An Overview of Learning from Imbalanced Datasets

Gonzalo A. Bello Lander

1. Introduction

The class imbalance problem occurs when, in a classification task, there are significantly more examples from one class than from the other. In these cases, classifiers tend to be biased towards the majority class and to ignore the minority one. This problem is present in many real world applications, such as fraud and failure detection [1,2], text classification [3] and medical diagnosis [4,5].

Furthermore, the minority class is usually the class of interest. For example, for medical diagnosis applications, where the main goal is to identify patients with a particular disease, there are usually more negative examples (i.e., healthy patients) than positive examples (i.e., sick patients).

Interest in the class imbalance problem first emerged when data mining and machine learning techniques began to be widely used for practical applications. Researchers quickly realized that many datasets were imbalanced and that this imbalance affected the performance of the classifiers [6].

This led to two workshops on learning from imbalanced datasets, one held at the Association for the Advancement of Artificial Intelligence (AAAI) conference in 2000 and the other one at the International Conference on Machine Learning (ICML) in 2003, as well as a special issue on the subject by the ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations in 2004 [6].

Since then, it has become apparent that the class imbalance problem is ubiquitous in the data mining and machine learning community. In fact, it has been identified as one of the 10 most challenging problems in data mining [7].

In recent years, the number of publications on the class imbalance problem has increased significantly [8] and many different solutions have been proposed.

The purpose of this report is to present an overview of the state of the art in learning from imbalanced datasets. It is worth noting that this report will focus exclusively on imbalanced learning for binary classification tasks, although some of the performance measures and solutions presented can be extended to multi-class classification.

Section 2 introduces some of the performance measures used for imbalanced learning. Section 3 describes several solutions proposed to the class imbalance problem, including sampling methods, ensemble methods and cost-sensitive methods. Some of these solutions are compared and discussed in section 4. Finally, section 5 presents the conclusions and recommendations of the report.

2. Performance Measures for Imbalanced Learning

Accuracy is the most commonly used performance measure in classification tasks.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where TP, TN, FP and FN are the number of true positives (number of positive examples correctly classified), true negatives (number of negative examples correctly classified), false positives (number of positive examples incorrectly classified) and false negatives (number of negative examples incorrectly classified), respectively[1].

The values of TP, TN, FP and FN are usually tabulated in a *confusion matrix*, as in Figure 1.

		True class	
		$\boldsymbol{P}$	$\boldsymbol{N}$
Predicted class	$\boldsymbol{P}$	TP	FP
	$\boldsymbol{N}$	FN	TN

Figure 1. Confusion matrix

[1] This report follows the convention of identifying the positive class as the minority class and the negative class as the majority class.

When the data is imbalanced, using accuracy to evaluate the performance of a classifier can be misleading. For example, given a dataset with 98 examples from the majority class and only 2 examples from the minority class, a classifier that assigns every example to the majority class would yield an accuracy of 98%, even though none of the minority class examples were correctly classified.

It is evident that accuracy is highly sensitive to the class distribution of the data, and thus it is biased towards the majority class [8,9,10].

For this reason, one focus of the research in imbalanced learning has been determining how to effectively measure the performance of a classifier when the data is imbalanced. Some of the measures that have been proposed to this end are introduced in the following sections.

2.1 Precision and Recall

Precision and *recall* are other common performance measures in classification tasks.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Precision measures how many examples classified as positive were indeed positive, while recall measures how many examples belonging to the positive class were correctly classified.

The goal of a classifier is to increase the recall, without decreasing the precision. The *F-measure* metric represents this trade-off between precision and recall, and has been used in some studies as a performance measure for imbalanced learning [11].

$$F - Measure = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision}$$

where β is a coefficient that specifies the relative importance of precision versus recall. This coefficient is usually set to 1 (*F1 Score*) or 2 (*F2 Score*).

