The Class Imbalance Problem: A Systematic Study

Nathalie Japkowicz and Shaju Stephen
School of Information Technology and Engineering
University of Ottawa
150 Louis Pasteur, P.O. Box 450 Stn. A
Ottawa, Ontario, Canada, B3H 1W5

Abstract ***In machine learning problems, differences in prior class probabilities—or class imbalances—have been reported to hinder the performance of some standard classifiers, such as decision trees. This paper presents a systematic study aimed at answering three different questions. First, we attempt to understand what the class imbalance problem is by establishing a relationship between concept complexity, size of the training set and class imbalance level. Second, we discuss several basic re-sampling or cost-modifying methods previously proposed to deal with class imbalances and compare their effectiveness. Finally, we investigate the assumption that the class imbalance problem does not only affect decision tree systems but also affects other classification systems such as Neural Networks and Support Vector Machines.***

Keywords: concept learning, class imbalances, re-sampling, misclassification costs, C5.0, Multi-Layer Perceptrons, Support Vector Machines

Introduction

As the field of machine learning makes a rapid transition from the status of “academic discipline” to that of “applied science”, a myriad of new issues, not previously considered by the machine learning community, is now coming into light. One such issue is the *class imbalance* problem. The class imbalance problem corresponds to the problem encountered by inductive learning systems on domains for which one class is represented by a large number of examples while the other is represented by only a few.[1]

The class imbalance problem is of crucial importance since it is encountered by a large number of domains of great environmental, vital or commercial importance, and was shown, in certain cases, to cause a significant bottleneck in the performance attainable by standard learning methods which assume a balanced class distribution. For example, the problem occurs and hinders classification in applications as diverse as the detection of oil spills in satellite radar images (Kubat et al., 98), the detection of fraudulent telephone calls (Fawcett and Provost, 97), in-flight helicopter gearbox fault monitoring (Japkowicz et al., 95), information retrieval and filtering (Lewis and Catlett, 94) and diagnoses of rare medical conditions such as thyroid diseases (Murphy and Aha, 94).

To this point, there have been a number of attempts at dealing with the class imbalance problem (Pazzani et al., 94; Japkowicz et al., 95; Ling and Li, 98; Kubat and Matwin, 97; Fawcett and Provost, 97; Kubat et al., 98; Domingos, 99; Chawla et al., 01; Elkan, 01); However, these attempts were mostly conducted in isolation. In particular, there has not been, to date, much systematic strive to link specific types of imbalances to the degree of inadequacy of standard classifiers nor are there been many comparisons of the various methods proposed to remedy the problem. Furthermore, no comparison of the performance of different types of classifiers on imbalanced data sets has yet been performed.[2]

[1] In this paper, we only consider the case of concept-learning. However, the discussion also applies to multi-class problems.

[2] Two studies attempting to systematize research on the class imbalance problem are worth mentioning, nonetheless: One, currently in progress at AT&T Lab, links different degrees of imbalances to the performance of C4.5, a decision Tree learning system on a large number of real-world data sets. However, it does not study the effect of concept complexity nor training set size in the context of their relationship with class imbalances, nor does it look at ways to remedy the class imbalance problem or the effect of class imbalances on classifiers other than C4.5. The second study is that by [Lawrence et al., 1998], which does not study the effect of class imbalances on classifiers' performance but which compares a number of specific approaches proposed to deal with class imbalances in the context of Neural Networks and on a few real-world data sets. In their study, no classifier other than Neural Networks were considered and no systematic study conducted.

The purpose of this paper is to address these three concerns in an attempt to unify the research conducted on this problem. In a first part, the paper concentrates on explaining what the class imbalance problem is by establishing a relationship between concept complexity, size of the training set and class imbalance level. In doing so, we also identify the class imbalance situations that are most damaging for a standard classifier that expects balanced class distributions. The second part of the paper turns to the question of how to deal with the class imbalance problem. In this part we look at five different methods previously proposed to deal with this problem and, all assumed to be more or less equivalent to each other. We attempt to establish to what extent these methods are, indeed, equivalent and to what extent they differ. The first two parts of our study were conducted using the C5.0 decision tree induction system. In the third part, we set out to find out whether or not the problems encountered by C5.0 when trained on imbalanced data sets are specific to C5.0. In particular, we attempt to find out whether or not the same pattern of hindrance is encountered by Neural Networks and Support Vector Machines and whether similar remedies can apply.

The remainder of the paper is divided into six sections. Section 2 is an overview of the paper explaining why the questions we set out to answer are important and how they will advance our understanding of the class imbalance problem. Section 3 describes the part of the study focusing on understanding the nature of the class imbalance problem and finding out what types of class imbalance problems create greater difficulties for a standard classifier. Section 4 describes the part of the study designed to compare the five main types of approaches previously attempted to deal with the class imbalance problem. Section 5 addresses the question of what effect class imbalances have on classifiers other than C5.0. Sections 6 and 7 conclude the paper.

Overview of the Paper

As mentioned in the previous section, the study presented in this paper investigates the following three series of questions:

Question 1: What is the nature of the class imbalance problem? i.e., in what domains do class imbalances most hinder the accuracy performance of a standard classifier such as C5.0?

Question 2: How do the different approaches proposed for dealing with the class imbalance problem compare?

Question 3: Does the class imbalance problem hinder the accuracy performance of classifiers other than C5.0?

These questions are important since their answers may put to rest currently assumed but unproven facts, dispel other unproven beliefs as well as suggest fruitful directions for future research. In particular, they may help researchers focus their inquiry onto the particular type of solution found most promising, given the particular characteristics identified in their application domain.

Question 1 raises the issue of when class imbalances are damaging. While the studies previously mentioned identified specific domains for which an imbalance was shown to hurt the performance of certain standard classifiers, they did not discuss the questions of whether imbalances are always damaging and to what extent different types of imbalances affect classification performances. This paper takes a global stance and answers these questions in the context of the C5.0 tree induction system on a series of artificial domains spanning a large combination of characteristics.[3]

Question 2 considers five related approaches previously proposed by independent researchers for tackling the class imbalance problem[4]:

1. Upsizing the small class at random.
2. Upsizing the small class at "focused" random.
3. Downsizing the large class at random.
4. Downsizing the large class at "focused" random.
5. Altering the relative costs of misclassifying the small and the large classes.

In more detail, Methods 1 and 2 consist of re-sampling patterns of the small class (either completely randomly or randomly but within parts of the input space close to the boundaries with the other class) until there are as many data from the small class as from the large one.[5] Methods 3 and 4 consists of eliminating data from the large class (either completely randomly or, randomly but within parts of the input space far away from the boundaries with the large class) until there are as many data in both classes. Finally, method 5 consists

[3] The paper, however, concentrates on domains that present a "between-class imbalance" in that the imbalance affects each subcluster of the small class to the same extent. Because of lack of space, the interesting issue of "within-class imbalances"—which are special cases of the problem of small disjuncts (Holte, 89)—has been omitted here. This very important question is dealt with elsewhere (Japkowicz, 01).

[4] In this study, we focus on discrimination-based approaches to the problem which base their decisions on both the positive and negative data. The study of recognition-based approaches which base their decision on one of the two classes but not both has been attempted in (Japkowicz, 00) but did not seem to do as well as discrimination-based methods (this might be linked, however, to the fact that the recognition threshold was not chosen very carefully. Nonetheless, we leave it to future work to determine truly whether or not that is the case).

[5] (Estabrooks, 00) and the AT&T study previously mentioned in Footnote 2 show that, in fact, the optimal amount of re-sampling is not necessarily that which yields the same number of data in each class. The optimal amount seems to depend upon the input domain and does not seem easy to estimate a priori. In order to simplify our study, here, we decided to re-sample until the two classes are of the same size. This decision will not alter our results, however, since we are interested in the *relative* performance of the different remedial approaches we consider.

of reducing the relative misclassification cost of the large class (or, equivalently, increasing that of the small one) to make it correspond to the size of the small class.

These methods were previously proposed by (Ling and Li, 98; Kubat and Matwin, 97; Domingos, 99; Chawla et al., 00; and Elkan, 01) but were not systematically compared before. Here, we compare the five methods, once again, to the data sets used in the previous part of the paper. This was done to see whether or not the five approaches for dealing with class imbalances respond to different domain characteristics in the same way.

