POZNAN UNIVERSITY OF TECHNOLOGY

Mateusz Lango

Analysis of data difficulty factors for multi-class imbalanced problems and their application in classification methods

Doctoral dissertation

Supervisor: prof. dr hab. inż. Jerzy Stefanowski

Faculty of Computing and Telecommunications
Institute of Computer Science

Poznań, 2021

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Learning classifiers is the fundamental research area in supervised machine learning, which focuses on constructing systems that are able to automatically assign instances to a set of predefined classes basing on their description with a set of attributes [52]. Such a system, called a classifier, is not explicitly programmed but instead discovers the class assignment function basing on a dataset of learning examples. Such a generally posed problem led to the development of numerous algorithms that are widely used in many distinct application areas such as product recommendation [131], assessment of recidivism risk [148], authorship identification [58], traffic prediction [117], etc. Despite prominent research interest in the problem of classification and considerable successes in many applications, some issues still remain open and hinder widespread usage in specific yet important application domains. One such issue is *learning from class imbalanced data* [163, 76].

A dataset is called imbalanced if it contains classes of different sizes and at least one of the classes is insufficiently represented [25]. We call these underrepresented classes *minority* ones, whereas classes that are more prominent in the dataset are called *majority classes*. For instance, in the medical domain, one may need to construct a classifier that filters out urgent cases among patients coming to the hospital's emergency department. Typically, only a small fraction of patients must be immediately treated and later hospitalized[1] which results in imbalanced class distribution. Note that recognizing minority (urgent) cases correctly is critical for successfully implementing machine learning methods in this application.

Besides the medical domain, the effective recognition of minority classes is crucial in many more application domains of machine learning, such as sentiment classification [22], automatic graph construction [89], fraud detection [173] or cybersecurity incidents detection [142]. Contrary to the expectations of practitioners who seek high accuracy in recognizing minority classes, classical machine learning methods construct classifiers that recognize majority classes more accurately. In extreme cases, the constructed classifier completely ignores minority classes, not being able to classify any test example into these classes [89].

[1] According to the 2018 National Hospital Ambulatory Medical Care Survey (United States), only 12.4% of emergency department visits result in hospital admission.

Initially, it was thought that this undesirable behavior of the learning systems emerged only from *the global imbalance ratio*, i.e., from the significant discrepancy between class sizes in the training set. However, the experimental analyses demonstrated that for some simple classification problems the level of class imbalance has almost no effect on the finally constructed classifier [64, 127]. It was observed that the class imbalance affects the construction of classifiers and significantly deteriorates the recognition of minority classes only if it occurs together with other *data difficulty factors* [65, 127, 122]. Data difficulty factors can be roughly divided into global factors that affect all the examples in the dataset and local factors that concern only a certain subset of instances [86]. One example of a global data difficulty factor is the discussed class imbalance, whereas class overlapping, class decomposition into sub-concepts, or a significant number of outlier observations are typical examples of local factors.

Due to the great practical importance of learning from imbalanced data, a lot of dedicated methods have been proposed to mitigate this issue [16]. These methods can be divided into algorithmic methods, methods modifying data distribution, and cost-sensitive methods [54]. Algorithmic methods try to solve the difficulties resulting from the construction of classifiers on imbalanced data by modifying the existing learning systems, e.g., changing the decision rule or weighing the loss function. This category also includes specialized classifier ensembles that modify voting schemes or the way the base classifiers are constructed. Unlike algorithmic methods that focus on modifying specific algorithms, methods modifying the data distribution in the original training data are universal and independent of the selected classifier. These methods modify an imbalanced dataset in a way that a classifier learned on such preprocessed data achieves better results on the minority class. The last group of methods uses cost-sensitive learning algorithms, focusing the classification system on minority classes by assigning higher misclassification costs to their examples.

The vast majority of the methods included in the aforementioned categories are designed for binary imbalanced data only. Nevertheless, the phenomenon of class imbalance also occurs in multi-class datasets. Extending our previous example, the hospital may be interested not only in identifying urgent cases but also in detecting the cases that should be treated by general practitioners or those which should be handled in different hospitals (due to e.g. lack of a specialized unit). Again, the two just described classes will have fewer examples than the less urgent cases treated onsite that will constitute the majority class. The presented problem has several minority classes, but other problems with several majority classes or with several majority and minority classes also occur in practice in many important application domains, including text analysis [78], diagnosis of technical devices [152], or analysis of medical data [154, 135].

Even so, much of the research to date and the proposed learning methods only deal with two-class imbalanced classification problems and cannot be directly applied to problems having several classes. The few proposed methods are limited to problem decompositions into binary tasks and other specialized methods, mostly adaptations of binary preprocessing methods [130]. Nevertheless, straightforward modifications of the binary methods do not take into account more complex relationships that arise between classes in multi-class imbalanced problems and do not deal with additional sources of difficulties identified by practitioners.

Moreover, the proposed methods are often constructed rather heuristically, without a clear reference to the issues of multi-class imbalanced learning they are aiming to solve. This is because the sources of difficulty in classifying multiple imbalanced classes were not extensively studied. Such theoretical and experimental analyzes of imbalanced classification conducted so far, with very few exceptions, were limited to binary problems only and their results cannot be easily applied to multi-class data. Performing such analysis would not only deepen the understanding of the problem but also could lead to the development of new methods for multi-class imbalanced classification, designed in a principled way. It is worth mentioning that works on analyzing data difficulty factors in binary imbalanced problems were successfully conducted [122, 138, 125] and resulted in the proposition of several very effective classification methods [121, 124, 14, 109].

Even though imbalanced data classification has been intensively investigated in the past quarter of a century, it is still considered an open problem. Notably, there is a recently growing research interest in the more challenging multi-class variant of the problem, which is still under-researched and has many potential applications. There is a considerable need to design new specialized methods for multi-class imbalanced data and thoroughly characterize the problem.

Based on the above analysis, we formulate the following hypothesis of this dissertation:

> Novel, more effective methods for constructing classifiers from multi-class imbalanced data can be proposed. Such methods could take into account information about the sources of difficulties related to the data distribution both at the local and global level.

This hypothesis led us to formulate the following principal goals of this thesis:

- to extend the analysis of difficulty sources in multi-class imbalanced data, particularly by performing experimental analysis of several data difficulty factors, including global factors such as imbalance ratio, class size configurations, and the number of classes, as well as local factors such as class overlapping and interrelations between different class types,
- to extend and adapt a method for identifying data difficulty factors in real multi-class imbalanced datasets that allow for modeling interdependencies between classes
- to propose a new data-level method for multi-class imbalanced data classification that adapts their performance on the basis of the detected characteristics of the data distribution through the analysis of difficulty factors,
- to develop new methods for the classification of imbalanced multi-class data that are based on ensembles of classifiers.

The main contributions of the conducted research are:

- the experimental study investigating the influence of various data difficulty factors on the performance of classification algorithms on multi-class imbalanced problems, also in relation to the balanced optimal Bayes classifier,
- Similarity Oversampling and Undersampling Preprocessing (SOUP) algorithm, which uses difficulty factor analysis along with expert-based class dependency modeling to modify the data distribution such that the classifiers learned on it have better mi-

nority class prediction ability,

- the development of the Multi-class Roughly Balanced Bagging algorithm that extends bagging ensemble technique for multi-class imbalanced data, which achieves high prediction quality for complex multi-class imbalanced datasets (including these with a large number of features),
- the experimental evaluation of the usefulness of multi-class imbalanced data learning methods, including those proposed in this thesis, for a practical machine learning problem, namely sentiment analysis. This problem was chosen for the analysis as it has great practical importance and is considered relatively difficult [83].

Some contributions presented in this thesis have already been presented at international conferences and published in scientific journals. The list of the author's publications can be found in Appendix A.

1.2 Thesis Structure

The thesis is organized as follows: Chapter 2 introduces the reader to the problem of imbalanced data and provides a brief overview of algorithms for binary imbalanced data. The chapter also contains the discussion of the research on data difficulty factors as well as a review of methods for multi-class imbalanced data classification. Chapter 3 presents an experimental evaluation of the impact of several local and global data difficulty factors on classification performance on multi-class imbalanced problems. The conclusions from this analysis of data difficulty factors are used to propose a method for detecting them in real datasets in Chapter 4. That chapter also presents a new resampling method, called Similarity Oversampling and Undersampling Preprocessing, that exploits these detected factors to improve classifiers' performance. This resampling method is further extended in Chapter 5 to construct an even more effective bagging ensemble for multi-class imbalanced data. Furthermore, the chapter also presents another well-performing ensemble for imbalanced data. These new bagging ensembles are compared against each other and with other related ensemble methods, highlighting the benefits of exploiting data difficulty factors in the design of new learning algorithms. Finally, in Chapter 6 the proposed methods are used for a challenging and practical task of sentiment classification.

Chapter 2

Learning from Imbalanced Data

2.1 Imbalanced Data

A data set is called imbalanced when the number of instances representing different classes is not equal. However, to talk about the problem of imbalanced data classification, we usually have in mind situations where the differences in class cardinalities are significant [54]. The vast majority of related works consider binary imbalanced classification problems where the class with a smaller number of examples is called minority or positive class and their counterpart is referred to as majority or negative class.

While learning from an imbalanced dataset, typical learning systems tend to focus their operation on the most prominent classes at the same time disregarding the classification performance on the minority classes [16]. It usually comes from the focus on obtaining overall high classification accuracy that can be more easily achieved by assigning examples to more frequent classes, e.g., when the decision is uncertain. Such behavior of a learning system can stem from optimizing a loss function that is calibrated for 0-1 loss, but also from preferring more general rules that cover more instances during classifier construction. Even though classification accuracy is a relevant metric in some applications and such operation of learning systems is beneficial, in other domains, classifiers with insufficient recognition of minority classes are of little use.

One example of such an application domain is medicine. For instance, in cancer screening testing, the goal is to filter out as many healthy patients as possible while not discarding any infected patient from additional, more thorough testing. The dataset in this domain will be strongly imbalanced since the number of positive cases (infected patients) is comparatively small, making a classifier trained with standard classification methods unuseful since it will tend to classify more examples to the majority class and potentially fail to send unhealthy patients for proper cancer testing.

Imbalanced data classification aims at solving the aforementioned problem by constructing classifiers that are particularly focused on achieving high classification performance on minority classes. Therefore, apart from the imbalanced class distribution, the imbalanced classification problem is characterized by the practical importance of minority class recognition rather than the focus on the general recognition of any instances [138]. Besides the medical domain, such problems arise in many other domains such as fault diagnosis in engineering [146], object recognition [128], customer relationship management [8], power engineering [174], bioinformatics [93], finance [132] and natural language processing [89].

Imbalanced datasets are characterized in the literature by the imbalance ratio coefficient, which frequently is defined as:

$$IR = \frac{N_-}{N_+}$$

where N_- and N_+ are the cardinalities of majority and minority class, respectively. Such definition is standard for binary imbalanced problems, however, in the case of multi-class imbalanced data, N_- is the size of the biggest majority class and N_+ is the cardinality of the smallest minority class. The lowest possible value of imbalance ratio is 1, indicating a perfectly balanced dataset. Any value above 1 means that the dataset is, to some extent, imbalanced. Even though some researchers consider that a dataset poses an imbalanced classification problem when the imbalance ratio is bigger than a certain value, e.g., 1.5 [38], there is no consensus on the particular threshold value in the imbalanced learning community. The researchers usually point out that the most important characteristic of an imbalanced learning problem is that it requires constructing classifiers biased towards less frequent classes, rather than it has a particularly high value of imbalance ratio [39].

Note that some researchers use a different definition of the imbalance ratio, e.g., they use the percentage of examples belonging to the minority class or ratios like 1:100 [39].

