NICE ACTIMIZE

Rarity Problem in Supervised Fraud Detection: Undersampling Data-Level Approach for Imbalance Classification

Insights Article

Danny Butvinik
Chief Data Scientist | NICE Actimize

Abstract

Rarity event problems are one of the major challenges in supervised fraud detection and data mining tasks. Imbalanced datasets degrade the performance of data mining and machine learning techniques as the overall accuracy and decision making are biased to the majority class, which leads to misclassifying the minority class samples or, furthermore, treating them as noise. Building classification models from such imbalanced datasets is a relatively new challenge in the machine learning and data mining community because many traditional classification algorithms assume similar proportions of majority and minority classes. When the data is imbalanced, these algorithms generate models that achieve good classification accuracy for the majority class, but poor accuracy for the minority class. Such limitation is addressed by the methods which exploit main approaches, including data-level, algorithmic-level and hybrid approaches. This article provides an overview of the data-level approach and its main objective of better characterizing the financial transactions. This characterization will focus on two possible target classes – legitimate or fraudulent – that create information asymmetry and cause the problem of imbalanced datasets or curse of imbalanced data.

Introduction

In many supervised learning applications, there is a significant difference between the prior probabilities of different classes, i.e., between the probabilities with which an example belongs to the different classes of the classification problem. Known as the class imbalance problem, this situation is common where finance domains are considered one of the top problems in data mining today. Furthermore, it's worth pointing out that the minority class usually has the highest interest from a learning point of view and also a higher cost when it is not well classified. The problem with imbalanced datasets is that standard classification learning algorithms are often biased towards the majority class, resulting in a higher misclassification rate for the minority class instances. In recent years, many solutions have been proposed to deal with this problem, both for standard learning algorithms and for ensemble techniques. They can be categorized into three major groups:

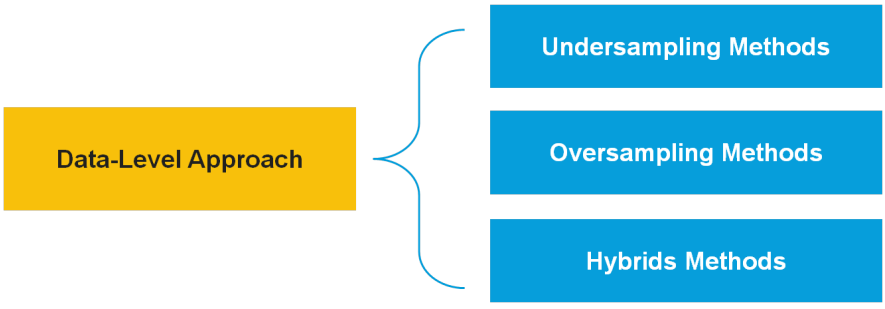

Figure 1: Data-level approach taxonomy: three families of resampling techniques for class imbalance problem.

This article will review these types of methodologies, presenting a taxonomy for each group, enumerating and briefly describing the main properties of the most significant approaches that have been traditionally applied in this field.

Copyright © 2020 NICE Actimize. All rights reserved.

Most of the studies on the behavior of several standard classifiers in imbalance domains have shown that significant loss of performance is mainly due to the skewed class distribution, given by the imbalance ratio (IR), defined as the ratio of the number of instances in the majority class to the number of examples in the minority class. However, there are several investigations which also suggest that there are other factors that contribute to such performance degradation. However, the scope of this article is to focus only on the problem of data skewedness and to overview the main methods and approaches.

Resampling methods are designed to change the composition of a training dataset for an imbalanced classification task. Most of the attention of resampling methods for imbalanced classification is focused on oversampling the minority class. Nevertheless, a suite of techniques has been developed for under sampling the majority class that can be used in conjunction with effective oversampling methods.

Imbalanced Datasets in Classification

In the classification problem field, the scenario of imbalanced datasets materializes frequently. The main property of this type of classification problem is that the examples of one class significantly outnumber the examples of the other one. The minority class usually represents the most important concept to be learned. However, it is difficult to identify because it might be associated with exceptional and significant cases or the data acquisition of these examples is costly. In most cases, the imbalanced class problem is associated with binary classification, but the multi-class problem often occurs and, since there can be several minority classes, is more difficult to solve.

