

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Classification Problem in Imbalanced Datasets

Aouatef Mahani and Ahmed Riad Baba Ali

Abstract

Classification is a data mining task. It aims to extract knowledge from large datasets. There are two kinds of classification. The first one is known as complete classification, and it is applied to balanced datasets. However, when it is applied to imbalanced ones, it is called partial classification or a problem of classification in imbalanced datasets, which is a fundamental problem in machine learning, and it has received much attention. Considering the importance of this issue, a large amount of techniques have been proposed trying to address this problem. These proposals can be divided into three levels: the algorithm level, the data level, and the hybrid level. In this chapter, we will present the classification problem in imbalanced datasets, its domains of application, its appropriate measures of performances, and its approaches and techniques.

Keywords: classification, imbalanced datasets, sampling, data mining, classifier

1. Introduction

Classification is the most popular task of data mining. It consists of assigning to each instance a class chosen from a set of predefined classes, according to the value of certain predictive attributes [1]. Its problem is to correctly classify an instance with indeterminate class. This classification can be done by several methods that are divided into two categories. The first category is based on the use of a model or a classifier such as decision trees and classification rules. However, the second category is based on the internal functioning of the learning algorithm such as neural networks [2] and support vector machines (SVMs). All these methods use large datasets to extract knowledge.

The used datasets are organized in the form of tables. The tables' columns are called the attributes, and they represent the characteristics of the dataset. Traditionally, the last attribute is called a class attribute. The tables' rows represent the data, and they are called instances. The number of instances varies from one class to another. So, the number of instances of one class is larger than that of the second class in some existing datasets. Therefore, datasets are divided into two categories: balanced and imbalanced datasets. In the latter, instances are divided into two sets: majority instances which are the most frequent and minority instances which are the less frequent.

Rule-based classification algorithms have a bias toward majority classes [3]. They tend to discover the rules with high values of accuracy and coverage. These rules are usually specific to majority instances, whereas specific rules that predict

minority instances are usually ignored or treated as noise. Consequently, minority instances are often misclassified. Generally, because most classifiers are designed to minimize the global error rate [4], many problems occur. First, they perform poorly on imbalanced datasets, and they either produce general rules or very specific ones. In the first case, the classifier has a bias toward majority instances, and it ignores the minority ones. In the second case, the classifiers tend to overfit the training data which provokes poor classification accuracy on unseen data. Next, the cost of misclassifying a minority instance is usually more expensive than misclassifying a majority one [5, 6]. Finally, in many applications misclassifying a rare event can result in more serious problems than a common event [7]. For example, in case of cancerous cell detection in medical diagnosis, misclassifying non-cancerous cells may lead to some additional clinical tests, but misclassifying cancerous cells leads to very serious health risks.

The class imbalance problem is a fundamental problem in machine learning, and it has received much attention [8–14]. This problem is known as partial classification [15], nugget discovery [16], classification problem with imbalanced datasets [17], or datasets with rare classes [18]. Considering the importance of this issue, a large amount of techniques have been developed trying to address this problem. These proposals can be divided into three groups which depend on how they deal with class imbalance. First, the algorithm-level approaches can either propose specific algorithms or modify the existing ones. Second, the data-level techniques introduce an additional processing step to decrease the effect of skewed class distribution such as undersampling and oversampling methods. Finally, the hybrid-level methods combine algorithm level and data level such as boosting and cost-sensitive learning.

This chapter is organized as follows. Section 2 presents the classification problem in imbalanced datasets. In Section 3, we present some domains in which the datasets appear. In Section 4, we present the evaluation metrics used in classification problem in imbalanced datasets. In Section 5, we detail the different approaches and techniques used to handle classification in imbalanced datasets. Finally, in Section 6, we make our concluding remarks.

2. Presentation of the classification problem

In the binary imbalanced datasets, the number of instances of one class is higher than that of the second class. Consequently, the first class is known as majority class and the second class as minority one. Therefore, this dataset contains two kinds of instances: majority and minority.

The distribution of instances in imbalanced binary datasets is measured by the imbalanced ratio (IR) [19] which is defined in Eq. (1):

$$IR = \frac{\text{Number of majority instances}}{\text{Number of minority instances}} \quad (1)$$

According to the value of IR, the imbalanced datasets are divided into three classes [20]: datasets with low imbalance (IR is between 1.5 and 3), datasets with medium imbalance (IR is between 3 and 9), and datasets with high imbalance (IR is higher than 9).

3. Application domains

The imbalanced datasets appear in the following several domains.

3.1 Risk management

Every year, the telecommunication industry suffers billions of dollars in unrecoverable debts. Therefore, uncollectible control is a major problem in the industry. One solution is to use large amounts of historical data to build models that are used to assess risk for each customer client or for each transaction to support risk management that reduces the level of unrecoverable debt. However, in a dataset, nonpayment of customers includes a few percent of the population [21].

3.2 Medical diagnosis

Clinical datasets store large amounts of patient information. Data mining technique is applied on these datasets to uncover the relationships and trends between clinical and pathological data. It aims to understand the evolution and characteristics of certain diseases. However, in these datasets, cases of disease are rarer than the normal population [22].

3.3 Intrusion detection in networks

Network-based computer systems are increasingly playing a vital role in modern societies. Attacks on computer systems and networks are growing. Different categories of network attacks exist; some are numerous, and others are rare. For example, the KDD-CUP'99 dataset contains four categories of network attacks: denial of service (DoS), monitoring (probe), root to local (R2L), and user to root. (U2R). The last two attacks are intrinsically rare [23].

4. Evaluation metrics

The classical performance measures used for evaluating the performances of classifiers when used with balanced datasets are not appropriate for imbalanced datasets. This is because they have a strong bias toward majority class and are sensitive to class skews [24–27]. For example, the accuracy measure is not appropriate for the problem of imbalanced datasets [28]. If we consider a dataset which contains only 1% of minority instances and 99% of majority instances, the accuracy is 99% if all majority instances are well classified. However, misclassified 1% minority instances may lead to an enormous cost, and 99% accuracy could be a disaster for a medical diagnosis. Consequently, other metrics are necessary for measuring the performances of classifiers.

Some measures are extracted directly from the confusion matrix. They measure the classification performance of the majority and minority classes independently. Some others are combined to measure the performance of a classifier. They are described below.

