Learning from Imbalanced Datasets (Supervised and Unsupervised Learning)

Arely Aceves Compean

Abstract—When faced with a classification process, data imbalance compromises the accuracy of the predictions. To deal with this issue, resampling methods such as oversampling the minority class or undersampling the majority class may be used. While resampling offers a simple solution to the problem, it is not the only one. The use of learning methods less sensitive to data imbalance, such as decision trees and random forests may also offer an effective solution. This paper proposes a new method that avoids manipulating data distribution by using a combination of unsupervised clustering along with random forest classifiers to tackle the classification of binary labeled data with different degrees of imbalance.

Index Terms—Imbalanced datasets, resampling, oversampling, undersampling, supervised learning, unsupervised learning, clustering, k-means, elbow method, silhouette method, decision trees, random forests.

1 Introduction

The continuous development of science and technology as driven data to row at an exponential rate, thus furthering the importance of knowledge discovery and data engineering research. When dealing with a classification problem, unbalanced data compromises the accuracy of the learning model. Data imbalance results when there is an unequal distribution that leads to one class dominating over the others. Underrepresented, or minority, classes may show less in predictions or be completely ignored. even though data imbalance is more evident in binary datasets, it may also present itself in multi-class datasets.

The most common ways of dealing with data imbalance is the use of resampling techniques where either examples are added or deleted to make distribution equal among all classes. The search of methods that are able to classify data disregarding whether it is balanced or not has resulted in the creation of a new method that combines an unsupervised learning method, such as clustering, with supervised learning using random forest classifier. This proposed method will be compared against a baseline of 2 classifiers, decision trees and random forests. Along with this baseline 2 resampling techniques, oversampling and undersampling, paired with the latter classifier will be 2 extra methods being analysed. These 5 method will be evaluated and compared on their average accuracy, precision, recall and F1-score.

2 Backround

2.1 Imbalanced Datasets

Imbalanced datasets are those that contain data with unequally represented classes. [9] expresses that the natural distribution is not ideal for learning a classifier. [3] points out that the result of this is that the data tends to be classified into the majority class which is usually the less important one.

2.1.1 Resampling

Resampling is a popular way of dealing with this issue [9], [3]. Different forms of this technique are random oversampling with replacement, random undersampling, directed oversampling, directed undersampling, oversampling with informed generation of new samples, and combinations [3].

Random undersampling aims to achieve a balance of the data by removing samples from the majority class until all classes match their desired percentages [9], [3]. The main idea behind this process is to ensure that the minority class has a larger presence in the set [9]. The main drawback of this technique is that it is prone to discard useful data that could be helpful for training; to avoid this analysing the relationships between the data points before deletion may be an option [3].

Oversampling aims to balance class distribution by replicating data from the minority class. [9] and [3] agree that this may lead to overfitting since the new examples are exact copies of existing one making the classifier learn about duplicated data rather than an existing pattern in the minority class. A way of overcoming this problem is by using techniques such as SMOTE (synthetic minority oversampling technique) which rather than duplicating, generates new examples based on existing data [9], [3].

Random over or undersampling may create noise since the random creation or deletion of instances may introduce noise for the classification process [8].

2.2 Classification for Imbalanced Datasets

Classification methods face different difficulties when fed with imbalanced data. Traditional maximum likelihood or bayesian network are dependent on data distribution and when presented with this issue their learning results in overfitting [7]. Neural networks, although more competent than other methods, have a more complex structure, therefore implementing and running them present a higher cost [7]. Kernel methods, especially SVM, are too sensitive to outliers, which affects their performance. Ensemble systems, such as decision tree-based ensembles (bagagge tree, random forest, oblique random forest and rotation forest) are faster, more flexible, simpler and more robust to outliers [7].

2.2.1 Decision Trees

Decision trees learn a model from a training set by inferring decision rules from the data features [1]. Each split is based on a single feature. When they receive a new example they go through the whole tree until they end on a leaf node and output this as their prediction. According to [6], decision trees' 2 main limitations are node splitting based on a single feature, which is inefficient in dealing with dependencies between features, and the use of orthogonal splits are not optimal when high correlation is present.