2.2 Geometric Mean

Another metric that has been adopted by some researchers as a performance measure for imbalanced learning is the *geometric mean* (G-Mean) [12]. The G-Mean represents the trade-off between the true positive and the true negative rates.

$$G - Mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$

A higher G-Mean indicates a better performance for both the positive and the negative classes.

The F-measure and the G-Mean provide a better measure of the performance of a classifier than accuracy when the data is imbalanced. However, most researchers believe that these metrics are not sufficient for imbalanced learning and recommend using a curve-based analysis instead [8].

2.3 ROC Curves

The Receiver Operating Characteristic curve (*ROC curve*) is a two-dimensional plot of the TP rate (or *sensitivity*) over the FP rate (or *1 – specificity*) that represents the tradeoff between TP and FP [13].

$$TP\ rate = Sensitivity = \frac{TP}{TP + FN}$$

$$FP\ rate = 1 - Specificity = \frac{FP}{FP + TN}$$

Note that the TP rate or sensitivity is equivalent to the recall.

A discrete classifier (i.e., a classifier that outputs a discrete class label), such as a decision tree, produces a single point in ROC space, such as point A in Figure 2 [13].

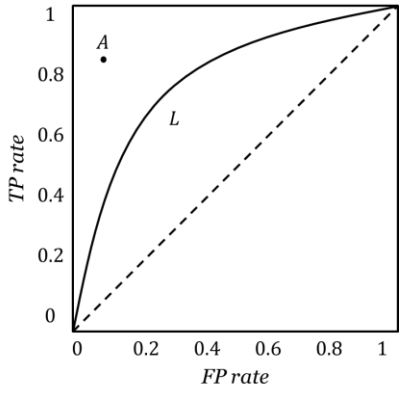

Figure 2. ROC curve

Note that in ROC space the point (0,0) represents a classifier that assigns all examples to the negative class, the point (1,1) represents a classifier that assigns all examples to the positive class, and the point (0,1) represents a classifier that achieves perfect classification. Thus, a discrete classifier is considered to be better than another if its corresponding point in ROC space is closer to the point (0,1).

For classifiers that output a continuous numeric value, such as neural networks, varying the classification threshold produces a series of points in ROC space, such as curve L in Figure 2. In these cases, the Area Under the ROC Curve (AUC) is commonly used to compare the performance of two classifiers. A classifier is considered to be better than another if its corresponding AUC is closer to 1 [13].

ROC curves are insensitive to the class distribution of the data, which is why they have been widely adopted to measure the performance of classifiers in the presence of imbalanced data [13].

However, for highly imbalanced datasets, ROC curves may offer an overly optimistic view of a classifier's performance [14]. Hence, some researchers propose using Precision-Recall curves (*PR curves*), which plot precision over recall, or *cost curves*, which plot performance over the range of possible class distributions and misclassification costs, as an alternative to ROC curves [14,15].

3. Solutions to the Class Imbalance Problem

Several solutions to the class imbalance problem have been proposed. Generally, these solutions are divided in two groups [6,16]:

(a) *Data level solutions*, which consist of modifying the data to remove or minimize the imbalance. These solutions do not modify the learning algorithms, so they are independent of the classifier used. Data level solutions include sampling methods, such as undersampling and oversampling.

(b) *Algorithmic level solutions*, which consist of modifying existing learning algorithms to deal with imbalanced datasets. Algorithmic level solutions include adjusting the class weights or the decision threshold of the classifier to improve the performance (i.e. cost-sensitive methods) and recognition-based learning.

Some of the solutions to the class imbalance problem are described in the following sections, with particular emphasis on data level solutions (e.g., sampling methods).

3.1 Sampling Methods for Imbalanced Learning

Sampling methods aim to improve the performance of a classifier by balancing the class distribution of the data, that is, by adding or removing examples from the minority or the majority class, respectively.

3.1.1 Random Sampling Methods

The basic sampling methods for imbalanced learning are random oversampling and random undersampling.

Random undersampling balances the class distribution by randomly removing majority class examples. The main limitation of this method is that it can discard examples that could be important for learning [6,9,17].