4.1 Precision

It is a measure of accuracy [29]. It represents the percentage of well-classified minority instances in relation to all instances whose predicted class is a minority. It is defined in Eq. (2):

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

4.2 Recall

It is the percentage of minority instances which are well classified as belonging to the minority class. In literature, this metric has several names such as sensitivity, true positive rate (TPrate), or positive accuracy [30]. It is defined in Eq. (3):

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4.3 Specificity

It is the percentage of majority instances which are well classified as belonging to the majority class. This measure is also known as true negative rate (TNrate) or negative accuracy. It is defined in Eq. (4):

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

4.4 False-positive rate (FPrate)

It is the percentage of majority instances misclassified as belonging to the minority class. It is defined in Eq. (5):

$$FPrate = \frac{FP}{FP + TN} \quad (5)$$

4.5 False-negative rate (FNrate)

It is the percentage of minority instances misclassified as belonging to the majority class. It is defined in Eq. (6):

$$FNrate = \frac{FN}{FN + TP} \quad (6)$$

4.6 G-mean

It indicates the balance between classification performances on the majority and minority classes [30]. A poor performance in the prediction of the positive instances will lead to a low G-mean value even if the negative instances are correctly classified by the model [31]. It has been used by several researchers for evaluating classifiers on imbalanced datasets [31–33]. G-Mean takes recall and specificity into account simultaneously. It is defined in Eq. (7). This metric will be used to test our approach:

$$G - \text{Mean} = \sqrt{Recall * Specificity} \quad (7)$$

4.7 F-measure

It is defined as the harmonic mean of precision and recall [34]. Its value increases proportionally with the increase of precision and recall; a high value of F-measure indicates that the model performs better on the minority class. This metric is defined in Eq. (8):

$$F - \text{Measure} = \frac{2 * Recall * Precision}{Recall + Precision} \quad (8)$$

4.8 Receiver operating characteristic curve (ROC)

The ROC curve [34, 35] is a technique for visualization, organization, and selection of classifiers based on their performances. It has long been used in signal detection to represent the trade-off between the success rate and false alarm rate of classifiers. It is a two-dimensional graph where TPrate is plotted on the y-axis and FPrate is plotted on the x-axis.

For a discrete classifier, the pair (FPrate, TPrate) is produced that corresponds to one point in the ROC space. However, a probabilistic classifier produces a continuous numerical value. Therefore, a threshold may be used to produce a series of points in the ROC space to produce a curve instead of one point.

4.9 Area under the ROC curve (AUC)

From the ROC curve, we define another measure called area under the curve (AUC) [35, 36] defined in Eq. (9) to compare the performance of two classifiers. If the area associated with classifier C1 is greater than that associated with classifier C2, then the performances of C1 are better than C2:

$$AUC = \frac{TPrate + TNrate}{2} = \frac{1 + TPrate - FPrate}{2} \quad (9)$$

5. Approaches and techniques

The several approaches have been proposed to handle the classification problem in imbalanced datasets. These approaches are divided into three levels [20]: data level, algorithm level, and hybrid level.

5.1 Data level methods

It consists of resampling the data in order to decrease the effect caused by the imbalance [3]. They are classified into three groups [3]: oversampling, undersampling, and hybrid methods.

5.1.1 Oversampling methods

Oversampling is used to increase the size of an imbalanced dataset by duplicating some minority instances. This duplication can be done by the following methods.

5.1.1.1 Random oversampling

It duplicates some minority instances chosen randomly [3]. Therefore, the multiple copies of minority instances increase the overlapping between these instances [37]. In particular, the overlapping appears when the produced classifier contains more specific rules for multiple copies of the same instance. As a result, the accuracy of learning is high in this scenario, and the performance of the classifier for the test is generally low [38].

5.1.1.2 Synthetic minority oversampling technique (SMOTE)

SMOTE [39] is a synthetic method with data generation. It has achieved several successes in various fields [3]. It creates a synthetic example x_{new} for each minority

instance x_i as follows. It determines the K -nearest neighbors (which are minority instances whose Euclidean distance between them and x_i is the smallest) of x_i . Then, it selects randomly one of K -nearest neighbors y_i . Finally, it applies Eq. (10), where δ is a random number $\in [0, 1]$. Therefore, we understand that x_{new} is a point of the segment joining x_i and y_i :

$$x_{new} = x_i + (y_i - x_i) * \delta \quad (10)$$

SMOTE will not ignore the minority instances because it generalizes decision regions for them. But SMOTE has two problems [40]: overgeneralization and variance. The first problem is due to the blind generalization of the minority area without taking into account the majority class, which increases the number of overlapping between classes. The second problem concerns the number of generated synthetic instances which is set in advance without taking into account the IR.

5.1.1.3 MSMOTE

SMOTE does not consider the distribution of minority instances and those that are noisy in a dataset. For this reason, MSMOTE [41] divides the minority instances into three groups: security, border, and latent noises.

An instance is security, if the number of its K -nearest neighbors belonging to the minority class is greater than those belonging to the majority class.

An instance is border, if the number of its K -nearest neighbors belonging to the minority class is lower than those belonging to the majority class.

An instance is latent noise, if all its K -nearest neighbors have the majority class.

MSMOTE generates synthetic instances for all security instances in the same way as SMOTE. However, for each border instance, it selects the most nearest neighbor to generate a synthetic example. But, it does not generate synthetic instances for noisy instances, because they decrease the classifier's performances.

5.1.1.4 Borderline-SMOTE

Border instances and those nearby are more likely to be misclassified than those that are far from the border, and they are the most important for classification. Based on this analysis, the border instances contribute little in the classification. Therefore, the Borderline-SMOTE [42] method has been proposed to apply oversampling to border minority instances instead of applying it to all minority instances. To do this, it constructs a set of border minority instances known as DANGER. Then, it applies SMOTE for each instance of the DANGER set.

5.1.1.5 Adaptive synthetic sampling approach (ADASYN)

ADASYN [43] uses a function called density as an automatic criterion to take a decision about the number of synthetic instances that may be generated of each minority instance.

5.1.2 Undersampling methods

It consists of reducing the data size by deleting some majority instances with the objective of equalizing the number of instances of each class [44]. There are several approaches of undersampling that differ in the way of selection of majority instances that will be deleted.

5.1.2.1 Random undersampling (RUS)

RUS [4, 44] removes some majority instances selected randomly. But it can potentially hinder learning [37, 38, 45]; the deleted majority instances can cause the classifier to ignore important concepts related to the majority class.

5.1.2.2 Informed undersampling

It is proposed to avoid the loss of information caused by RUS [46]. Among the algorithms of this kind of undersampling, we have the following.