Question 3, finally, asks whether the observations made in answering the previous questions for C5.0 also hold for other classifiers. In particular, we study the effect of class imbalances on Multi-Layer Perceptrons (MLPs), which could be thought of being capable of more flexible learning than C5.0, and thus, be less sensitive to class imbalances. We then repeat this study with Support Vector Machines (SVMs) which could be believed, not to be affected by this problem given that they base their classification on a small number of support vectors and, thus, may not be sensitive to the number of data representing each class. We look at the performance of MLPs and SVMs on a subset of the series of domains used in the previous part of the paper so as to see whether the three approaches are affected by different domain characteristics in the same ways.

Question 1: What is the nature of the Class Imbalance Problem?

In order to answer Question 1, a series of artificial concept-learning domains was generated that varies along three different dimensions: the degree of *concept complexity*, the *size* of the training set, and the level of *imbalance* between the two classes. The standard classifier system tested on this domain in this section was the C5.0 decision tree induction system (Quinlan, 93). This classifier has previously been shown to suffer from the class imbalance problem (e.g., (Kubat et al., 98)), but not in a completely systematic fashion. The study in this section aims at answering the question of what different faces a class imbalance can take and which of these faces hinders C5.0 most.

This part of the paper first discusses the domain generation process followed by a report of the results obtained by C5.0 on the various domains.

Domain Generation

For the experiments of this section, 125 domains were created with various combinations of *concept complexity*, *training set size*, and *degree of imbalance*. The generation method used was inspired by Schaffer who designed a similar framework for testing the effect of overfitting avoidance in sparse data sets (Schaffer, 93). From Schaffer's study, it was clear that the complexity of the concept at hand was an important part of the data overfitting problem and, given the relationship between the problem of overfitting the data and dealing with class imbalances (see (Kubat et al., 98)), it seems reasonable to assume that, here again, concept complexity is an important piece of the puzzle. Similarly, the training set size should also be a factor in a classifier's ability to deal with imbalanced domains given the relationship between the data overfitting problem and the size of the training set. Finally, the degree of imbalance is the obvious other parameter expected to influence a classifier's ability to classify imbalanced domains.

The 125 generated domains of our study were generated in the following way: each of the domain is one-dimensional with inputs in the [0, 1] range associated with one of the two

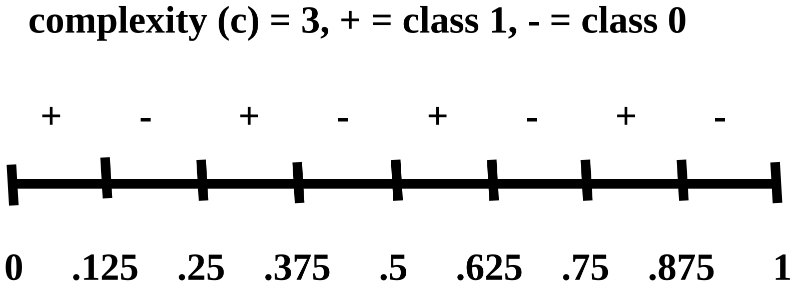


Figure 1: A Backbone Model of Complexity 3

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 on the size of the domain as well as on its degree of imbalance. An example of a backbone model is shown in Figure 1.

Five different complexity levels were considered ($c = 1..5$) 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, .5) is associated with a class value of 1, while every point whose input is in range (.5, 1] is associated with a class value of 0; At complexity level $c = 2$, points in intervals [0, .25) and (.5, .75) are associated with class value 1 while those in intervals (.25, .5) and (.75, 1] are associated with class value 0; etc., regardless of the size of the training set and its degree of imbalance.[6]

[6] In this paper, complexity is varied along a single very simple dimension. Other more sophisticated models could be used in order to obtain finer-grained results. In (Estabrooks, 00), for example, a k-DNF model using several dimensions was used to generate a few artificial domains presenting class imbalances. The study was less systematic than the one in this paper, but it yielded results corroborating those of this paper.

Five training set sizes were considered ($s = 1..5$) where each size, s, corresponds to a training set of size $round((5000/32) * 2^s)$. Since this training set size includes all the regular intervals in the domain, each regular interval is, in fact, represented by $round(((5000/32) * 2^s)/2^c)$ training points (before the imbalance factor is considered). For example, at a size level of $s = 1$ and at a complexity level of $c = 1$ and before any imbalance is taken into consideration, intervals [0, .5) and (.5, 1] are each represented by 157 examples; If the size is the same, but the complexity level is $c = 2$, then each of intervals [0, .25), (.25, .5), (.5, .75) and (.75, 1] contains 78 training examples; etc.

Finally, five levels of class imbalance were also considered ($i = 1..5$) where each level, i, corresponds to the situation where each sub-interval of class 1 is represented by all the data it is normally entitled to (given c and s), but each sub-interval of class 0 contains only $1/(32/2^i)$th (rounded) of all its normally entitled data. This means that each of the sub-intervals of class 0 are represented by $round((((5000/32)*2^s)/2^c)/(32/2^i))$ training examples. For example, for $c = 1$, $s = 1$, and $i = 2$, interval [0, .5) is represented by 157 examples and (.5, 1] is represented by 79; If $c = 2$, $s = 1$ and $i = 3$, then [0, .25) and (.5, .75) are each represented by 78 examples while (.25, .5) and (.75, 1] are each represented by 20; etc.

The number of testing points representing each sub-interval was kept fixed (at 50). This means that all domains of complexity level $c = 1$ are tested on 50 positive and 50 negative examples; all domains of complexity level $c = 2$ are tested on 100 positive and 100 negative examples; etc.

Results for Question 1

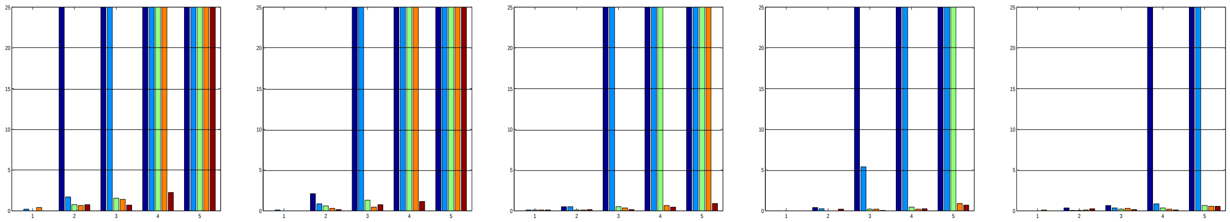
The results for C5.0 are displayed in Figures 2, 3, 4 and 5 which plots the error C5.0 obtained for each combination of concept complexity, training set size, and imbalance level, on the entire testing set. For each experiment, we reported four types of results: 1) the *corrected results* in which no matter what degree of class imbalance is present in the training set, the contribution of the false positive error rate is the same as that of the false negative one in the overall report.[7] 2) the *uncorrected results* in which the reported error rate reflects the same imbalance as the one present in the training set.[8] 3) the *false positive* error rate; and 4) the *false negative* error rate. The corrected and uncorrected results are provided so as to take into consideration two out of any possible number of situations: one in which, despite the presence of an imbalance, the cost of misclassifying the data of one class is the same as that of classifying those of the other class (the corrected version); the other situation is the one where the relative cost of misclassifying the two classes correspond to the class imbalance.[9]

Each plot in each of these figures represents the plot obtained at a different training set size. The leftmost plot corresponds to the smallest size ($s = 1$) and progresses until the rightmost plot which corresponds to the largest ($s = 5$). Within each of these plots, each cluster of five bars represent the concept complexity level. The leftmost cluster corresponds

[7] For this set of results, we simply report the error rate obtained on the testing set corresponding to the experiment at hand.