Since most of the standard learning algorithms consider a balanced training set, this may generate suboptimal classification models, i.e., a good coverage of the majority examples, but one where minority ones are frequently misclassified. Therefore, those algorithms, which obtain a good behavior in the framework of standard classification, do not necessarily achieve the best performance for imbalanced datasets. There are several reasons behind this behavior:

- The use of global performance measures for guiding the learning process, such as the standard accuracy rate, may provide an advantage to the majority class.
- Classification rules that predict the positive class are often highly specialized and thus their coverage is very low, so they are discarded in favor of more general rules, i.e., those that predict the negative class.
- Very small clusters of minority class examples can be identified as noise, and therefore they could be wrongly discarded by the classifier. On the contrary, a few real noisy examples can degrade the identification of the minority class since it has fewer examples with which to train.

Addressing Classification with Imbalanced Data

A large number of methods and approaches have been proposed to handle the class imbalance problem. These approaches can be categorized into two groups:

- Internal approaches that create new algorithms or modify existing ones to account for the class imbalance problem; and
- External approaches that pre-process the data to diminish the effect of their class imbalance.

Furthermore, cost-sensitive learning solutions. incorporating both the data-level and algorithmic-level approaches, assume higher misclassification costs for samples in the minority class and seek to minimize the high-cost errors. Ensemble methods are also frequently adapted to imbalanced domains, either by modifying the ensemble learning algorithm at the data-level to pre-process it before the learning stage of each classifier or by embedding a cost-sensitive framework in the ensemble learning process.

Copyright © 2020 NICE Actimize. All rights reserved.

Fig.1 illustrates the taxonomy of resampling techniques that may affect skewed class distribution to handle the problem of imbalanced datasets.

i. Under sampling methods, which create a subset of the original dataset by eliminating instances (usually majority class instances).
ii. Over sampling methods, which create a superset of the original dataset by replicating some instances or creating new instances from existing ones.
iii. Hybrid methods, which combine both sampling approaches.

Undersampling Methods

There are many different types of undersampling techniques, although most can be grouped into those that select examples to keep in the transformed dataset, those that select examples to delete, and hybrids that combine both types of methods.

Undersampling refers to a group of techniques designed to balance the class distribution for a classification dataset that has a skewed class distribution. An imbalanced class distribution will have one or more classes with few examples (the minority classes) and one or more classes with many examples (the majority classes). It is best understood in the context of a binary (two-class) classification problem where class “0” is the majority class and class “1” is the minority class.

Undersampling techniques remove examples from the training dataset that belong to the majority class to better balance the class distribution, such as reducing the skew from a 1:10000 to a 1:100, 1:2, or even a 1:1 class distribution. This is different from over sampling that involves adding examples to the minority class to help reduce the skew in the class distribution.

Undersampling, that consists of reducing the data by eliminating examples belonging to the majority class with the goal of equalizing the number of examples of each class.

Undersampling methods can be used directly on a training dataset that can then, in turn, be used to fit a machine learning model. Typically, undersampling methods are used in conjunction with an oversampling technique for the minority class, and this combination often results in better performance than using oversampling or undersampling alone on the training dataset.

The simplest undersampling technique involves randomly selecting examples from the majority class and deleting them from the training dataset. This is referred to as random undersampling. Although simple and effective, a limitation of this technique is that examples are removed without any concern for how useful or important they might be in determining the decision boundary between the classes. This means it is possible, or even likely, that useful information will be deleted.

The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the induction process. The removal of data is a critical decision to be made, hence many who propose undersampling use heuristics to overcome the limitations of the non-heuristic decisions.

An extension of this approach is to be more discerning regarding the examples that are deleted from the majority class. This typically involves heuristics or learning models that attempt to identify redundant examples for deletion or useful examples for non-deletion. There are many undersampling techniques that use these types of heuristics. In the following sections, we will review some of the more common methods and develop an intuition for their operation on a synthetic imbalanced binary classification dataset.

Copyright © 2020 NICE Actimize. All rights reserved.

