were designed so as to display an alternance of the classes in the five-dimensional space considered, and so as to avoid any overlap, which was previously shown to interfere with the class imbalance problem [11] and should be treated separately. In the column labeled "cluster centers", in Table 3, each vector indicates the center of each cluster along with its class label. Each cluster is defined as a five dimensional hyper-cube with its center and its length (indicated in the last column of Table 3). The actual cluster points are distributed around their centers using a uniform distribution. In case some of the elements of the feature vectors are randomly assigned values smaller than zero or greater than one, their values are set to zero or one, respectively.

3.3 Real-World Domains: Wisconsin Breast Cancer, Pima Indian Diabetes, and Currency Risk Management

Two of the real-world domains for the experiments of this paper were obtained from the UCI machine learning repository. This repository is often used as a standard test bed to evaluate the performance of classifiers. Two domains, Wisconsin Breast Cancer and Pima Indian Diabetes were selected for the experiments of this paper. As previously mentioned, these data sets present a high class imbalance and are particularly complex. This makes them good candidates to test our hypothesis.

The first one of these domains is concerned with the diagnosis of breast cancer, based on information about cells. Each example's label is '2' or '4'; label, '2', indicates that the cell described in the example is not cancerous, while label, '4', indicates that it is cancerous. Therefore, examples with label '2' represent the negative examples while the others represent the positive examples. The distribution of the training examples we used for this domain is illustrated in table 4. As shown in this table, the degree of class imbalance is defined in the same way as it was in the artificial domains. The size of the training sets is a bit different, however: a total number of training examples of 40 represents the small size, while one of 140 represents the large size. The size of the test set is 100 with complete class balance: it consists of 50 positive examples and 50 negative examples.

Table 4. The Distribution of Training Examples in Wisconsin Breast Cancer

		Positive ('4')	Negative ('2')
1:9	40	4	36
	140	14	126
1:3	40	10	30
	140	35	105
1:1	40	20	20
	140	70	70

The second real domain is concerned with the diagnosis of diabetes based on physical information about Pima Indians. Each example has one of two labels, '0' and '1'; '0' indicates no diabetes and '1' represents diabetes. Examples with label '1', are set as positive examples. Those with label '0' represent negative ones. The distribution of training examples is illustrated in table 5. In this domain, 40 training examples are defined as the small set size, 100 represent the medium size, while 200 represent the large size. The size of the test set is 100 with a perfect class balance, like in the previous domain.

Table 5. The Distribution of Training Examples in Pima Indian Diabetes

		Positive ('1')	Negative ('0')
1:9	40	4	36
	100	10	90
	200	20	180
1:3	40	10	30
	100	25	75
	200	50	150
1:1	40	20	20
	100	50	50
	200	100	100

A disadvantage of the two UCI domains is that they are relatively small. Furthermore, their degree of imbalance is not very high. In order to test our methods in a more realistic context, we selected a third domain: Customer Foreign Exchange data for Currency Risk Mangement. This domain is concerned with the prediction of fraudulent foreign exchange transactions based on the transactions' profile. It is available from: www.sis.uncc.edu/%7Emirsad/itcs6265/resource.htm. It is composed of thirteen discrete input attributes and one output class that can take three values: "bad", "average", and "good". In order to turn this problem into a binary classification problem, the two values, "bad" and "average" were mapped into a negative outcome while the value, "good", represents the positive one.

This domain was used in our evaluation of resampling methods (section 6) as a single experiment set. We did not use it in the experiments of Sections 3 and 4 because of its high degree of imbalance which prevented the more flexible experimentation conducted there. In our experiments of Section 5, the currency risk management domain contained 1700 negative and 50 positive examples in the training set while the test set was composed of 100 negative and 100 positive examples.

4. THE EFFECT OF CLASS IMBALANCE

This first set of experiments attempts to determine whether the class imbalance problem always causes a degradation in performance or whether it does so only in certain cases.[3] In order to answer the question, we ran C4.5 and back propagation on the artificial and real domains described in the previous section. The results of our experiments are displayed in figures 2, 3, and 4, which plot the accuracy of C4.5 and back propagation for each combination of concept complexity, training set size and imbalance level, on the entire test set. As mentionned previously, for each experiment, we reported a single type of result: results in which no matter what degree of class imbalance is present in the training set, the contribution of the false positive rate is the same as that of the false negative one in the overall report.

