UNIVERSIDADE DE ÉVORA

Departamento de Informática

Data Mining

Classification of Unbalanced Datasets

Štěpán Pešout

Universidade de Évora
17/12/2021

Index

Introduction

Unbalanced dataset usually means that the different classes are neither equally represented nor close to the equal representation. If the classification problem is binary, this could be a situation where, for example, class A has 90% representation in the dataset, while class B has only the remaining 10%.

Of course, this problem can also occur in multi-class classification tasks; however, these differ only minimally from binary classification problems in this discussed aspect, as most techniques are applicable to both types.

In the real world, these situations are quite common. The 9:1 class distribution and even much more unbalanced ones are easy to encounter, for instance, when assessing whether certain people suffer from some diseases. A good example can also be recognizing credit card frauds.

Unbalanced classes cause problems especially in situations, when they are distributed in the ratio of 4:1 or worse *(Brownlee, 2015)*.

If Weka is used for binary classification, this unbalance can easily be detected using Zero Rule because it always returns the most frequent class. In Python's scikit-learn package, DummyClassifier can be used for the same purpose. If the classification accuracy of these simple algorithms is surprisingly good, then it is certain that the dataset is unbalanced. The percentage value of accuracy in this case even exactly corresponds to the ratio in which the classes are distributed in the input dataset.

Problems of unbalanced datasets

If the main goal is to create clusters – divide the samples into certain natural groups, then the disproportion between the classes is not a big problem. For some datasets, unbalance may not even be a problem in classification. This occurs when the classes are easily splittable using some available features *(Lahera, 2019)*.

Nevertheless, in most of the classification tasks, it is inappropriate to train models on datasets which are unbalanced, because the model will probably be poor. Sometimes not even the chosen algorithm matters.

The reason is quite simple. The algorithm gets a large number of samples of one class prompting it to be biased towards that particular one. Then, it fails to recognize what makes samples of the minority classes different from the others. Thus, it does not reveal hidden patterns that would lead to the correct distinction of one class from another. Therefore, if the dataset is significantly unbalanced, the model may be overfitted with the data of the majority class. *(Lahera, 2019)*.

In this case, also the accuracy paradox may appear *(Brownlee, 2015)*. Since the test dataset should have the same distribution as the training one, this may result in all samples being classified as the majority class. This problem obviously persists even when using the cross validation. Although such a prediction may show high accuracy, it does not change the fact that it is very unsatisfactory. The problem is in the way accuracy is calculated, because it is only a ratio of the correct predictions to all of them.

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}$$

In a nutshell, it is very problematic to train a prediction model on unbalanced datasets. The resulting models may then have high accuracy while testing, but it is often a biased and misleading number.

Possible solutions

Changing performance metrics

The primary thing to do is to use other metrics instead of accuracy to see if the model is actually good or if the performance result is just a biased value. One of the satisfactory and frequently used ways to show the quality of a model after applying the test set is a confusion matrix – a table showing the correct predictions and the types of incorrect predictions that occurred during model testing *(Tan, Steinbach and Kumar, 2014)*. However, sometimes it is useful to represent the model performance just by one number.

The first possibility is **precision (p)**, which can be calculated as the fraction of true positives among all of the examples that were predicted to belong to a certain class. In other words, it shows how many of the selected ones are relevant *(Tan, Steinbach and Kumar, 2014)*.

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Recall (r) is also commonly used. It shows how many relevant items are selected. It is defined as the fraction of examples which were predicted to belong to a class respecting all of the examples that actually belong to the class *(Tan, Steinbach and Kumar, 2014)*.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Precision and recall can be combined into another metric known as the **F-measure**. It represents a harmonic mean between these two metrics. It ensures that both precision and recall are high enough *(Tan, Steinbach and Kumar, 2014)*.

$$F\text{–}measure = \frac{2rp}{r + p}$$

There are many other metrics, which may be used for this purpose; for instance ROC curves or Normalized Gini Score.