Random oversampling balances the class distribution by randomly replicating minority class examples. The main limitation of this method is that because it duplicates minority class examples, it can lead to overfitting [6,9,17].

To overcome the limitations of random sampling, several informed sampling methods have been proposed.

3.1.2 Informed Undersampling Methods

Informed undersampling methods aim to remove examples that either hinder the classifier's performance (e.g. noisy or borderline examples) or are not important for learning (e.g. examples distant from the decision border).

3.1.2.1 Tomek Links

A *Tomek link* is a pair of minimally distanced nearest neighbors belonging to different classes.

Tomek links can be used as an undersampling method by removing the examples in the Tomek links belonging to the majority class. The goal is to remove majority class examples that are noisy or close to the decision border [17].

Tomek links can also be used as a data cleaning method by removing both examples in the Tomek links [17].

3.1.2.2 Condensed Nearest Neighbor Rule

A subset of examples is consistent if it correctly classifies all examples in the dataset by using the 1-nearest neighbor rule.

The *Condensed Nearest Neighbor Rule* (CNN) finds a consistent subset of examples *S* by (1) randomly selecting one majority class example and all minority class examples, (2) applying the 1-nearest neighbor rule over the examples in *S* to classify the examples in the dataset, and (3) adding the misclassified examples to *S*. Finally, the subset of examples *S* is used for training the classifier [18].

The CNN rule aims to remove majority class examples that are distant from the decision border, because they might be less important for learning. However, it tends to include noisy examples, because they are more likely to be misclassified [18]. For this reason, methods that combine the CNN rule with other sampling strategies that remove noisy examples have been proposed (see section 3.1.2.3).

3.1.2.3 One-sided Selection

One-sided selection (OSS) consists of applying Tomek links as an undersampling method followed by the CNN rule. The goal is to remove majority class examples that are noisy or either close or distant from the decision border [17].

Because Tomek links undersampling is computationally expensive, a variation of OSS proposes the application of the CNN rule first to reduce the size of the dataset [17].

3.1.2.4 Edited Nearest Neighbor Rule and Neighborhood Cleaning Rule

The *Edited Nearest Neighbor Rule* (ENN) is an undersampling method that aims to discard noisy examples by removing majority class examples that are incorrectly classified by at least two of its three nearest neighbors [18].

Similarly, the *Neighborhood Cleaning Rule* (NCL) finds all the examples misclassified by its three nearest neighbors. If an example belongs to the majority class, it is removed. If it belongs to the minority class, its nearest neighbors from the majority class are removed [17].

3.1.3 Informed Oversampling Methods

Informed oversampling methods aim to add either replicated or synthetically generated examples to improve the classifier's performance.

3.1.2.5 Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) is an oversampling method which consists of generating new examples by interpolating two minority class examples. New examples are generated by subtracting the feature vector of a minority example *E* from the feature vector of its nearest neighbor, multiplying it by a random number between 0 and 1 and adding it to the feature vector of *E* [10].

SMOTE aims to avoid the overfitting characteristic of random oversampling and improve generalization by inducing larger and less specific decision regions for the minority class [10].

One of the limitations of SMOTE is that because it interpolates minority class examples without considering the class of their neighbors, it can increase the overlapping between the classes [8]. Several extensions of SMOTE have been proposed to address this issue.

SMOTE + Tomek links consists of applying SMOTE followed by Tomek links as a data cleaning method. The goal is to not only balance class distribution, but to also address the problem of overlapping classes by removing noisy and borderline examples [17].

Similarly, *SMOTE + ENN* consists of applying SMOTE followed by the ENN rule to remove all examples misclassified by its three nearest neighbors. ENN usually removes more examples and thus provides a more thorough data cleaning than Tomek links [17].

Borderline-SMOTE generates examples by interpolating only minority class examples that are close to the decision boundary, that is, minority class examples that have more neighbors belonging to the majority class than to

the minority class. The idea is that because borderline minority class examples are more likely to be misclassified, these new examples will be more useful for learning. It is worth noting that if all the nearest neighbors belong to the majority class, the minority class example is considered noisy and no new examples will be generated [8].