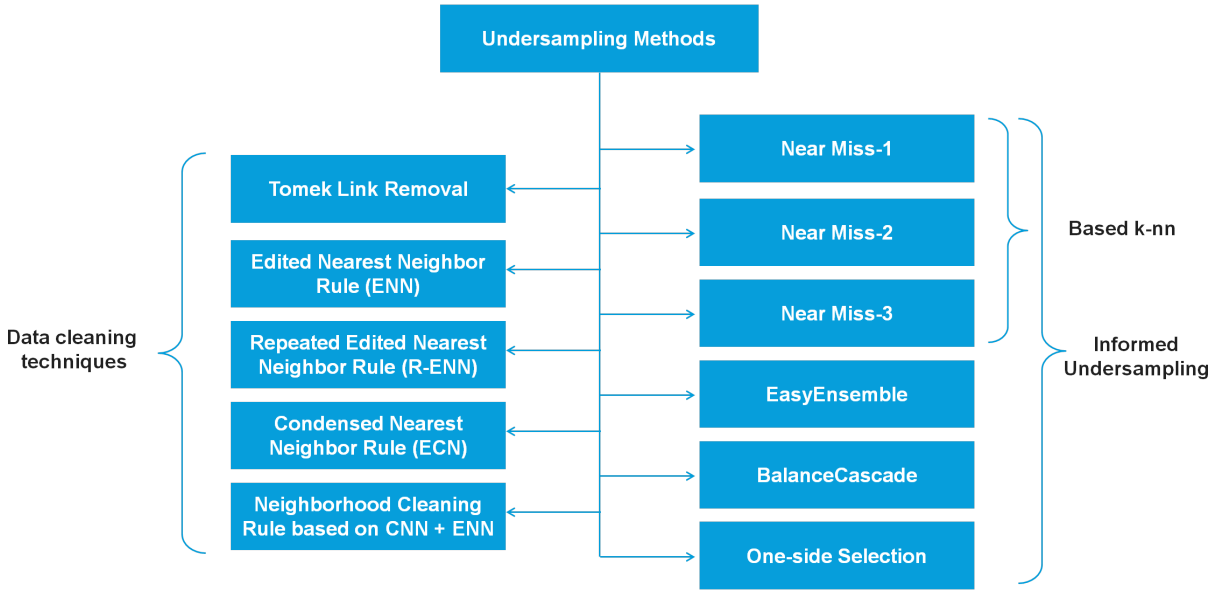

Figure 2: Taxonomy of data-level undersampling methods for class imbalance classification

EasyEnsemble: The idea behind EasyEnsemble is simple. Similar to the balanced Random Forests, EasyEnsemble generates T balanced sub-problems. The output of the ith sub-problem is AdaBoost classifier H_i , an ensemble with s_i weak classifiers $\{h_{i,j}\}$. An alternative view of $h_{i,j}$ is to treat it as a feature that is extracted by the ensemble learning method and can only take binary values. H_i, in this viewpoint, is simply a linear classifier built on these features. Features extracted from different subsets N_i thus contain information of different aspects of the original majority training set *N* . Finally, instead of counting votes from $\{H_i\}_{i=1..T}$, we collect all the features $h_{i,j}$ $(i = 1..T$, j= $1..s_i)$, and form an ensemble classifier from them. The output of EasyEnsemble is a single ensemble, but it looks like an 'ensemble of ensembles'. It is known that "Boosting" mainly reduces bias while "Bagging" mainly reduces variance.

BalanceCascade: The idea is as follows. After H_1 is trained, if an example $x_1 \in N$ is correctly classified to be in the majority class by H_1, it is reasonable to conjecture thatx_1 is somewhat redundant in *N* , given that we already have H_1. Thus, we can remove some correctly classified majority class examples from *N* . As in EasyEnsemble, we use AdaBoost in this method. This method is called BalanceCascade since it is somewhat like the cascade classifier. The majority training set *N* is shrunk after every H_i is trained, and every node H_i is dealing with a balanced sub-problem ($|N_i| = |P|$). However, the final classifier is different. A cascade classifier is the conjunction of all $\{H_i\}_{i=1..T}$, i.e.$H(x)$ predicts positive if and only if all $H_i(x)$ $(i = 1..T)$ predict positive. BalanceCascade is similar to EasyEnsemble in their structures.

Both EasyEnsemble and BalanceCascade are very efficient. Their training time is roughly the same as that of under-sampling when the same number of weak classifiers are used.