5.1.2.2.1 EasyEnsemble

It aims to a better exploitation of majority instances ignored by RUS. At first, it divides the training dataset into minority set P and majority set N of sizes n and p, respectively [46]. Then, it builds T subsets $N_1, N_2 \dots, N_T$ of size p by applying random sampling with replacement on N. After that, it generates T classifiers $H_1, H_2 \dots, H_T$. The classifier H_i is produced by applying AdaBoost on N_i and P, and it contains the concepts of all majority and minority instances. Finally, it constructs the final classifier H by combining the T generated classifiers.

5.1.2.2.2 BalanceCascade

The training dataset is composed of the sets P of minority instances of size p and N of majority instances of size n [46]. BalanceCascade constructs at each iteration the classifier H_i from all the set P and the subset E chosen randomly from N, with $|E| = p$. Then, it updates N by deleting all majority instances which are well classified by H_i . This algorithm explores the majority instances in a supervised way because the set of majority instances is updated after generation of each classifier.

5.1.2.2.3 Informed undersampling with KNN

This technique [44] is based on the distribution characteristics of data by applying KNN algorithm [47]. The following three methods of this technique have been proposed:

NearMiss-1 selects majority instances as follows:

- For each majority instance x_i .
- For each minority instance x_j : Computes the distance d_{ij} between x_i and x_j .
- Identify the three nearest neighbors x_k ($1 \leq k \leq 3$) for x_i that represents minority instances.
- Compute the average distance d_i defined in Eq. (11):

$$d_i = \frac{1}{3} \sum_{k=1}^3 d_{ik} \quad (11)$$

- Select majority instances x_i whose average distance to the three closest minority class instances is the smallest.

NearMiss-2 method has the same steps as the previous method. But, it selects the majority instances whose average distance to the three farthest minority class instances is the smallest.

NearMiss-3 selects a given number of the closest majority instances for each minority instance to guarantee that every minority instance is surrounded by some majority instances.

5.1.2.3 Undersampling with data cleaning techniques

The data cleaning techniques were applied to eliminate the overlapping between classes. In the following four subsections, we represent some methods.

5.1.2.3.1 Tomek links

Tomek links method [48] may be used as an undersampling method. It deletes the noisy majority instances and those that are close to the border. The obtained training dataset after removing the Tomek links is organized into set of clusters. This method may be used as a data cleaning technique to delete majority and minority instances.

5.1.2.3.2 Condensed nearest neighbor (CNN) rule

CNN [49] is an instance reduction algorithm proposed by Hart. It deletes the redundant majority instances. An instance is considered as redundant if it can be deduced from other instances. CNN uses the initial training dataset E to construct the consistent dataset E' that contains instances that correctly classify all instances of E using 1-NN algorithm. Its steps are:

1. Copy the first majority instance x and all the minority instances of the training dataset E into the sub dataset E' .
2. While there are misclassified instances in E , do:
 - a. Classify the instance y (belonging to E) using E' and 1-NN.
 - b. Add y to E' , if it is misclassified.

CNN is sensitive to noise. However, noisy instances are more susceptible to be misclassified [50], and they will misclassify the instances of test dataset [50, 51].

5.1.2.3.3 Neighborhood cleaning rule (NCL)

NCL is an undersampling technique introduced by Laurikkala [52] to balance a dataset by applying data reduction. Its main advantage is that it takes into account the quality of the data with a focus on data cleaning more than reduction. It removes noisy majority instances using the edited nearest neighbor (ENN) algorithm, which is an instance reduction algorithm developed by Wilson [53]. It is used to delete all instances whose class differs at least twice from the class of its three nearest neighbors.

5.1.2.3.4 One-sided sampling (OSS)

OSS [17] is the result of using CNN followed by Tomek links. CNN is applied to remove redundant majority instances. However, Tomek links deletes the noisy majority instances and border minority instances.

5.1.2.4 Evolutionary undersampling (EUS)

EUS [20] results from the application of prototype selection [54] and genetic algorithm. It has eight models that depend on the objective that EUS aims to reach. For the first objective, there are two purposes. The first one is to balance a dataset without losing the accuracy, and then EUS is known as evolutionary balancing undersampling (EBUS). In the second one, EUS aims to obtain an optimal power of classification without taking in the consideration the balance of a dataset; it is called evolutionary undersampling guided by classification measures (EUSCM). For the second objective, there are two possibilities: majority selection (MS) instances only or global selection (GS) of both majority and minority instances.

5.1.3 Hybrid methods

These methods combine undersampling and oversampling. They aim to eliminate the overfitting [3] caused by oversampling methods. For examples, SMOTE+Tomek links [17] applies Tomek links after generation of synthetic minority instances by SMOTE, and SMOTE+ENN [17] uses ENN to delete minority and majority instances. For this, each misclassified instance of training dataset by its three nearest neighbors is deleted.

5.2 Algorithm level

Most approaches are based on either modifying the existing complete classification algorithms in order to adapt them to the imbalanced datasets or proposing specific ones.

5.2.1 Modification of the existing algorithms

5.2.1.1 Decision trees

A decision tree [55–58] is the most popular form of rule-based classifiers. It allows to model simply, graphically, and quickly a phenomenon more or less complex. Its readability, speed of execution, and the few necessary hypotheses a priori explain its current popularity. All the methods of constructing a decision tree have these operators: deciding if a node is terminal, selection of a test to associate to a node, and assignation a class to a leaf.

The existing methods of construction the decision trees differ by the choices made for different operators. CART [59] and C4.5 [60] are the most popular algorithms for decision trees.

In the construction phase of a tree, C4.5 selects the node attribute that maximizes the information gain [60], that is, a high value of confidence. However, this measure is not suitable for imbalanced datasets because the most confident rules do not imply that they are the most significant, and some of the most significant rules may not be the most confident (may not have high confidence). The same problem arises for CART, which uses the Gini function [60]. These algorithms focus on the

antecedent to find the class. Also, they use sensitive measures to the class distribution. For these reasons, some approaches have been proposed which apply nonsensitive measures [61] or modify the construction phase.

For example, *class confidence proportion decision tree* (CCPDT) approach is a robust and insensitive approach. It generates rules that are statistically significant [62]. It focuses on each class to find the most significant antecedent. In this way, all instances are partitioned according to their classes. Therefore, the instances that belong to the different classes will not have an impact on the others. For this, the new class confidence (CC) measure has been proposed to find the most interesting antecedents of each class. It is defined in Eq. (12):

$$CC(X \rightarrow y) = \frac{Supp(X \cup y)}{Supp(y)} = \frac{TP}{TP + FN} \quad (12)$$

However, obtaining rules which have a high CC value is still insufficient to solve the problem. So, it is necessary to make sure that the classes implied by these rules not only have great confidence but are more interesting than their alternative classes. As a result, the new class confidence proportion (CCP) measure has been proposed. It is defined in Eq. (13):

$$CCP(X \rightarrow y) = \frac{CC(X \rightarrow y)}{CC(X \rightarrow y) + CC(X \rightarrow \bar{y})} \quad (13)$$

Therefore, the CCPDT approach modifies the C4.5 algorithm by replacing the entropy (the attribute partition criterion) by CCP.