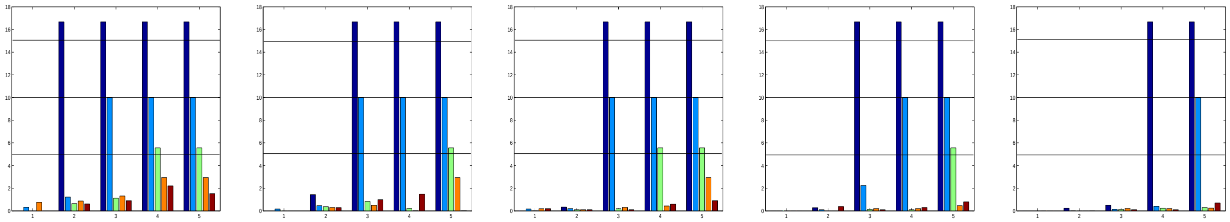
[8] For this set of results, we modify the ratio of false positive to false negative error obtained on the original testing set to make it correspond to the ratio of positive to negative examples in the training set.

[9] A more complete set of results could have involved comparisons at other relative costs as well. However, given our large number of experiments, this would have been unmanageable. We thus decided to focus on two meaningful and important cases only. Similarly, and for the same reasons, we decided not to vary C5.0's decision threshold across the ROC space (Swets et al., 2000). Since we are seeking to establish the relative performance of several classification approaches we believe that all the results obtained using the same decision threshold are representative of what would have happened along the ROC curves. We leave it to future work, however, to verify this assumption.



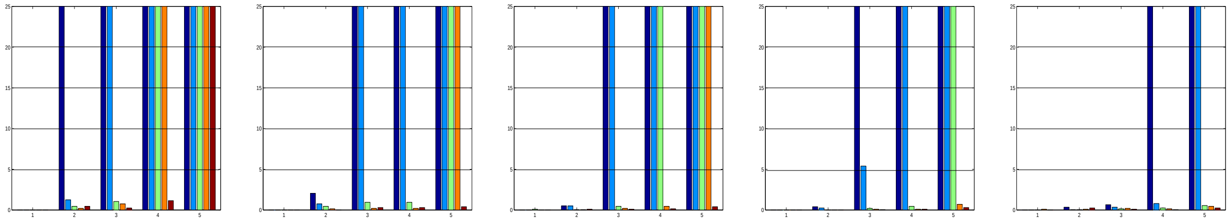
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 2: **C5.0 and the Class Imbalance Problem—Corrected**



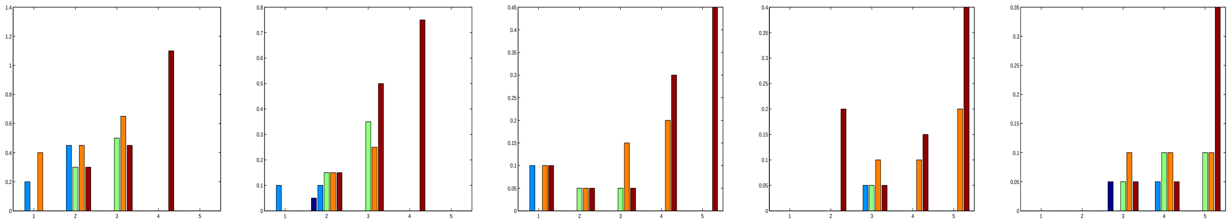
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 3: **C5.0 and the Class Imbalance Problem—UnCorrected**



(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 4: **C5.0 and the Class Imbalance Problem— False Positive Error Rate**



(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 5: **C5.0 and the Class Imbalance Problem— False Negative Error Rate: Very Close to 0**

to the simplest concept ($c = 1$) and progresses until the rightmost one which corresponds to the most complex ($c = 5$). Within each cluster, finally, each bar corresponds to a particular imbalance level. The leftmost bar corresponds to the most imbalanced level ($i = 1$) and progresses until the rightmost bar which corresponds to the most balanced level ($i = 5$, or no imbalance). The height of each bar represents the average percent error rate obtained by C5.0 (over five runs on different domains generated from the same backbone model) on the complexity, class size and imbalance level this bar represents. To make the comparisons easy, horizontal bars were drawn at every 5% marks. If a graph does not display any horizontal bars, it is because all the bars represent an average percent error below 5%, and we consider the error negligeable in such cases.

Our results reveal several points of interest: first, no matter what the size of the training set is, linearly separable domains (domains of complexity level $c = 1$) do not appear sensitive to any amount of imbalance. As a matter of fact, as the degree of concept complexity increases, so does the system's sensitivity to imbalances. Indeed, we can clearly see both in Figure 2 (the corrected results) and Figure 3 (the uncorrected results) that as the degree of complexity increases, high error rates are caused by lower and lower degrees of imbalances. Although the error rates reported in the corrected cases are higher than those reported in the uncorrected cases, the effect of concept complexity on class imbalances is clearly visible in both situations.

A look at Figures 4 and 5 explains the difference between Figures 2 and 3 since it reveals that most of the error represented in these graphs actually occurs on the negative testing set (i.e., most of the errors are false positive errors). Indeed, none of the average percents of false negative errors over all degrees of concept complexity and levels of imbalance ever exceed 5%. This is not surprising since we had expected the classifier to overfit the majority class, but the extent to which it does so might be a bit surprising.

As could be expected, imbalance rates are also a factor in the performance of C5.0 and, perhaps more surprisingly, so is the training set size. Indeed, as the size of the training set increases, the degree of imbalance yielding a large error rate decreases. This suggests that in very large domains, the class imbalance problem may not be a hindrance to a classification system. Specifically, the issue of relative cardinality of the two classes—which is often assumed to be the problem underlying domains with class imbalanced—may in fact be easily overridden by the use of a large enough data set (if, of course, such a data set is available and its size does not prevent the classifier from learning the domain in an acceptable time frame).

All in all, our study suggests that the imbalance problem is a *relative* problem depending on both the complexity of the concept represented by the data in which the imbalance occurs and the overall size of the training set, in addition to the degree of class imbalance present in the data. In other words, a huge class imbalance will not hinder classification of a domain whose concept is very easy to learn nor will we see a problem if the training set is very large. Conversely, a small class imbalance can greatly harm a very small data set or one representing a very complex concept.

Question 2: A Comparison of Various Strategies

Having identified the domains for which a class imbalance does impair the accuracy of a regular classifier such as C5.0, this section now proposes to compare the main methodologies that have been proposed to deal with this problem. First, the various schemes used for this comparison are described, followed by a comparative report on their performance. In all the experiments of this section, once again, C5.0 is used as our standard classifier.

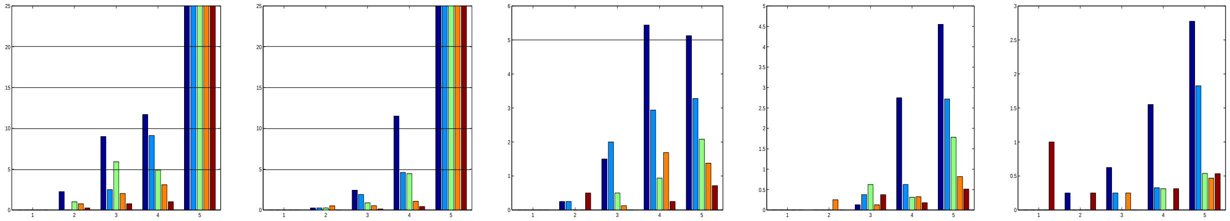
Schemes for Dealing with Class Imbalances

Over-Sampling Two oversampling methods were considered in this category. The first one, *random oversampling*, consists of oversampling the small class at random until it contains as many examples as the other class. The second method, *focused oversampling*, consists of oversampling the small class only with data occurring close to the boundaries between the concept and its negation. A factor of $\alpha = .25$ was chosen to represent closeness to the boundaries.[10]

Under-Sampling Two under-sampling methods, closely related to the over-sampling methods were considered in this category. The first one, *random undersampling*, consists of eliminating, at random, elements of the over-sized class until it matches the size of the other class. The second one, *focused undersampling*, consists of eliminating only elements further away (where, again, $\alpha = .25$ represents closeness to the boundaries)

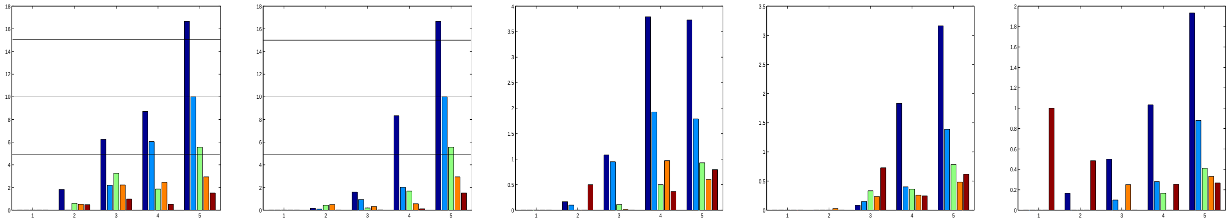
Cost-Modifying The cost-modifying method used in this study consists of modifying the relative cost associated to misclassifying the positive and the negative class so that it compensates for the imbalance ratio of the two classes. For example, if the data presents a 1:10 class imbalance in favour of the negative class, the cost of misclassifying a positive example will be set to 9 times that of misclassifying a negative one.

