Concept-Learning in the Presence of *Between-Class* and *Within-Class* Imbalances

Nathalie Japkowicz*

School of Information Technology and Engineering
University of Ottawa
150 Louis Pasteur, P.O. Box 450 Stn. A
Ottawa, Ontario, Canada K1N 6N5
nat@site.uottawa.ca

Abstract. In a concept learning problem, imbalances in the distribution of the data can occur either *between* the two classes or *within* a single class. Yet, although both types of imbalances are known to affect negatively the performance of standard classifiers, methods for dealing with the class imbalance problem usually focus on rectifying the between-class imbalance problem, neglecting to address the imbalance occuring within each class. The purpose of this paper is to extend the simplest proposed approach for dealing with the between-class imbalance problem—random re-sampling—in order to deal simultaneously with the two problems. Although re-sampling is not necessarily the best way to deal with problems of imbalance, the results reported in this paper suggest that addressing both problems simultaneously is beneficial and should be done by more sophisticated techniques as well.

1 Introduction

Imbalanced data sets are inductive learning domains in which one class is represented by a greater number of examples than the other. [1] Several methods have previously been proposed to deal with this problem including stratification (re-sampling or down-sizing approaches), cost-based learning, and one-sided learning. In this paper, we will only focus on stratification methods, though the close relationship between cost-based and stratification based learning makes the observations made in this paper applicable to cost-based learning as well.

Although stratification approaches have previously been shown to increase classification accuracy [[Kubat and Matwin1997], [Ling and Li1998]], none of these studies took into consideration the fact that both *between-class* and *within-class* imbalances may occur. In the context of this study, a between-class imbalance corresponds to the case where the number of examples representing the positive class differs from the number of examples representing the negative class;

* This research was supported by an NSERC Research Grant.

[1] Throughout this paper, we focus on concept-learning problems in which one class represents the concept while the other represents counter-examples of the concept.

and a within-class imbalance corresponds to the case where a class is composed of a number of different subclusters and these subclusters do not contain the same number of examples. The within-class imbalance problem along with the between-class imbalance problem are instances of the general problem known as the *problem of small disjuncts* [Holte *et al.*1989] which can be stated as follows: Since classification methods are typically biased towards classifying large disjuncts (disjuncts that cover a large number of examples) accurately, they have a tendency to overfit and misclassify the examples represented by small disjuncts.

The purpose of this paper is to show that the within-class imbalance problem and the between-class imbalance problem both contribute to increasing the misclassification rate of multi-layer perceptrons. More specifically, the study distinguishes between different types of imbalances and observes their effects on classification accuracy with respect to perfectly balanced situations or rebalanced ones in artificial domains. It then derives an optimal re-balancing strategy which it tests on a real-world domain.

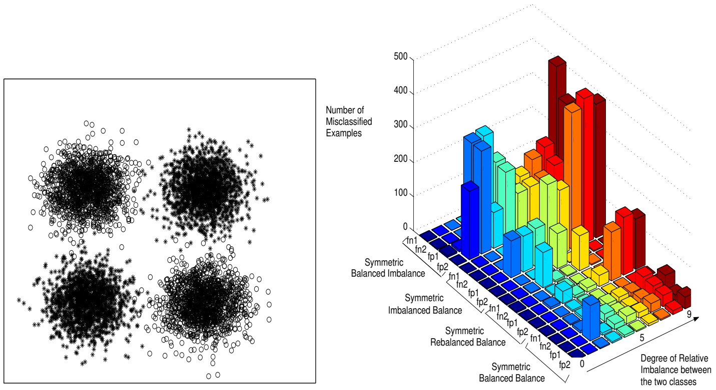
2 Experiments on Artificial Domains

This section presents a systematic study of the generalized imbalance problem in two cases. The first case, the symmetrical case, involves data sets that have as many subclusters in each class. The second case, the asymmetrical case, involves data sets that have more subclusters in one class than in the other.