Table 6. The Design of Back Propagation for this set of Experiments

	Artificial Domains		Real Domains	
	One dimension	Five dimensions	Breast	Pima
#input nodes	1	5	10	8
#hidden nodes	1	2	2	2
#output nodes	1	1	1	1
Learning rate	0.2	0.2	0.2	0.2
#Training Iterations	100	100	100	100

Table 6 summarizes the design of the back propagation procedure for this set of experiments on the artificial and UCI domains of the previous section. In Table 6, rows indicate the parameters of the back propagation procedure and columns indicate the domains

[3] Similar experiments were already conducted in [10] and [12] on different data sets and using different evaluation measures. We repeat them here on other domains for the sake of conciseness and completeness of the paper.

considered. The number of input nodes indicates the dimension of feature vectors of each domain. The number of hidden nodes in each domain was set to the values presented in table 6. Since we performed binary classification only, a single output node was used to indicate the class. In order to compare the domains based on their combination of concept complexity, training set size, and class imbalance (and no other factor), the learning rate and the number of training iterations were fixed at 0.2 and 100, respectively, in all the domains.

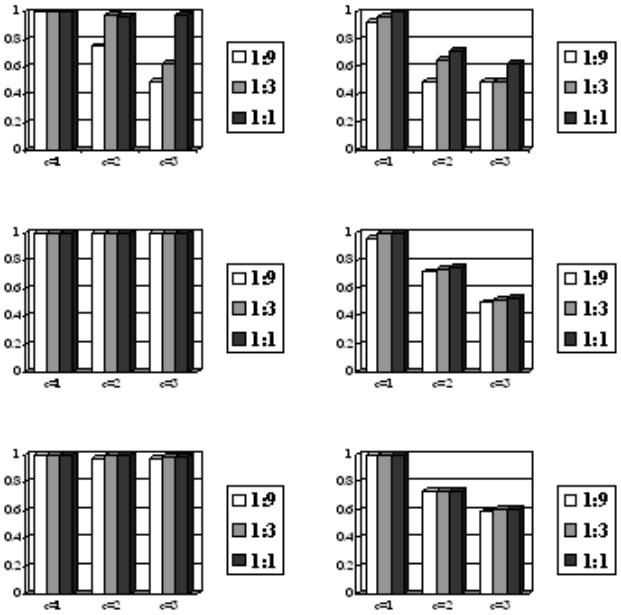

Figure 2. The Results of Class Imbalance in Backbone Model with Classifiers (Left-C4.5, Right-Back Propagation) and Training Set Size (Top-80, Middle-400, Bottom-1600)

Figure 2 displays the result of the performance of the two classifiers, C4.5 and back propagation, on the first artificial domain (with single dimension). In figure 2, each column indicates the result of a specific classifier; the graphs in the left column are the results of C4.5 and those in the right column are the results of back propagation. The rows in the figure indicate the training set size; the graphs in the top row are the results obtained with 80 training examples (i.e., the small size), those in the middle row are those obtained with 400 training examples (i.e., the middle size), and those in the bottom row are those obtained with 1600 training examples (i.e., the large size). In each graph, the y-axis indicates the "balanced accuracy" (i.e., the correct classification rate on the balanced test set) of each classifier, and the x-axis corresponds to the concept complexity (i.e., each cluster of three bars represents the results obtained at a different complexity level). In each cluster, the white bar indicates a high class imbalance, the gray bar indicates a middle class imbalance, and the black bar indicates a low class imbalance (i.e., a complete class balance). The results in figure 2 show that while class imbalances hinder classification performance in small data sets (and more so in domains with high concept complexity), this loss of performance gets gradually eradicated as the training set size increases. This suggests that the class imbalance problem may not always be to blame for the often observed performance loss that accompanies it. Rather, we suggest that this performance loss may be caused by the small disjuncts that can develop due to the presence of rare cases in domains with a small training set size and high complexity settings (as shown in Table 1 and discussed in Section 2).[4] The next section will test this hypothesis, but we first look at whether the phenomena just observed on our first artificial domains also recur in the second artificial domains and the real-world domains.