5.2.1.2 Support vector machines (SVMs)

The Kernel-based learning methods are inspired by the statistical theory of learning and the dimensions of Vapnik-Chervonenkis (VC) [63], such as support vector machines (SVMs). These latter are supervised learning methods. They are used for classification in binary datasets in order to find a classifier that separates the data and maximizes the distance between these two classes. This classifier is linear, and it is called hyperplane. SVMs aim to find the most optimal hyperplane, which passes in the middle of the points of two classes and that maximizes the margin in order to minimize the classification error [64].

In imbalanced datasets, the ideal hyperplane is close to the majority instances, and the decision boundary is very close to the minority instances. In this case, the support vectors representing the minority instances are far from the ideal hyperplane. Thus, their contribution to the final hypothesis is little [32, 65, 66]. To solve this problem, several methods have been proposed that differ from the mechanism used. For example, in [67] the following three approaches were presented:

- Boundary movement (BM) that modifies the coefficient b in the kernel function.
- Biased penalties (BP) introduce different penalty factors for the minority and majority classes in objective function in the Lagrangian formulation. These factors reflect the importance of classes during the learning phase.
- Border class alignment (BCA) expands the border around the minority class much more than the border around the majority class.

5.2.2 Specific algorithms

The specific algorithms have been proposed to deal with classification problem in imbalanced datasets. Among them, we present RLSD and LUPC.

5.2.2.1 Rule learning for skewed datasets (RLSD)

RLSD [62] is an efficient algorithm for handling imbalanced datasets. Its discovery process leads from a specific search to a general search. First, it discovers rules for minority instances used in learning. Then, it compares them with majority instances. It has the following three research phases:

1. Discretization phase consists of dividing the values of a numerical attribute into a small number of intervals. Each interval is mapped by a discrete symbol.
2. Rule generation phase is a frequent data discovery process for minority instances. This algorithm summarizes this phase:

Input: the set of minority instances P and the maximum number of allowed rules

Output: the set of rules: Rules.

Begin

1. Initially, the rule set is empty : Rules := \emptyset ;
2. for each minority instance $p \in P$ do
 - 2.1. The rule $R := p$;
 - 2.2. Consider R as an initial rule;
 - 2.3. If R does not belong to Rules then
 - 2.3.1. Merge R :
For each rule $RE \in$ Rules:
If R and RE have common conditions then generate the new rule NR with common conditions and apply the procedure Add_Rules to add NR to Rules.
 - 2.3.2. Apply the procedure Add_Rules to add R to Rules.
3. for each rule $RE \in$ Rules do if $TPrate(RE) < Min_TPrate$ then delete RE

End.

The procedure Add_Rules adds the concerned rule R to the set of rules if it does not belong to this set. After that, if the number of rules exceeds M , then it deletes a rule selected randomly.

3. The evaluation and rule selection phase: in rule evaluation step, RLSD calculates the accuracy of each generated rule by the correspondence with each majority instance. A rule is deleted if its precision (defined in Eq. (3)) is less than the minimum precision. In rule selection step, RLSD selects the rule with the highest F-measure value (defined in Eq. (8)). Then, it deletes all minority instances covered by this rule. After that, F-measure is recalculated for the remaining rules using the rest of minority instances. This process is repeated until there are no more minority instances or there is no rule that covers the remaining minority instances.

5.2.2.2 Learning minority classes in unbalanced datasets (LUPC)

The main feature of LUPC [67] is the combination of the separate and conquer rule induction method [68] and the association rules [69].

It lets an imbalanced dataset of size D composed of the set of positive (minority) instances Pos and the set of negative (majority) instances Neg .

LUPC uses three measures of performances: accuracy [70] (acc), error rate (err) [64], and positive cover ratio (PCR). They are defined in Eqs. (14)–(16):

$$acc(R) = \frac{|Cov^+(R)|}{|Cov(R)|} \quad (14)$$

$$err(R) = 1 - acc(R) \quad (15)$$

$$PCR(R) = \frac{|Cov^+(R)|}{|D|} \quad (16)$$

where the coverage (Cov) [70] of the rule R is the percentage of instances that are covered by this rule. It is defined in Eq. (17):

$$Cov(R) = \frac{\text{number of covered instances}}{|D|} \quad (17)$$

$Cov^+(R)$ is the number of covered instances that have the same class as that of R .

A rule is $\alpha\beta$ -strong if the conditions given in Eqs. (18) and (19) are checked, where parameters α and β are the thresholds, with $0 \leq \alpha$ and $\beta \leq 1$.

A rule is $non\text{-}\alpha\beta$ -forte if the condition given in Eq. (20) is checked:

$$acc(R) \geq \alpha \quad (18)$$

$$PCR(R) \geq \beta \quad (19)$$

$$Cov^-(R) \geq \frac{1 - \alpha}{\alpha} * Cov^+(R) \quad (20)$$

The steps of LUPC are:

Input: The sets Pos and Neg , the minimum threshold of accuracy : min_acc and the minimum PCR: min_cov

Output: The set of rules: $Rules_sets$.

Begin

$Rules_sets := \emptyset$;

$\alpha, \beta := \text{Initialize}(Pos, min_acc, min_cov)$;

while ($Pos \neq \emptyset$ and $(\alpha, \beta) \neq (min_acc, min_cov)$)

 Rule $R := \text{Best rule}(Pos, Neg, \alpha, \beta)$;

 If ($R \neq \emptyset$) then

$Pos := Pos - \{\text{instances covered by } R\}$

$Rules_sets := Rules_sets \cup R$;

 else

$Reduce(\alpha, \beta)$;

$Rules_sets := \text{Post traitement}(Rules_sets)$; // It is optional

end.