[10] This factor means that for interval [a, b], data considered close to the boundary are those in [a, a+ .25 × (b-a)] and [a+.75 × (b-a), b]. If no data were found in these intervals (after 500 random trials were attempted), then the data were sampled from the full interval [a, b] as in the *random oversampling* methodology.



(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 6: **Oversampling: Error Rate, Corrected**

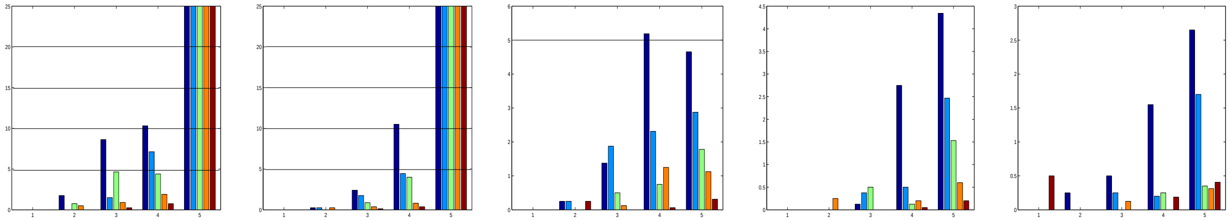


(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 7: **Oversampling: Error Rate, Uncorrected**

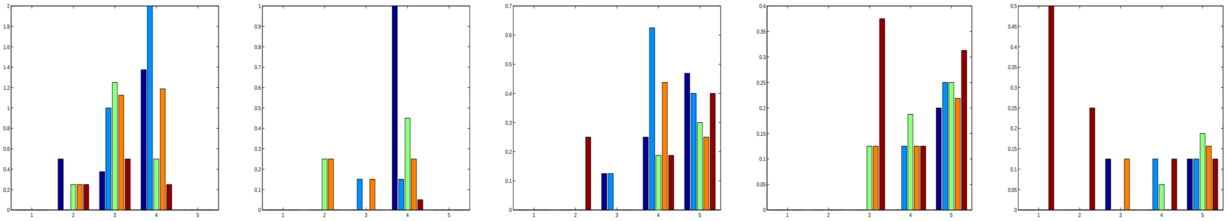
Results for Question 2

Like in the previous section, four series of results are reported in the context of each scheme: the corrected error, the uncorrected error, the false positive error and the false negative error. The format of the results is the same as that used in the last section. The results for random oversampling are displayed in Figures 6 to 9; those for focused oversampling, in Figures 10-13; those for random undersampling in Figures 14-17; those for focused undersampling in Figures 18-21; and those for cost-modifying, in Figures 22-25.



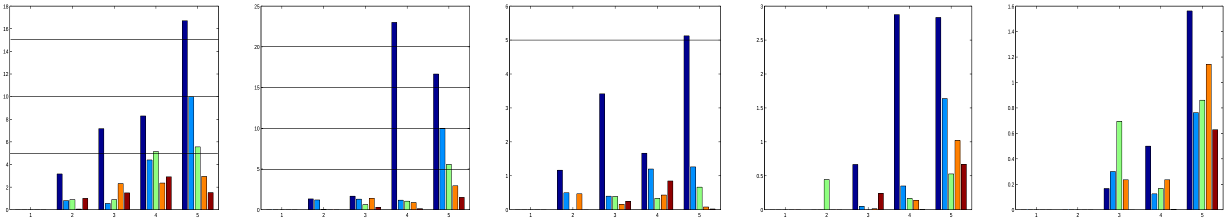
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 8: **Oversampling: False Positive Error Rate**



(a) **Size=1** (b) **Size=2** (c) **Size=3** (d) **Size=4** (e) **Size=5**

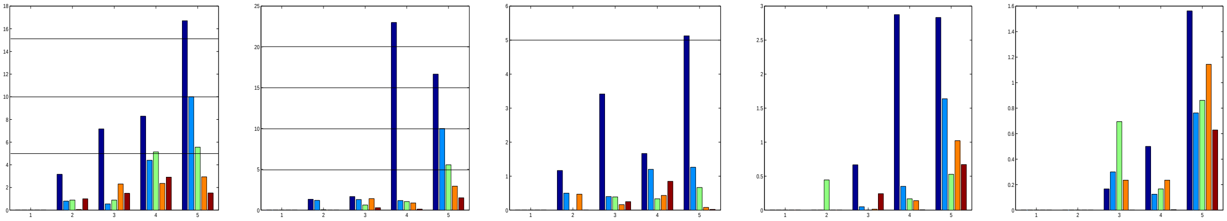
Figure 9: **Oversampling: False Negative Error Rate**



(a) **Size=1** (b) **Size=2** (c) **Size=3** (d) **Size=4** (e) **Size=5**

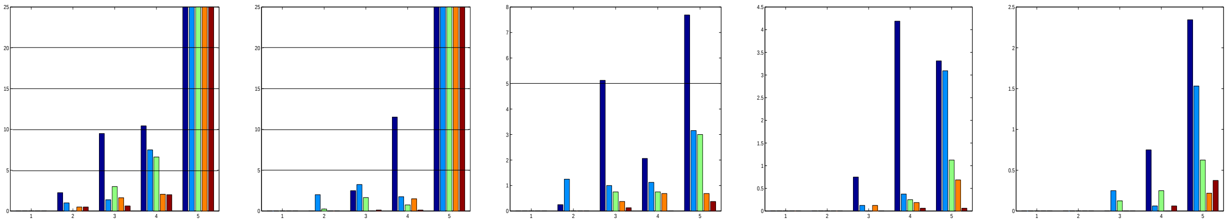
Figure 10: **Focused Oversampling: Error Rates, Corrected**

The results indicate a number of interesting points. First, all the methods proposed to deal with the class imbalance problem present an improvement over C5.0 used without any type of re-sampling nor cost-modifying technique both in the corrected and the uncorrected versions of the results. Nonetheless, not all methods help to the same extent. In particular, of all the methods suggested, undersampling is by far the least effective. This result is actually at odds with previously reported results (e.g., (Domingos, 99)), but we explain this disparity by the fact that in the applications considered by (Domingos, 99), the minority class is the class of interest while the majority class represents everything other than these examples



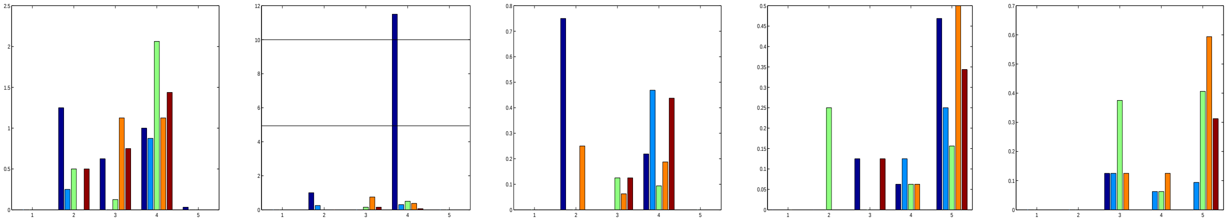
(a) **Size=1** (b) **Size=2** (c) **Size=3** (d) **Size=4** (e) **Size=5**

