Learning with Skewed Class Distributions

Maria Carolina Monard and Gustavo E.A.P.A. Batista

Laboratory of Computational Intelligence - LABIC
Department of Computer Science and Statistics - SCE
Institute of Mathematics and Computer Science - ICMC
University of São Paulo - Campus of São Carlos
P. O. Box 668, 13560-970, São Carlos, SP, Brazil
Phone: +55-16-273-9692. FAX: +55-16-273-9751.
{mcmonard, gbatista}@icmc.usp.br

Abstract. Several aspects may influence the performance achieved by a classifier created by a Machine Learning system. One of these aspects is related to the difference between the numbers of examples belonging to each class. When this difference is large, the learning system may have difficulties to learn the concept related to the minority class. In this work[1], we discuss several issues related to learning with skewed class distributions, such as the relationship between cost-sensitive learning and class distributions, and the limitations of accuracy and error rate to measure the performance of classifiers. Also, we survey some methods proposed by the Machine Learning community to solve the problem of learning with imbalanced data sets, and discuss some limitations of these methods.

Keywords: Class imbalance, cost-sensitive learning, machine learning.

1 Introduction

Supervised learning is the process of automatically creating a classification model from a set of examples, called the *training set*, which belong to a set of classes. Once a model is created, it can be used to automatically predict the class of other unclassified examples.

In other words, in supervised learning, a set of n training examples is given to an inducer. Each example $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \ldots \times F_m$ where F_j is the domain of the jth feature. Training examples are tuples $(\mathbf{X}, Y)$ where Y is the label, output or class. The Y values are typically drawn from a discrete set of classes $\{1, \ldots, K\}$ in the case of *classification*. Given a set of training examples, the learning algorithm (*inducer*) outputs a *classifier* such that, given a new example, it accurately predicts the label Y.

[1] This work was previously published in LAPTEC-2002, Frontiers in Artificial Intelligence and its Applications, IOS Press.

In this work, we reserve our discussion to concept-learning[2], so Y can assume one of two mutually exclusive values. We use the general labels **positive** and **negative** to discriminate between the two class values.

For a number of application domains, a huge disproportion in the number of cases belonging to each class is common. For instance, in detection of fraud in telephone calls [7] and credit card transactions [15], the number of legitimate transactions is much higher than the number of fraudulent transactions. In insurance risk modelling [12], only a small percentage of the policyholders file one or more claims in any given time period. Also, in direct marketing [11], it is common to have a small response rate (about 1%) for most marketing campaigns. Other examples of domains with intrinsic imbalance can be found in the literature. Thus, learning with skewed class distributions is an important issue in supervised learning.

Many traditional learning systems are not prepared to induce a classifier that accurately classifies the minority class under such situation. Frequently, the classifier has a good classification accuracy for the majority class, but its accuracy for the minority class is unacceptable. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. Unfortunately, that is the norm for most applications with imbalanced data sets, since these applications aim to profile a small set of valuable entities that are spread in a large group of "uninteresting" entities.

In this work we discuss some of the most frequently used methods that aim to solve the problem of learning with imbalanced data sets. These methods can be divided into three groups:

1. **Assign misclassification costs**. In a general way, misclassify examples of the minority class is more costly than misclassify examples of the majority class. The use of cost-sensitive learning systems might aid to solve the problem of learning from imbalanced data sets;
2. **Under-sampling**. One very direct way to solve the problem of learning from imbalanced data sets is to artificially balance the class distributions. Under-sampling aim to balance a data set by eliminating examples of the majority class;
3. **Over-sampling**. This method is similar to under-sampling. But it aims to achieve a more balanced class distributions by replicating examples of the minority class.