The procedure “Initialize” depends on the user-specified bias on PCR or accuracy. It initializes α and β as min_acc and min_cov , respectively. Otherwise, α is initialized to 0.95 or min_acc if min_acc is greater than 0.95, and β is initialized to the maximum value of PCR of the attribute-value pairs available on the minority instances. To find the best rule, LUPC follows these steps:

1. Construct the set of attribute-value pairs: build the set $E1$ of all pairs, where each available pair (attribute, value) for positive instances is considered as a

part of condition whose class is C^+ . Choose the set $E2$ (is a subset of $E1$) of the candidate pairs: each pair belonging to $E1$ is considered as a candidate pair if it covers more than $\alpha * \beta * |D|$ minority instances. Identify the $\alpha\beta$ -strong pairs: each candidate pair will be checked on the Neg instances to see if it is $\alpha\beta$ -strong. Order the $\alpha\beta$ -strong pairs either by their precision or by their PCR. Choose η attribute-value pair candidates which are $\alpha\beta$ -strong, and add them to the set of attribute-value pairs. In the case where the number of $\alpha\beta$ -strong pairs $< \eta$, then add the pairs that are not $\alpha\beta$ -strong and that have either a high accuracy or a high PCR.

2. Generate the set of candidate rules that contains γ rules belonging to the set of attribute-value pairs as follows:

- Order all attribute-value pairs according to the accuracy and/or PCR.
- If the number of $\alpha\beta$ -strong pairs is greater than or equal to γ , then add γ pairs to the set of candidate rules otherwise:
 - First, put the $\alpha\beta$ -strong pairs and the non- $\alpha\beta$ -strong pairs in the set of candidate rules. Then, delete non- $\alpha\beta$ -strong rules whose PCR is lower than β . After that, improve the set of candidate rules by iteratively executing the following procedure:
 1. Generate new rules by combining each non- $\alpha\beta$ -strong rule of the set of candidate rules with the pairs which are in the set of attribute-value pairs.
 2. If the generated rules become $\alpha\beta$ -strong, then they will be inserted in the first part of the set of candidate rules.
 3. The procedure stops if there is no change in the non- $\alpha\beta$ -strong rules or the number of rules in the set of candidate rules is greater than γ .
 4. Reject the rules that satisfy the condition given in Eq. (20).

The values of α and β are gradually reduced by the rate Δa and Δc , respectively. The default quantities used are $\Delta a = 2\%$ and $\Delta c = 1\%$.

5.3 Hybrid level

Some methods of the complete classification cannot deal with the classification in imbalanced datasets without being combined with other techniques. Among these methods, we present ensemble methods, the cost-sensitive learning, and some other approaches based on metheuristics.

5.3.1 Ensemble methods

Ensemble methods build a series of N classifiers and combine them to produce the final classifier C^* using voting strategies. They aim to obtain a high precision classifier. They are divided into two classes: boosting and bagging.

5.3.1.1 Boosting

Boosting algorithms [64] focus on difficult instances to classify without differentiating their classes. According to Prof. Zhou Zhi-Hua [71], the boosting algorithms are very efficient and able to deal with classification in imbalanced datasets, because the minority instances are likely to be misclassified, and therefore, they will have weights higher in the following iterations.

However, Mikel G. et al. [72] considered that the integration of data sampling methods can reduce the additional costs of automatically detecting the optimal distribution of representative classes and samples and also reduce the bias of a specific learning algorithm. Among these methods, we have SMOTEBoost, RUSBoost, and DataBoost-IM.

5.3.1.1.1 SMOTEBoost

It alters the distribution of the training dataset by adding minority instances generated by SMOTE [73] in order to provide it to the algorithm AdaBoost.M2 [74].

5.3.1.1.2 RUSBoost

It operates in a similar way to SMOTEBoost, but it applies random undersampling on the training dataset [75].

5.3.1.1.3 DataBoost-IM

It combines AdaBoost.M1 [76] with a data generation strategy [31]. It differs from the two previous algorithms, because it performs the balancing process for majority and minority instances after identifying the difficult instances. Its steps are as follows:

1. Produce the classifier C to detect the misclassified instances which are called seeds.
2. Order the seeds in ascending order of their weight.
3. Construct the sets MAJ and MIN, which contain M_j majority instances and M_m minority instances, respectively, that have the highest weights.
4. Generate N synthetic instances for each majority seed and M synthetic instances for each minority seed.
5. Update the weights taking into consideration the newly added synthetic instances.

5.3.1.2 Bagging

It constructs N classifiers on N distinct datasets [77]. Each dataset is known as bag; it is obtained by random sampling with replacement.

In imbalanced datasets, the number of majority instances in a bag is also high. The main factor to apply on bagging to adapt it to this kind of datasets is the way of collecting the instances. We distinguish three main algorithms in this family:

5.3.1.2.1 OverBagging

The distribution of instances may be taken into consideration in order to equalize the number of minority instances N_{\min} and the number of majority instances N_{maj} [78]. Instead of constructing the bags randomly, we apply the oversampling according to the following two possibilities. In the first one, the minority instances are duplicated by oversampling, and majority instances are added directly, or they are selected by random sampling with replacement to increase the diversity. In the second one, the SMOTEBagging [78] is applied, where $A\% \cdot N_{\text{maj}}$ minority instances are selected by random drawing with replacement and the remaining instances are generated by SMOTE. The factor A is called resampling rate. It is equal to 10% in the first iteration and 100% in the last (it is multiple of 10).

5.3.1.2.2 UnderBagging

The number of majority instances is reduced to the number of minority instances in each bag [79]. All minority instances may be in the same bag. But for increasing the diversity, they can be selected by random sampling with replacement.

5.3.1.2.3 UnderOverBagging

It follows the two previous methodologies, but it is identical to SMOTEBagging [79].

5.3.2 Cost-sensitive learning methods

Most classification algorithms ignore various misclassification errors and consider that all these errors have the same cost. In many real-world applications, this hypothesis is not true because the difference between different classification errors can be quite large.

Cost-sensitive learning methods [80, 81] have been given a lot of attention in recent years to address this problem. They have been divided into two categories: direct cost-sensitive learning [80] and cost-sensitive meta learning [80, 82]. They are also used for imbalanced datasets such as boosting and SVMs.

5.3.2.1 Cost-sensitive learning with boosting

In each iteration of boosting, the weights of misclassified instances increase by the same ratio whatever their classes. However, in imbalanced datasets, the number of misclassified minority instances is higher. Hence, it is necessary to distinguish between different sorts of instances in the weight attribution phase. Therefore, the higher weights may be attributed to the minority instances in order to be well classified. To achieve this goal, misclassification costs are introduced into the weight update equation. Among these algorithms [83, 84], we have AdaC1, AdaC2, and AdaC3. They differ in the way of introducing the misclassification costs into the weight update formula within the exponential part and into the equation of the calculation of classifier performance.