2.2 Classification Measures for Imbalanced Data

Since our aim is to construct classifiers that accurately classify minority class instances, without losing too much on the classification performance on majority classes, the use of standard accuracy measure for imbalanced problems in unsuitable. The classifier assigning everything to the majority class on a dataset with $IR = 99$ has very high accuracy of 99%, despite completely ignoring minority instances. Therefore, in imbalanced learning we use other more appropriate classification metrics. Even though many metrics have been proposed for imbalanced problems, in this chapter we will present only the most frequent ones that will be later used in this thesis. For the more comprehensive review of classification metrics for imbalanced problems, one should refer to e.g. [39, 55].

To assess the classifier performance on binary imbalanced data, one of the most popular measures is F-score, which is defined as the harmonic mean of two other measures, namely precision and recall. These two measures are focused on assessing the predictive performance of a classifier on minority class. Precision is calculated as the ratio of examples correctly assigned to the minority class to all examples assigned to the minority class by the classifier. On the other hand, recall (also called true positive rate or sensitivity) is the number of minority instances correctly recognized as such divided by the number of all minority examples. More concretely, these measures can be defined by the following equations:

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

$$Recall_+ = TP_{Rate} = \frac{TP}{TP + FN}$$

$$\text{F-score} = \frac{2}{\frac{1}{Precision_+} + \frac{1}{Recall_+}}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives. Aside from reporting aggregated precision and recall in the form of the F-score, both these measures are often reported additionally to give a better insight into classifier's working on the minority class.

Another very popular measure for binary imbalanced classification is G-mean i.e. the geometric mean of recall and true negative rate (also called specificity). The latter is simply a recall calculated for the majority class instances and can be defined as

$$Recall_- = TN_{Rate} = \frac{TN}{TN + FP}$$

where TN is the number of true negatives and FP is the number of false positives. Therefore, the G-mean can be defined as

$$\text{G-mean} = \sqrt{Recall_+ \cdot Recall_-}$$

In the recent study analyzing theoretical properties of many different metrics for imbalanced data classification, G-mean was indicated as the measure possessing many useful properties [19]. Also, the analysis of metrics' visualizations showed the superiority of G-mean over e.g. F-score [19].

G-mean can be also easily extended for the multi-class classification, where it can be defined as the geometric mean of all classes' recalls. Nevertheless, while using such defined G-mean on multiple classes one problem arises. Given that the data set is imbalanced, there is a chance that one of the heavily underrepresented minority classes will have recall of zero, which causes collapse of the whole G-mean value to 0. In the binary case, this property, i.e. zeroing measure for a classifier ignoring any of the classes, was beneficial because it allowed discarding classifiers that ignore the minority class. However, in multi-class data this property makes indistinguishable classifiers that work well on many minority classes (except one) with classifiers that completely ignore all minority classes. This issue is usually solved by replacing zero recalls with a small constant value e.g. $\epsilon = 0.001$ during geometric mean computation. The G-mean values reported in this work are calculated in this way, i.e. with equation:

$$\text{G-mean} = \sqrt[|C|]{\prod_{i=1}^{|C|} \max\{Recall_i, \epsilon\}}$$

where $|C|$ is the number of classes and $Recall_i$ is calculated with the same equation as $Recall_+$ while treating instances of class i as positives and all the other classes as negatives.

The extension of F-score for multi-class imbalanced data is called macro-averaged F-score. In order to compute this measure, one should compute average precision and average recall over all classes first. Then, macro-averaged F-score is simply a harmonic mean of these two values.

$$\text{F-score}_{macro} = \frac{2}{\frac{1}{\frac{1}{|C|}\sum_{i=1}^{|C|} Precision_i} + \frac{1}{\frac{1}{|C|}\sum_{i=1}^{|C|} Recall_i}}$$

Other measures used in multi-class imbalanced classification include: AveragedAccuracy [38] (computed as an arithmetic mean over each class recall), Kappa coefficient [6] or MAUC [172] (sometimes called M-measure) for classifiers with probabilistic output.

2.3 Data Difficulty Factors of Imbalanced Data

In the first subsection, we characterized imbalanced classification problems by means of their global imbalance ratio. Even though this property of imbalanced data is the most distinctive one, it was discovered to be not the most influential one. For instance, Wallace et al. [153] shown that on linearly separable imbalanced problems, optimal separating hyperplane has zero cost regardless of the value of imbalance ratio. Similarly, standard classifiers obtain a pretty good classification performance on some real datasets with a high imbalance ratio [127]. Therefore, the imbalance ratio itself does not determine the difficulty of the problem.

Naturally, in real datasets, aside from class imbalance, may occur other *data difficulty factors* that also pose difficulties for constructing classifiers. These difficulty factors include various forms of *class overlapping*, division of classes into several non-homogeneous *subconcepts*, and *data rarity* [39]. Even though these factors alone lead to the deterioration of classifiers' performance, adding to them the class imbalance results in much more significant performance drops. Moreover, the performance loss comes primarily from the reduction of minority class recognition that is critical in practice. Therefore, many researchers [66, 68, 64, 65, 157, 127, 121] consider understanding the relationship between these additional data difficulty factors and class imbalance crucial for the development of algorithms dealing with imbalanced data.

The first data difficulty factor that will be considered here is class overlapping, by which we mean a situation in which there are regions in the feature space that do not clearly belong to any of the classes since they contain mixed instances from several classes. Since there are no clear boundaries between classes, then naturally the induction of decision boundary is more difficult. Nevertheless, in the case of imbalanced data, the induced decision boundary will usually be moved too much towards the minority class region, i.e., assigning a lower number of instances to this class. This is a consequence of the fact that in the overlapping region, the minority class will be weaker represented due to its rarity, so the overlapping region typically will be assigned by the classifier to the more prevalent majority class. For instance, experiments conducted on C4.5 decision tree classifier and the completely overlapped classes from a normal distribution (both classes are generated from normal distributions with the same mean and standard deviations) demonstrated that the classifier learned to always respond with the majority class when class imbalance was present [127]. Similarly, experiments performed on other standard classifiers like SVM, J48, MLP, or kNN demonstrated a substantial negative impact of class overlapping on the minority class recognition and pointed out the local characteristics of data distribution can be more important than the global imbalance [45].

Another difficulty factor is the decomposition of minority class into several subconcepts, i.e., examples of the class do not constitute one compact group, but they are rather split into several, potentially remote, groups in the feature space. This difficulty is related to two other phenomena: 1) construction of small disjuncts by algorithms such as decision trees and rules that are less accurate [57] and 2) more complex forms of class imbalance, namely within-class imbalance [54]. The latter characterizes datasets with a minority class separated into subconcepts of which some are much less dense. Since the minority class has a relatively low number of examples, in extreme cases, very small subconcepts can be interpreted as noise and, consequently, not be reflected in the learned model. The problem of very small subconcepts was investigated e.g. in [68], where it was experimentally demonstrated that they are more responsible for the decrease of classification performance than global class imbalance.

Yet another difficulty factor that is often considered in the imbalanced data literature is so-called data rarity or the lack of data, which is related to the small number of examples of a class or a concept that makes accurate classifier construction more challenging [157]. Unfortunately, imbalanced data often arise in domains such as medicine where the collection of large datasets is problematic.

Detecting and measuring the prevalence of these difficulty factors in real datasets is not a trivial task. Some authors proposed using the number of support vectors in the SVM classifier as a measure of class overlapping [32] or using the k-means algorithm to detect class subconcepts [68]. Others suggest estimating noise examples by checking if they are misclassified by the majority of classifiers in an ensemble [150].

Nevertheless, Napierała and Stefanowski [123] proposed a more general method that allows assessing many different difficulty factors at once. The method consists of calculating the prevalence of four example types in a dataset, called safe, borderline, rare, and outlier. Safe examples lie in a homogeneous region in the feature space, dominated by its class. Borderline examples belong to the class overlapping region where locally neither of the classes is strongly dominating the other. Rare examples are very small groups of two (or three) examples that potentially represent some underrepresented subconcept. Finally, outlier examples are single instances that lie in the region dominated by the other class. The distribution of example types is usually reported for the minority class only, since the majority classes have mostly safe examples. These four example types are illustrated in Figure 2.1.

Each of these example types can be identified by analyzing the classes of the example's nearest neighbors. For instance, if $k = 5$ nearest neighbors are used for assigning the example types, then five or four neighbors that belong to the example's class are indicators of a safe example. If only three or two neighbors belong to the same class as the considered example, it is labeled as a borderline example, etc. Another variant of this method is using kernel estimation with predefined thresholds on locally measured conditional class probability. The authors of these methods demonstrated that for datasets with different dominating example types, the best performing method handling class imbalance is different [123].

Another method that allows not only for the detection of the example types but also detects the class subconcepts and allows for measuring the with-in class imbalanced was

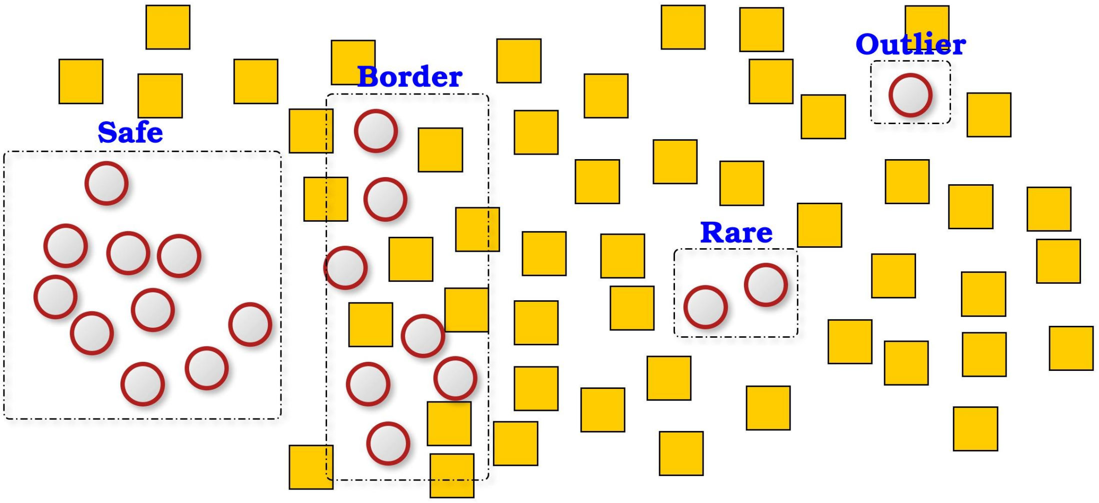


Figure 2.1: Four types of examples for binary imbalanced classification. Minority instances are depicted as circles and majority ones as yellow squares.

proposed by us in [84]. ImGrid is a dedicated grid-based clustering algorithm that, rather than dividing a dataset into clusters of similar examples, splits the minority class examples into groups of the same type. An additional procedure of the algorithm constructs representations of subconcepts from the type-oriented clusters.

Note that all the aforementioned methods for detecting data difficulty factors in imbalanced data are dedicated to binary problems only. Previous to this work, to the best of our knowledge, there was only one work using the methodology of Napierała and Stefanowski directly on multi-class data by treating all the other classes as one majority class [130]. In that work, Sáez et al. obtained promising results by using the obtained example types to direct oversampling towards examples of one selected type, which indicates that designing methods for multi-class imbalanced data that take into account other data difficulty factors can be beneficial. Nevertheless, the impact of data difficulty factors on multi-class imbalanced data as well as more advanced methods for detecting them were not proposed in the literature.

2.4 Selected Methods for Imbalanced Data Classification

2.4.1 Methods for Binary Imbalanced Data

The methods for binary imbalanced classification are usually divided into data-level, algorithmic-level, and cost-sensitive methods [54]. Data-level methods are the most general ones since they alter the original dataset with the goal of constructing a more balanced dataset that would be more convenient for learning classifiers. Therefore, they can be used with virtually any learning method. Some of the simplest approaches of this type are *random undersampling* (RUS) and *random oversampling* (ROS).

RUS preprocesses the original dataset by randomly eliminating majority instances until the dataset is perfectly balanced or until a given proportion between class cardinalities is achieved. Therefore, RUS constructs a much smaller dataset that makes the training faster while improving the classifier performance on the minority class. Nevertheless, it can potentially discard useful majority instances and, in the case of the majority class with several sub-concepts, there is a risk of removing one of them entirely from the dataset.