This work is organised as follows: Section 2 discusses why accuracy and error rate are inadequate metrics to measure the performance of learning systems when data have asymmetric misclassification costs and/or class imbalance; Section 3 explains the relationship between imbalanced class distributions and cost-sensitive learning; Section 4 discusses which class distributions are best for learning; Section 5 surveys some methods proposed by the Machine Learning community to balance the class distributions; Section 6 presents a brief discus-

[2] However some of the methods discussed here can be applied to multi-class problems.

sion about some evidences that balancing a class distributions has little effect in the final classifier; finally, Section 7 shows the conclusions of this work.

2 Why Accuracy and Error Rate are Inadequate Performance Measures for Imbalanced Data Sets

Different types of errors and hits performed by a classifier can be summarised in a *confusion matrix*. Table 1 illustrates a confusion matrix for a two class problem, with classes labelled **positive** and **negative**:

	Positive Prediction	*Negative Prediction*
Positive Class	True Positive (a)	False Negative (b)
Negative Class	False Positive (c)	True Negative (d)

Table 1. Different types of errors and hits for a two classes problem.

From such matrix it is possible to extract a number of metrics to measure the performance of learning systems, such as error rate $E = \frac{(c+b)}{(a+b+c+d)}$ and accuracy $Acc = \frac{(a+d)}{(a+b+c+d)} = 1 - E$.

The error rate (E) and the accuracy (Acc) are widely used metrics for measuring the performance of learning systems. However, when the prior probabilities of the classes are very different, such metrics might be misleading. For instance, it is straightforward to create a classifier having 99% accuracy (or 1% error rate) if the data set has a majority class with 99% of the total number of cases, by simply labelling every new case as belonging to the majority class.

Other fact against the use of accuracy (or error rate) is that these metrics consider different classification errors as equally important. For instance, a sick patience diagnosed as healthy might be a fatal error while a healthy patience diagnosed as sick is considered a much less serious error since this mistake can be corrected in future exams. On domains where misclassification cost is relevant, a cost matrix could be used. A cost matrix defines the misclassification cost, i.e., a penalty for making a mistake for each different type of error. In this case, the goal of the classifier is to minimize classification cost instead of error rate. Section 3 discusses more about the relationship between cost-sensitive learning and imbalanced data sets.

It would be more interesting if we could use a performance metric that disassociates the errors (or hits) occurred in each class. From Table 1 it is possible to derive four performance metrics that directly measure the classification performance on the positive and negative classes independently, they are:

– **False negative rate**: $FN = \frac{b}{a+b}$ is the percentage of positive cases misclassified as belonging to the negative class;

- **False positive rate**: $FP = \frac{c}{c+d}$ is the percentage of negative cases misclassified as belonging to the positive class;
- **True negative rate**: $TN = \frac{d}{c+d} = 1 - FP$ is the percentage of negative cases correctly classified as belonging to the negative class;
- **True positive rate**: $TP = \frac{a}{a+b} = 1 - FN$ is the percentage of positive cases correctly classified as belonging to the positive class;

These four class performance measures have the advantage of being independent of class costs and prior probabilities. It is obvious that the main objective of a classifier is to minimize the false positive and negative rates or, similarly, to maximize the true negative and positive rates. Unfortunately, for most "real world" applications, there is a tradeoff between FN and FP and, similarly, between TN and TP. The *ROC*[3] *graphs* [13] can be used to analyse the relationship between FN and FP (or TN and TP) for a classifier.

Consider that the minority class, whose performance will be analysed, is the positive class. On a ROC graph, TP $(1 - FN)$ is plotted on the Y axis and FP is plotted on the X axis. Some classifiers have parameter for which different settings produce different ROC points. For instance, a classifier that produces probabilities of an example being in each class, such as Naive Bayes classifier, can have a threshold parameter biasing the final class selection[4]. Plotting all the ROC points that can be produced by varying these parameters produces a ROC curve for the classifier. Typically this is a discrete set of points, including (0,0) and (1,1), which are connected by line segments. Figure 1 illustrates a ROC graph of 3 classifiers: A, B and C. Several points on a ROC graph should be noted. The lower left point (0,0) represents a strategy that classifies every example as belonging to the negative class. The upper right point represents a strategy that classifies every example as belonging to the positive class. The point (0,1) represents the perfect classification, and the line $x = y$ represents the strategy of random guessing the class.