Figure 11: **Focused Oversampling: Error Rates, UnCorrected**



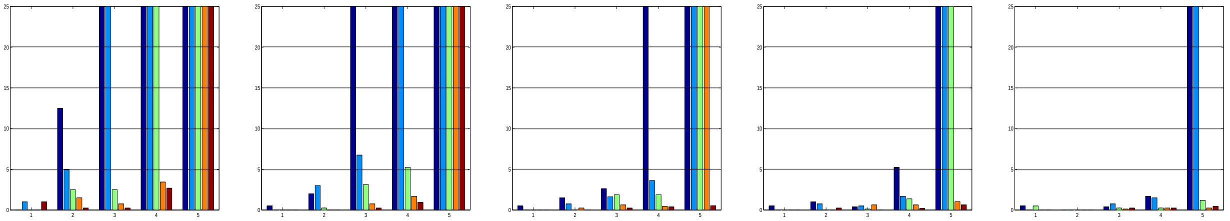
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 12: **Focused Oversampling: Error Rates, False Positives**



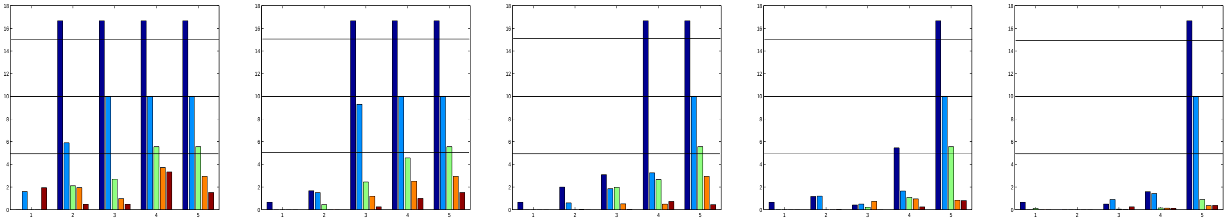
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 13: **Focused Oversampling: Error Rates, False Negatives**



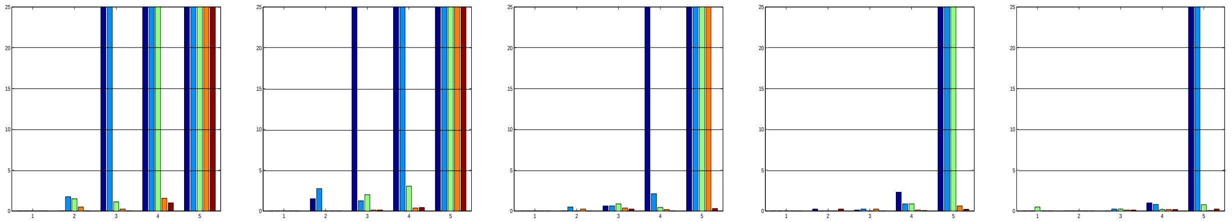
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 14: **Undersampling: Corrected Error Rate**



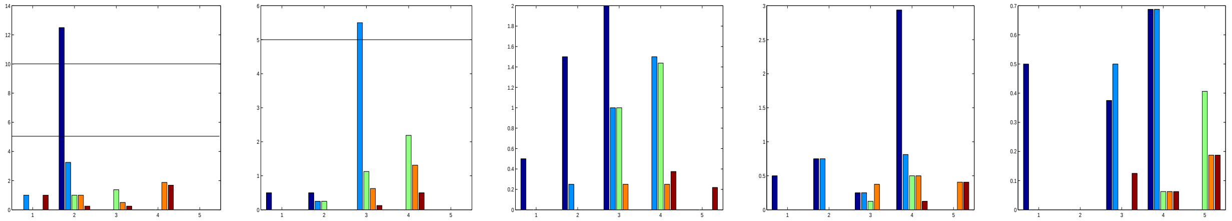
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 15: **Undersampling: Uncorrected Error Rate**



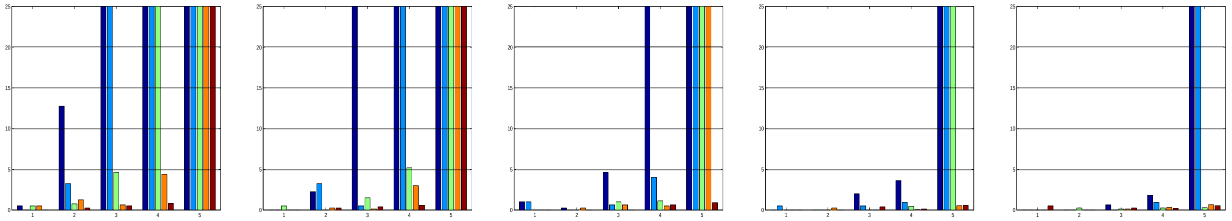
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 16: **Undersampling: False Positive Error Rate**



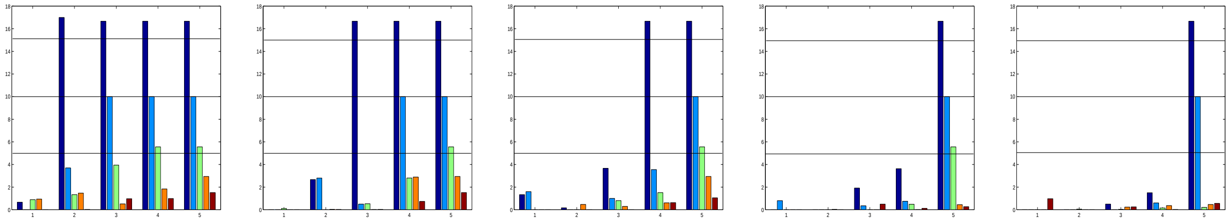
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 17: **Undersampling: False Negative Error Rate**



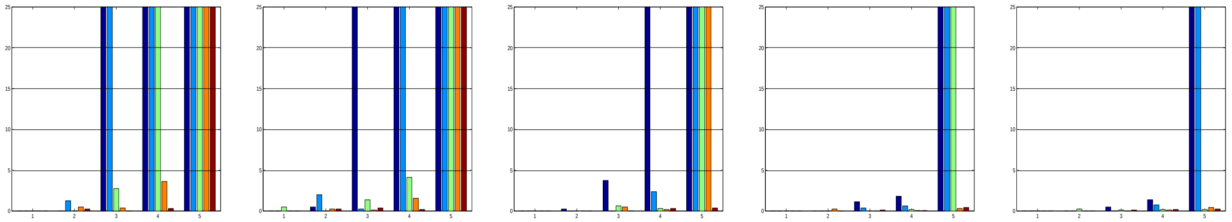
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 18: **Focused Undersampling: Error Rate, Corrected**



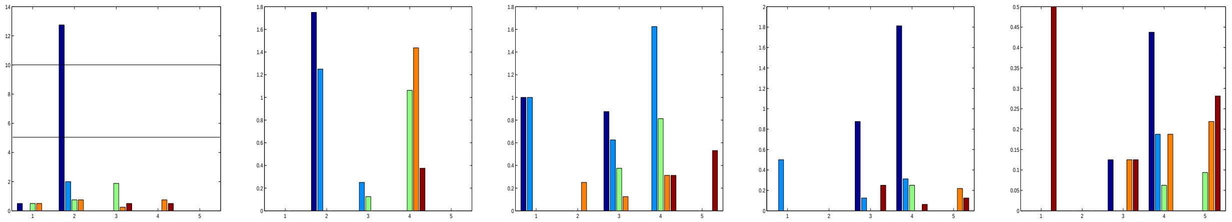
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 19: **Focused Undersampling: Error Rate, Uncorrected**



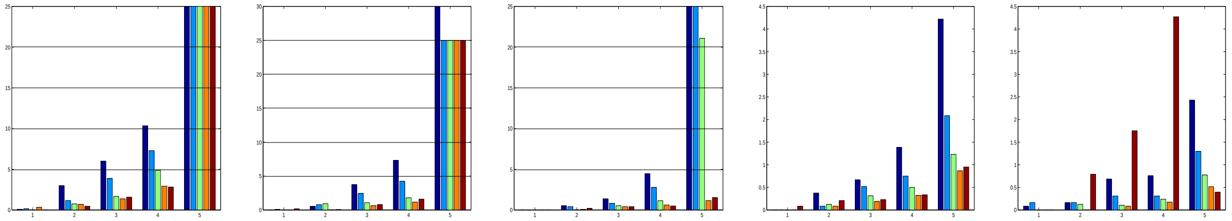
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 20: **Focused Undersampling: False Positive Error Rate**