Contrary to RUS, random oversampling duplicates randomly selected minority instances as long as the cardinalities of the classes are not equal. Even though ROS does not remove any important knowledge from the dataset, learning on the randomly oversampled dataset can lead to classifier overfitting, especially if there are outliers in the dataset that will be repeatedly duplicated during the preprocessing.

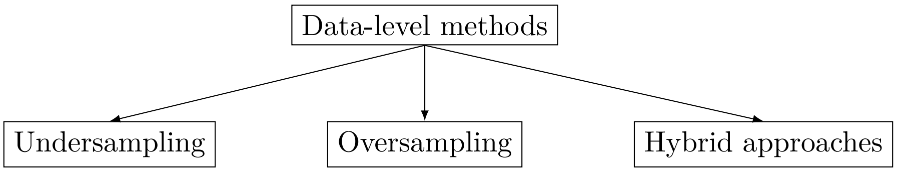
Aside from random methods, also more informed under- and oversampling approaches have been proposed. The most popular such method is *Synthetic Minority Oversampling Technique* (SMOTE) [24], that instead of duplicating minority instances during oversampling, automatically constructs completely new, artificial ones. In order to create a new minority instance, SMOTE selects one of the minority instances and looks for its k nearest neighbors that belong to the same class. Then, a new instance is constructed as the linear interpolation between the instance and one randomly selected nearest neighbor. Typically, $k = 5$ is used and the new instances are constructed until the dataset is fully balanced (even though, originally the number of generated instances was a parameter of the method). In the case of the nominal features, the constructed instance is given the most frequent value among the k nearest neighbors.

During dataset preprocessing, SMOTE completely ignores majority class examples and sometimes leads to overgeneralization of the minority class, e.g., by constructing examples between two minority class subconcepts separated by the concept of the majority class. Therefore, many extensions of SMOTE has been proposed such us Borderline-SMOTE [49], MWMOTE [10] and SMOTE-IPF [145]. The interested reader can find the list of about 85 different SMOTE extensions in [73].

Learning algorithm	Examples of modifications for imbalanced learning
rule-based learners	BRACID [121]
decision trees	Hellinger-distance decision trees [28], DKM splitting criterion [35], Class Confidence Proportion Decision Tree [105]
random forest	Weighted Random Forest [27]
k nearest neighbours	k Exemplar-based Nearest Neighbor [98], gravitational-based Nearest Neighbor [23]
support vector machines	asymmetric kernel scaling [112], zSVM [61], Fuzzy SVM [101]
neural networks	weighting loss functions [6, 29]

Table 2.1: Examples of algorithm-level approaches that modify existing learning algorithms for imbalanced learning.

Another group of methods, algorithmic-level approaches, modify a particular learning algorithm in order to improve its performance on imbalanced tasks. One common way of making the classifier focus on the minority class is to modify the optimized loss function, for example, by weighting examples proportionally to the inverse of its class frequency [6]. Other methods include asymmetric margins in SVM [60], special modifications of the



→ random undersampling

→ approaches cleaning decision boundary: NCR [91], ENN [159], NearMiss-{1,2,3} [111], Tomek links [149]

→ approaches removing instances far from decision boundary: CNN [50]

→ approaches mixing both above strategies: OSS [81]

→ clustering-based approaches: SBC [164], CPM [165]

→ approaches selecting instances through optimization: ACOSampling [166], CBEUS [70]

→ random oversampling

→ Syntetic Minority Oversampling Technique (SMOTE) [24]

→ SMOTE extensions: Borderline-SMOTE [49], ADASYN [53], Safe-level SMOTE [21], MWMOTE [10]

→ other approaches generating synthetic data: noisy replication [92]

→ cluster-based approaches: CBO [68]

→ direct combinations of undersampling and oversampling methods: SMOTE + Tomek links [11], SMOTE + ENN [11]

→ SPIDER [139]

→ ROSE [114]

→ SMOTE-IPF [145]

→ approaches based on evolutionary optimization [110]

Figure 2.2: The categorization of data-level approaches along with references to several illustrative approaches of each type.

distance function in k nearest neighbor classifier [98], changing the splitting criterion in decision trees [105] and modifications of hypothesis search strategies and evaluation measures in rule induction algorithms [121].

The algorithmic-level group also includes specialized classifier ensembles that are usually based on the extensions of bagging [17] or boosting [41]. As we mentioned before, one of the disadvantages of random undersampling is that it produces smaller datasets, therefore increasing the variance of the learner. This disadvantage can be easily overcome by bagging classifiers constructed on datasets produced by the undersampling technique since bagging is known for its variance reduction abilities. In fact, one of the few theoretical works on imbalanced data classification [153] recommends using bagging together with undersampling to obtain good performance on imbalanced problems.

One of such approaches, which will be explored in further chapters, is *Roughly Balanced Bagging* (RBBag) [56]. RBBag constructs classifier ensembles by training component classifiers on different datasets constructed from the original data by a special undersampling procedure. That procedure works as follows: N_+ examples from minority classes are copied to the new dataset by random sampling with replacement, then majority examples are randomly copied to the new dataset, but their amount is sampled from the Pascal distribution (negative binomial distribution) with the number of successes $k = N_+$ and probability of success $p = \frac{1}{2}$. The procedure will, on average, produce balanced datasets since the expected value of the Pascal distribution is equal to $\frac{(1-p)k}{p} = \frac{(1-0.5)N_+}{0.5} = N_+$.

Besides RBB, many other bagging extensions were proposed for imbalanced data such as OverBagging [155], Unevenly Balanced Bagging [99], Neighborhood Balanced Bagging [14]. Also, various modifications of boosting have been proposed, such as RUSBoost [134] or SMOTEBoost [26].

The last group of methods transforms the problem to cost-sensitive learning by obtaining the cost matrix from the expert or estimating it from data [39]. Cost-sensitive learning methods include: methods converting any learning algorithm to a cost-sensitive one (e.g. MetaCost [36]) and direct approaches (e.g. AdaCost [37], RBF with costs [5]).

For a more detailed review of class imbalanced methods, please refer to [16, 39, 54].

2.4.2 Methods for Multi-class Imbalanced Data

The problem of multi-class imbalanced data is considered as more difficult than its binary counterpart [171, 76, 156] as one cannot focus the algorithm only on one minority class to improve classifier's performance. The problem also received considerably less research attention, therefore the number of proposed methods is much smaller. The approaches for multi-class imbalanced data can be roughly divided into:

- methods that transform the multi-class problem into the binary classification ones, relying on previously proposed algorithms for binary imbalanced data;
- specialized approaches, dedicated for multi-class data.

2.4.2.1 Methods that cast the multi-class problem into binary classification

The most popular way to cast the multi-class problem into binary classification is through classifier ensembles that decompose the problem into a series of binary problems. Such methods have initially been proposed for standard classification tasks to enable binary classifiers to handle multi-class problems [7]. In imbalanced learning, they allow using previously proposed binary imbalanced classification methods to tackle multi-class imbalance.

One of such methods is *one-versus-all* ensemble (OVA, also: one-against-all OAA) [113, 129] that constructs $|C|$ base classifiers, each responsible for detecting a particular class. The training set for each base classifier contains all the instances, but all the classes apart from the selected one are aggregated to one common negative class. Then, a binary classifier is trained to distinguish between positive and negative class, i.e., between the selected class and all the others. During prediction, the test instance is classified by all base classifiers and is assigned to the class of the most confident base classifier.

In the case of multi-class imbalanced data, OVA constructs binary problems that often have a much higher imbalance ratio than the original problem, potentially constructing a more challenging problem for the binary classification methods. This issue does not occur in *one-versus-one* ensemble (OVO, one-against-one OAO) [42, 67], that constructs a base classifier for all pairs of classes. In OVO, the training set for the base classifier contains only examples of the selected pair of classes. Therefore, it constructs binary problems that have smaller or equal imbalance ratios than the original problem. Then, similar to OVA, a binary classifier is trained to distinguish between the two classes in the prepared training data. The final decision during the test phase can be taken by majority voting or another function aggregating base classifiers decision such us the sum of confidence scores [51], the sum of scores weighted with class priors [133] or calculating weighted votes against each class [30].

Another possibility to construct an ensemble of binary classifiers to handle multi-class data is through predicting *error-correcting output codes* (ECOC) [34]. ECOC assigns a binary code of a constant length to each class and trains classifiers for predicting each bit of that code. The prediction is made by assigning the class with the most similar code to the predicted one (measured e.g. by Hamming distance). One can say that OVA and OVO are just special cases of ECOC. For instance, OVA is an ECOC ensemble that uses codes constructed by one-hot encoding the class value.

Since all aforementioned techniques train multiple classifiers on binary imbalanced problems, they can be easily used in combination with standard techniques for binary imbalanced data. OVO and OVA are often used with oversampling or undersampling approaches applied on each binary training set of base classifiers [38, 100]. The application of preprocessing technique can depend on the severity of class imbalance in the constructed problem, measured e.g. with imbalance ratio [170] or different preprocessing approaches can be used depending on the subproblem characteristic [94]. Besides data-level approaches, also algorithmic-level approaches for binary imbalanced data can be used in decomposition ensembles. For instance, as base classifiers in OVO/OVA ensembles specialized approaches such as SMOTEBagging [169] as well as one-class methods [79] has been used.

There were also several works on the design of functions aggregating base classifiers decisions in decomposition ensembles working on imbalanced data. These works include optimizing voting weights for each classifier in an aggregation function with genetic algorithms [44], taking into account global class affinity [151] and adjusting decision thresholds [75, 168]. In the case of more general ECOC ensembles, the authors of [107] proposed a weighted distance function between the class codes whose weights are obtained by solving a quadratic programming problem.

Apart from decomposing the problem with classifier ensembles, one can also adapt binary data-level methods to multi-class problems by applying them iteratively to a single dataset with only temporally binarized labels. One of such methods is Static-SMOTE [40] that in each iteration selects the smallest class as the minority class and all the other classes are treated as a single majority class. The method makes $|C|$ iterations, in which it applies the standard SMOTE technique to duplicate the selected minority class. Since the currently smallest class is oversampled in each iteration, a single class can be selected for oversampling in several iterations of the Static-SMOTE procedure. In such a case, SMOTE algorithm is run only on the original instances (i.e. excluding artificial instances constructed in previous iterations).

In a similar vein are designed *Multi-Class Combined Cleaning and Re-sampling* (MC-CCR) [74] and *Multi-Class Radial-Based Oversampling* (MC-RBO) [77] methods. Both works present a new binary resampling algorithm that is later applied to multi-class data by a very similar iterative procedure. The algorithm iterates over all classes in descending order and runs resampling only if there are classes that are bigger than the currently processed class C_i. The resampling procedure is run on a specially constructed dataset consisting of all instances of C_i class and a randomly selected subset of instances from bigger classes. The same number of examples is taken from each class bigger than C_i, while their total number is equal to the size of the biggest class in the dataset. The instances from bigger classes are treated as one common majority class in the binary resampling method, and the instances from C_i are considered minority ones. All modifications made to such constructed binary dataset by the resampling algorithm (i.e. newly constructed instances) are transferred back to the original multi-class dataset after the resampling is performed.

Mahalanobis distance-based over-sampling technique (MDO) [3] is yet another example of informed resampling technique for multi-class imbalanced data that one can see as a binary method applied iteratively to each class. MDO oversamples randomly selected minority instances by generating an artificial instance within the same Mahalanobis distance from the class center. The method iterates over all the classes and oversample them to the size of the biggest class in the dataset.

2.4.2.2 Specialized methods for multi-class imbalanced data

The specialized methods for multi-class imbalanced data can be divided, similarly to the methods for binary imbalanced data, into data-level, algorithmic-level, and cost-sensitive methods. Nevertheless, their number is very limited.