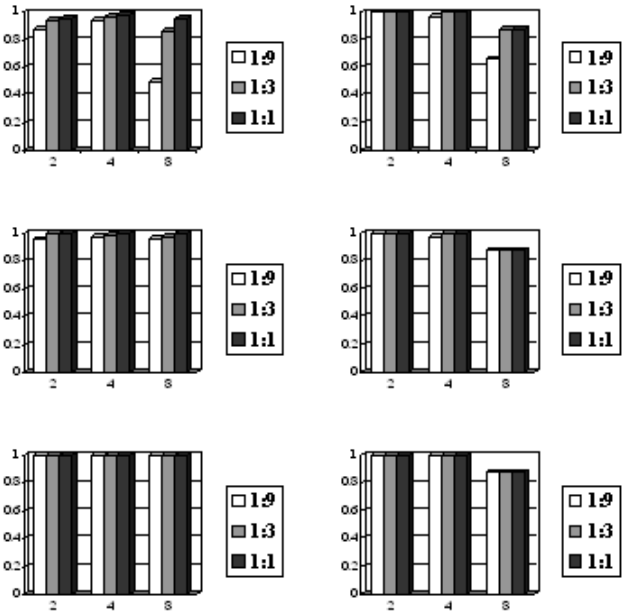

Figure 3. The Results of Class Imbalance in Five Dimensional Artificial Domains with Classifiers (Left-C4.5, Right-Back Propagation) and Training Set Size (Top-80, Middle-400, Bottom-1600)

Figure 3 displays the results observed on the second artificial domains with five dimensions. The arrangement of these graphs is identical to that in figure 2. Each cluster of bars in each graph corresponds to the number of clusters in the domain (that represents the concept complexity of these domains). These results also show that the performance loss experienced by the classifiers in small, complex and imbalanced domains disappears as the size of the training sets increases. As previously, these results suggest that the small disjunct problem may be more to blame than the class imbalance problem.

Figure 4 displays the results obtained by the classifiers on the two UCI domains. These results are based on the degree of class imbalance and the training set size (in real-world settings, we cannot vary the concept complexity of our domains, but our assumption is that it is quite high). In the arrangement of the graphs in figure 4, the columns indicate which classifier is considered (as in the two previous figures), but the rows indicate the domains of application. The top row represents the results obtained for the Wisconsin Breast Cancer Domain while the

[4] The fact that such small disjuncts are a problem for accurate classification is well documented in [14] which discusses the negative effect of a small training set size on the error rate of small disjuncts.

bottom row represents the Pima Indian Diabetes domain. Each cluster of bars in each graph indicates the size of the training set.

These results show that the performance of classifiers, though hindered by class imbalances, is repaired as the training set size increases. This, again, suggests that small disjuncts play a role in the performance loss of class imbalanced domains.

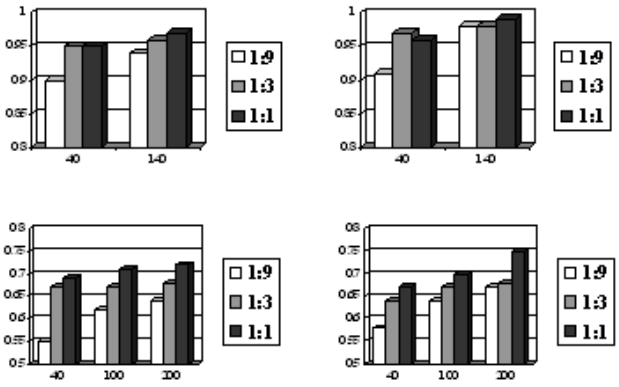

Figure 4. The Results of Class Imbalance in two Real Domains with Classifiers (Left-C4.5, Right-Back Propagation) (Top-Wisconsin Breast Cancer, Bottom-Pima Indian Diabetes)

5. CLASS IMBALANCE VERSUS SMALL DISJUNTS

The experiments of this section were designed to test the hypothesis suggested in the previous section. Namely, we will verify whether or not the loss of performance experienced by our classifiers is caused by the presence of rare cases in our domains (which lead to small disjuncts) rather than, simply, by the class imbalance. In order to test this hypothesis in our artificial domains, we kept the same generation scheme with respect to concept complexity and class imbalance, but we eliminated the notion of training set size. Instead, we set the cluster or interval size to 50 training examples in the small class and generated the number of data points necessary per cluster or interval to implement the particular imbalance level considered in the large class; This means that no rare cases remain.

In the real-world domains, the process was a little bit more complex since the rare cases are not explicit. We approximated these rare cases using a clustering algorithm (k-means, with k=4) on each class.[5] Unlike in the artificial domains, the generative model of the UCI domains is unknown. We approximated it by generating artificial examples based on the rare cases isolated by the k-means procedure. This example generation was done by injecting random values into the rare cases' input vectors following An's resampling method [9] (See Section 6.1). In these domains, rare cases consisted of the clusters (learned by k-means) of size smaller than 3 and such clusters were inflated to a cluster size of 30 examples.