2.1 The Symmetric Case

In order to study the effect of between-class and within-class imbalances as well as to choose an appropriate solution to these problems in the case of a symmetrical domain, we generated a series of variations of the X-OR problem in which the data distribution differed from one experiment to the other. We then tested the relative accuracy performance of a standard Multi-Layer Perceptron with fixed parameters. These experiments gave a sense of which tasks are more or less difficult to learn by this standard classifier.

Task The X-OR domain used in this experiment is depicted in Figure 1(a). Each class is composed of two subclusters located at the bottom left and top right corner in the case of the positive class (positive instances are represented by '$\star$') and at the top left and bottom right corners in the case of the negative class (negative instances are represented by 'o'). The subclusters are non-overlapping and, in the original domain, each subcluster contains 1,500 training examples. The testing set is distributed in the same way, except for the size of each subcluster which is of 500 examples. Unlike for the training set, the size of the testing set remains fixed for all the experiments. This means that even if the training set contains less data in one subcluster than in the other, we consider the small subcluster to be as important to classify accurately as the larger one. In other words, the cost of misclassifying the small subcluster is considered to be as high as the cost of misclassifying the larger one.



(a) Test Set **(b) Results**

Fig. 1. Experiment on a Symmetric Artificial Domain

Experiments Starting from the original domain, four series of experiments were conducted which changed the between-class balance or the within-class balance of the negative class by modifying the size of the negative subclusters either at the same rate or at a different one while the size of the positive subclusters was either kept fixed or modified simultaneously. These experiments were named: 1) Symmetric Balanced Balance (SBB), 2) Symmetric Balanced Imbalance (SBI), 3) Symmetric Imbalanced Balance (SIB), and 4) Symmetric Rebalanced Balance (SRB), where the first term indicates that there are as many subclusters in the positive and negative class, the second one represents the status of the *within-class* cluster relation and the third one represents the status of the *between-class* cluster relation.

In other words, SBB corresponds to the experiment in which the four subclusters (positive or negative) are of the same size; SBI corresponds to the case where although there are as many examples in the two positive subclusters and the two negative subclusters respectively, there are overall more positive examples than negative ones; SIB corresponds to the case where the size of the overall positive set equals that of the negative one, but although the two positive subclusters are of the same size, the two negative ones are of different sizes; finally, SRB corresponds to the case where the SIB data set has been re-balanced by resampling each subcluster (positive and negative) to make it match the size of the largest subcluster present in the training set (this largest subcluster is necessarily one of the negative subclusters).

Within each experiment set, 10 different degrees of between-class or within-class imbalance were considered, following an exponential rate of size decrease. More specifically, the imbalance was created by decreasing the size of the subcluster(s) targetted by the particular approach at hand at a rate of $\frac{original\_subcluster\_size}{2^i}$ with $i = 0..9$. For example, when $i = 5$, the SBB set is composed of two positive and two negative subclusters of size $ceiling(\frac{1{,}500}{2^5}) = 47$; the SBI set contains two positive subclusters of size $1,500$ each and two negative subclusters of size $ceiling(\frac{1,500}{2^5}) = 47$, each; The SIB set contains two positive subclusters of size $1,500$ each one negative subcluster of size $ceiling(\frac{1,500}{2^5}) = 47$ and one negative cluster of size $3,000 - 47 = 2,953$.

As mentioned previously, the size of the parameters of the neural networks used for these experiments were kept fixed since we are not interested in whether a neural network can solve the X-OR problem (which we know is always possible given sufficient ressources), but rather in which tasks cause it more or less difficulty. The parameters we chose—since they were adequate for the original domain—were of 4 hidden units and 200 training epochs. The training procedure used was Matlab's default optimization algorithm: the Levenberg-Marquardt procedure. The network used sigmoidal functions in both its hidden and output layer. After being trained, the networks were tested on the testing set. The experiments were all repeated 5 times and the results of each trial averaged.