2.2.2 Random Forests

[2] defines random forests as a classifier that consists of a collection of tree-structured classifiers where each of this casts a unit vote for the most popular class for a given input. Random forests are well suited for handling high dimensional data [6]. It reduces bias and variance and uses randomization to control tree diversity [6]. This algorithm shares the limitations of its sub-components, decision trees.

2.3 Clustering

Clustering is a method of finding groups in multivariate data [5]. Since the number of clusters in different datasets is not commonly known, this is categorized as an unsupervised learning method. In order to find the optimal amount of clusters, clustering algorithms like the "simple" k-means, tend to be paired with approaches such as the silhouette and elbow methods [5].

2.3.1 K-Means

This is the simplest clustering algorithm, its main purpose is to identify k clusters in any given dataset [5]. [5] also identifies the following steps as part of this algorithm:

1) Randomly place k points in the space represented by the objects to cluster. These points represent initial centroids for each group.
2) Assign each object to its closest centroid.
3) Use the mean of the distances of all points in a cluster to calculate its new centroid.
4) Repeat previous 2 steps until all objects have been assigned to a cluster and centroids no longer move.

2.3.2 Silhouette Method

The silhouette width or silhouette score was introduced by Kaufman and Rousseeuw and is known to have a good performance [5]. This concept involves the difference of within-cluster tightness and separation from the rest [5]. It is defined as $s(i)$ where $i \in I$ such that:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

Where $a(i)$ is the average distance between i and all the other elements within the cluster it belongs to and $b(i)$ is the minimum of the average distances between i and all the elements from the rest of the clusters [5].

The average silhouette score of all i in I can characterize a clustering [5]. The largest the silhouette score, the best the number of clusters; with this in mind, the optimal k for figure 1 is 9.

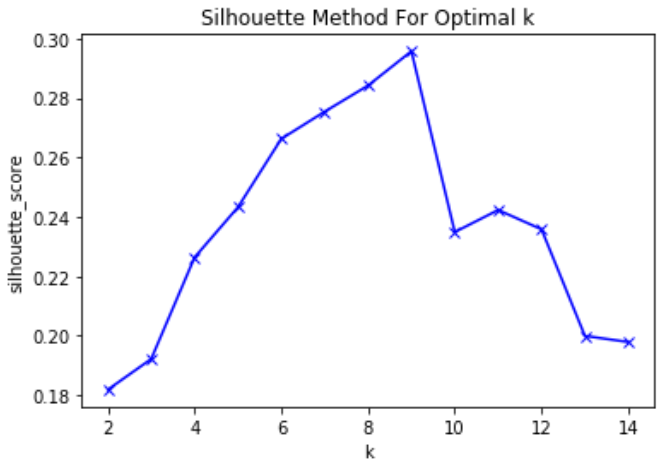

Fig. 1. Silhouette Method for Optimal k.

2.3.3 Elbow Method

It is the oldest method of calculating the optimal number of clusters in the data [5]. It is a visual method that consists of calculating the sum of squared errors of a range of clusters starting from k = 2 and plotting it into a graph (as shown in figure 2 in order to identify the "elbow" or the inflection point, where the difference between errors of k and k + 1 is too small to be significant [5]. The main issue with this method is that sometimes this "elbow" is not as evident as expected and is hard to identify, such is the case of figure 2.