The goal of our experiments in both the artificial and the UCI domains is to observe whether the performance of the classifiers improves as we inflate the rare cases, in an effort to eliminate the small disjuncts, without altering the degrees of class imbalance or concept complexity.

Table 7. The Distribution of Training Examples in Artificial Domain

Concept Complexity	Positive	Negative
c=1, #Clusters =2	50 * 1 = 50	450 * 1 = 450
c=2, #Clusters =4	50 * 2 = 100	450 * 2 = 900
c=3, #Clusters =8	50 * 4 = 200	450 * 4 = 1800

Table 7 shows the distribution of the training examples in the artificial domains used in these experiments. The degree of class imbalance is set to 1:9 and each cluster or interval contains 50 positive examples or 450 negative examples.

Figure 5 shows the results obtained by the classifiers with and without inflation of rare cases using the means just described on the artificial domains. In the arrangement of Figure 5, the graphs in the top row indicate the results obtained on the one-dimensional domain; the white bars correspond to the case where the rare cases were not inflated (white bars of top row in figure 2) while the black bars correspond to the results of the experiments of this section, with inflation of rare cases); The graphs in the bottom row show results obtained on the 5-dimensional domain. Once again, the white bars correspond to the white bars appearing in the graph of the top row of figure 3 (the case where the rare cases are not inflated) and the black bars correspond to the results obtained in this set of experiments (with inflation of the rare cases). In each graph, each cluster of bars corresponds to the concept complexity, i.e., the degree of complexity in the backbone model at the top and the number of clusters at the bottom. All the results are computed for a 1:9 class imbalance. Figure 5 shows that the inflation of rare cases improves the performance of the classifiers by eliminating the need for small disjuncts. The results are more notable at high concept complexity where the number of small disjuncts is the greatest. These results, thus, support our hypothesis stipulating that the loss of performance observed in the case of class imbalances is not caused by the class imbalance per se, but rather, by the fact that in cases of class imbalances, the small class contains rare cases leading to small disjuncts which, in turn, cause a decrease in performance.[6]

[5] As pointed out by both Thomas Dietterich and Peter Turney (in separate comments during presentations of this work), this solution is problematic since there is not guarantee that the clusters learned in the unsupervised step correspond to the small disjuncts created by the supervised step. However, as mentioned in Section 2.4, this is just an approximation which we assume sheds light on the identity of the rare cases. Future work will look at how to improve upon this approximation.

[6] Note, however, that in the single dimension artificial domain, the class imbalance problem seems to be more relevant than the small disjunct problem (in the other domains, this is not the case). This is seen at concept complexities c=2 and c=3 and for C4.5 where the performance obtained by inflating the small disjuncts remains inferior to the performance obtained by the fully balanced set (compare the black bar in the top left graph of Figure 5 to the black bar in the top left graph of Figure 2). This

Figure 6 shows the results obtained by the classifiers on the UCI domains with the same 1:9 class imbalance as in the artificial domains, and with and without the inflation of rare cases. The left graph indicates the result obtained in the Wisconsin Breast Cancer domain while the right graph corresponds to the results obtained in the Pima Indian Diabetes Domain. In both graphs, the white bar corresponds to the case where the rare cases remain (these bars correspond to the white bars in the left clusters of the graphs of Figure 4), while the black bars show the results obtained in the new experiments of this section, with the rare cases inflated. Each cluster of bars corresponds to a specific classifier: C4.5 or backpropagation. We note that the inflation of rare cases improves upon the performance of both classifiers in both domains and is especially pronounced in the Pima Indian Diabetes Domain.