Copyright © 2020 NICE Actimize. All rights reserved.

Methods that Select Examples to Keep

In this section, we will take a closer look at two methods that choose which examples from the majority class to keep, the "near-miss" family of methods, and the popular condensed "nearest neighbor" rule.

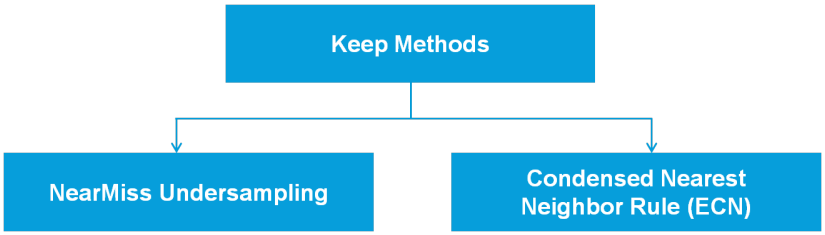

Figure 3: An illustration for Keep methods

NearMiss Undersampling

The **NearMiss** family of methods performs undersampling of instances in the majority class based on their distance to other instances in the same class. In NearMiss-1, those instances are retained whose mean distance to the K nearest instances is lowest, where K is a tunable hyperparameter. NearMiss-2, in contrast to Nearmiss-1, keeps those instances whose mean distance to the K farthest instances is lowest. NearMiss-3 selects K-NN in majority class for every instance in minority class. In this case, the undersampling ratio is directly controlled by K and is not separately tuned.

The NearMiss-3 seems desirable, given that it only keeps those majority class examples that are on the decision boundary.

Condensed Nearest Neighbor Rule Undersampling (CNN)

Condensed Nearest Neighbor (or CNN) is an undersampling technique where the goal is to choose a subset U of training dataset T, such that for every instance in training dataset T, its nearest neighbor in U is of the same class. Undersampling via CNN can be slower compared to other methods since it requires many passes over the training data. Further, because of randomness involved in the selection of instances at each iteration, the subset selected can vary significantly. A variant of CNN is only undersample L (majority class) – i.e., retain all instances from S (minoity class), but retain only those instances in L that belong to U.

CNN undersampling technique seeks a subset of a collection of samples that results in no loss in model performance, referred to as a minimal consistent set. It is achieved by enumerating the examples in the dataset and adding them to the *"store"* only if they cannot be classified correctly by the current contents of the store. This approach was proposed to reduce the memory requirements for the k-Nearest Neighbors (KNN) algorithm. When used for imbalanced classification, the store is comprised of all examples in the minority set and examples from the majority set that cannot be classified correctly.

Copyright © 2020 NICE Actimize. All rights reserved.

Methods that Select Examples to Delete

In this section, we will take a closer look at methods that select examples from the majority class to delete, including the popular Tomek Links method and the Edited Nearest Neighbors rule.

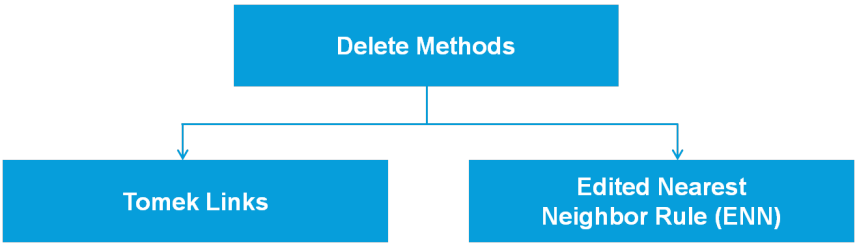

Figure 4: An illustration of Delete methods

Understanding of Tomek Links
A criticism of the Condensed Nearest Neighbor Rule is that examples are selected randomly, especially at the outset. This results in a) retention of unnecessary samples and b) occasional retention of internal rather than boundary samples.

Two modifications to the CNN procedure were proposed by Ivan Tomek in his 1976 paper [1]. One of the modifications (method 2) is a rule that finds pairs of examples, one from each class; they together have the smallest Euclidean distance to each other in feature space.

A pair of examples is called a "Tomek link" if they belong to different classes and each other's nearest neighbor. Undersampling can be done by removing all Tomek links from dataset. An alternate method is to only remove the majority class instances that are part of a Tomek link.