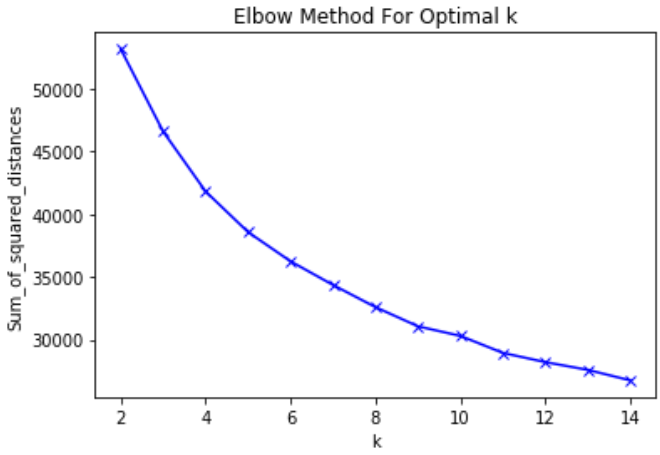

Fig. 2. Elbow Method for Optimal k.

2.4 Evaluation

2.4.1 Measures

2.4.1.1 Accuracy

Shows the ratio of correct predictions (true positives and true negatives) done by the model [4].

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

TABLE 1
Confusion Matrix.

2.4.1.2 Precision

Shows the ratio of true positives to predicted positives (false positives and true positives) labeled by the model [4].

$$precision = \frac{tp}{tp + fp}$$

2.4.1.3 Recall

Shows the ratio of true positives to actual positives (false negatives and true positives) labeled by the model [4].

$$recall = \frac{tp}{tp + fn}$$

2.4.1.4 F1-score

Shows the balance between precision and recall with a value of 1 showing full balance greater than 1 favouring precision and lower than one favouring recall [4].

$$F1 = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * precision + recall}$$

3 METHODOLOGY

The task at hand was to create a classification method that was able to label data withstanding different degrees of imbalance. The baseline for this task was set by using decision trees and random forests as classifiers. Alongside these, 2 resampling techniques coupled with a random forest classifier were also tested and compared.

3.1 Datasets

In order to see the behaviour of each method at varying degrees of imbalance, three datasets were selected; one with low imbalance (65%), another with high imbalance (94.98%) and a last one which lays in-between (75.92%).

3.1.1 Credit Card

This dataset was modified in order to represent a high imbalance, 94.98%. It was undersampled, which means that 275000 transactions were removed from the majority class. After this, the size of the dataset was 9808 examples, each with 30 features. It contains only quantitative data that did not need any extra transformation before turning it into feature vectors of size 30.

3.1.2 Income Evaluation

This dataset had a mid range imbalance of 75.92%. It is composed of 32561 examples. It contains both categorical and quantitative data which means that it had to be transformed into binary data before being turned into feature vectors of size 109.

3.1.3 Mushroom

This dataset was modified in order to represent a low imbalance, 65%. It was undersampled, which means that 1655 examples were removed from the majority class. After this, the size of the dataset was 6470 examples. It contains only categorical data that had to be transformed into binary data before being turned into feature vectors of size 117.

3.2 Methods

3.2.1 Baseline

The baseline methods were chosen due to them being known to perform well regardless of class distribution. They were trained using the imbalanced data directly.

3.2.2 Resampling

From the 2 classifiers used, random forest was selected to be paired with resampling techniques, which are typically used to deal with imbalanced data. This classifier was favoured over decision tree due to the fact that it was also employed in the proposed method, thus making it more comparable. Resamplings involved either oversampling the majority class or undersampling the minority class before training the classifier. The result of this process is an equal distribution of all classes.

3.2.2.1 Oversample

It consisted of the generation of new examples of the minority class by duplicating existing ones randomly. To do so, the function RandomOverSampler from the imblearn library was used. The oversampling was applied after splitting the data and it was only done to the training set to avoid data bleeding, duplicated examples in both the train and test sets.

3.2.2.2 Undersample

Some examples were randomly deleted from the majority class using the RandomUnderSampler function from the imblearn library. This was applied to just the training set to match the methodology of the oversampling technique.

3.2.3 Proposed Method

It consists of a k-means clusterer paired with random forest classifiers. The main idea was to train a k-means clusterer to find the clusters in the data and then train a random forest classifier for each of them.

K-means receives the desired k number of clusters as one of its parameters. The optimal k for each dataset was selected from a range of 2 to 14 by means of the silhouette and elbow methods. The best performing k was selected from each of the displayed graphs and manually input. The silhouette graph plots the silhouette scores against their corresponding k as shown in figure 1; while the elbow graph plots the sum of squared errors against their corresponding k as shown in figure 2.