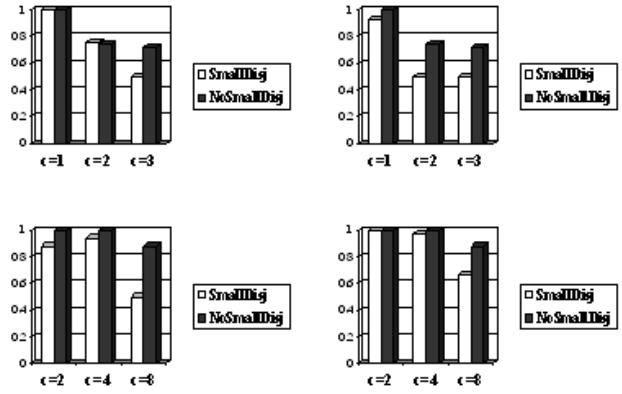

Figure 5. The Results of Removing Small Disjuncts in the Artificial Domains with Classifiers (Left-C4.5, Right-Back Propagation) (top-one-dimensional domain, bottom- 5-dimensional domain)

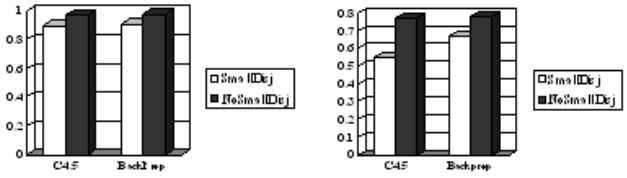

Figure 6. The Results of Removing Small Disjuncs in Real Domains with Classifiers (Left-Wisconsin Breast Cancer, Right-Pima Indian Diabetes)

The results on both the artificial and real-world domains, thus, show that the elimination of small disjuncts through inflation of rare cases improved the performance of both classifiers. In the artificial domains, the performance of the classifiers is improved when they are applied to the domains with the highest concept complexity. This is because these domains cause the largest number of small disjuncts. In the real-world settings, the Pima Indian Diabetes domain is more complex than the Wisconsin Breast Cancer domain (as shown in the experiments reported in Figure 4 by the fact that the accuracy obtainable in the Pima Indian Diabetes domain is smaller than that obtainable in the Wisconsin Breast Cancer domain). This leads to a greater amount of improvement in the Pima Indian Diabetes data set. These observations, thus, support the hypothesis that the elimination of small disjuncts through the inflation of rare cases is useful and particularly effective in imbalanced domains of high concept complexity.

6. METHODS CONSIDERED IN OUR EXPERIMENTS

This section describes the various methods tested in our experiments. The first category of methods described corresponds to standard methods that have previously been applied to the class imbalanced problem. These methods do not concern themselves with the problem of small disjuncts. The second type of method considered here considers the small disjunct problem, but does not consider the class imbalanced problem. Furthermore, it treats the small disjuncts in quite a radical way, simply removing them from the learned hypothesis. The last type of methods is the one we proposed in [7] and [8], cluster-based oversampling. It considers both the class imbalance and the small disjunct problem and, rather than removing the small disjuncts, it inflates the rare cases from which they are based so as to enhance them.

The purpose of our experiments will be to pit cluster-based oversampling against the large variety of other approaches described in this section. The comparison against the first category of methods is meant to compare cluster-based oversampling to other kinds of re-sampling methods as well as to cost-sensitive learning, which was previously reported (e.g., [10]) to be slightly superior to random re-sampling. The comparisons against the method of removing small disjuncts was done to verify that our somewhat complicated handling of small disjuncts is worthwhile.

6.1 Methods for Handling Class Imbalances with no regard for the Small Disjuncts

Three of the methods described in this subsection are re-sampling methods while the fourth one is a cost-based approach. Resampling is the process of manipulating the distribution of the training examples in an effort to improve the performance of classifiers. There is no guarantee that the training examples occur in their optimal distribution in practical problems and, thus, the idea of resampling is to add or remove examples with the hope of reaching the optimal distribution of the training examples and, thus, realizing the potential ability of classifiers. We describe two simple and one advanced methods designed for the class imbalanced problem without regard to the small disjunct problem (oversampling, undersampling and An's oversampling) as well as one method that seeks to address the class imbalance problem without re-sampling the data: cost-based learning.

Before discussing our various methods, let's assume that the given problem is one of binary classification in an unbalanced distribution of training examples. The minority class is the class

observation corroborates the observation made in [6] and supports our use of a method that handles the small disjunct and class imbalance problems simultaneously. (In, [7] we had tested two variations of the same method: the one described in this paper as well as one that handles rare cases but does not guarantee that the classes will ultimately be balanced. The method tested in this paper performed better, which is why we retained it).

with the lowest number of training examples and the majority class is the class with the highest number of training examples.

The first simple resampling method is (random) oversampling, which is the process of adding examples to the minority class by copying existing examples at random. It has one parameter, the addition rate, which corresponds to:

(Number of examples to add) /
(Difference in number of examples between the majority and the minority class)

If the addition rate is set to 1.0, then the strategy produces a completely balanced distribution.