The simplest resampling method for multi-class imbalanced data is Global-CS [172] that duplicates the whole set of instances for each class smaller than the biggest. The duplication is repeated iteratively until the class does not reach the size of the biggest class. In the last iteration, if the further class duplication would exceed the biggest class size, only the number of instances required to achieve a balanced dataset is randomly selected and oversampled.

SMOTE and Clustered Undersampling Technique (SCUT) [4] integrates oversampling of minority classes with the undersampling of majority classes to obtain a balanced dataset in which every class has the cardinality equal to average class size. The method runs expectation-maximization algorithm to fit Gaussian Mixture Model for each majority class, and randomly undersamples instances from each cluster. The number of removed instances depends on the cluster size, and is set to finally obtain the same number of samples from each class' cluster. On the other hand, the minority class instances are oversampled with SMOTE algorithm.

Another idea of modifying SMOTE for multi-class data denominated *Synthetic Minority Oversampling Method* (SMOM) is presented in [175]. As mentioned earlier, the classic SMOTE algorithm can construct instances that lead to minority class overgeneralization. SMOM tries to solve this issue by assigning lower weights to neighboring examples whose linear interpolation would go through the region dominated by other classes. Additionally, SMOM uses the NBDOS clustering algorithm to detect safe[1] regions of minority class whose oversampling will not lead to overgeneralization.

Lin et al. [102] proposed random oversampling of all classes, followed by a dynamic under-sampling procedure for classifiers trained in an online fashion (e.g., MLP with stochastic gradient descent). In every epoch of training, only the randomly selected examples are used for updating classifier parameters. The probability of being selected depends not only on the instance's class size but also on the classifier's confidence score of the correct class. In particular, the misclassified examples are never undersampled i.e. they are always selected to update the model.

The second group of approaches, i.e. algorithmic-level includes various modifications of loss functions such as class-balanced cross-entropy [29], symmetric focal loss [103] and symmetric margin loss [106], often applied in the context of deep neural networks. Other algorithms have also been adjusted for multi-class imbalanced problems. One of such algorithms are decision trees, for which a special split construction criterion, concretely adapted Hellinger distance [28] were proposed. Another example is Extreme Learning Machine with decision outputs compensation [167].

Moreover, several specialized ensemble algorithms have been proposed. MDOBoost [2] incorporates Mahalanobis distance-based over-sampling into boosting algorithm to improve minority classes recognition. Wang and Yao [156] use a version of AdaBoost that increases base classifier diversity through negative correlation learning on randomly oversampled dataset to obtain good results on multi-class imbalanced data.

Another method is *Dynamic Ensemble Selection-Multiclass Imbalance* (DES-MI) [46] that adjust bagging ensemble to multi-class data by modifying training sets construction strategy for component classifiers and by introducing a new dynamic voting schema. The training set for each component classier is a balanced dataset whose size is determined by $r|C|$, where r is a random integer between 1 and the size of the biggest majority class. The dataset is constructed by randomly oversampling or undersampling instances to get exactly r instances from each class. During the prediction, the prediction is made basing only on the decision of selected, most locally competent classifiers. The competence of each base classifier is evaluated with weighted accuracy of k nearest neighbors of testing

[1] The authors of SMOM call such instances „outstanding" and refer to instances in the overlapping regions as „trapped".

instance, with weights inversely proportional to class frequencies in the neighborhood.

The last group of methods are cost-sensitive methods, with the most popular Rebalance [172] that assigns equal costs to each class and later uses multi-class cost-sensitive learners. This method was a direct inspiration for the aforementioned Global-CS method. The cost matrix for multi-class problem can also be optimized by a genetic algorithm in order to maximize some quality measure appropriate for imbalanced data. For instance, [143] optimize cost matrix in order to obtain the highest possible G-mean with AdaC2 [144] boosting algorithm.

Chapter 3

Studying Data Difficulty Factors of Multi-class Imbalanced Data

3.1 Motivation

Most of the current research on multi-class imbalanced data is devoted to proposing new methods that improve the classifiers' predictive performance. Meanwhile, the characteristics of multi-class imbalanced data have not been still sufficiently studied.

Multi-class imbalanced problems are usually characterized only with some form of the imbalance ratio. Wang and Yao [156] also distinguished between *multi-majority* and *multi-minority* problems, referring to the datasets with only one minority or only one majority class having all the other classes of the opposite type. One more type of multi-class distribution is defined in [20], namely: linear imbalance or *gradual imbalance*, which contains classes of linearly growing sizes. Note that in such class size configuration, there is no clear separation of minority and majority classes since the dataset also has some classes of *intermediate size*.

Furthermore, a more general question about difficulty factors in multi-class imbalanced data that are the most influential in the degradation of classifier's performance was also not thoroughly studied. In general, the multi-class learning problems are recognized as harder than the two-class ones. Even for balanced data, decision boundaries between many classes are considered as more complex and challenging to learn [82]. In the context of multi-class imbalanced data, some authors also claim that multi-class imbalanced problems are more challenging than their binary counterparts [171] and a few hypotheses about possible difficulty factors can be found in the literature. However, most of them were not validated in experimental nor theoretical studies, and the works on the characteristics of these data are very limited.

The impact of increasing the number of classes in different configurations of so-called multi-majority and multi-minority datasets was investigated in [156]. Other class configurations, especially with several minority and/or majority classes, were not studied, even though such datasets with small, big, and intermediate-sized classes can often be encountered in many related works. All experiments were performed on the artificial datasets with the same imbalance ratio of 10, and the results showed that multi-majority class configuration is more difficult than multi-minority one in this limited setup.

Other issues such as overlapping between two (or more) minority classes, changes of global imbalance ratio (IR), and types of local class imbalance were not thoroughly investigated. Similarly, the hypothesis stated e.g. in [76] that interrelations between classes are difficult as some classes may be majority ones with respect to smaller ones but at the same time act as minority ones for the remaining classes was also not experimentally studied.

The authors of [130] adopted Napierala's example types described in Section 2.3, to the multi-class setting in a one-vs-all manner. They examined the performance of classifiers trained on a dataset with oversampled minority examples of one type. This strategy was effective for almost all datasets, although the authors did not present methods for tuning the degrees of oversampling nor selecting the type to be oversampled automatically. That work shows that analyzing data difficulty factors can lead to improvements in classifier performance in multiclass imbalanced setting, however it also does not answer questions about the impact of various difficulty factors on classifiers' performance.

Based on the analysis of the literature, we focus attention on the following *multi-class data difficulty factors*:

- overlapping between multiple classes,
- imbalance ratio,
- different configurations of class sizes (including multi-minority, multi-majority ones as well as these with gradually growing class sizes),
- increasing the number of classes.

The current research on these factors in multiple class data is insufficient. On the other hand, the data difficulty factors in the binary problems have already been more intensively studied as mentioned in Chapter 2.3, also see experimental studies [66, 68, 45, 127]. Most of these studies were carried out firstly with synthetic data, but then their observations were transferred to real-world datasets, see e.g. [123]. These studies demonstrated that considering aforementioned data difficulty factors helps in estimating the difficulty of the given data, explains differences in classifiers' classification performance, and assists in evaluating their competence areas. Finally, the results of such analyses were used to develop new algorithms for learning classifiers or developing pre-processing methods for binary imbalanced data, see e.g. [14, 130, 121].

Interestingly, such kind of analysis has not been still carried out for multiple class imbalanced problems. Furthermore, several hypotheses on the nature of these problems were often just mentioned in related papers but not supported with the appropriate experimental examination.

3.2 Research Questions

The analysis of related works has led us to the conclusion that the hypotheses on the difficulty of multi-class imbalanced data are based almost only on the observation of unsatisfactory classification results. Furthermore, it is unclear what is the impact of different difficulty factors on the classifier's performance. In order to fill this gap in the current body of knowledge, we will carry out comprehensive experiments, where we want to answer the following research questions:

1. How much does the increase of overlapping between multiple imbalanced classes influence classifier performance? Does its impact depend on imbalance ratios?
2. What is the impact of different class size configurations on classification?
   - Is the multi-majority class configuration more difficult than the multi-minority one?
   - Is the gradual change of class size more influential than a sharp transition between minority and majority classes?
   - Does the overlapping between classes interact with the different class size configurations?
3. Does the overlapping between minority and majority classes decrease classification performance more than the overlapping between minority classes alone?
4. What is the influence of increasing the number of imbalanced classes on the classification of multiple classes.

In order to examine these issues, we carry out the experiments in a controlled environment based on synthetically generated data (similarly to many earlier studies on binary data e.g. [127, 65]), studying the impact of the multi-class imbalanced data difficulty factors on classification performance of two popular classifiers. Using artificial data, each factor can be easily modeled and parameterized, which allows us to analyze each isolated factor (i.e. single factor without the presence of others in the data). The usage of pre-defined class distributions will also allow us to compare classifiers' predictions to the optimal Bayesian classifier.

3.3 Experimental Setup

3.3.1 Generators of synthetic data

Three synthetic data generators were constructed. *Triangle* and *square* generators were designed to answer research questions 1-3, while *line* generator is specially constructed for examining question 4. All of them generate data from multivariate normal distributions, each associated with a different class. By default, the covariance matrices of normal distributions are set to be multiplicities of an identity matrix with the same multiplier for all classes.

The triangle generator produces data with three classes whose centroids (means) are set to be at the apexes of an equilateral triangle. The length of each triangle edge was chosen to be 5 units as setting the standard deviations to 1 unit produces a dataset with virtually[1] no overlapping cases (see Fig. 3.1a). Using the generator with growing values of the standard deviation produces datasets with an increasing level of overlapping

[1] Gaussian distribution has non-zero density function for the whole domain, so it is not possible to achieve no overlapping at all.

between classes (see Fig. 3.1b). The standard deviation can be interpreted in our case as an equivalent measurement of overlapping in the generated data.

The square generator is designed in a similar way. It is a generalization of the triangle one and produces datasets with four classes from normal distributions whose means are located at the apexes of a square. The side length also equals to 5 units.

On the other hand, the line generator produces datasets with an arbitrary number of classes. The classes in the generated datasets are arranged along the line at equal intervals of 5 units.

Even though the generators can produce data with practically any number of dimensions, we decided to use them in a simplistic 2-dimensional scenario for the experiments. The training sets consisted of 1000 examples. The test sets contained 4000 examples and were balanced i.e. each class had the same number of test examples to better estimate evaluation measures.

In order to better explain a class overlapping, we refer to measures of overlapping already proposed in the literature, see e.g. [147]. One of the most popular measures is Fisher's discriminant ratio, which is defined as follows:

$$F = \min_{c_1, c_2 \in C} \max_{i} \frac{\mu_{c_1,i} - \mu_{c_2,i}}{\sigma^2_{c_1,i} + \sigma^2_{c_2,i}}$$

where C is the set of all classes, μ_c is the mean vector of the examples belonging to class c, analogously σ_c is the vector of standard deviations of these examples and i iterates over features. The notation $\mu_{c,i}$ denotes i-th element of the vector μ_c. Other measures of overlapping proposed in the literature include calculating the volume of overlap region (approximated by a hypercube) or estimating feature efficiency [147].

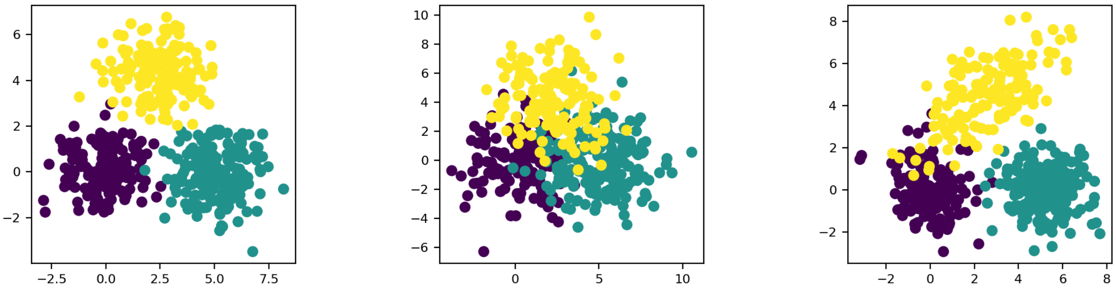
In further experiments, we measure the class overlapping by the standard deviations of normal distributions used to generate datasets. They can be easily transformed into Fisher's discriminant ratio, but have a simple interpretation in our setting. For instance, generating data with $\sigma = 1$ will result in a dataset with $F = \frac{5}{1^2+1^2} = 2.5$ regardless of the type of generator (triangle, square, linear) being used. Similarly, $\sigma = 3$ transforms to $F = 0.27$ which can be interpreted as moderate class overlapping and $\sigma = 10$ to $F = 0{,}025$ which indicates a very strong class overlapping.