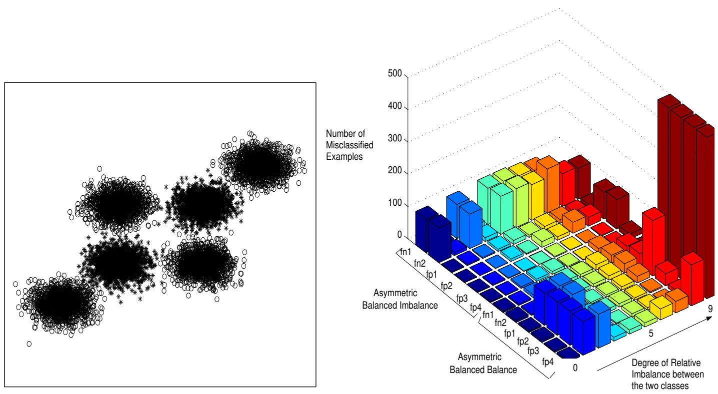
Results The results of all the experiments in this section are reported in Figure 1(b). In this figure, the results are reported in terms of four quantities: number of false negatives for positive subcluster 1 (fn1), number of false negatives for positive subcluster 2 (fn2), number of false positives for positive subcluster 1 (fp1), number of false positives for positive subcluster 2 (fp2). The results are also reported for each level of imbalance starting at level 0 (no imbalance) reported in the front row to level 9 (largest imbalance) reported in the back row. The results are reported in terms of number of misclassified examples in each subcluster. In each case, the maximum possible number of misclassified examples is 500, the size of each testing subcluster. The results were reported in the following order: SBI, SIB, SRBand SBB. This order corresponds to the least accurate to the most accurate strategy and was chosen to allow for the best perspective on a single graph.

In more detail, the results indicate that the results on the SBI strategy are the least accurate because it causes both negative subclusters a high degree of misclassification. The positive class, on the other hand, is generally well classified. This can be explained by the fact that, in this experiment, both negative subclusters have smaller sizes than the positive ones.The degree of imbalance observed between the two classes, however, does not appear to be an important factor in the misclassification rates observed (remember, however, that the imbalance level grows exponentially which means that the absolute difference between two consecutive levels is greater at the begining than it is at the end). The results on the SIB domain are a little more accurate than those on the SBI domain since this time, only one of the two negative subclusters—the one represented by the smallest number of examples—suffers from some misclassification error. The third set of results, the set of results obtained when using a rebalancing sampling strategy so as to rectify the SIB problem is shown to be effective although, as the degree of within-class imbalance in the negative class increases, the re-sampling strategy is shown to loose some accuracy in the originally small but re-sampled subcluster, though this loss is much smaller than the loss incurred when no re-sampling is used.[2] This result can be explained by the fact that re-sampling the same data over and over is not as useful as having individual data points belonging to the same distribution (as shown by the fourth set of results on the SBB domain). Indeed, a re-sampling strategy may rectify the imbalance problem but it does not introduce all the information necessary to prevent the inductive learner to overfit the training examples. On the contrary, it probably encourages some amount of overfitting in the originally small negative subcluster.

These results, thus, suggest that balancing a domain with respect to the between-class problem is not sufficient since, if within-class imbalances are present, the classifier will not be very accurate.

2.2 The Asymmetric Case

Although the experiments of the previous section gave us an idea of the effect of between-class and within class imbalances on the classification accuracy of a multi-layer perceptron, they only considered the case where there are as many subclusters in the positive and the negative class. The question asked in this section is how to handle the case of within-class and between-class imbalance when the number of subclusters in each class is different. In particular, we are interested in finding out whether, in such cases, better classification can be expected when all the subclusters (independently of their class) are of the same size and, thus, the two classes are represented by different numbers of examples or when all the subclusters within the same class are of the same size, but altogether, the class sizes are the same.



(a) Test Set **(b) Results**

Fig. 2. Experiment on an Asymmetric Artificial Domain

---

[2] In a couple of isolated cases, one of the positive subclusters also seems to be affected, but the significance of this observation is unclear.

Task In order to answer this question, we generated a new test set closely related to the X-OR problem of the previous section. As a matter of fact, the test set represents the X-OR problem plus two new negative subclusters, both located on the same diagonal as the two positive subclusters, but just outside the square formed by linking the four subclusters of the X-OR problem. This new problem is depicted in Figure 2(a) with '$\star$' representing positive examples and 'o"s representing negative ones.