3.1.2.5 Cluster-based Oversampling

Cluster-based oversampling aims to balance the data between and within the classes by (1) creating independent clusters from the minority and majority classes using K-means clustering, (2) randomly oversampling each majority class cluster until all of them contain the same number of examples, and (3) randomly oversampling each minority cluster until all of them contain the same number of examples and the two classes are balanced [8].

Cluster-based oversampling can also be implemented using SMOTE to oversample each cluster instead of random oversampling [8].

3.2 Ensemble Methods for Imbalanced Learning

It has been noted that some sampling methods, particularly random undersampling, have a high variance. That is, classifiers trained with different samples of the data obtained with these methods, tend to exhibit significantly different performances [11].

For this reason, some authors recommend combining sampling methods with ensemble learning techniques to reduce the variance and improve the performance of the classifiers [11].

3.2.1 Bagging Methods for Imbalanced Learning

A simple approach that has been shown to yield good results is to combine random undersampling with bagging [8,11].

Bagging consists of implementing an ensemble of classifiers trained with different bootstrap samples of the data [19]. For imbalanced learning, each of these samples is generated independently by randomly undersampling the majority class. The results of each classifier are then combined via voting.

3.2.2 Boosting Methods for Imbalanced Learning

Other solutions that combine sampling methods with boosting have also been proposed.

SMOTEBoost integrates SMOTE with boosting by generating minority class examples at each iteration of the boosting algorithm [10]. The idea is to focus each successive classifier on the minority class examples, which are more difficult to learn. This way, the final ensemble will have a larger and better-defined decision region.

Some authors have pointed out that data generation methods, such as SMOTE, are computationally expensive [8], especially if combined with ensemble methods. For this reason, alternative methods to oversample the minority class have been proposed.

One of these methods is *JOUS-Boost*, which consists of randomly oversampling the minority class at each iteration of the boosting algorithm and introducing independently and identically distributed noise to replicated minority class examples [20].

3.3 Cost-sensitive Methods for Imbalanced Learning

Cost-sensitive methods do not balance the class distribution of the data. Instead, these methods deal with imbalanced datasets by using a cost matrix that contains the misclassification costs for each class. The goal of cost-sensitive methods is to minimize the total cost associated with the dataset [21].

For imbalanced datasets, the cost of misclassifying minority class examples is usually higher than the cost of misclassifying majority class examples. For example, for medical diagnosis applications, a false negative (i.e., a sick patient classified as healthy) is considered more costly than a false positive (i.e., a healthy patient classified as sick).

There are two main approaches to cost-sensitive methods for imbalanced learning. One works at the data level, by incorporating cost information to sampling or ensemble methods in order to bias the classifier towards the minority class. The other approach works at the

algorithmic level, by directly modifying the learning algorithms to account for unequal misclassification costs [8].

At the data level, cost-sensitive methods include *AdaC1*, *AdaC2*, *AdaC3* and *AdaCost* [8,22]. The basic idea of these methods is to incorporate the misclassification costs to the boosting algorithm. At each iteration of the algorithm, the examples with a higher cost will be more likely to be sampled. This way, the classifiers will focus on the more costly minority class examples.

At the algorithmic level, cost-sensitive methods are specific to a particular learning algorithm. For example, for decision trees, these methods include adjusting the decision threshold, the node splitting criteria or the pruning scheme of the tree [8]. For neural networks, these methods include adjusting the outputs or the learning rate of the network or replacing the error minimization function with an expected cost minimization function [8].

The main limitation of cost-sensitive methods is that they usually require the cost matrix and the corresponding misclassification costs to be known, and in practice this information is rarely available [8,9].

4. Discussion

The use of sampling methods is one of the most common approaches to the class imbalance problem, probably because these methods are easy to implement and are classifier-independent.