To measure the level of class imbalance, we use the imbalance ratio as defined in Chapter 2.1.

3.3.2 Evaluation measures

The performance of the classifiers trained on the generated data will be assessed with two standard measures that are appropriate for multi-class imbalanced data. Firstly, we choose *Recall* of each class, being a recognition rate of correctly classified examples with respect to all examples from a class. Secondly, we use the *balanced accuracy* to aggregate the single class Recalls into one measure. Both measures have a good intuitive meaning, provide a more detailed view on the classifier performance, and allow for comparing the impact of difficulty factors for different classes.

Having the artificially generated data, we can compare the performance of the learned classifier to a theoretically computed optimal solution. More concretely, we refer to the



(a) Dataset generated with $\sigma = 1$.

(b) Dataset generated with $\sigma = 3$.

(c) Dataset generated with $\sigma = 1$ and one-sided overlapping $\sigma_{left} = 3$ between top (yellow) and left (navy blue) class.

Figure 3.1: Exemplary datasets from the triangle generator with the imbalance ratio $IR = 1$

optimal Bayes classifier that uses knowledge about the true data distribution. Its typical formulation [52] maximizes classification accuracy and takes the following form:

$$\hat{y} = \arg\max_{y \in C} P(y|x) = \arg\max_{y \in C} \frac{P(x|y)P(y)}{\sum_{y' \in C} P(x|y')P(y')}$$

where the probability distributions are the true distributions behind the data generation process. Since classification accuracy is not a proper measure for imbalanced data, we used the optimal Bayes classifier derived for *balanced* accuracy – a measure often used in related works [38]. This modification is based on replacing the class frequencies $P(y)$ to uniform distribution, i.e. $\frac{1}{|C|}$.

By comparing class recalls obtained by a real classifier to the recalls of such a defined optimal model, one can notice whether a classifier insufficiently learns concepts related to a class or focuses too much on recognizing another. In the experiments, we report a ratio of class recall obtained by an optimal model to the recall of real classifier, which is called OC Ratio (Optimal to Classifier Ratio):

$$OCRatio_y = \frac{Recall_{\text{optimal}}(y)}{Recall_{\text{classifier}}(y)}$$

The value of OC Ratio = 1 for a given class y would mean that the classifier obtains True Positive Rate for y as high as the best classifier calibrated for the imbalanced distribution. Values higher than one indicate that the class is recognized poorer than in the optimal classifier – that means that values slightly higher than 1 are expected even for a pretty successful real classifier. The values below 1 mean that the recall obtained by a learning algorithm for a particular class is actually too high to get a well-balanced classification performance. We will show in the experiments that values below 1 are characteristic for majority classes.

The usage of OC Ratio (so a reference to known optimal classification) extends knowledge from earlier typical analysis of real classifiers with such measures as class recalls. For illustration, let's analyze a simple situation with two datasets, one with no overlapping and the other with the medium overlapping, where the optimal classifier can achieve 100% and 80%, respectively. If one looks at the studied real classifiers with results of 90% and 78%, it may be claimed that the classifier does not work well with the overlapping data. However, following OC ratio measurements, its classification performance could be treated as sufficiently acceptable while comparing to the optimal one.

3.3.3 Classifiers

We chose two popular standard classifiers: CART tree[2] and k-Nearest Neighbors. CART was always run with default parameters, especially without pruning, whereas Euclidean metric and $k = 5$ was used in kNN. Both classifiers were selected as they are often used in the most related multiple class imbalanced works [130, 63] or even more often in experiments with binary imbalanced data [45]. Moreover, they represent different learning and representation paradigms.

Due to the considerable amount of results and to make this chapter more compact, we present selected detailed plots of decision trees only. The observations and conclusions drawn from these results remained consistent with those for kNN. Nevertheless, the results of both classifiers will be commented in the text, and the complete results are available at the supplement website[3].

All reported results were averaged over 20 runs.

3.4 Experimental Results

3.4.1 Impact of class overlapping and imbalance ratio on the performance of classifiers

In order to address the first research question, the experiments have been carried out with a series of datasets produced by the triangle and square data generators with the imbalance ratio systematically being increased from IR = 1 to 20 and with overlapping ranging from $\sigma = 1$ to 10.

The most representative results on triangle data are presented in Figure 3.2, where classes no. 2 and 3 are minority ones while class 1 denotes the majority one. The first row of the plots in Figure 3.2, depicts changes of class recalls with respect to an increasing overlapping for selected three imbalance ratios.

As expected, problems with the highest class overlapping are more difficult. Recalls of all classes (minority and majority ones) decrease gradually with growing overlapping. However, in the imbalanced datasets ($IR > 1$) the recognition of minority classes decreases much faster than for majority ones. Analyzing the corner cases, one can notice that adding high class imbalance seems to have a relatively small effect for non-overlapping data distributions, resulting in a few percent lower recall of minority classes only, e.g.,

[2] More precisely its implementation in sklearn Python package.

[3] http://www.cs.put.poznan.pl/mlango/publications/difficulty-analysis.html

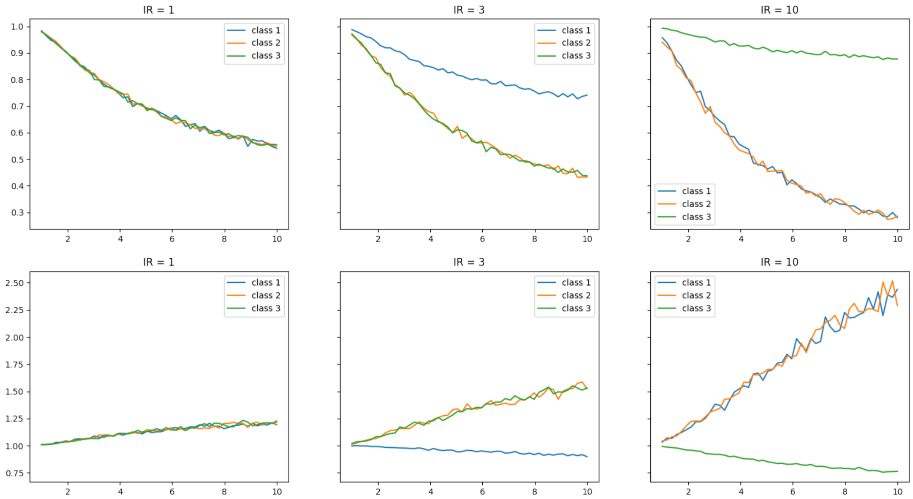


Figure 3.2: Class recalls (first row) and OC Ratio (second row) for the triangle datasets with three selected imbalance ratios (columns) versus class overlapping measured by σ. In the second and third column ($IR = 3$ and $IR = 10$), class 1 is the majority class whereas classes 2 and 3 are minority classes while the first column corresponds to the balanced case.

from 97.8% for the balanced multi-minority triangle dataset to 95.3% for $IR = 10$ and 94.1% for $IR = 20$. On the other hand, for highly overlapping data ($\sigma = 10$) it is visible that increasing IR from 1 to 3 results in approx. 10% drop of minority class recall. Increasing the imbalance ratio further up to 10 results in a similar decrease of 13%.

Looking merely at recall plots, one may say that the class overlapping degrades the class recognition more significantly than the imbalance ratio. Nevertheless, the additional analysis with the optimal classifier demonstrates that it is a more complex phenomenon and the imbalance ratio can be recognized as the influential difficulty factor as well. Referring to the second row of plots in Figure 3.2 that visualize OC Ratio, note that the tree classifier achieves about 25% lower recall than the optimal Bayes classifier on datasets with balanced and highly overlapping classes. That seems to be only a moderate decrease comparing it to approximately 250% lower recall when a high class imbalance of $IR = 10$ to this highly overlapped dataset is added. On the other hand, incorporating class imbalance to the non-overlapping distribution has a minor impact.

The next observation from these plots is that the relation between the class overlapping measured in standard deviations of class distributions and OC Ratio (Optimal to Classifier Ratio) seems to be linear for minority classes in the studied datasets. The slope of the least-squares regression line is 0.024 for balanced datasets, 0.05 for $IR = 3$, 0.077 for $IR = 6$ and 0.128 for $IR = 9$ demonstrating that the impact of class overlapping grows considerably for the imbalanced datasets.

This result is also confirmed for square generated datasets with four classes where the slopes of the linear regression for the same imbalance ratios are 0.028 ($IR = 1$), 0.074 ($IR = 3$), 0.110 ($IR = 6$) and 0.169 ($IR = 9$), respectively. All obtained slopes were statistically significant using standard F-test [52] with $p < 0.0001$.

Such trend lines and statistically significant coefficients are also obtained in our experiments with k-NN classifier. The recalls obtained by this classifier are a bit higher than these of the decision trees for the balanced dataset. However, while increasing the imbalance ratio, the advantage of kNN over trees decreases to finally become a disadvantage of about 8% for minority classes ($IR = 20$, $\sigma = 10$). Nevertheless, the observations and conclusions made here for decision trees generalize smoothly to the kNN classifier.

Conclusions and answers to the research question:

- The increase of class overlapping strongly deteriorates the recognition of the minority classes, particularly on datasets with higher imbalance ratios. Changing the imbalance ratio presented a limited impact on the recognition of datasets without class overlapping, whereas class overlapping impacts the classifiers' performance even for balanced datasets. Results demonstrated that that combination of both factors degrades classifiers' performance more significantly, which is consistent with earlier experimental studies on binary imbalanced data such as [45].
- The majority class is recognized more accurately by considered standard classifiers than the optimal classifier (see OC Ratio below 1), which supports designing methods for imbalanced data that improve recognition of minority classes at the cost of slight worsening the classifier performance on majority classes.

Finally, we claim that the possibility of using a reference of the optimal classifier's recall for the performance of real classifiers extends our knowledge on dealing with overlapping by the latter. When its recall decreases (what is expected), we may now estimate whether it is closer or farther from the optimal solution.

3.4.2 The role of the different class size configurations

We proceed with the next set of experiments to answer the second research question. Recall that in the binary datasets, we do not face any problem in indicating which class is the minority one and which one is not, but the situation becomes a different one when dealing with multiple classes.

For instance, if data contains three classes of 100, 300, and 900 instances, respectively, the role of the intermediate class is not clear. On the one hand, it is three times smaller than the biggest class, which would suggest its minority status, but on the other hand, it is also three times bigger than the smallest class. The presence of such intermediate classes was often noticed in the related works [76, 130, 63], but their impact on classification difficulty was not evaluated.

In our further experiments, apart from two class configurations without intermediate classes i.e. sharp ones consisting of clearly distinguishable minority and majority classes, we also consider configurations with intermediate classes.

We again use the triangle and square data generators and measure the classifier performance on three types of class distributions: multi-majority, multi-minority (with a sharp difference in class sizes), and gradual (with intermediate classes) for varying imbalance ratio IR from 1 to 20 and three class overlapping setups: class separation ($\sigma = 1$), moderate overlapping ($\sigma = 3$), strong overlapping ($\sigma = 10$). Multi-majority datasets contain only one minority class and several majority classes of equal size. On the contrary, multi-minority datasets have one majority class with the highest cardinality and several same-sized smaller classes, according to the selected imbalance ratio.