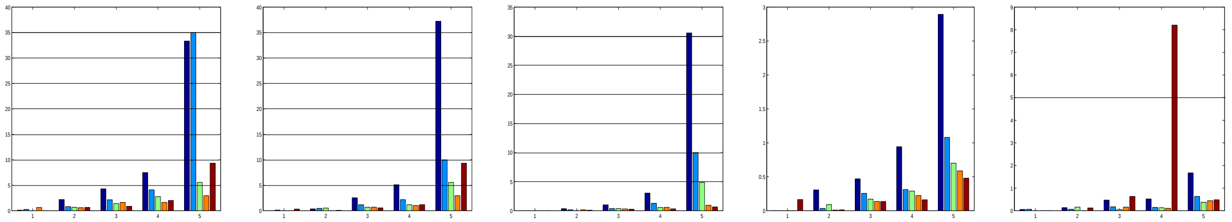
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 21: **Focused Undersampling: False Negative Error Rate**



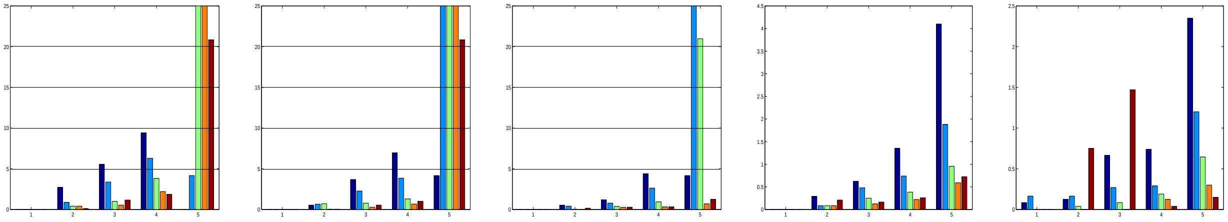
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 22: **Cost Modifying: Corrected Error Rate**



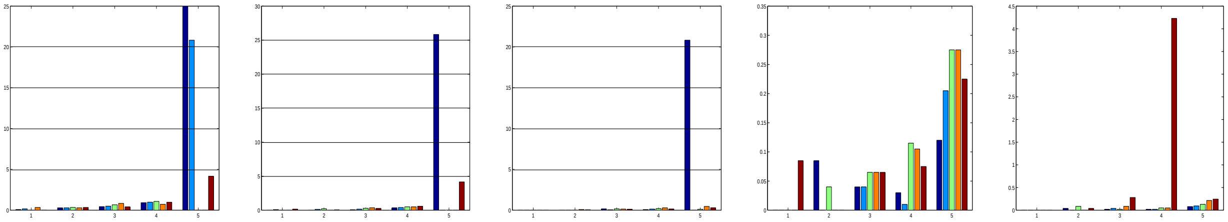
(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 23: **Cost Modifying: Uncorrected Error Rate**



(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 24: **Cost Modifying: False Positive Error Rate**



(a) Size=1 (b) Size=2 (c) Size=3 (d) Size=4 (e) Size=5

Figure 25: **Cost Modifying: False Negative Error Rate**

of interest. It follows that in domains such as (Domingos, 99)'s the majority class includes a lot of data irrelevant to the classification task at hand that are worth eliminating by undersampling techniques. In our data sets, on the other hand, the roles of the positive and the negative class are perfectly symmetrical and no examples are irrelevant. Undersampling is, thus, not a very useful scheme in these domains. Focused undersampling does not present any advantages over random undersampling on our data sets either and neither methods are recommended in those cases where the two classes play symmetrical roles and do not contain irrelevant data.

The situation, in the case of oversampling, is quite different. Indeed, oversampling is shown to help quite dramatically at all complexity and training set size level. Just to illustrate this fact, consider, for example, the situation at size 2 and degree of complexity 4: while in this case, any degree of imbalance (other than the case where no imbalance is present) causes C5.0 difficulties (see figures 2(b) and 3(b)), none but the highest degree of imbalance do so when the data is oversampled at random (see figures 6(b) and 7(b)). Contrarily to the case of undersampling, the focused approach does make a difference—albeit, small—in the case of oversampling. Indeed, at sizes 1 and 2, focused oversampling deals with the highest level of complexity better than random oversampling (compare the results at degree of difficulty 5 in figures 6(a, b) and 7(a, b) on the one hand and figures 10(a, b) and 11(a, b), on the other hand). Interestingly, the improvement in overall error does not seem to affect the distribution of the error. Indeed, as Figures 8, 9, 12 and 13 will attest, while the false positive rate has decreased, the false negative one has not significantly increased despite the fact that the size of the positive training set has increased dramatically. This is quite an important result since it contradicts the expectation that oversampling would have shifted the error distribution and, thus, not much helped in the case where it is essential to preserve a low false negative error rate while learning the false positive error rate. In summary, oversampling and focused oversampling seem quite effective ways of dealing with the problem, at least in situations such as those represented in our training set.

The last method, cost-modifying, is more effective than both random oversampling and focused oversampling in all but a single observed case, that of concept complexity 5 and Size 3 (compare the results for concept complexity 5 in figures 6(c), 7(c), 10(c) and 11(c) on the one hand to those of figures 22(c) and 23(c) on the other). In this case both random and focused oversampling are more accurate than cost-modifying. The generally better results obtained with the cost-modifying method over those obtained by oversampling are in agreement with (Lawrence et al., 98) who suggest that modifying the relative cost of misclassifying each class allows to achieve the same goals as oversampling without increasing the training set size, a step that can harm the performance of a classifier. Nonetheless, although we did not show it here, we assume that in those cases where the majority class contains irrelevant examples, undersampling methods may be more effective than cost modifying ones.

Question 3: Are other classifiers also sensitive to Class Imbalances in the Data?

Sections 1 and 2 studied the question of how class imbalances affect classification and how they can be countered all in the context of C5.0, a decision tree induction system. In this section, we are concerned about whether classification systems using other learning paradigms are also affected by the class imbalance problem and to what extent. In particular, we consider two other paradigms which, a priori, may seem less prone to hindrances in the face of class imbalances than decision trees: Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs).

MLPs can be believed to be less prone to the class imbalance problem because of their flexibility. Indeed, they may be thought to be able to compute a less global partition of the space than decision tree learning systems since they get modified by each data point sequentially and repeatedly and thus follow a top-down as well as a bottom-up search of the hypothesis space simultaneously. Even more convincingly than MLPs, SVMs can be believed to be less prone to the class imbalance problem than C5.0 because boundaries between classes are calculated with respect to only a few support vectors, the data points located close to the other class. The size of the data set representing each class may, thus, be believed not to matter given such an approach to classification.

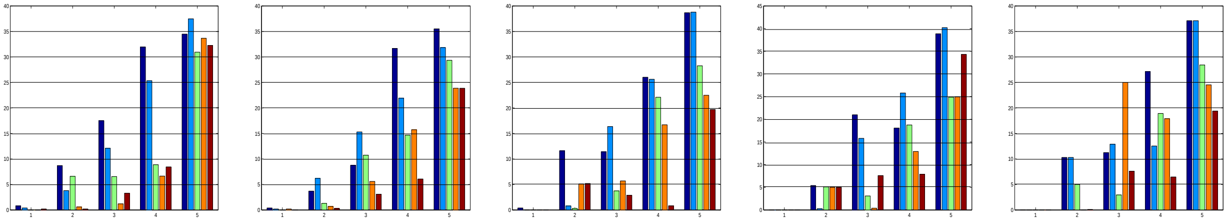
The point of this section is to assess whether indeed MLPs and SVMs are less prone to the class imbalance problem and if so, to what extent. Again, we used domains belonging to the same family as the ones used in the previous section to make this assessment. Nonetheless, because MLP and SVM training is much less time-efficient than C5.0 training and because SVM training was not even possible for large domains on our machine (because of a lack of memory), we did not conduct as extensive a set of experiments as we did in the previous sections. In particular, because of memory restrictions, we restricted our study of the effects of class imbalances to domains of size 1 for SVMs (for MLPs, we actually conducted our study on all sizes for the imbalance study since we did not have memory problems) and, because of low training efficiency, we only looked at the effect of random oversampling and undersampling for size 1 on both classifiers.[11]

MLPs and the Class Imbalance Problem

Because of the nature of MLPs, more experiments needed to be ran than in the case of C5.0. Indeed, because the performance of MLPs depends upon the number of hidden units it uses, we experimented with 2, 4, 8 and 16 hidden units and reported only the results obtained with the optimal network capacity. Other default values were kept fixed (i.e., all the networks were trained by the Levenberg-Marquardt optimization method, the learning rate was set at 0.01; the networks were all trained for a maximum of 300 epochs or until the performance gradient descended below 10^{-10}; and the threshold for discrimination between the two classes was set at 0.5). This means that the results are reported a-posteriori (after checking all the possible network capacities, the best results are reported).