The second simple resampling method is (random) undersampling, which is the process of removing examples in the majority class at random. It has one parameter, the removal rate, which corresponds to:

(Number of examples to remove) /
(Difference in number of examples between the majority and the minority class)

If the removal rate is set to 1.0, then the majority class is reduced to the size of the minority class to produce a completely balanced distribution.

The third strategy is more sophisticated than the two just described. It was proposed in 1996 by G. An [8]. This strategy contrasts with simple oversampling in that it generates additional training examples different from the existing ones. Such examples are called artificial training examples [8] and they are built by creating random vectors that follow the Gaussian distribution with a mean of zero and a covariance matrix that takes into consideration the values of each of the original training examples. For each original training example, an artificial training example is built by adding a random vector following the Gaussian distribution just mentioned (which corresponds to random noise [8]) to its input vector and keeping the label of the original data point. There are, thus, as many artificial training examples as there are original training examples. Both artificial and original training examples are used to train classifiers. This strategy of resampling doubles the number of training examples, but does not change the ratio of the minority to the majority class.

Cost modification consists of weighing errors made on examples of the minority class higher than those made on examples of the majority class in the calculation of the training error. This, in effect, rectifies the bias given to the majority class by standard classifiers when the training error corresponds to the simple (non-weighted) accuracy. In this experiment, errors on the positive examples (the minority class) have a weight of 1.0, while those on the negative ones (the majority class) have a weight of 0.1. These parameters were chosen so as to counter the 1:9 imbalance present in the domains we tested in the next section.

6.2 A Method for Eradicating Small Disjuncts

This section describes the radical method used to dispose of small disjuncts. It is based on the assumption that small disjuncts are unimportant to the learning task.

Punning of small disjuncts, which is only applicable to the decision trees of our experiments, consists of pruning branches of our decision trees, not based on the reduction of training error (regular pruning), but based on the number of covered training examples (our implemented pruning). In a decision tree, the branches containing a small number of covered training examples correspond to small disjuncts. In our experiments, branches covering fewer than three training examples were removed during the pruning step.

6.3 Cluster-Based Oversampling: A Method for Inflating Small Disjuncts

The resampling strategy proposed in this section consists of clustering the training data of each class (separately) and performing random oversampling cluster by cluster. This method was previously described and applied to several artificial domains as well as to letter recognition [10] and text classification [8] tasks. Its idea is to consider not only the *between-class imbalance* (the imbalance occurring between the two classes) but also the *within-class imbalance* (the imbalance occurring between the subclusters of each class) and to oversample the data set by rectifying these two types of imbalances simultaneously.

Before performing random oversampling, the training examples in the minority and the majority classes must be clustered. In this study, the k-means algorithm was adopted as our clustering algorithm, but many other algorithms could have been selected. K-means works as follows: k training examples are first selected at random as representative of each cluster. The input vector of these representative examples represents the mean of each cluster. The other training examples are processed one by one. For each of these examples, the distance between it and the k cluster centers is calculated. The example is attributed to the cluster closest to it. The cluster that received the example, has its mean vector updated by averaging the input vectors of all its corresponding examples.

Once the training examples of each class have been clustered, oversampling starts. In the majority class, all the clusters, except for the largest one, are randomly oversampled so as to get the same number of training examples as the largest cluster. Let *maxclasssize* be the overall size of the large class. In the minority class, each cluster is randomly oversampled until each cluster contains *maxclasssize/Nsmallclass* where *Nsmallclass* represents the number of subclusters in the small class.

For example, please consider that the training examples of majority and minority classes are respectively clustered as follows (the numbers represent the number of examples in each cluster of each class).

Majority Class: 10, 10, 10, 24

Minority Class: 2, 3, 2

This resampling strategy turns the above distribution into the one below.

Majority Class: 24, 24, 24, 24

Minority Class: 32, 32, 32

In the majority class, all the size 10 clusters get oversampled to 24 training examples, the largest majority subcluster. Since the minority class contains three clusters and since the size of the majority class after re-sampling is 96, all the minority class clusters are randomly oversampled until they contain 96/3= 32 examples. In this strategy, we see that oversampling is applied to both classes so that in the end, no between-class and no within class imbalance remains.