This means that in a binary classification problem with classes "0" and "1," a pair would have an example from each class and would be closest neighbors across the dataset. In words, instances *a* and *b* define a Tomek Link if:

i. instance *a*'s nearest neighbor is *b*,
ii. instance *b*'s nearest neighbor is *a*, and
iii. instances *a* and *b* belong to different classes.

These cross-class pairs are now generally referred to as "*Tomek Links*" and are valuable as they define the class boundary.

Method 2 has another potentially important property: It finds pairs of boundary points which participate in forming the (piecewise-linear) boundary. Such methods could use these pairs to generate progressively simpler descriptions of acceptably accurate approximations of the original completely specified boundaries.

The procedure for finding Tomek Links can be used to locate all cross-class nearest neighbors. If the examples in the minority class are held constant, the procedure can be used to find all examples in the majority class that are closest to the minority class, and then removed. These would be the ambiguous examples.

From this definition, we see that instances that are in Tomek Links are either boundary instances or noisy instances. This is because only boundary instances and noisy instances will have nearest neighbors, which are from the opposite class.

Copyright © 2020 NICE Actimize. All rights reserved.

So while finding the ambiguous examples on the class boundary is useful, it is not a great undersampling technique on its own. In practice, the Tomek Links procedure is often combined with other methods, such as the Condensed Nearest Neighbor Rule. The choice to combine Tomek Links and CNN is natural, as Tomek Links can be said to remove borderline and noisy instances, while CNN removes redundant instances.

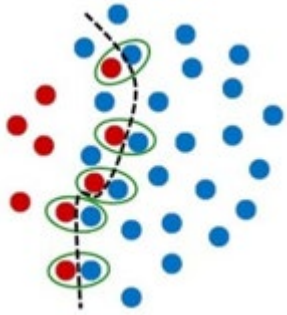

Figure 5: An illustrative example for Tomek link

Edited Nearest Neighbors Rules for Understanding (ENN)

Edited Nearest Neighbor or (ENN) is undersampling of the majority class is done by removing point whose class label differs from a majority of its K nearest neighbors.

Another rule for finding ambiguous and noisy examples in a dataset is called Edited Nearest Neighbors or ENN. This rule involves using *k=3* nearest neighbors to locate those examples in a dataset that are misclassified and that are then removed before a k=1 classification rule is applied. This approach of resampling and classification was proposed by Dennis Wilson in his 1972 paper [2].

The modified three-nearest neighbor rule is particularly attractive, as it uses the three-nearest neighbor rule to edit the pre-classified samples and then uses a single-nearest neighbor rule to make decisions.

When used as an undersampling procedure, the rule can be applied to each example in the majority class, allowing those examples that are misclassified as “minority class” to be removed, and those correctly classified to remain.

It is also applied to each example in the minority class, where misclassified examples have their nearest neighbors from the majority class deleted. For each instance “a” in the dataset, its three nearest neighbors are computed. If “a” is a majority class instance and is misclassified by its three nearest neighbors, then “a” is removed from the dataset. Alternatively, if “a” is a minority class instance and is misclassified by its three nearest neighbors, then the majority class instances among its neighbors are removed.

Repeated Edited Nearest Neighbor (R-ENN) is ENN algorithm that applied successfully until ENN can remove no further instances.

Copyright © 2020 NICE Actimize. All rights reserved.

Combinations of Keep and Delete Methods

In this section, we will take a closer look at previously reviewed techniques that both keep and delete examples from the majority class, such as One-Sided Selection and the Neighborhood Cleaning Rule.

One-sided Selection for Undersampling (OSS)

There are also other types of informed undersampling methods. For instance, the one-sided selection or (OSS) method selects a representative subset of the majority class E and combines it with the set of all minority examples S_{min} to form a preliminary set $N, N = \{E \cup S_{min}\}$. This set N is further refined by using a data cleaning technique – OSS resulting from the application of Tomek links followed by the application of CNN.