5.3.2.2 Cost-sensitive learning with support vector machines (SVMs)

SVMs [63] have been integrated with sampling methods to deal with the classification problem in imbalanced datasets. Among these methods, we have the following.

5.3.2.2.1 SMOTE with different costs (SDC)

SDC results from the application of SMOTE with different error costs (DEC) [32]. This method aims to shift the decision boundary far from minority instances and to increase their number. To achieve the first objective, Veropoulos et al. [85] proposed the use of different costs for the minority and majority classes. The minority instances are also duplicated by SMOTE to make them densely distributed in order to guarantee the most well-defined boundary.

5.3.2.2.2 Ensembles of over-/undersampling SVMs

These methods [86] balance the training dataset by preprocessing and providing it to SVM for building an optimal classifier. For instance, the ensemble of undersampling SVMs (EUS-SVM) applies SVM, N times on N different training datasets. It contains all minority instances and some majority instances selected by random sampling. The final classifier is built by the combination of N produced classifiers.

5.3.3 Approaches based on metheuristics

5.3.3.1 Undersampling by genetic algorithm (USGA)

This approach [87] applies an intelligent method to the extraction of the classification rules from imbalanced binary datasets based on three phases.

In phase 1, a learning algorithm is developed based on a genetic algorithm with the aim of extracting the first classifier noted C_1 , which covers only majority instances when available. Majority instances, which are well classified by the rules of C_1 , are removed. This approach balances the imbalanced dataset and prevents the loss of information contained in the deleted majority instances, which are replaced by the classification rules of C_1 . The number of deleted majority instances depends on the value of IR ; the process is carried out until the IR is equal to 1. The genetic algorithm is used to find the “best” rule for the majority class from the imbalanced dataset. The quality of each rule is evaluated by satisfying specificity (defined in Eq. (4)). A rule is the best if it has a high number of well-classified majority instances.

In phase 2, the same procedure is applied to the obtained balanced dataset using a fivefold cross-validation to construct the classifier C_2 , which contains rules that represent both majority and minority instances. In this phase, the quality of each rule is evaluated by maximizing the accuracy (Eq. (14)).

In the third phase, they merge C_1 and C_2 to produce the classifier C_3 at first, and then they process the obtained classifier C_3 by eliminating the specific and contradictory rules.

5.3.3.2 ACOSampling

ACOSampling [13] is a method of undersampling based on ant colony optimization [88]. It handles imbalanced DNA microarray datasets. It consists of extracting the balanced dataset S' for the original dataset S as follows:

- For T times, it divides the dataset S into two datasets, training and validation.
- Each training dataset S_i is processed by applying modified ACO algorithm [88] to filter less informative majority instances and search the corresponding optimal training dataset S'_i .

- The statistical results from the T optimal training datasets are given in the form of a list of frequencies, where each frequency indicates the importance of the corresponding majority instance. The extracted instances are those with high frequency, which will be combined with all minority instances to construct the final balanced training dataset S'.
- It produces one classifier by support vector machines using S'.

6. Conclusion

In this chapter, we have presented the classification problem in imbalanced datasets, which are composed of two kinds of instances: majority instances and minority ones. We have also presented the different approaches and techniques used to handle this problem which are divided in three levels: data level, algorithm level, and hybrid level.


In future work, we are planning to present a state of the art about different approaches and techniques used to handle the classification problem in multi-class imbalanced datasets. Also, we will extend our proposed approach to this kind of datasets.

Author details

Aouatef Mahani* and Ahmed Riad Baba Ali
LRPE, FEI, University of Science and Technology of Algiers Houari Boumediene
USTHB, Algiers, Algeria

*Address all correspondence to: mahani.aouatef@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Parpinelli RS, Lopes HS, Freitas AA. An ant colony based system for data mining: applications to medical data. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01); 7-11 July 2001; San Francisco, California; 2001. pp. 791-797
- [2] Lu H, Setiono R, Liu H. Effective data mining using neural network. IEEE Transactions on Knowledge and Data Engineering. 1996;**86**:957-961. DOI: 10.1109/69.553163
- [3] Batista G, Prati RC, Monard MC. A study of the behaviour of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter. 2004;**61**:20-29. DOI: 10.1145/1007730.1007735
- [4] Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intelligent Data Analysis. 2002;**65**:429-449. DOI: 10.3233/IDA-2002-6504
- [5] Japkowicz N, Holte RC, Ling CX, Matwin S. Learning from Imbalanced Data Sets Workshop (ICML'2003). Washington, DC; 2003
- [6] Weiss GM, Provost F. Learning when training data are costly: The effect of class distribution on tree induction. Artificial Intelligence Research archive. 2003;**191**:315-354. DOI: 10.1613/jair.1199
- [7] Tang Y, Zhang Y, Chawla NV, Krasser S. SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2009; **391**:281-288. DOI: 10.1109/TSMCB.2008.2002909
- [8] Alejo R, García V, Sotoca JM, Mollineda RA, Sánchez JS. Improving the performance of the RBF neural networks trained with imbalanced samples. Lecture Notes in Computer Science. 2007;**4507**:162-169. DOI: 10.1007/978-3-540-73007-1_20
- [9] Fu X, Wang L, Chua KS, Chu F. Training rbf neural networks on unbalanced data. In: Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), 18-22 November 2002; Singapore. Singapore: IEEE Xplore; 2003. pp. 1016-1020
- [10] Murphey YL, Wang H, Ou G, Feldkamp LA. OAHO: An effective algorithm for multi-class learning from imbalanced data. In: IEEE International Joint Conference on Neural Networks (IJCNN 2007); August 12-17, 2007; Renaissance Orlando Resort. 2007. pp. 406-411
- [11] Qiong G, Xian-Ming W, Zhao W, Bing N, Chun-Sheng X. An improved smote algorithm based on genetic algorithm for imbalanced data classification. Digital Information Management. 2016;**142**:92-103
- [12] Yoon K, Kwek S. A data reduction approach for resolving the imbalanced data issue in functional genomics. Neural Computing and Applications. 2007;**16**:295-306. DOI: 10.1007/s00521-007-0089-7
- [13] Yu H, Ni J, Zhao J. ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. Neurocomputing. 2013;**101**:309-318. DOI: 10.1016/j.neucom.2012.08.018
- [14] Zhou ZH, Liu XY. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering. 2006;**181**:63-77. DOI: 10.1109/TKDE.2006.17
- [15] Ali K, Manganaris S, Srikanth R. Partial classification using association