7. THE EFFECT OF CLUSTER-BASED OVERSAMPLING

This section reports on the effect of the cluster-based oversampling method described in the previous section. This effect is contrasted to that of all the other methods discussed in Section 6. For our experiments, all these methods were applied to the small training set (80 training examples) in the artificial domains and to the small (40 training examples) and the large/medium (140/100 training examples) training sets in the Wisconsin Breast Cancer and Pima Indian Diabetes domains, respectively. The degree of class imbalance in all cases is 1:9. In the testing set, the distribution is completely balanced with 50 positive and 50 negative examples in all cases. In addition, we report our results on the currency risk management problem which, as mentionned before, is a larger real world data set with a higher class imbalance. For both oversampling and undersampling, the resampling rate is set to 0.8[7]. In cluster-based oversampling, the k-means algorithm was run with k=4. In An's oversampling approach, the random values added to the elements of the input vector of each example were based on the Gaussian distribution of mean zero and variance 0.1. All our results are reported in Figures 8 and 9.

Figure 8 shows the results obtained on the artificial domains of 1) no resampling, 2) cost modification, 3) pruning the small disjuncts, and our four resampling methods (4) Oversampling, 5) Undersampling, 6) An, and 7) Cluster-Based Oversampling [Our proposed method]), in this order. The top row in this figure represents the first artificial domain and the bottom row represents the second. The graphs in the left column are the results obtained with C4.5, while those in the right column are those obtained with back propagation [Please, note that the backpropagation graphs have one less column than the C4.5 graphs (Column 3). This is because pruning of small disjuncts cannot be precisely implemented with backpropagation]. As before, each cluster of columns represents a different complexity level, and each column in each cluster represents a different method, displayed in the order listed above.

In the artificial domains, oversampling and cluster based oversampling show better result than cost modification and pruning based on small disjuncts. But undersampling and An's based oversampling show worse result than them. Since the artificial domains are simple problems, there is no notable difference between oversampling and cluster-based oversampling.

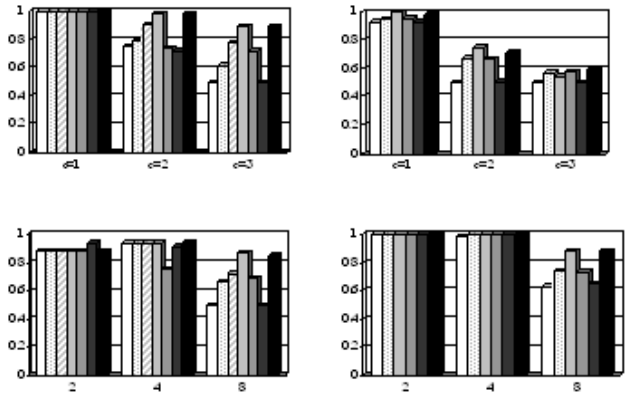

Figure 7. The Result of Performance of Classifiers with Resampling Methods in Artificial Domains with Classifiers (Left-C4.5, Right-Back Propagation) and Dimension (Top-Single Dimension, Bottom – Five Dimensions)

Figure 9 shows the results obtained on the three real-world domains. In figure 9, the top row represents the first domain, "Wisconsin Breast Cancer", the middle row represents the second one, "Pima Indian Diabetes", and the bottom row represents the third one, "Customer Decision in Foreign Currency Exchange".

In all three figures, we find that with the exception of An's oversampling, the resampling methods work generally better than cost modification and pruning of small disjuncts; An's oversampling shows the worst results among the four resampling methods. In these real-world domains, cluster based oversampling works better than any other method. In the first and second domains, cluster based oversampling is particularly effective in the small training set size case. This shows that this approach is quite tolerant to small domains.

Altogether, these experiments support the hypothesis that cluster based oversampling works better than simple oversampling or other methods for handling class imbalances or small disjuncts, especially when the number of training examples is small and the problem, complex. The reason is that cluster-based resampling identifies rare cases and re-samples them individually, so as to avoid the creation of small disjuncts in the learned hypothesis.

[7] If the resampling rate is 1.0, then the resampling methods build data sets with complete class balance. However, it was shown previously [12], that the perfect balance is not always the optimal rate. In this paper, we set the resampling rate to 0.8, since there were a number of cases in [12] where 0.8 was the optimal rate.