Several studies have been conducted comparing different sampling methods for imbalanced learning, including [17] and [23]. These two studies used datasets with various degrees of imbalance from the UCI repository. Ten of these datasets are described in Table 1.

Datasets with more than two classes (e.g., the Letter dataset) were converted to two-class datasets by identifying the class with fewer examples as the minority class and the remaining classes as the majority class.

In [17], Batista et al. compared the following sampling methods: random undersampling and oversampling, Tomek links, CNN, OSS, CNN + Tomek links, NCL, SMOTE, SMOTE + Tomek links and SMOTE + ENN.

Table 1. UCI Datasets

Dataset	Minority Examples	Majority Examples
Pima	268 (34.90%)	500 (65.10%)
German	300 (30.00%)	700 (70.00%)
Haberman	81 (26.47%)	225 (73.53%)
Vehicle	212 (25.06%)	634 (74.94%)
Letter-vowel	3878 (19.39%)	16122 (80.61%)
E.Coli	35 (10.42%)	301 (89.58%)
Satimage	626 (9.73%)	5809 (90.27%)
Glass	17 (7.94%)	197 (92.06%)
Letter-a	789 (3.95%)	19211 (96.05%)
Nursery	328 (2.53%)	12632 (97.47%)

In [23], Hulse et al. compared the following sampling methods: random undersampling and oversampling, OSS, ENN, SMOTE, borderline-SMOTE and cluster-based oversampling. For random undersampling, random oversampling, SMOTE and borderline-SMOTE, various rates of undersampling/oversampling were considered.

In both studies, decision trees constructed with the C4.5 learning algorithm were used as classifiers, and AUC was used as a performance measure. Note that in [23], other classifiers, such as support vector machines, were also evaluated, and the G-Mean and the F-Measure were also calculated.

The results obtained in [23] indicate that for decision trees, the best performance in terms of AUC is achieved with random undersampling. On the other hand, in terms of the F-measure, the best performance is achieved with SMOTE. This shows that the results are highly dependent on the measure selected to evaluate the performance of the classifiers.

Additionally, the results obtained in [23] also indicate that for support vector machines, the best performance is achieved with SMOTE and random oversampling. This highlights the fact that even though sampling methods are classifier-independent, some methods perform better for some classifiers than others.

Conversely, the results obtained in [17] indicate that for decision trees, the best performance in terms of AUC is achieved with oversampling methods, such as random oversampling, SMOTE and different variations of SMOTE.

At the same time, other studies on imbalanced learning using UCI datasets, such as [11], show that for support vector machines, the best results are obtained using random undersampling instead of SMOTE.

This inconsistency in the results of similar studies demonstrates that even though many different solutions to the class imbalance problem have been proposed, this is still an open and challenging problem in data mining.

This also emphasizes the need for standardized evaluation practices and benchmark problems for imbalanced learning that facilitate the comparison between different sampling methods and other solutions for the class imbalance problem [8].

It is worth noting that some studies comparing cost-sensitive methods with sampling methods have also been conducted [22], but the results obtained have not allowed researchers to determine with certainty which of these two approaches deals with the class imbalance problem more effectively.

Furthermore, because cost information is rarely available, sampling methods are used more often than cost-sensitive methods to deal with the problem of imbalanced data in practical applications.

5. Conclusions

This report presented an overview of the class imbalance problem, including some of the proposed solutions to the problem and some of the performance measures used to evaluate classifiers in the presence of imbalanced data.

It was noted that the class imbalance problem is very common in data mining applications, and that in recent years many different solutions to the problem have been proposed. In this report, more emphasis was placed on sampling methods, which are one of the most widely used solutions to the problem of imbalanced data.

It was also shown that many studies comparing different solutions to the class imbalance problem have been either inconsistent with previous studies or inconclusive.

There is clearly a need for a deeper understanding of the problem that allows researchers to propose more effective methods to deal with imbalanced data.

Additionally, the results obtained in these studies indicate that determining which solution to the class imbalance problem is more effective depends on the classifier used, as well as on the performance measure selected to evaluate the classifier.