For the gradual dataset, we generate three or four classes of different sizes for a selected imbalance ratio IR (calculated as a ratio between the largest and smallest of them). The size of the intermediate class is defined so that the ratio between its size and the majority class size is equal to the analogous inverted ratio with the size of the minority class (just as in the example in the previous paragraph). If there are more intermediate classes, they are generated to maintain the same ratio between classes of consecutive sizes e.g. 60, 120, 240, 480 for four classes and $IR = 8$. The values of OC Ratio obtained for the triangle generator and decision trees are presented in Figure 3.3 while the other results are available in the online supplement.

Analyzing the first column of Figure 3.3, one can again observe that the impact of increasing the imbalance ratio on data with virtually no class overlapping is almost none. When class distributions do overlap, growing the class imbalance ratio causes a higher discrepancy between the predictive performance of the decision tree classifier and the optimal Bayes classifier. Independently of the class size configuration type, increasing the imbalance ratio degrades the recognition of minority classes. However, the extent of that degradation depends on the class configurations.

Let us analyze the dataset with IR=20 and highly overlapping classes ($\sigma = 10$) for different configurations of class sizes. For the smallest class in the multi-minority configuration, the tree classifier achieves OC Ratio of 3.1. On the other hand, in the same setting but multi-majority class distribution, the tree classifier recognizes the smallest class 7 times worse than the Bayes classier. That demonstrates a significant impact of the type of class size configuration on the performance of classifiers. Generalizing this observation to other cases, the OC Ratio of the smallest class in the multi-majority datasets is always higher than for the multi-minority configuration for all studied values of IR with moderate or strong class overlapping. Our experiments show that the classification performance of minority classes more strongly deteriorates in the multi-majority variant.

The difficulty of the gradual configuration of classes lies somewhat in-between the two earlier discussed configurations with the smallest class obtaining OC Ratio of 4 on the dataset with IR=20 and $\sigma = 10$. Interestingly, the recognition of the intermediate sized class only slightly deteriorates with the increase of the imbalance ratio. Its OC Ratio did not exceed 1.5 even for IR=20 which is comparable to the recognition of the smallest class in the easiest, i.e., multi-minority configuration with IR of only 2.5 with the same class overlapping. Similarly, looking at the raw values of class recalls, the intermediate class losses only 5% of recall in comparison to the balanced data for highly overlapping distribution whereas the smallest class lost almost 20% in the same setting.

The same conclusions can be drawn from the similar experiments with k-NN classifier, even though this classifier was less successful in dealing with class overlap in our datasets i.e. the OC Ratio values were generally higher, obtaining even 14 in the worst case (multi-majority with IR = 20 and $\sigma = 10$). Similarly to the decision trees, the minority class obtained better OC Ratios in multi-minority than in the gradual setup, however, the differences were smaller (for high IR from 0.5 to 0.76). The intermediate class in the gradual setting again had almost the same OC Ratio independently of growing class imbalance.

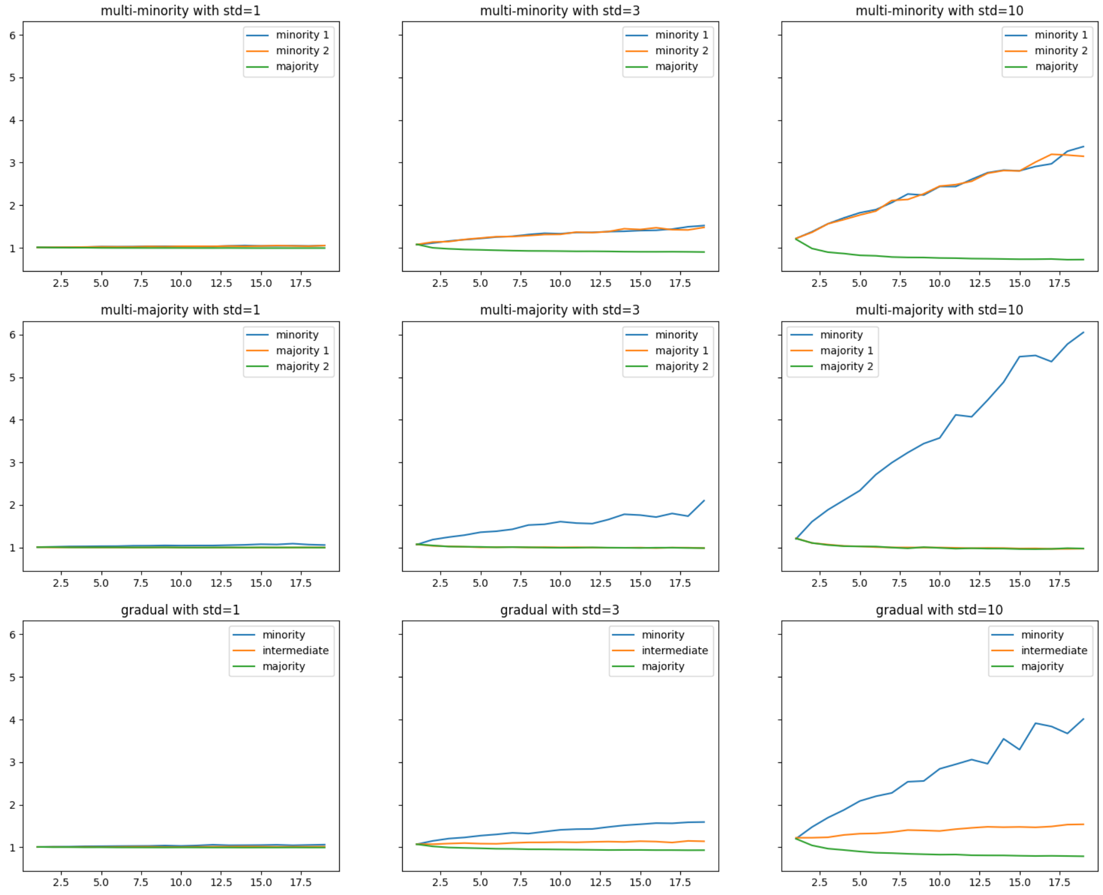


Figure 3.3: The values of OC Ratio obtained by the decision tree classifier on triangle datasets for the imbalance ratio IR growing from 1 to 20 with three different class overlapping $\sigma \in \{1, 3, 10\}$ (columns) and three different types of class distributions (represented in next rows).

This experiment was repeated for four-class square data where the observations were in-line with previously discussed ones for multi-minority and multi-majority data. On gradual data we observed that the second biggest class has almost constant OC Ratio while moving towards more imbalanced data. On the other hand, the recall of the second smallest class was deteriorating for gradual datasets with increasing imbalance ratio. The smallest class was the one loosing recall the most quickly.

Conclusions and answers to the research question:

- The type of class size configuration strongly influences recognition of minority classes. The multi-majority class configuration is more difficult than multi-minority one. The difficulty of the gradual class case (with intermediate size classes) falls between two of them, but is more similar to the easier multi-minority case.
- In the gradual class sizes configuration, the intermediate class is rather well recognized by both considered classifiers, which may suggest that the development of new methods for imbalanced data should be focused on improving recognition of mainly the smallest classes.

3.4.3 Overlapping and interrelation between different class types

The results of experiments discussed in the previous section demonstrated that class overlapping strongly deteriorates the classifier performance. In this section we take a closer look on the influence of class overlapping with respect to interrelations between different types of classes. More precisely, we want to examine whether the overlapping between two minority classes is less harmful for the classifier performance than overlapping of minority and majority classes. Furthermore, overlapping with the intermediate class will be assessed.

To address these questions, we use the triangle generator with several configurations of class sizes. These classes were generated in different class configurations scenarios: either the sharp difference between two equal sized minority class vs. the majority one; or the gradual changes of class sizes with the intermediate one. For each of them the series of experiment was organized as follows: the class overlapping was systematically increased only between the selected pair of classes, leaving the level of overlap between other classes without a change. More precisely, we select one class from dataset and perform two experiments. Firstly we gradually increase the overlap of selected class with another class in the dataset (look at Fig. 3.1c for an example of such generated dataset), measuring classifier performance. Then, as a control experiment, we repeat this procedure for analogous datasets but with overlap being increased between the selected class and the third class. Referencing triangle generator, the class in the upper apex of the triangle is increasingly overlapped with the class on the left-down apex (look at Fig. 3.1c) and later, analogous experiment is performed where the upper class is overlapped with right-down class. We have studied various imbalance ratios between classes.

Figure 3.4 presents results of experiments on datasets representing two different class size configurations. The first investigated dataset (first row of plots) is multi-minority one (class sizes: 800, 100 and 100 examples), and the second is of gradual type, with class cardinalities of 640, 280, and 80 (two next rows). Consider, for instance, the second row in the first column, where we present recalls of the selected minority class while increasing overlapping toward the majority class and in the second column its recall while it overlaps with the intermediate class. The third column compares results of the selected class in both experiments. In both datasets, IR between the majority and the smallest minority is equal to 8. We do not present results on multi-majority datasets here since the overlap between a minority class and two equal-sized majority classes does not cause differences in the results.

As one could expect, a continuous increase of overlap between any of the two classes always decreases recalls. However, the extent of the decrease depends on the type of overlapping classes. While increasing the overlapping between two minority classes causes a strong decrease of Recall of about 12% for both minority classes, the overlap between minority and majority class is considerably more harmful (the decline is approx. 26%).

The role of the intermediate class is more sophisticated. Increasing overlapping between the minority and intermediate class causes decreases of the recognition of the former only slightly smaller than while overlapping minority with the majority class. On the other hand, when the intermediate class overlaps with the majority class, its recognition quickly

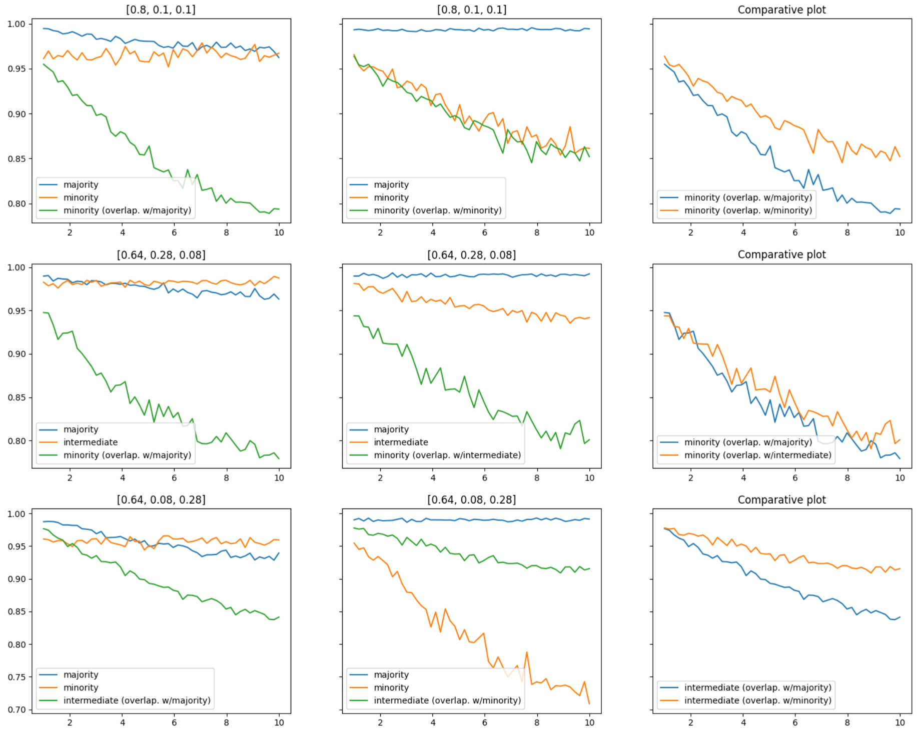


Figure 3.4: Results of Recalls on the triangle datasets (IR=8), where the class overlap was increased only with the selected class. Each row presents the results for one dataset configuration (the class cardinalities distribution is indicated in the plot title). The first two plots in each row (form the left) shows class recalls when the selected class (green line) was overlapped with one of the two other classes. The third plot (the right column) shows comparison of green lines from the first two plots for the selected class.

deteriorates (like the minority class). The overlap between the intermediate and minority class results in slow decrease of the intermediate class recall and a faster worsening of the recognition of the minority one. It seems that the intermediate class has a "twofold" role– it acts like a minority class when interacting with bigger classes and works like a majority class in the context of smaller classes. Indeed, looking at the OC Ratios we see that the intermediate class is recognized too strongly by the classifier while overlapping with the minority class (OC $= 0.97$ with $\sigma = 10$) and recognized too poorly while interacting with the majority class (OC=1.04 with $\sigma = 10$).