From these 2 values a range of possible candidates was extracted and the one with the best F1 score was selected. The optimal k was used to train the k-means clusterer that was used.

Moving onto the classification stage, a random forest classifier was trained for each of the clusters containing data from more than one class; the training examples were the members of the cluster. Clusters with a single class use this as their prediction.

4 Results

The evaluation process consisted of a 10 fold cross-validation where the data was divided into 10 bins with 9 being used as the training set and the remaining one as the test set. The sets for each iteration are composed of different permutations of the bins. The data was split using stratified k fold to keep the data imbalance on the sets. The measures chosen to analyse the performance of the methods were accuracy, precision, recall and F1-score, which were calculated for each iteration. The mean of each of these measures were assigned as final scores for each of the methods that were tested.

4.1 High Imbalance

The scores of each measure for the tested methods on the high imbalanced dataset, credit card, are displayed in table 2. The proposed method was not the most accurate, it was the third best with an accuracy of 0.9913. It presented the highest precision and F1-scores with 0.9971 and 0.9940 respectively; therefore even if it was not the most accurate it had the 99.71% of its predicted positives were correctly labeled and it had the most balanced recall and precision. The most accurate classifier was random forest with an accuracy of 0.9929, bot it presented the worst recall, 0.8716; even with the best accuracy, it had the lowest amount of actual positives correctly labeled. Not far behind comes the accuracy of this classifier paired with the undersampling technique, 0,9928, it also presents the best overall performance on the rest of the measures. The worst performing classifier was the decision tree with an accuracy of 0.9195 and an F1-score of 0.8813.

High Imbalance					
Method		Measure			
		Accuracy	Precision	Recall	F1
Baseline	Decision Tree	0.9195	0.9004	0.9571	0.8813
	Random Forest	0.9929	0.9866	0.8716	0.9214
Resampling	Oversampling	0.9827	0.9852	0.9827	0.9790
	Undersampling	0.9928	0.9932	0.9928	0.9920
Proposed Method		0.9913	0.9971	0.9913	0.9940

TABLE 2
Results from Credit Card Dataset (High Imbalance).

4.2 Mid Imbalance

The scores of each measure for the tested methods on the mid imbalanced dataset, income evaluation, are displayed in table 3. The proposed method had the third best accuracy for the mid imbalanced dataset and presented the highest precision with a score of 0.9983, which means that 99.83% of its predicted positives were correctly labeled. The best overall performance was that of the decision tree classifier followed by the random forest classifier. The worst performing methods were those that employed resampling, they had the lowest scores in all 4 measures.

Mid Imbalance					
Method		Measure			
		Accuracy	Precision	Recall	F1
Baseline	Decision Tree	0.8436	0.8606	0.9476	0.9020
	Random Forest	0.8254	0.8201	0.9864	0.8956
Resampling	Oversampling	0.7360	0.7673	0.7360	0.7174
	Undersampling	0.7356	0.7672	0.7356	0.7169
Proposed Method		0.7609	0.9983	0.7609	0.8625

TABLE 3
Results from Income Evaluation Dataset (Mid Imbalance).

4.3 Low Imbalance

The scores of each measure for the tested methods on the low imbalanced dataset, mushroom, are displayed in table 4. The proposed method had the worst performance on the low imbalanced dataset, the scores of the 4 measures were less than half of the rest of the methods. The decision tree classifier was the best performing out of all the 5 methods, it was closely followed by the method featuring the random forest with undersampled data. There was no much difference between the random forest classifier without resampling and it paired with oversampled data. Both baseline methods had a recall of 1, all of the actual positives were correctly labeled.