One-Sided Selection, or OSS for short, is an undersampling technique that combines Tomek Links and the Condensed Nearest Neighbor (CNN) Rule. Specifically, Tomek Links are ambiguous points on the class boundary and are identified and removed in the majority class. The CNN method is then used to remove redundant examples from the majority class that are far from the decision boundary. OSS is an undersampling method resulting from the application of Tomek links followed by the application of US-CNN. Tomek links, used as an undersampling method, remove noisy and borderline majority class examples. US-CNN aims to remove examples from the majority class that are distant from the decision border.

This combination of methods was proposed by Miroslav Kubat and Stan Matwin in their 1997 paper. The CNN procedure occurs in one-step and involves first adding all minority class examples to the store and some number of majority class examples (e.g. 1), then classifying all remaining majority class examples with KNN (*k=1*) and adding misclassified examples to the store.

Neighborhood Cleaning Rule for Undersampling (NCL)

The Neighborhood Cleaning rule or (NCL) method proposed is among the most popular undersampling methods. NCL is an undersampling technique that combines both the Condensed Nearest Neighbor (CNN) Rule to remove redundant examples and the Edited Nearest Neighbors (ENN) Rule to remove noisy or ambiguous examples.

NCL, a technique introduced by J. Laurikkala, balances imbalanced class distribution by performing data reduction. NCL proved its efficiency in identifying difficult small classes. The main advantage of NCL is that it considers the quality of the removed data by focusing on data cleaning more than data reduction. NCL is based on the concept of OSS, a data reduction technique used when class distribution is imbalanced. OSS applies instance-based methods to reduce the larger classes while the class of interest (the smaller class) is intact.

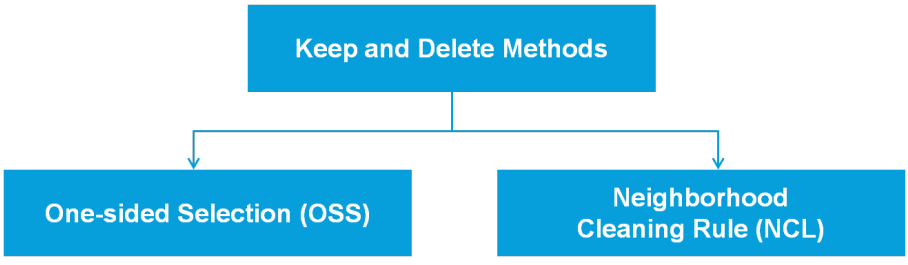

Figure 6: An illustration for combination of Keep & Delete Methods

Copyright © 2020 NICE Actimize. All rights reserved.

Summary

This article discussed most significant and well-known undersampling techniques to improve classification performance on the minority class in the presence of data imbalance. However, the methods discussed are by no means an exhaustive list. Several other techniques, which have been proposed in literature, have had success in handling data imbalance.

Undersampling has become the de facto strategy to deal with skewed distributions, but, though easy to be justified, it conceals two major effects: i) increasing the variance of the classifier and ii) producing warped posterior probabilities. The first effect is typically addressed using averaging strategies to reduce the variability. The second requires the calibration of the probability to the new priors of the testing set. Despite the popularity of undersampling for unbalanced classification tasks, it is not clear how these two effects interact and when undersampling leads to better accuracy in the classification task.

Citations

1. *Tomek, I. (1976). Two Modifications of CNN. IEEE Transactions on Systems, Man and Cybernetics, 6:769-772.*
2. *Dennis Wilson (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data*
3. *Miroslav Kubat and Stan Matwin (1997). Addressing the curse of Imbalanced. Training Sets – One-Sided Selection.*

Learn more about NICE Actimize Financial Crime Analytics research here.

ABOUT NICE ACTIMIZE

NICE Actimize is the largest and broadest provider of financial crime, risk and compliance solutions for regional and global financial institutions, as well as government regulators. Consistently ranked as number one in the space, NICE Actimize experts apply innovative technology to protect institutions and safeguard consumers and investors assets by identifying financial crime, preventing fraud and providing regulatory compliance. The company provides real-time, cross-channel fraud prevention, anti-money laundering detection, and trading surveillance solutions that address such concerns as payment fraud, cybercrime, sanctions monitoring, market abuse, customer due diligence and insider trading.

Copyright © 2020 Actimize Ltd. All rights reserved. No legal or accounting advice is provided hereunder and any discussion of regulatory compliance is purely illustrative.

NICE · ACTIMIZE