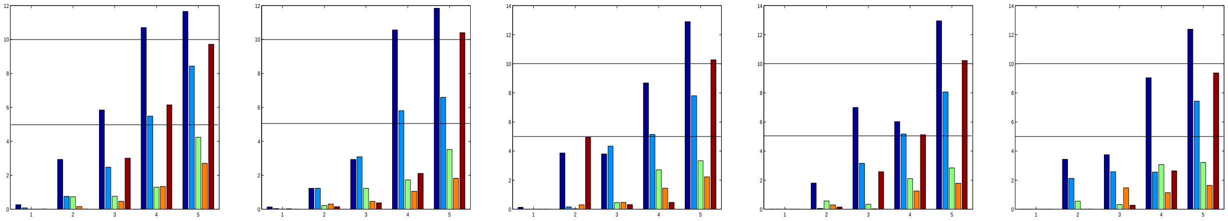
The results are presented in Figures 26, 27, 28 and 29 for concept complexities c=1..5,

[11] Unlike in Question 2 for C5.0, our intent here is not to compare all possible techniques for dealing with the class imbalance problem with MLPs and SVMs. Instead, we are only hoping to shed some light on whether these two systems do suffer from class imbalances and get an idea of whether some simple remedial methods can be considered for dealing with the problem.



(a) **Size=1** (b) **Size=2** (c) **Size=3** (d) **Size=4** (e) **Size=5**

Figure 26: **MLPs and the Class Imbalance Problem—Corrected**

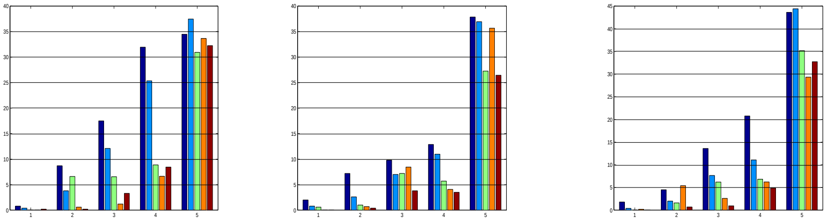


(a) **Size=1** (b) **Size=2** (c) **Size=3** (d) **Size=4** (e) **Size=5**

Figure 27: **MLPs and the Class Imbalance Problem—Uncorrected**

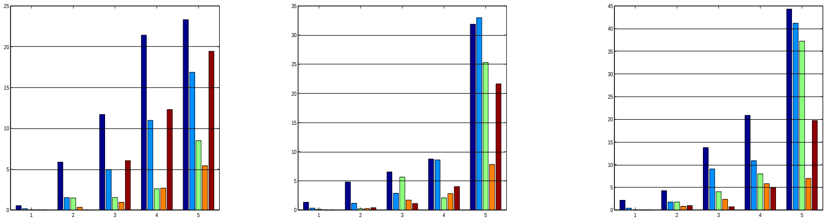
training set sizes s=1..5, and imbalance levels i=1..5. The format used to report these results is the same as the one used in the previous two sections.

There are several important differences between the results obtained with C5.0 and those obtained with MLPs. In particular, in all the MLP graphs a large amount of variance can be noticed in the results despite the fact that all results were averaged over five different trials. The conclusions derived from these graphs thus should be thought of reflecting general trends rather than specific results. Furthermore, a careful analysis of the graphs reveals that MLPs do not seem to suffer from the class imbalance problem in the same way as C5.0.



(a) **Imbalanced** (b) **Oversampling** (c) **Undersampling**

Figure 28: **Lessening the Class Imbalance Problem in MLP Networks—Corrected**



(a) Imbalanced **(b) Oversampling** **(c) Undersampling**

Figure 29: **Lessening the Class Imbalance Problem in MLP Networks—UnCorrected**

Looking, for example, at the graphs for size 1 for C5.0 and MLP (see figures 2(a) and 3(a) on the one hand and figures 26(a) and 27(a) on the other hand), we see that C5.0 displays extreme behaviors: it either does a perfect (or close to perfect) job or it misclassifies 25% of the testing set wrongly (see figure 2(a)). For MLP, this is not the case and misclassification rates span an entire range. As a result, MLP seems less affected by the class imbalance problem than C5.0. For example, for size=1, and concept complexity 4, C5.0 ran with imbalance levels 4, 3, 2, and 1 (see figure 2(a)) misclassify 25% of the testing set whereas MLP (see figure 26(a)) misclassifies the full 25% of the testing set for only imbalance levels 2 and 1—the highest degrees of imbalance (some misclassification also occurs at imbalance levels 3 and 4, but not as drastic as for levels 2 and 1). Note that the difficulty displayed by MLPs at concept complexity 5 for all sizes is probably caused by the fact that, once again for efficiency reasons, we did not try networks of capacity greater than 16 hidden units. We, thus, ignore these results in our discussion.

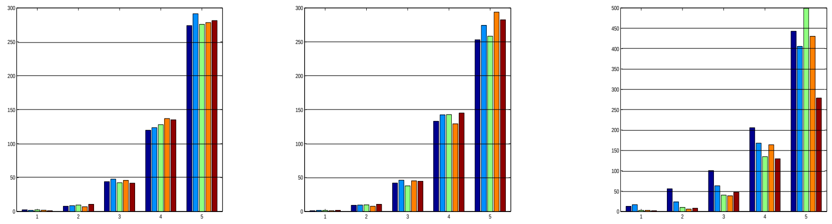
Another important difference that can be seen by looking at the graphs for size 5 of both C5.0 (figures 2(e) and 3(e)) and MLP (figure 26(e) and 27(e)) is that while the overall size of the training set makes a big difference in the case of C5.0, it doesn't make any difference for MLP: except for the highest imbalance levels combined with the highest degrees of complexity, C5.0 does not display any noticeable error at training set size 5—the highest. MLPS's on the other hand do. This may be explained by the fact that it is more difficult for MLP networks to process large quantities of data than it is for C5.0.

Because MLP generally suffers from the class imbalance problem, we asked whether, like for C5.0, this problem can be lessened by simple techniques. For the reason of efficiency noted earlier and for reasons of conciseness of report, we restricted our experiments to the cases of random oversampling and random undersampling and to the smallest size (size 1) case. The results of these experiments are shown in Figures 28 and 29 which display the results obtained with no re-sampling at all (a repeat of figures 26(a) and 27(a)), random oversampling and random undersampling. Only the corrected and uncorrected results are reported.

The results in these figures show that both oversampling and undersampling have a noticeable effect for MLPs, though once again, oversampling seems more effective. The difference in effectiveness between undersampling and oversampling, however, is less pronounced in the case of MLPs than it was in the case of C5.0. As a matter of fact, undersampling is much less effective than oversampling for MLP in the most imbalanced cases, but it has comparable effectiveness in all the other ones. This suggests that like for C5.0, simple methods for counteracting the effect of class imbalances should be considered when using MLPs.

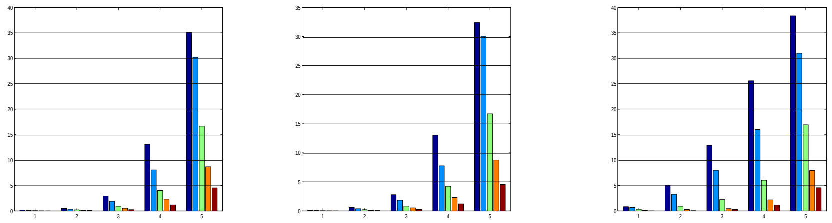
SVMs and the Class Imbalance Problem

Like for MLPs, more experiments needed to be ran with SVMs than in the case of C5.0. Actually, even more experiments were ran with SVMs than with MLPs. We ran SVMs with a Gaussian Kernel but since the optimal variance of this kernel is unknown, we tried 10



(a) Imbalanced **(b) Oversampling** **(c) Undersampling**

Figure 30: The Class Imbalance Problem in SVMs—Corrected.