- rules. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97); 14-17 August 1997; Newport Beach, California. 1997. pp. 115-118
- [16] Riddle P, Segal R, Etzioni O. Representation design and brute-force induction in a Boeing manufacturing domain. *Applied Artificial Intelligence*. 1994;**8**:125-147. DOI: 10.1080/08839519408945435
- [17] Fernández A, García S, del Jesus MJ, Herrera F. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*. 2007; **159**:2387-2398. DOI: 10.1016/j.fss.2007.12.023
- [18] Weiss GM. Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter*. 2004;**6**:7-19. DOI: 10.1145/1007730.1007734
- [19] Orriols-Puig A, Bernadó-Mansilla O, Goldberg DE, et al. Facetwise analysis of XCS for problems with class imbalances. *IEEE Transaction on Evolutionary Computation*. 2009;**13**:1093-1119. DOI: 10.1109/TEVC.2009.2019829
- [20] García S, Herrera F. Evolutionary undersampling for classification with imbalance datasets: Proposals and taxonomy. *Evolutionary Computation*. 2009;**17**:275-306. DOI: 10.1162/evco.2009.17.3.275
- [21] Ezawa K, Singh M, Norton SW. Learning goal oriented Bayesian networks for telecommunications risk management. In: Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96); 3-6 July 1996; Bari, Italy; 1996. pp. 139-147
- [22] Cohen G, Hilario M, Sax H, Hugonnet S, Geissbühler A. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*. 2006;**37**:7-18. DOI: 10.1016/j.artmed.2005.03.002
- [23] Tavallaee M, Stakhanova N, Ghorbani A. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2010;**40**:516-524. DOI: 10.1109/TSMCC.2010.2048428
- [24] Daskalaki S, Kopanas I, Avouris N. Evaluation of classifiers for an uneven class distribution problem. *Applied Artificial Intelligence*. 2006;**20**:38-417. DOI: 10.1080/08839510500313653
- [25] Huang J, Ling CX. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*. 2005; **17**:299-310. DOI: 10.1109/TKDE.2005.50
- [26] Landgrebe TCW, Paclick P, Duin RPW, Bradley AP. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In: Proceedings of the Eighteenth International Conference on Pattern Recognition (ICPR'06); 20-24 August 2006; Hong Kong, China: IEEE; 2006. pp. 123-127
- [27] Provost F, Fawcett T. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97); 14-17 August 1997; Newport Beach, California; 1997. pp. 43-48
- [28] Joshi M. On evaluating performance of classifiers for rare classes. In: Proceedings of IEEE International Conference on Data Mining; 9-12 December 2002; Maebashi City, Japan; 2002. pp. 641-644
- [29] Buckland M, Gey F. The relationship between recall and precision. *American Society for Information Science*. 1994;**45**:12-19. DOI: 10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L

- [30] Kubat M, Matwin S. Addressing the curse of imbalanced training sets: One-sided selection. In: Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97); 8-12 July 1997; Nashville, Tennessee; 1997. pp. 179-186
- [31] Guo H, Viktor HL. Learning from imbalanced data sets with boosting and data generation: The Databoost-IM approach. ACM SIGKDD Explorations Newsletter. 2004;**6**1:30-39. DOI: 10.1145/1007730.1007736
- [32] Akbani R, Kwek S, Japkowicz N. Applying support vector machines to imbalanced data sets. Lecture Notes in Computer Science. 2004;**3201**:39-50. DOI: 10.1007/978-3-540-30115-8_7
- [33] Wu G, Chang EY. KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on Knowledge and Data Engineering. 2005;**17**6:786-795. DOI: 10.1109/TKDE.2005.95
- [34] Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006;**27**8:861-874. DOI: 10.1016/j.patrec.2005.10.010
- [35] Fawcett T. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Technical Report HPL-2003-4. Palo Alto: HP Labs; 2003
- [36] Egan JP. Signal Detection Theory and ROC Analysis. New York: Academic Press; 1975. 277 p
- [37] Mease D, Wyner AJ, Buja A. Boosted classification trees and class probability/Quantile estimation. Journal of Machine Learning Research. 2007;**8**:409-439
- [38] Holte RC, Acker LE, Porter BW. Concept learning and the problem of small disjuncts. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI'89); 20-25 August 1989; Detroit, Michigan, USA; 1989. pp. 813-818
- [39] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. Artificial Intelligence Research. 2002;**16**: 321-357. DOI: 10.1613/jair.953
- [40] Wang BX, Japkowicz N. Imbalanced data set learning with synthetic samples. In: Proceedings of IRIS Machine Learning Workshop; 09 June 2004; Ottawa, Canada. 2004
- [41] Hu S, Liang Y, Ma L, He Y. MSMOTE: Improving classification performance when training data is imbalanced. In: Computer Science and Engineering, International Workshop (IWCSE'09); 28-30 October 2009; Qingdao, China; 2009. pp. 13-17
- [42] Han H, Wang W, Mao B. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Proceedings of the International Conference on Intelligent Computing (ICIC'05); 3-6 August 2005; Nanchang, China; 2005. pp. 878-887
- [43] He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the International Joint Conference on Neural Networks; 1-8 June 2008; Hong Kong, China; 2008. pp. 1322-1328
- [44] Zhang J, Mani I. KNN approach to unbalanced data distributions: A case study involving information extraction. In: Proceeding of International Conference on Machine Learning (ICML'03); 21-24 August 2003; Washington DC; 2003
- [45] Drummond C, Holte R. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Proceeding of International Conference on Machine Learning (ICML'03); 21-24 August 2003; Washington DC; 2003. pp. 1-8