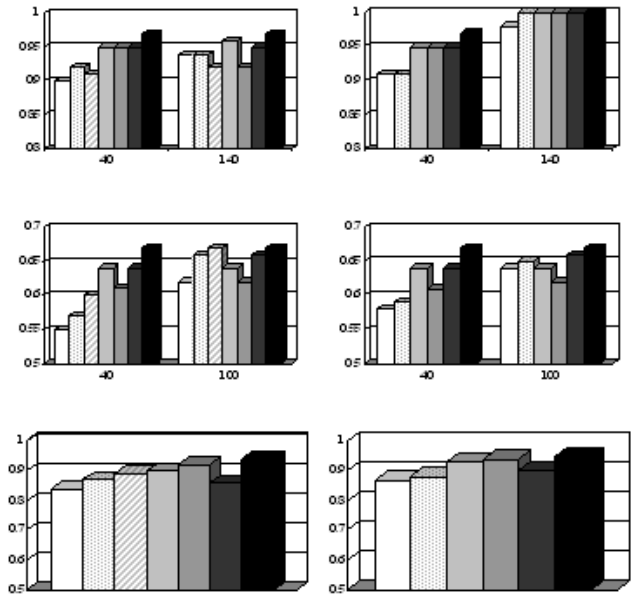

Figure 8. The Result of Performance of Classifiers with Resampling Methods in Real Domains with Classifiers (Left-C4.5, Right-Back Propagation) (Top-Wisconsin Breast Cancer, Middle – Pima Indian Diabetes, Bottom – Customer Decision in Foreign Currency Exchange)

8. CONCLUSION

The purpose of this study was to question whether the loss of performance incurred by classifiers faced with a class imbalance problem stems from the class imbalance per se or from a particular condition often caused by the class imbalance problem: the small disjunct problem. After showing that it is the small disjunct problem more than the class imbalance problem that is responsible for this decrease in accuracy, the paper questions whether it is more effective to use solutions that address both the class imbalance and the small disjunct problem simultaneously than it is to use solutions that address the class imbalance problem or the small disjunct problem, alone. The method we propose to deal with class imbalances and small disjuncts simultaneously, cluster-based oversampling, is shown to outperform all the class imbalance geared methods used in this study in the real-world domains. In the artificial domains, it was comparable to simple random oversampling. This preliminary result shows that, indeed, taking both problems into consideration is a worthy pursuit.

9. REFERENCES

[1] M. Kubat, R. Holte, and S. Matwin. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning* 30: 195-215, 1998.

[2] T.E. Fawcett and F. Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 3 (1): 291-316, 1997.

[3] D. Lewis and J. Catlett. Heterogeneous Uncertainty Sampling for Supervised Learning", *In Proceedings of the Eleventh International Conference of Machine Learning*, pages 148-156, 1994.

[4] P.M. Murphy and D.W. Aha. UCI Repository of Machine Learning Databases. *University California at Irvine, Department of Information and Computer Science*.

[5] R.C. Holte, L. E. Acker, and B.W. Porter. Concept Learning and the Problem of Small Disjuncts, In *Proceedings of the Eleventh Joint International Conference on Artificial Intelligence*, pages 813-818, 1989.

[6] Pearson R.K, Gonye, G.E and Schwaber, J.S., "Imbalanced Clustering of Microarray Time-Series", In *ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003.

[7] N. Japkowicz. Concept Learning in the Presence of Between-Class and Within-Class Imbalance. *Advances in Artificial Intelligence: Proceedings of the 14th Converences of the Canadian Society for Computational Studies of Intelligence*, pages 67-77, 2001.

[8] A. Nickerson, N. Japkowicz, and E. Millos, "Using Unsupervised Learning to Guide Resampling in Imbalanced Data Sets", In *Proceedings of the 8th International Workshop on AI and Statistics*, pages 261-265, 2001.

[9] G. An. The Effects of Adding Noise During Backpropagation Training on a Generalization Performance, *Neural Computation*, 8: 643-674, 1996.

[10] N. Japkowicz and S. Shaju, "The Class Imbalance Problem: A Systematic Study", *Intelligent Data Analysis*, Volume 6, Number 5, pp. 429-450, 2002.

[11] Visa, S. and Ralescu, A., "Learning Imbalanced and Overlapping Classes using Fuzzy Sets", In *ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003.

[12] A. Estabrooks, T. Jo, and N. Japkowicz, "A Multiple Resampling Method for Learning from Imbalanced Data Sets", *Computational Intelligence*, 28 (1): in press, 2004.

[13] G. M. Weiss, "Learning with Rare Case and Small Disjuncts", In *Proceedings of 17th International Conference on Machine Learning*, 558-565, 1995

[14] G. M. Weiss "A Quantitive Study of Small Disjuncts", In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 665-670, 2000