Once again, each subcluster of the training set is originally represented by 1,500 examples, independently of its class. Like in the previous section, the testing set is distributed in the same way, except for the size of each subcluster which is of 500 examples.

Experiments In this section, two series of experiments were conducted that only changed the between-class balance. The within-class balance was untouched (since its effect was already tested in the previous section) although the size of the subclusters belonging to each class, respectively, was allowed to differ. These experiments were named: Asymetric Balanced Balance (ABB) and Asymetric Balanced Imbalance (ABI), where the first term indicates that there are different numbers of subclusters per class, the second term represents the status of the within-class cluster relation and the third one represents the status of the between-class cluster relation. In other words, ABB corresponds to the experiment in which the two positive subclusters are of the same size and the four negative subclusters are of the same size, but the negative subclusters are half the size of the positive ones so that, altogether, the two classes have the same number of training instances; ABI corresponds to the case where all the positive and negative subclusters are of the same size and, thus, the two classes have different numbers of training instances.

Again within each experiment set, 10 different degrees of between-class or within-class imbalance were considered, following an exponential rate of size decrease. As before, the imbalance was created by decreasing the size of the subcluster(s) targetted by the particular approach at hand at a rate of $\frac{original\_subcluster\_size}{2^i}$ with $i = 0..9$. For example, when $i = 5$, the ABB Set has two positive subclusters of size $ceiling(\frac{1,500}{2^5}) = 47$ and four negative subclusters of size $floor(\frac{1,500}{2\times 2^5}) = 23$, with no between-class imbalance; Similarly, the ABI set is composed of two positive and four negative subclusters of size $ceiling(\frac{1,500}{2^5}) = 47$, each, thus creating a between-class imbalance of 94 examples.

Like previously and for the same reasons, the size of the parameters of the neural networks used for these experiments were kept fixed, though, due to the increased difficulty of the test domain, we increased the number of hidden units to 8. All the other parameters remained the same. These parameters were adequate for the original domain of Figure 2(a). After being trained, the networks were again tested on a testing set. The experiments were all repeated 5 times and the results of each trial averaged.

Results The results of all the experiments in this section are reported in Figure 2(b). In this figure, the results are reported in terms of six quantities: number of false negatives for positive subcluster 1 (fn1) and positive subcluster 2 (fn2), number of false positives for positive subcluster 1 (fp1), positive subcluster 2 (fp2), positive subcluster 3 (fp3) and positive subcluster 4 (fp4). The results are also reported for each level of imbalance starting at level 0 (no imbalance) reported in the front row to level 9 (largest imbalance) reported in the back row. The results are reported in terms of number of misclassified examples in each subcluster. In each case, the maximum number of misclassified examples is 500, the size of each testing subcluster. The results were reported in the following order: ABI and ABB since this order corresponds to the least accurate to the most accurate strategy and was, again, chosen to allow for the best perspective on a single graph.

In more detail, the results indicate that the results on the ABI domain are less accurate than those obtained on the ABB domain because they suggest that the two positive subclusters are prone to misclassification errors whereas they are generally not in the case of the ABB domain. This can be explained by the fact that in the ABI domain, the size of the positive class is half that of the negative one. In most cases, it thus, appears that it is generally better to be in a situation where the two classes are balanced (with no between- nor within-class imbalance), even if that means that the size of the subclusters of the class composed of the greater number of subcluster is smaller than that of its counterparts in the other class.[3]

3 An Optimal Re-Balancing Strategy

Based on the results obtained in the symmetrical and asymmetrical domains of section 2, we can now hypothesize on an optimal re-balancing strategy for the cases where both within-class and between-class imbalances are present in a domain. The benefits of this strategy is then tested in a grouped-letter recognition problem.