- [46] Liu XY, Wu J, Zhou ZH. Exploratory under sampling for class imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2006;**39**:539-550. DOI: 10.1109/TSMCB.2008.2007853
- [47] Cover TM, Hart PE. Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*. 1967;**13**:21-27. DOI: 10.1109/TIT.1967.1053964
- [48] Tomek I. Two modifications of CNN. *IEEE Transaction System, Man, Cybernetics*. 1976;**6**:769-772. DOI: 10.1109/TSMC.1976.4309452
- [49] Hart P. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*. 1968;**14**:515-516. DOI: 10.1109/TIT.1968.1054155
- [50] Wilson DR, Martinez TR. Reduction techniques for instance-based learning algorithms. *Machine Learning*. 2000; **38**:257-286. DOI: 10.1023/A:1007626913721
- [51] Aha DW, Kibler D, Albert MK. Instance-based learning algorithms. *Machine Learning*. 1991;**6**:37-66. DOI: 10.1007/BF00153759
- [52] Laurikkala J. Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*. 2001;**21**:63-66. DOI: 10.1007/3-540-48229-6_9
- [53] Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems Man and Cybernetics*. 1972;**2**:408-421. DOI: 10.1109/TSMC.1972.4309137
- [54] Ho SY, Liu CC, Liu S. Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Pattern Recognition Letters*. 2002;**23**:1495-1503. DOI: 10.1016/S0167-8655(02)00109-5
- [55] Witten IH, Frank E. *Data Mining Practical Machine Learning Tools and Techniques*. 2nd ed. San Fransisco, CA: Morgan Kaufmann Publishers; 2005. 560 p
- [56] Mitchell T. *Machine Learning*. 1st ed. New York: McGraw-Hill Education; 1997. 432 p
- [57] Adriaans P, Zantinge D. *Data Mining*. 1st ed. Harlow, England: Addison-Wesley Professional; 1996
- [58] Tuffery S. *Data Mining et Statistique Décisionnelle: l'intelligence dans les bases de données*. 2nd ed. France: Editions Technip; 2005. 400 p
- [59] Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. 1st ed. London: Chapman and Hall/CRC; 1984. 368 p
- [60] Quinlan JR. *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann Publishers; 1993. 302 p
- [61] Geisser S. The predictive sample reuse method with applications. *American Statistical Association*. 1975; **70**:320-328. DOI: 10.2307/2285815
- [62] Cieslak DA, Chawla NV. Learning decision trees for unbalanced data. *Machine Learning and Knowledge Discovery in Databases*. 2008;**52**:241-256. DOI: 10.1007/11893028_93
- [63] Michalewicz Z, Fogel DB. *How to Solve it: Modern Heuristics*. 2nd ed. Berlin Heidelberg: Springer-Verlag; 2004. 554 p. DOI: 10.1007/978-3-662-07807-5
- [64] Schapire RE. The strength of weak learnability. *Machine Learning*. 1990;**5**:197-227. DOI: 10.1007/BF00116037
- [65] Raskutti B, Kowalczyk A. Extreme Re-balancing for SVMs: A case study. *ACM SIGKDD Explorations Newsletter*.

2004;**61**:60-69. DOI: 10.1145/1007730.1007739

[66] Wu G, Chang EY. Adaptive feature-space conformal transformation for imbalanced-data learning. In: Proceedings of the Twentieth on International Conference on Machine Learning (ICML'03); 21-24 August 2003; Washington, DC, USA. AAAI Press; 2003. pp. 816-823

[67] Ho TB, Nguyen D, Kawasaki S. Mining prediction rules from minority classes. In: Proceedings of the Fourth International Conference on Applications of Prolog (INAP2001); 20-22 October 2001; Tokyo, Japan; 2001. pp. 254-265

[68] Furnkranz J. Separate-and-conquer rule learning. Artificial Intelligence Review. 1999;**13**:3-54. DOI: 10.1023/A:1006524209794

[69] Agrawal A, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD'93); 25-28 May 1993; Washington, D.C., USA; 1993. pp. 207-216

[70] Han J, Kamber M. Data Mining Concepts and Techniques. 2nd ed. San Francisco, CA, USA: Morgan Kaufmann; 2006. 800 p

[71] Zhou ZH. Ensemble Methods: Foundations and Algorithms. 1st ed. Florida: Chapman and Hall/CRC; 2012. 236 p. DOI: doi.org/10.1201/b12207

[72] Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: Bagging, boosting and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 2012;**42**:463-484. DOI: 10.1109/TSMCC.2011.2161285

[73] Chawla NV, Lazarevic A, Hall LO, Bowyer KW. SMOTEBoost: Improving prediction of the minority class in boosting. Knowledge Discovery in Databases. 2003;**28**:107-119. DOI: 10.1007/978-3-540-39804-2_12

[74] Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning. 1999;**37**:297-336. DOI: 10.1023/A:1007614523901

[75] Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 2010;**40**:185-197. DOI: 10.1109/TSMCA.2009.2029559

[76] Freund Y, Schapire RE. A decision theoretic generalization of on-line learning and an application of boosting. Computer and System Sciences. 1997;**55**:119-139. DOI: 10.1006/jcss.1997.1504

[77] Breiman L. Bagging predictors. Machine Learning. 1996;**24**:123-140. DOI: 10.1023/A:1018054314350

[78] Wang S, Yao X. Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE Symposium Series on Computational Intelligence and Data Mining (CIDM 2009); 30 March-2 April 2009; Nashville, TN, USA; 2009. pp. 324-331

[79] Barandela R, Valdovinos R, Sánchez R. New applications of ensembles of classifiers. Pattern Analysis and Applications. 2003;**6**:245-256. DOI: 10.1007/s10044-003-0192-z

[80] Turney PD. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Artificial Intelligence Research. 1995;**2**:369-409. DOI: 10.1613/jair.120

- [81] Turney PD. Types of cost in inductive concept learning. In: Proceedings of the Workshop on Cost Sensitive Learning at the Seventeenth International Conference on Machine Learning (ICML'00); 29 June-02 July 2000; Stanford, California, USA; 2000. pp. 15-21
- [82] Ling CX, Yang Q, Wang J, Zhang S. Decision trees with minimal costs. In: Proceedings of the Twenty-First International Conference on Machine Learning (ICML'04); 4-8 July 2004; Banff, Alberta, Canada; 2004. pp. 544-551
- [83] Zadrozny B, Langford J, Abe N. Cost sensitive learning by cost-proportionate instance weighting. In: Proceedings of the Third International Conference on Data Mining (ICDM'03); 19-22 November, 2003; Melbourne, Florida, USA; 2003. pp. 155-164
- [84] Sheng VS, Ling CX. Roulette sampling for cost-sensitive learning. In: Proceedings of the European Conference on Machine Learning (ECML-2007); 17-31 September 2007; Warsaw, Poland; 2007. pp. 724-731
- [85] Veropoulos K, Campbell C, Cristianini N. Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99); 31 July 31-6 August 1999; Stockholm, Sweden; 1999. pp. 55-60
- [86] Wang BX, Japkowicz N. Boosting support vector machines for imbalanced data sets. *Foundations of Intelligent Systems*. 2008; **4994**:38-47. DOI: 10.1007/978-3-540-68123-6_4
- [87] Mahani A, Baba-Ali AR. A new rule-based knowledge extraction approach for imbalanced datasets. *Knowledge and Information Systems*. DOI: 10.1007/s10115-019-01330-9
- [88] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Proceedings of the First European Conference on Artificial Life (ECAL'91); 11-13 December 1991; Paris. France; 1991. pp. 134-142