From a ROC graph is possible to calculate an overall measure of quality, the *under the ROC curve area (AUC)*. The AUC is the fraction of the total area that falls under the ROC curve. This measure is equivalent to several other statistical measures for evaluating classification and ranking models [8]. The AUC effectively factors in the performance of the classifier over all costs and distributions.

[3] ROC is an acronym for *Receiver Operating Characteristic*, a term used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel.

[4] In a similar way, other learning system can be adapted to produce such posterior probabilities estimates. In decision trees, the class distributions at the leaves can be use as an estimate. Rule learning systems can make similar estimates. Neural networks produce continuous outputs that can be mapped to probability estimates.

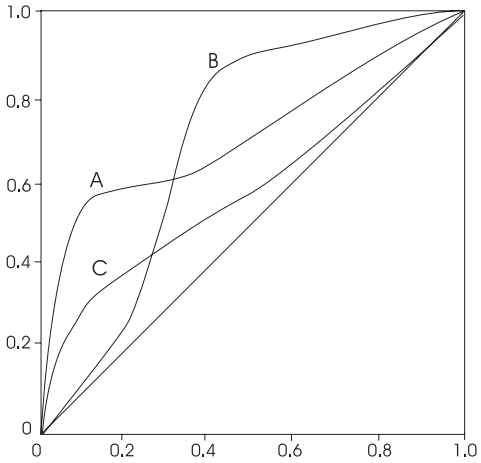

Fig. 1. A ROC graph for 3 classifiers.

3 Imbalance and Cost-Sensitive Learning

A classifier induced from an imbalanced data set has, typically, a low error rate for the majority class and an unacceptable error rate for the minority class. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. In this situation, it is important to accurately classify the minority class in order to reduce the overall cost.

A cost-sensitive learning system can be used in applications where the misclassification costs are known. Cost-sensitive learning systems attempt to reduce the cost of misclassified examples, instead of classification error.

Unfortunately, some learning systems are not able to integrate cost information into the learning process. However, there is a simple and general method to make any learning system cost-sensitive for a concept-learning problem [2]. The idea is to change the class distributions in the training set towards the most costly class. Suppose that the positive class is five times more costly than the negative class. If the number of positive examples are artificially increased by a factor of five, then the learning system, aiming to reduce the number of classification errors, will come up with a classifier that is skewed towards the avoidance of error in the positive class, since any such errors are penalised 5 times more. In [6] is provided a theorem that shows how to change the proportion of positive and negative examples in order to make optimal cost-sensitive classifications for a concept-learning problem. Moreover, a general method to make learning system cost-sensitive is presented in [4]. This method has the advantage of being applicable to multi-class problems.

Thus, class imbalance and cost-sensitive learning are related to each other. One way to learn with an imbalanced data set is to train a cost-sensitive learning system with the misclassification cost of the minority class greater than the ma-

jority class. One way to make a learning system cost-sensitive is to intentionally imbalance the training set.

Most methods that deal with imbalanced data sets aim to improve the learning of the minority concept by balancing the data set. In this way, the minority class becomes more costly, and we can expect it will be better classified. From this point, two different situations can be identified: first, there are plenty of data, and the problem can be understood as "which proportion of positive/negative examples is the best for learning?"; second, data are scarce, and there is another additional problem: "how discard negative examples/duplicate positive examples without introducing much bias in the learning process?". In the following sections these two questions are discussed in more detail.

4 Which Proportion of Positive/Negative Examples is the Best for Learning?

The problem of determining which proportion of positive/negative examples is the best for learning goes beyond the problem of learning from imbalanced data sets. Much of the Machine Learning community has assumed that the naturally occurring class distributions are the best for learning. However, because of leaning from the naturally occurring class distributions yield bad results for highly imbalanced data sets, this assumption started to be studied in deep.