3.1 Formulation of the Strategy

Let L be a concept-learning problem with two classes A and B each composed of N_A and N_B subclusters respectively. Class A is composed of subclusters a_i of size n_a^i, respectively (with $i \in \{1, 2, ...N_A\}$) and class B is composed of subclusters b_j of size n_b^j, respectively (with $j \in \{1, 2, ...N_B\}$). Let $maxcluster_A = max(n_a^1, n_a^2, ...n_a^{N_A})$ and $maxcluster_B = max(n_b^1, n_b^2, ..., n_b^{N_B})$. Let, further, $maxclasssize = max(maxcluster_A \times N_A, maxcluster_B \times N_B)$ and $maxclass$ be the class corresponding to maxclasssize (i.e., class A in case $maxcluster_A \times N_A \geq maxcluster_B \times N_B$ and class B, otherwise. Let $altclass$ be the class *not* corresponding to maxclass (i.e., altclass=A if maxclass=B and vice-versa). According to the results of Section 2, L will be learned more accurately by multi-layer perceptrons if the training set is transformed as follows:

[3] Note, however, that the graph shows several cases where the ABB strategy causes misclassification to the negative class. These cases, however, are rarer than the cases where the positive class is negatively affected in the ABI situation.

> Each subcluster of class *maxclass* is re-sampled until it reaches size $maxcluster_{maxclass}$. At this point, the overall size of *maxclass* will be *maxclasssize* and there will be no within-class imbalance in class *maxclass*. In order to prevent a between-class imbalance as well as within-class imbalances in altclass, each subcluster of altclass is re-sampled until it reaches size $maxclasssize/N_{altclass}$.

This procedure will guarantee no between-class imbalance and no within-class imbalance although, like in the asymmetrical case above, the size of A's subclusters may differ from that of B's.

3.2 Testing the Stategy

In order to determine whether the strategy just derived is practical, we tested our approach on a real world-domain. In particuliar, we tested the multi-layer perceptron on the letter recognition problem consisting of discriminating between a certain number of vowels and consonnants.

More specifically, we used the letter recognition data set available from the UC Irvine Repository. However, we defined a subtask which consisted of recognizing vowels from consonnants and, in order to make our task more tractable, we reduced the vowel set to the letters a, e, and u and the consonnant set to the letters m, s, t and w. In addition, rather than assuming the same number of examples per letter in the training set, we constructed the training data in a way that reflects the letter frequency in English text.[4] The testing set was always fixed and consisted of 250 data points per letter. The reason why the distribution of the testing set differs from that of the training set is because the cost of misclassifying a letter is independent of its frequency of occurence. For example, confusing "war" for "car" is as detrimental as confusing "pet" for "pat" even though "w" is much more infrequently used than "e".

In the experiments we conducted on these data, the performance of the multi-layer perceptron was compared to its performance in three different training-set situations: Imbalance, Naive Re-Balance, Informed Re-Balance, and Uninformed Re-Balance. The purpose of the Imbalance, Naive Re-Balance, and Informed-Rebalance experiments is simply to verify whether our optimal-resampling strategy also helps on a real-world domain. The Imbalance experiment consisted of running the multi-layer perceptron on the letter recognition domain without practicing any type of re-balancing. The Naive Re-Balance strategy consisted of re-sampling from the negative class (containing 750 training data), ignoring its internal distribution, until its size reached that of the positive class (containing 942 training data). The Informed-Rebalance experiment assumes that the subcluster division of each class is fully known and the data sets are rebalanced

[4] In particular, we relied on the following frequencies: a: .0856, e: .1304, u: .0249, m: .0249, s: .0607, t: .1045, w: .0017 and consequently built a training set containing the following corresponding number of training examples per letter: a= 335 points, e= 510 points, u= 97 points, m= 97 points, s= 237 points, t= 409 points, w= 7 points. These letters were chosen because of their interesting differing frequencies.

according to our optimal re-balancing strategy.[5] In the Uninformed Re-Balance strategy, no prior knowledge about the data is assumed. In this case, the k-means unsupervised learning algorithm is used to determine the inner-distribution of each class, followed by our optimal re-balancing method.[6]

In all three experiments, the neural networks were optimized using Matlab's default optimization algorithm: Levenberg-Marquardt, and the network used sigmoidal units in both their hidden and output layers. In each experiment, four different networks were ran five times each with 2, 4, 8 and 16 hidden units. The results were averaged over the five runs and the best results were reported.