This observation is also true for k-Nearest Neighbors classifier, where the intermediate class obtained OC $= 0.96$ ($\sigma = 10$) while overlapping with the minority class and OC=1.05 ($\sigma = 10$) for overlapping with the majority one. Comparing the difference between the recognition of the minority class while overlapping with majority and intermediate class shows that, as expected, the overlapping with the latter is a bit easier. For the most severe investigated overlapping ($\sigma = 10$), the minority class was recognized by kNN with $OC = 1.17$ and $OC = 1.13$ while overlapping with majority and intermediate class, respectively.

Conclusions and answers to the research question:

- An increase of overlapping between minority and majority classes is more harmful for performance of the classifier than the analogous increase of overlap between minority classes.
- For the gradual class size configuration, the recognition of the intermediate class quickly worsen while increasing overlapping with the majority class. Conversely, while interacting with the minority class it causes fast deterioration of the recognition of minority class, similarly to majority classes.

3.4.4 Increasing the number of classes

In this section, we will study the research question regarding the impact of the number of classes with data produced by the line generator.

Figure 3.5 collects the selected results for a decision tree classifier. Again three types of class size configurations correspond to rows of plot in this figure, i.e. multi-minority, multi-majority, and gradual increase of class size. In y axis we report values of recall of the worst recognized class (which is selected among all classes in the given case). Trend lines in these plots are categorized with respect to three levels of overlapping ($\sigma = 1, 3, 10$, columns); three imbalance ratios $IR = 1, 3, 10$ (line colors) and class number ranging from 2 to 20 (x axis).

The main observation is that increasing the number of classes always decreases the recall of the worst recognized minority class. However, this impact depends mainly on the size of overlapping. The recall decreases are much more substantial for the higher overlapping levels, while being much smaller for no overlapping cases (e.g., only 0.3 – 0,4 for std 1; see the left column in Fig 3.5).

The most interesting and unexpected observation is that the highest decrease for datasets with class overlapping always occurs while moving from the binary imbalanced case to the variants with four or five classes. Adding the next class to the generated dataset also causes slightly weaker recognition, but the trend line's shape is clearly flatter. Furthermore, such trend characteristic of recall values occurs for all studied imbalance ratios. Naturally, datasets with the higher imbalance between classes are more difficult, hence, the recall values are lower for them, but again the general trend lines follow the same patterns.

Moreover, the class size configurations do not seem to influence this result – compare rows in Figure 3.5. Finally, the quite similar observation could be made for k-NN classifier.

Conclusions and answers to the research question:

- Datasets with more classes are more difficult than ones with a smaller number. Imbalance ratios and class size configurations do not seem to enlarge this effect.
- The increasing size of class overlapping is the next influential data factor with respect to the recall decreases.
- Interestingly, the increase of the number of imbalanced classes from two to a four / five classes leads to a stronger decrease of recall than adding more classes to an already multi-class imbalanced dataset.

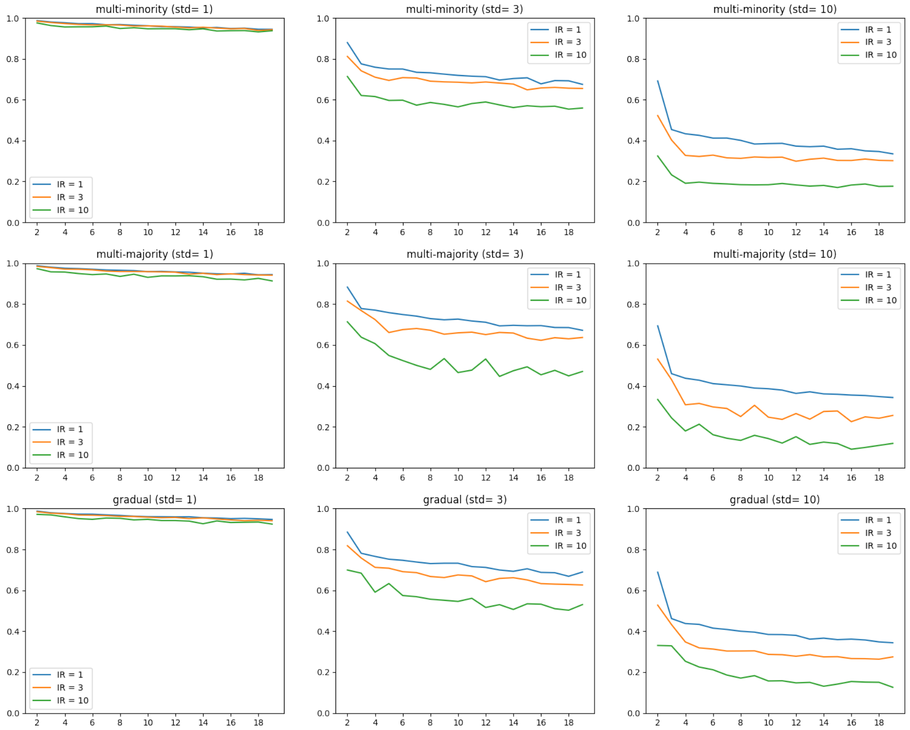


Figure 3.5: The Recall of the worst recognized class vs. the number of classes in the dataset for the tree classifier. Each row presents the results for datasets with different type of class size configuration (multi-minority, multi-majority and gradual), whereas each column presents results for datasets with different level of class overlapping (no-overlapping - std=1, moderate overlapping - std = 3 and strong overlapping - std=10).

3.5 Relation to real data

In this chapter, we are focused on the experiments in a controlled framework with specially generated synthetic data, which allows us to parametrize data difficulty factors. Moreover, it was possible to refer the results of the classifiers to the Bayesian optimal ones. As a result, one may better understand the impact of data factors on the classifier's performance. Nevertheless, it is interesting to verify our observations also in the context of real-world data.

Firstly, we analyzed datasets often considered in experiments with multi-class imbalanced data, such as [38, 63, 130]. However, there were no discussions on class size configuration, an important difficulty factor identified in our study. Only the paper [156] studies multi-majority and multi-minority datasets. Nevertheless, all of them completely ignore gradual datasets with intermediate classes to whose our research brings attention to.

Interestingly, despite the lack of research attention, such data with gradual class characteristics seem to be more common in the real world than multi-majority or multi-minority types. For instance, in the work [156] authors perform experiments with 12 standard real-word multi-class imbalanced datasets. Nine of them have gradual characteristics, two datasets are multi-majority ones, and only one represents the multi-minority group. However, this data factor is not used in the discussion of experiments and to analyze the differences in the classifiers' predictions more deeply.

dataset	class config.	# classes	b. acc.	majmaj	majmin	minmin	intmaj	intmin	intint
cleveland	gradual	5	0,30	0,00	0,37	0,06	0,36	0,12	0,00
winequality	gradual	4	0,37	0,55	0,09	0,00	0,25	0,01	0,00
yeast	gradual	9	0,42	0,40	0,04	0,01	0,30	0,06	0,07
cmc	multi-maj	3	0,46	0,44	0,48	0,00			
balance-scale	multi-maj	3	0,57	0,14	0,27	0,00			
glass	gradual	6	0,67	0,31	0,29	0,02	0,02	0,04	0,00
ecoli	gradual	5	0,73	0,00	0,01	0,00	0,14	0,27	0,02
led7digit	multi-maj	6	0,78	0,29	0,29	0,02			
vehicle	multi-min	3	0,90	0,00	0,23	0,00			
dermatology	gradual	6	0,93	0,00	0,01	0,00	0,07	0,02	0,32
newthyroid	multi-min	3	0,94	0,00	0,20	0,00			

Table 3.1: Balanced accuracy (b. acc.), class size configuration, number of classes and overlapping for selected datasets. Overlapping is reported between different class types: minority (min), majority (maj) and intermediate (int) classes.

The role of overlapping between different types of classes has been highlighted in our experiments with synthetic data. To examine it and its connections with other factors on real-world data, we carried out additional experiments with 11 real datasets coming from [38, 63]. For each dataset, we manually recognized its distribution type (gradual, multi-minority, multi-majority) and calculated the prevalence of different types of overlapping. We detect these types of class overlapping with a very simple method based on the analysis of 5-nearest neighbors, which provides only a rough estimation of it but still, we hypothesize that this approximation of the overlapping is sufficient in this experiment. For each example, we checked the class labels of its neighbors. If most of these neighbors did not belong to the example's class, we labeled it to be in the overlapping region[4]. Then, we identify the types of classes between which the overlapping has occurred based on each class's previous manual assignment to distribution types. The result of our analysis is presented in Table 3.1, where the number of examples from different types of overlapping is presented as the ratio of all examples in the dataset. For each dataset, we evaluated the predictions of the decision tree by means of the balanced accuracy measure.

First, we observe that datasets of multi-minority types are among the easiest ones, which is in line with our previous observations done on synthetic data. The `dermatology` dataset, which is of the gradual type, is put among the few easiest data. However, most of its classes do not overlap, while there is a medium overlapping between intermediate classes only. Next datasets in the ranking, such as `led7digits` represent slightly bigger overlapping (over 0,5) of minority and majority classes, which is associated with the lower balanced accuracy. It is followed by other gradual type datasets, such as `glass`, which has the stronger overlapping, including relations with majority classes. More difficult ones are

[4] This analysis may also cover so-called rare single examples located more deeply in the region of another class, which is one of the limitations of this simplistic method. In the following chapter, a more complex method will be proposed.

multi-majority data, such as `cmc` and `balance-scale`. For instance, `balance-scale` has an average amount of overlapping (41%), however, contrary to easier datasets with a similar amount of overlapping, see e.g. `dermatology`, its overlapping is between minority and majority classes that has been identified as more difficult in our earlier experiments. As expected, the classification performance of this dataset is much lower. The most difficult dataset is `cleveland` which has a very high overlapping of approx. 91%. Additionally, this overlapping is of majority-minority and intermediate-majority types, which are most difficult. A bit easier is `winequality` data which also has a very strong class overlapping of 90%, but its considerable amount is between majority classes.

To sum up, although these real-world datasets have more complex characteristics with many factors occurring together and compound relations between different types of classes, we claim that the experimental results are consistent with earlier experiments on the influence of isolated data factors.

3.6 Discussion

The main aim of this chapter is to investigate the following *mutli-class data factors*: overlapping between multiple classes, its interaction with various imbalance ratios, different configurations of class sizes (including multi-minority, multi-majority ones and those with gradual increases of the size), and the increasing number of classes. We experimentally study their role to characterize the difficulty of imbalanced multi-class data. To the best of our knowledge, these data factors were not previously systematically investigated.

The obtained experimental results led us to the following main observations:

- The class overlapping was the very influential factor when combined with the high imbalance or some types of class size configurations.
- The types of the class size configurations with multiple majority classes were more difficult than multi minority ones (the recalls of the smallest classes were clearly worse). The gradual class size configurations with the intermediate classes played a special role between them.
- The analysis of interrelations between different types of classes showed that the increase of overlapping between the minority and majority classes led to the stronger deterioration of classifier performance than between minority ones. The impact of the intermediate classes depended on the direction of overlapping.
- An increasing the number of classes was the most influential while considering changes between two and five classes.

Some of our results provided confirmations of previous hypotheses stated by few researchers. However, we think that other results extended the current body of knowledge on the difficulty of multi-class imbalanced data. For instance, knowing earlier literature on the binary imbalanced data one could expect that an increase of overlapping strongly deteriorates the recognition of the minority class. Still, our experiments showed that its impact is much stronger for various types of multiple class interrelations.