(a) Imbalanced **(b) Oversampling** **(c) Undersampling**

Figure 31: The Class Imbalance Problem in SVMs—Uncorrected.

different possible variance values for each experiment. We experimented with variances 0.1, 0.2, etc. up to 1. We did not experiment with modifications to the soft margin threshold (note that such experiments would be equivalent to the cost-modification experiments of C5.0). Like for MLPs, the results are reported a-posteriori (after checking the results with all the possible variances, we report only the best results obtained).

As mentioned before, because of problems of memory capacity, the results are reported for a training set size of 1 and it was not possible to report results similar to those reported for MLP in Figures 26 and 27. Instead, we report results similar to those of Figures 28 and 29 for MLPs in Figures 30 and 31 for SVMs. In particular, these figures show results obtained by SVMs with no resampling at all, random oversampling and random undersampling for size 1.

The results displayed in Figures 30(a) and 31(a) show that there is a big difference between C5.0 and MLPs on the one hand and SVMs on the other. Indeed, while for both

C5.0 and MLPs, the leftmost column in a cluster of columns—those columns representing the highest degree of imbalances—were higher than the others (figures 2(a) and 26(a)), in the case of SVMs (figure 30(a)) the 5 columns in the clusters display a flat value or the leftmost columns have lower values than the rightmost ones (see the case of concept complexity 4 in particular). The uncorrected results in Figure 31(a) reflect the fact that SVMs are completely insensitive to class imbalances and make, on average, as many errors on the positive and the negative testing set, and are shown to suffer if the relative cost of misclassifying the two classes is altered in favour of one or the other class.

This is quite interesting since it suggests that SVMs are absolutely not sensitive to the class imbalance problem (this, by the way, is similar to the property of the decision tree splitting criterion introduced by (Drummond and Holte, 00)). As a matter of fact, Figures 30 and 31 (b) and (c) show that oversampling the data at random does not help in any way and undersampling it at random even hurts the SVM's performance.

All in all, this suggests that when confronted to a class imbalance situation, it might be wise to consider using SVMs since they are robust to such problems. Of course, this should be done only if SVMs fare well on the particular problem at hand as compared to other classifiers. In our domains, for example, up to concept complexity 3 (included), SVMs (figure 30(a)) are competitive with MLPs (figure 28(a)) and only slightly less competitive with oversampled C5.0 (figure 6(a)). At complexity 4, oversampled MLPs (figure 28(b)) and C5.0 (figure 6(a)) are more accurate than SVMs (figure 30(a)).

Conclusion

The purpose of this paper was to explain the nature of the class imbalance problem, compare various simple strategies previously proposed to deal with the problem and assess the effect of class imbalances on different types of classifiers.

Our experiments allowed us to conclude that the class imbalance problem is a relative problem that depends on 1) the degree of class imbalance; 2) the complexity of the concept represented by the data; 3) the overall size of the training set; and 4) the classifier involved.

More specifically, we found that the higher the degree of class imbalance the higher the complexity of the concept and the smaller the overall size of the training set, the greater the effect of class imbalances in classifiers sensitive to the problem. The three types of classifiers we tested were not sensitive to the class imbalance problem in the same way: C5.0 was the most sensitive of the three, MLPs came next and displayed a different pattern of sensitivity (a grayer-scale type compared to C5.0's which was more categorical); and SVMs came last since they were shown not to be at all sensitive to this problem.

Finally, for classifiers sensitive to the class imbalance problem, it was shown that simple re-sampling methods could help a great deal whereas they do not help, and in certain cases, even hurt the classifier insensitive to class imbalances. An extensive and careful study of the classifier most affected by class imbalances, C5.0, reveals that while random re-sampling is an effective way to deal with the problem, random oversampling is a lot more useful than random undersampling. More "intelligent" oversampling helps even further, but more "intelligent" undersampling does not. The cost-modifying method seems more appropriate than the over-sampling and even focused over-sampling method except in one case of very high complexity and medium-range training set size.

Future Work

The work in this paper presents a systematic study of class imbalance problems on a large family of domains. Nonetheless, this family does not cover all the known characteristics that a domain may take. For example, we did not study the effect of irrelevant data in the majority class. We assume that such a characteristic should be important since it it may make undersampling more effective than oversampling or even cost-modifying on domains presenting a large variance in the distribution of the large class. Other characteristics should also be studied since they may reveal other strengths and weaknesses of the remedial methods surveyed in this study.

In addition several other methods for dealing with class imbalance problems should be surveyed. Two approaches in particular are 1) over-sampling by creation of new synthetic data points not present in the original data set but presenting similarities to the existing data points and 2) learning from a single class rather than from two classes, trying to *recognize* examples of the class of interest rather than *discriminate* between examples of both classes.

Finally, it would be interesting to combine, in an "intelligent" manner, the various methods previously proposed to deal with the class imbalance problem. Preliminary work on this subject was previously done by (Chawla et al., 01) and (Estabrooks and Japkowicz, 01), but much more remains to be done in this area.

Bibliography

Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, P. SMOTE: Synthetic Minority Oversampling TEchnique *International Conference on Knowledge Based Computer Systems*, 2000

Domingos, P. Metacost: A General Method for Making Classifiers Cost-sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155-164

Drummond, Chris and Holte, Robert Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria, *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 239-249, 2000.

Elkan, Charles The Foundations of Cost-Sensitive Learning *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001

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

Estabrook, A. and Japkowicz, N. A Mixture-of-Experts Framework for Concept-Learning from Imbalanced Data Sets, *Proceedings of the 2001 Intelligent Data Analysis Conference.*, 2001

Tom E. Fawcett and Foster Provost Adaptive Fraud Detection *Data Mining and Knowledge Discovery*, 3(1):291–316, 1997.

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

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

Nathalie Japkowicz Learning from Imbalanced Data Sets: A Comparison of Various Solutions *Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets*, 2000.

Japkowicz, N. Concept-Learning in the Presence of Between-Class and Within-Class Imbalances *Advances in Artificial Intelligence: Proceedings of the 14th Conference of the Canadian Society for Computational Studies of Intelligence,* pp. 67-77, 2001.

Lawrence, S., Burns, I., Back, A.D., Tsoi, A.C., Giles, C.L., Neural Network Classification and Unequal Prior Class Probabilities G. Orr, R.-R. Muller, and R. Caruana, editors, *Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys*, pp. 299-314. Springer Verlag, 1998.

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

Miroslav Kubat, Robert Holte and Stan Matwin Machine Learning for the Detection of Oil Spills in Satellite Radar Images *Machine Learning*, 30:195–215, 1998.

Murphy, P.M., and Aha, D.W. UCI Repository of Machine Learning Databases. University of California at Irvine, Department of Information and Computer Science, 1994.

Lewis, D. and Catlett, J. Heterogeneous Uncertainty Sampling for Supervised Learning *Proceedings of the Eleventh International Conference of Machine Learning*, pp. 148-156, 1994.

Charles X. Ling and Chenghui Li Data Mining for Direct Marketing: Problems and Solutions *International Conference on Knowledge Discovery & Data Mining*, 1998.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. a nd Brunk, C. Reducing Misclassification Costs *Proceedings of the Eleventh International Conference on Machine Learning*, 217–225, 1994.

David E. Rumelhart, Geoff E. Hinton and R. J. Williams Learning Internal Representations by Error Propagation *Parallel Distributed Processing*, David E. Rumelhart and J. L. McClelland (Eds), MIT Press, Cambridge, MA, 318–364, 1986.

Cullen Schaffer Overfitting Avoidance as Bias *Machine Learning*, 10:153–178, 1993.

Swets, J., Dawes, R., and Monahan, J. Better Decisions through Science *Scientific American*, October 2000: 82-97.