In [17] is made a throughout study about the proportion of positive/negative examples in learning. This study analyses the scenario that there are plenty of data, however because of computational restriction, the training set should be limited to n examples. In this scenario, which class distributions should be the best for training?

Using the area under the ROC curve (AUC) as performance measure, [17] shows that the optimal distribution generally contains between 50% and 90% of minority class examples. Also, allocating 50% of the training examples to the minority class, while it will not always yield optimal results, will generally lead to results which are no worse than, and often superior to, those which use the natural class distributions.

5 How to Discard Negative Examples/Duplicate Positive Examples Without Introducing Much Bias in the Learning Process?

As already mentioned, one of the most direct ways for dealing with class imbalance is to alter the class distributions toward a more balanced distribution. There are two basic methods for balancing the class distributions:

1. **Under-sampling**: these methods aim to balance the data set by eliminating examples of the majority class, and;
2. **Over-sampling**: these methods replicate examples of the minority class in order to achieve a more balanced distribution.

Both, under-sampling and over-sampling, have known drawbacks. Under-sampling can throw away potentially useful data, and over-sampling can increase the likelihood of occurring overfitting, since most of over-sampling methods make exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually, cover one replicated example.

Some recent research has focused in overcoming the drawbacks of both under-sampling and over-sampling. In [3] under-sampling and over-sampling methods are combined, and, instead of over-sampling by replicating minority class examples, new minority class examples are formed by interpolating between several minority class examples that lie together. Thus, they avoid the overfitting problem and cause the decision boundaries for the minority class to spread further into the majority class space.

In [10, 1] an under-sampling technique is analysed in order to minimize the amount of potentially useful data discarded. The majority class examples are classified as "safe", "borderline" and "noise" examples. Borderline and noisy cases are detected using *Tomek links* [16], and are removed from the data set. Only safe majority class examples and all minority class examples are used for training the learning system. A Tomek link can be defined as follows: given two examples x and y belonging to different classes, and be $d(x, y)$ the distance between x and y. A (x, y) pair is called a Tomek link if there is not a case z, such that $d(x, z) < d(x, y)$ or $d(y, z) < d(y, x)$. Figure 2 illustrates the process of cleaning a data set with Tomek links.

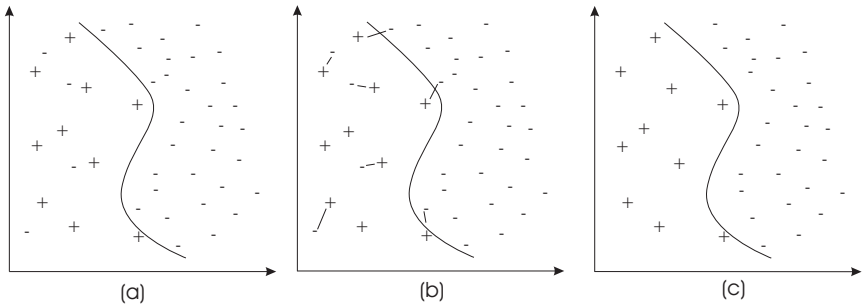

Fig. 2. Applying Tomek links to a data set. Original data set *(a)*, Tomek links identified *(b)*, and Tomek links removed *(c)*.

Other methods for reducing the training set size based on k-nearest neighbor, like Tomek links, are surveyed in [18].

6 Discussion

Much of research done to solve the problem of learning from imbalanced data sets is based on balancing the class distributions. However, recent research has shown

that many learning systems are insensitive to class distributions. Drummond and Holte [5] showed that there are decision tree splitting criteria that are relatively insensitive to a data set class distributions. Elkan [6] makes similar statements for Naive Bayes and decision tree learning systems. If a learning system is insensitive to the class distributions, then changing the class distributions — or balancing a data set — might have little effect in the induced classifier.