Low Imbalance					
Method		Measure			
		Accuracy	Precision	Recall	F1
Baseline	Decision Tree	0.9839	0.9764	1.0000	0.9879
	Random Forest	0.9430	0.9285	1.0000	0.9607
Resampling	Oversampling	0.9436	0.9655	0.9436	0.9479
	Undersampling	0.9465	0.9675	0.9465	0.9507
Proposed Method		0.4017	0.4413	0.4017	0.4091

TABLE 4
Results from Mushroom Dataset (Low Imbalance).

5 Discussion

The proposed method had the best performance when the data was highly imbalanced and it started decreasing along with the data imbalance; the more balanced the data, the worse scores this method achieved. The dataset size may have had an effect on this. The most balanced dataset was also the smallest one, so whether the performance was affected by the data distribution or its size is unclear. The biggest dataset was the one with mid imbalance, but this did not increase the performance of the proposed method and the other 4 methods also lowered their scores with this data.

Another factor could be the amount of features present in the data, the higher the number, the worse the performance of the proposed method. This could have been due to them playing an essential role during the clustering process since they are used to discover a relationship between the data points and the high dimensionality may have produced noisier results. This can be confirmed since the dataset with more dimensions produced the greater number of clusters. This dataset was also the one where the random forest performed worse than the decision tree classifier.

The bad performance of the proposed classifier on the low imbalance could be due to the equal representation of classes on clusters, making it hard to choose a label per cluster, the prediction could be more a casualty than a causality.

During the initial undersampling, used to fit the imbalance specifications, some representative examples may have been removed, affecting the learning process. To avoid this from happening, techniques like NearMiss could have been used, these analyse the data before proceeding with the deletions so the most significant data points may be spared. Another alternative could have been to do an initial oversampling, but whether to use the simple random oversampling or a more specialised method like SMOTE remains unknown, this would have to be tested. Even if this is not supposed to be considered as part of the process, it still affected the original state of the data and had to be commented on.

6 Conclusion

More standardized tests may be done with datasets of the same size and shape, but the use of different datasets presents a more realistic situation since not all data is the same and the methods will have to deal with all types of datasets. The number of feature present in the data seems to have affected the results of the proposed method, but did not have such effect on the rest of the tested methods. It is hard to determine whether the data imbalance does have an effect on the proposed method or it was just the difference in dimensionality, since both the number of features and the amount of imbalance are inversely proportional. There is no conclusive evidence on whether the size of the dataset interfered with the obtained results.

References

[1] L. Breiman, Ed., *Classification and regression trees*, The Wadsworth & Brooks/Cole statistics/probability series, Pacific Grove: Wadsworth, 1984, 358 pp., ISBN: 978-0-534-98054-2 978-0-534-98053-5.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 1, 2001, ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324 (visited on 04/28/2020).

[3] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," p. 12, 2006.

[4] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation," in *AI 2006: Advances in Artificial Intelligence*, A. Sattar and B.-h. Kang, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2006, pp. 1015–1021, ISBN: 978-3-540-49788-2. DOI: 10.1007/11941439_114.

[5] T. Kodinariya and P. Makwana, "Review on determining of cluster in k-means clustering," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, pp. 90–95, Jan. 1, 2013.

[6] N. Poona, A. Van Niekerk, and R. Ismail, "Investigating the utility of oblique tree-based ensembles for the classification of hyperspectral data," *Sensors*, vol. 16, no. 11, p. 1918, Nov. 2016. DOI: 10.3390/s16111918. [Online]. Available: https://www.mdpi.com/1424-8220/16/11/1918 (visited on 02/18/2020).

[7] I. Khosravi and Y. Jouybari-Moghaddam, "Hyperspectral imbalanced datasets classification using filter-based forest methods," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 4766–4772, Dec. 2019, ISSN: 2151-1535. DOI: 10.1109/JSTARS.2019.2914668.

[8] A. Puri and M. K. Gupta, "Comparative analysis of resampling techniques under noisy imbalanced datasets," in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, ISSN: null, vol. 1, Sep. 2019, pp. 1–5. DOI: 10.1109/ICICT46931.2019.8977650.

[9] N. V. Chawla, "C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure," p. 9,