For this reason, it is important to establish standardized evaluation practices for learning from imbalanced datasets that allow researchers to adequately compare existing and new solutions to the class imbalance problem.

In conclusion, even though there are many publications on this topic, the class imbalance problem is still a challenging and promising area for research in data mining and machine learning.

References

[1] C. Phua, D. Alahakoon and V. Lee. Minority Report in Fraud Detection: Classification of Skewed Data. *ACM SIGKDD Explorations Newsletter*. Vol. 6, No. 1 (2004), 20-29.

[2] M. Kubat, R. Holte and S. Matwin. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*. Vol. 30, No. 2 (1998), 195-215.

[3] A. Sun, E. Lim and Y. Liu. On strategies for imbalanced text classification using SVM: comparative study. *Decision Support Systems*. Vol. 48, No. 1 (2009), 191-201.

[4] M. Mazurowski, P. Habas, J. Zurada, J. Lo, J. Baker and G. Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*. Vol. 21, No. 2-3 (2008), 427-436.

[5] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*. Vol. 37, No. 1 (2006), 7-18.

[6] N. Chawla, N. Japkowicz and A. Kotcz. Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter*. Vol. 6, No. 1 (2004), 20-29.

[7] Q. Yang and X. Wu. 10 Challenging Problems in Data Mining Research. *International Journal of*

Information Technology & Decision Making. Vol. 5, No. 4 (2006), 597-604.

[8] H. He and E. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*. Vol. 21, No. 9 (2009), 1263-1284.

[9] G. Weiss. Mining with Rarity: A Unifying Framework. *ACM SIGKDD Explorations Newsletter*. Vol. 6, No. 1 (2004), 20-29.

[10] N. Chawla. Data Mining for Imbalanced Datasets: An Overview. *Data Mining and Knowledge Discovery Handbook* (2005), 853-867.

[11] B. Wallace, K. Small, C. Brodley and T. Trikalinos. Class Imbalance, Redux. *Proceedings of the IEEE 11th International Conference on Data Mining* (2011), 754-763.

[12] L. Cleofas, R. Valdovinos, V. Garcia and R. Alejo. Use of Ensemble Based on GA for Imbalance Problem. *Proceedings of the 6th International Symposium on Neural Networks: Advances in Neural Networks* (2009).

[13] T. Fawcett. ROC graphs: Notes and practical considerations for researchers (2003). Available at: http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf

[14] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning* (2006), 233-240.

[15] R. Holte and C. Drummond. Cost Curves: An Improved Method for Visualizing Classifier Performance. *Machine Learning*. Vol. 65, No. 1 (2006), 95-130.

[16] S. García and F. Herrera. Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy. *Evolutionary Computation*. Vol. 17, No. 3 (2009), 275-306.

[17] G. Batista, R. Prati and M. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD Explorations Newsletter*. Vol. 6, No. 1 (2004), 20-29.

[18] R. Wilson and T. Martinez. Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning*. Vol. 38, No. 3 (2000), 257-286.

[19] L. Breiman. Bagging Predictors. *Machine Learning*. Vol. 24, No. 2 (1996), 123-140.

[20] D. Mease, A. Wyner and A. Buja. Boosted Classification Trees and Class Probability/Quantile Estimation. *Journal of Machine Learning Research*. Vol. 8 (2007), 409-439.

[21] K. McCarthy, B. Zabar and G. Weiss. Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? *Proceedings of the 1st International Workshop on Utility-Based Data Mining* (2005), 69-77.

[22] W. Fan, S. Stolfo, J. Zhang, and P. Chan. AdaCost: Misclassification Cost-Sensitive Boosting. *Proceedings of the 16th International Conference on Machine Learning* (1999), 97-105.

[23] J. Hulse, T. Khoshgoftaar and A. Napolitano. Experimental Perspectives on Learning from Imbalanced Data. *Proceedings of the 24th International Conference on Machine Learning* (2007), 935-942.