The results obtained on these experiments are reported in Figure 3. In particular, Figure 3 is composed of 8 clusters of 4 columns each. Within each cluster, each column corresponds to the performance of each of our 4 strategies. The leftmost column of each cluster represents the results obtained on the Imbalance experiment; next are the results obtained with the Naive Re-Balance Strategy; this is followed with the Informed Re-Balance strategy; and the last column was obtained using the Uninformed Re-Balance strategy. The rightmost cluster represents the cumulative results obtained on the overall testing set, while each of the preceeding cluster represents the results on particular subclusters.

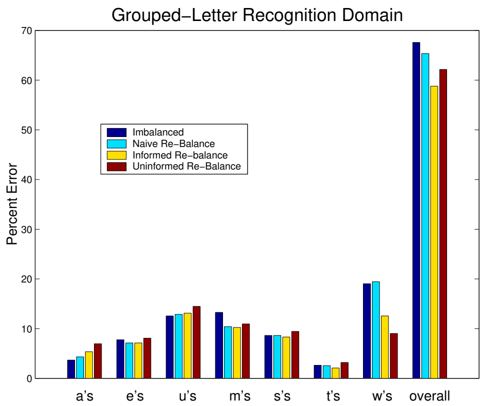


Fig. 3. Results

The results shown in Figure 3 suggest that, overall, as can be expected the Imbalance experiment shows a slightly larger error rate than the Naive Re-Balance experiment. The Informed Re-balance experiment shows a lower error rate than the Naive Re-Balance experiment and the Uninformed Re-Balance ex-

[5] Although this situation is unrealistic, this case is considered since it represents a lower bound on the results that can be obtained using our re-sampling strategy.

[6] An estimate of the number of clusters per class was determined prior to running the k-means algorithm. Our estimation procedure, however, is sub-optimal and will be refined in future work, using cross-validation experiments.

periment falls in-between the two results, helping to improve on the imbalanced results, but not performing quite as well as in the case where the composition of each class is fully known. In more detail, the results show that the Informed and Uninformed Re-Balance strategies are particularly effective in the case of a very small subcluster (w), but that the Uninformed strategy causes a slight decrease in accuracy in the other subclusters. This is usually not the case for the Informed strategy and we hope that improving our clustering approach in the Uninformed strategy will help in reducing this problem.

4 Conclusion and Future Work

It is not uncommon for classification problems to suffer from the problem of class imbalances. In particular, these imbalances can come in two forms: *between-class* and *within-class* imbalances. Though both problems are well-known and have been previously considered in the machine learning literature, they have not been previously considered simultaneously. The purpose of this paper was to derive a re-sampling strategy that considers both imbalances simultaneously and demonstrate that even a very simple method for dealing with the problem can be helpful in the case of drastically imbalanced subclusters.

There are many extensions of this work. First, the experiments on artificial domains were conducted on *unnaturally* imbalanced data sets. It would be useful to repeat these experiments on *naturally* imbalanced ones. Second, rather than testing our re-balancing strategy on a balanced domain, it would be more representative to test it on a range of class distributions using ROC Hulls or Cost Curves [[Provost and Fawcett2001], [Drummond and Holte2000]]. Third, it would be interesting to test our strategy on other classifiers and other domains.[7] Finally, we should try to adapt our strategy to cost-based algorithms that usually perform more accurately on imbalanced data sets than stratification methods.

References

[Drummond and Holte2000] Chris Drummond and Robert Holte. Explicitely representing expected cost: An alternative to roc representation. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207, 2000.

[Holte *et al.*1989] R. C. Holte, Acker L. E., and B. W. Porter. Concept learning and the problem of small disjuncts. In *IJCAI-89*, 1989.

[Kubat and Matwin1997] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced data sets: One-sided sampling. In *ICML-97*, 1997.

[Ling and Li1998] Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *KDD-98*, 1998.

[Nickerson *et al.*2001] Adam Nickerson, Nathalie Japkowicz, and Evangelos Milios. Using unsupervised learning to guide resampling in imbalanced data sets. In *AISTATS-01 (to appear)*, 2001.

[Provost and Fawcett2001] Foster Provost and Tom E. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

[7] Some preliminary work in this area can be found in [Nickerson *et al.*2001].