On the other hand, under- and over-sampling have been empirically analysed in several domains, with good results. In [9] several approaches for dealing with imbalanced data sets are compared, and it concludes that under- and over-sampling are very effective methods for dealing with imbalanced data sets.

Moreover, Drummond and Holte [5] stated that under- and over-sampling should be reconsidered in terms of how they affect pruning and leaf labelling of decision trees. However, on several experiments performed in [14], classifiers generated from balanced distributions obtained results that were, frequently, better than those obtained from the naturally occurring distributions. These experiments were conducted with no pruning, and adjusting the leaf labelling to account the changes made in class distributions.

7 Conclusion

Learning from imbalanced data sets is an important issue in Machine Learning. A direct method to solve the imbalance problem is to artificially balance the class distributions. This balance can be obtained by under-sampling the majority class, over-sampling minority class, or both. There are several works in the literature that confirm the efficiency of these methods in practice. However, there is some evidence that re-balancing the class distributions artificially does not have much effect on the performance of the induced classifier, since some learning systems are not sensitive to differences in class distributions. It seems that we still need a clearer understanding of how class distributions affect each phase of the learning process. For instance, in decision trees, how class distributions affect the tree construction, pruning and leaf labelling. A deeper understanding of the basics will permit us to design better methods for dealing with the problem of learning with skewed class distributions.

Acknowledgements. This research is partially supported by Brazilian Research Councils CAPES and FINEP.

References

1. G. E. A. P. A. Batista, A. Carvalho, and M. C. Monard. Applying One-sided Selection to Unbalanced Datasets. In O. Cairo, L. E. Sucar, and F. J. Cantu, editors, *Proceedings of the Mexican International Conference on Artificial Intelligence – MICAI 2000*, pages 315–325. Springer-Verlag, April 2000. Best Paper Award Winner.
2. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth & Books, Pacific Grove, CA, 1984.

3. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Jounal of Artificial Intelligence Research*, 16:321–357, 2002.
4. Pedro Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In *Knowledge Discovery and Data Mining*, pages 155–164, 1999.
5. Chris Drummond and Robert C. Holte. Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria. In *Proceedings of the 17th International Conference on Machine Learning (ICML'2000)*, pages 239–246, 2000.
6. Charles Elkan. The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.
7. Tom Fawcett and Foster J. Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
8. David J. Hand. *Construction and Assessment of Classification Rules*. John Wiley and Sons, 1997.
9. Nathalie Japkowicz. Learning from Imbalanced Data Sets: a Comparison of Various Strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, 2000. AAAI Press.
10. M. Kubat and S. Matwin. Addressing the Course of Imbalanced Training Sets: One-Sided Selection. In *Proceedings of 14th International Conference in Machine Learning*, pages 179–186, San Francisco, CA, 1997. Morgan Kaufmann.
11. Charles X. Ling and Chenghui Li. Data Mining for Direct Mining: Problems and Solutions. In *Proceedings of The Forth International Conference on Knownledge Discovery and Data Mining*, pages 73–79, 1998.
12. Edwin P. D. Pednault, Barry K. Rosen, and Chidanand Apte. Handling Imbalanced Data Sets in Insurance Risk Modeling. Technical Report RC-21731, IBM Research Report, March 2000.
13. Foster J. Provost and Tom Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Knowledge Discovery and Data Mining*, pages 43–48, 1997.
14. Foster J. Provost and Tom Fawcett. Robust Classification for Imprecise Environments. *Machine Learning*, 42(3):203–231, 2001.
15. S. J. Stolfo, D. W. Fan, W. Lee, A. L. Prodromidis, and P. K. Chan. Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. In *AAAI-97 Workshop on AI Methods in Fraud and Risk Management*, 1997.
16. I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems Man and Communications*, SMC-6:769–772, 1976.
17. Gary M. Weiss and Foster Provost. The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.
18. D. R. Wilson and T. R. Martinez. Reduction Techniques for Exemplar-Based Learning Algorithms. *Machine Learning*, 38(3):257–286, March 2000.