Collecting more data

Large datasets are generally important for training, because it helps to improve the performance of the resulting model. In the classification process, in most cases it is

not mandatory for the ratio of the samples of the different classes to exactly match the real world. For instance, when the goal is to recognize some kind of fraud, which happens in one of 100 cases, it is not always necessary to have one “fraud sample” per 99 regular ones.

On the contrary, in order to improve the model, it is desirable to obtain more data that belongs to the underrepresented classes in particular. This makes it possible for the algorithm to find the hidden patterns in the data and thus classify the samples using a model with higher performance.

Resampling the dataset

The main goal of resampling is also to reduce the ratio of samples of the majority and minority classes. Obviously, the following techniques may be reasonably combined with one another.

Undersampling

The idea of undersampling is to reduce instances of the majority class. This is usually done by keeping instances of classes that have lower representation and selecting a certain number of elements from the majority class, so that the representation is almost equal *(Lahera, 2019)*.

However, there is a risk that some useful samples will be eliminated. This problem can be solved, for example, by making the elimination not in a random but in an informed way with respect to the minority classes. Reliable way is for instance to remove those located far away from the decision boundary. Also, the k-nearest neighbors algorithm is a good choice to do this *(Tan, Steinbach and Kumar, 2014)*.

Oversampling

Oversampling is quite the opposite. It usually replicates the samples of the minority classes in order to get closer to equality between them and the major class. However, with noisy data, oversampling may be dangerous – it may cause model overfitting, because some of the noise examples may be replicated many times. In principle it means that oversampling does not add any new information into the original training set *(Tan, Steinbach and Kumar, 2014)*.

It is even possible to create new synthetical observations using some algorithms. It should be always based primarily on the available data. Variational Autoencoders (VAE) or SMOTE (Synthetic Minority Over-sampling Technique) may be used to do this *(Lahera, 2019)*.

Choosing different algorithm

It is always a rule of thumb to choose different classifiers according to the task and the input data, because there may be observed significant differences between them. Not the same algorithm fits every task and every dataset.

Another option is to use special algorithms that can divergently penalize different types of wrong decisions. These penalties can force the classifier to take account of the minority classes. When working with Weka it is possible to use, for example, the CostSensitiveClassifier. To avoid such problems it is often necessary to try various types of cost matrices in order to get the best results *(Brownlee, 2015)*.

Moreover, in some situations it is possible to change the whole approach to the problem. Instead of building a classifier it may be beneficial to analyze the input dataset with some algorithm which is capable of detecting anomalies. Suitable algorithms are, for instance, One Class SVM or Local Outlier Factor, that are available in Python's scikit-learn package *(Lahera, 2019)*.

Conclusion

In unbalanced datasets the different classes are not equally represented. This may occur in different data for both binary and multi-class classification tasks. When working with them, it is common to face certain challenges.

If unbalanced data is used for training, the resulting model will probably perform very poorly. It is also necessary to pay attention to the possible occurrence of the accuracy paradox; a situation where there is a high accuracy even though the model is poor. In these tasks, emphasis must therefore be placed on choosing appropriate performance metrics that should be used instead of accuracy.

In these situations, when it is not possible to obtain more data from the minority classes, undersampling (removing majority class samples), oversampling (adding minority class samples), or a combination of these techniques can be used.

Sometimes using a different algorithm or a particular change of the approach to the specific problem may also help.

When solving real problems, it is often best in practice to choose a combination of different techniques, but it varies according to the specific situation – the task and the dataset.

Bibliography

- Brownlee, J. (2015) *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset, Machine Learning Mastery.* Available at: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/ (Accessed: December 17, 2021).

- Lahera, G. (2019) *Unbalanced Datasets & What To Do About Them.* Available at: https://medium.com/strands-tech-corner/unbalanced-datasets-what-to-do-144e0552d9cd (Accessed: December 17, 2021).

- Tan, P.-N., Steinbach, M. and Kumar, V. (2014) *Introduction to Data Mining.* Boston: Pearson Education Limited.