Krawczyk [76] made a statement on the possibility that some classes may act as majority and minority ones with respect to different classes. This hypothesis was confirmed and extended in our experiments by showing that while adding overlap between minority and intermediate classes, the recognition of the latter was improving in terms of OC Ratio – similarly to majority classes. On the other hand, increasing overlapping between the intermediate and majority classes caused a decrease in classifier performance on the intermediate class, just like for minority classes.

Some earlier works discussed that majority classes were usually better recognized at the cost of poor classifier's performance for the minority classes. By using the reference to the optimal Bayes classifier, we showed that, in fact, investigated classifiers achieved higher recalls of majority classes than the optimal solution. This observation justifies the usage of imbalanced learning methods such as re-sampling that may worsen recognition of the majority class and focus the learning process more on minority classes.

Note that the usage of OC Ratio (so reference to known optimal classification) extends knowledge coming from the typical analysis of classifiers with such measures as Recalls, that were performed for evaluating difficulty factors in the binary imbalanced data setting.

The experiments also highlighted differences in handling overlapping between different class types. We postulate a deeper consideration of the gradual sized class configuration and the specific role of the intermediate classes. In our opinion, they should be taken into account while testing the performance of new classification methods. This additionally raises the question about measuring class imbalance in the multi-class setting. The typical imbalance ratio, which was also used in this study, seems too simplistic as it hides the important information about the gradual, multi-majority or multi-minority characteristics of the classes.

This research also push forward the necessity of a deeper analysis and better handling of different types of class overlapping in preprocessing methods for multi-class imbalanced data. As the overlapping between various types of classes has proved to be an important factor, it seems reasonable to construct methods that focus oversampling in the overlapping regions of particular class types. This has not been still exploited in the current preprocessing methods. In the following chapter, we will address these issues.

Chapter 4

Similarity Oversampling and Undersampling Preprocessing

4.1 Motivation

As we have shown in the previous chapter, multi-class imbalance problems contain additional data difficulty factors that impact the performance of learned classifiers and do not occur in binary problems. These additional difficulty factors come from the more complex relations that occur between classes in multi-class imbalanced datasets. Therefore, they were not considered while designing binary imbalanced learning methods.

Such more complex relations between classes also can not be taken into account while using typical decomposition methods that are very popular for handling multi-class imbalanced data. These methods lose information about the complexities of decision boundaries with several neighboring classes and are unable to model the regions with overlapping of several classes. In particular, the characteristic of one class influencing several neighboring classes in different ways, such as in the case of intermediate classes, is completely lost. Therefore, there is a need to design new methods for imbalanced data that consider these additional difficulty factors that were not previously exploited in the studies on binary imbalanced data.

The first step towards exploiting the characteristic of data difficulty factors is to detect and model them in real datasets properly. Note that our experimental studies on data difficulty factors were performed on artificial data, which permitted us to control data distributions and occurring difficulty factors, but no concrete method for real data was presented.

In this chapter, we extend one popular method for analyzing data difficulty factors in real binary imbalanced datasets [122] to the multi-class scenario. The proposed extension models interactions between various classes with the *class similarity factor* that can be predefined by the user or computed heuristically exploiting the observations from Chapter 3. Basing on this data difficulty factors detection method, a new resampling algorithm called Similarity Oversampling and Undersampling Preprocessing (SOUP) is proposed. The method takes advantage of the local and global difficulty factors, oversampling the most important instances of minority class and undersampling these majority samples that harm the minority classes' recognition the most. The experimental evaluation of the method demonstrates its effectiveness and shows that exploiting the information about data difficulty factors in the design of resampling methods can lead to improvements in classification performance.

4.2 Modelling Data Difficulty Factors in Multi-class Datasets

The global data difficulty factors of imbalanced data such as imbalanced ratio, number of classes, or class type configuration can be relatively simply identified from basic dataset statistics. However, measuring local data difficulty factors can be more challenging and requires more sophisticated methods.

4.2.1 Detecting local data difficulty factors in real datasets

Among the most important local data difficulty factors of multi-class datasets identified in the previous chapter were class overlapping and interrelations between different classes. In particular, increasing class overlapping between minority and majority classes resulted to be more harmful to the classifier performance than the overlapping between minority classes. Additionally, the overlapping between intermediate and minority classes more strongly hinders the recognition of minority class, whereas the overlapping between intermediate and majority classes more significantly influences the classifier's performance on the intermediate class. These observations will be taken into account while designing our method for detecting data difficulty factors in multi-class imbalanced datasets.

Recall that Napierała and Stefanowski [122] proposed to characterize real imbalanced datasets with the distribution of four so-called example types (see Chapter 2.3). These example types can be identified by thresholding the value of locally estimated conditional probability $P(y|x)$, denominated as *safe level.* For instance, in the most popular case of estimating this probability with k nearest neighbors, the safe level takes the following form:

$$safe\_level(x) = P(y = C_x|x) = \frac{k_{C_x}}{k} \tag{4.1}$$

where C_x is the true class of the example x, k_{C_x} is the number of neighbors belonging to the class C_x and k is the total number of analyzed neighbors. The value of an example's safe level can be converted to a particular example type (safe, borderline, rare, outlier) through discretization, however, it can also be used as-is to measure the example's difficulty.

The aforementioned definition of safe level via measuring k nearest neighbors can be straightforwardly applied to multi-class data [130]. Nevertheless, such usage of the safe level is equivalent to using one of the decomposition methods. More concretely, the safe level values of given class examples will not change if one will merge all the other classes to one big majority class and convert the problem to a binary one. Clearly, the more nuanced difficulty of class overlapping that depends on class types and their cardinalities is not modeled.

Our approach will extend the aforementioned definition of the safe level to model various forms of class overlapping with pre-defined similarity values between classes. The goal of introducing additional class similarity degrees is to capture the fact that the difficulty of class overlapping varies depending on which classes overlap. The method assumes that neighboring with examples of more similar classes is safer, i.e., leads to smaller performance degradation, than having interactions with more dissimilar classes. Note that the degree of similarity between classes is a globally defined value and, understandably, the set of nearest neighbors most probably contains examples that all are similar to each other in terms of the feature values.

Example. In order to give some better intuition, let us start with an example previously discussed in our paper [63]. Assume that we want to analyze the safeness of an example x that belongs to the minority class C_{min1} in a classification problem between C_{min1} and two other classes: the minority class C_{min2} and the majority class C_{maj}. We will consider three possible configurations of nearest neighbors described in Table 4.1.

Table 4.1: Considered three possible class distribution among the neighbors of example x

No.	Class C_{min1}	Class C_{min2}	Class C_{maj}
a	5	0	0
b	1	2	2
c	1	0	4

Clearly, situation (a) is the most preferred since the considered example x is in a non-overlapping region occupied by examples from its class. The situations (b) and (c) are more interesting, especially because they would be treated as the same ones while measuring the safeness of x with the binary method directly applied to multi-class data. However, following our observations from Chapter 3 the neighborhood (c) should be considered as more difficult than the neighborhood (b), as overlapping with majority class more strongly impacts minority class performance than overlapping with other minority classes. Therefore, while analyzing various example's neighborhoods, the ones with a higher number of neighbors from class C_{min2} should be considered safer than those with more neighbors from C_{maj}. Evidently, the neighbors from the example's class C_{min1} are the most preferred ones to construct safe neighborhoods. The strength of this preference between having neighbors from different classes in the neighborhoods of class C_{min1} is modeled with the similarity degrees between C_{min1} and other classes. In this example, the similarity between C_{min1} and C_{min2} should be set to a higher value than the similarity between C_{min1} and C_{maj}, and the similarity between the class itself should be fixed to the highest possible value.

More formally, we assume that the degree of similarity $\mu_{i,j}$ between each pair of classes C_i and C_j can be defined. We restrict the possible values of similarity degree $\mu_{i,j}$ to be in the range from 0 to 1 ($\mu_{i,j} \in [0, 1]$). The similarity between class itself is a priori set to its maximal value $\mu_{i,i} = 1$ and, for simplicity, in this work, we assume that the similarity degrees are symmetric, i.e., $\mu_{i,j} = \mu_{j,i}$.

Even though the degrees of similarity can be defined individually for each dataset, our general recommendation is to select higher values of similarity between minority classes ($\mu_{min1,min2} \rightarrow 1$) and relatively low values of similarity degrees between majority and minority classes ($\mu_{min,maj} \rightarrow 0$). Moreover, to capture the specific characteristics of overlapping with intermediate classes, one should select $\mu_{i,j}$ values in such a way that similarity between smaller minority class and the intermediate class is higher than the similarity between intermediate and majority class ($\mu_{min,inter} > \mu_{inter,maj}$). These recommendations come from our observations from Chapter 3.

Apart from these general recommendations, we also propose a heuristic formula for $\mu_{i,j}$ that not only produces similarity configuration consistent with our guidelines but also captures additional intuition that overlapping between classes with higher disproportion in their cardinalities should be treated as less safe than the overlapping between classes of similar size (e.g. two minority classes). The proposed formula is as follows:

$$\mu_{i,j} = \frac{\min\{|C_i|, |C_j|\}}{\max\{|C_i|, |C_j|\}} \tag{4.2}$$

where $|C_i|$ denotes the cardinality of class i.

Example. To better illustrate the working of this formula, consider the standard `car` dataset from the UCI repository, which contains four classes. The cardinalities of these classes are as follows: $|C_1| = 65$, $|C_2| = 69$, $|C_3| = 384$ and $|C_4| = 1210$. Following the recommendations mentioned above, the similarity between minority classes should be high and effectively $\mu_{1,2}$ is equal to 0.94 according to the presented heuristic. On the other hand, the similarity degree between minority classes and majority one is of only $\mu_{1,4} \approx \mu_{2,4} \approx 0.05$ as expected. Our heuristic also captures the fact that intermediate class examples can act as majority ones in minority class regions and as minority ones in a region with a high prevalence of majority examples. The similarity values between intermediate class C_3 and minority classes are approximately equal to $\mu_{1,3} \approx \mu_{2,3} \approx 0.17$, whereas the similarity to the majority class is $\mu_{3,4} \approx 0.32$.

Having defined the degree of similarity, we present the modified formula for computing the safe level of an example in multi-class imbalanced dataset.

$$safe\_level(x) = \frac{1}{k} \sum_{c \in C} k_c \mu_{C(x),c} \tag{4.3}$$

where k is the size of the analyzed neighborhood, k_c is the number of neighbors belonging to the class c, C is the set of all classes and $\mu_{C(x),c}$ is the similarity degree between class c and the true class of example x.

Example. Returning to the example from the beginning of this section, we will revisit the situations from Table 4.1 additionally assuming that similarity between minority classes is set to 0.5 and their similarity to the majority class is set to 0. Then, the safe level for the situation (a) is equal to

$$safe\_level(x_a) = \frac{1}{5}(5 \times 1 + 0 \times 0.5 + 0 \times 0) = 1$$

i.e., the highest possible safety. Note, that the similarity of class C_{min1} to itself is equal to 1. This was the easiest considered scenario, whereas cases (b) and (c) were more complex. Recall that (c) should be recognized as more difficult than (b).

$$safe\_level(x_b) = \frac{1}{5}(1 \times 1 + 2 \times 0.5 + 2 \times 0) = 0.4$$

$$safe\_level(x_c) = \frac{1}{5}(1 \times 1 + 0 \times 0.5 + 4 \times 0) = 0.2$$

To sum up, the situation (a) is recognizes as the most safe one, the next easiest situation is (b) and the most difficult situation among considered is (c), as expected.

4.2.2 Experimental study

We conclude this section by presenting a computational experiment on three real datasets in which we calculate the proposed safe levels and show their relation to classification performance. In the experiment, we consider three popular imbalanced datasets from UCI repository: `new thyroid`, `ecoli`, and `cleveland` which binarized versions were used in many related works. According to these studies, they represent different levels of difficulty [123, 138]. `Cleveland` was identified as a difficult dataset with a high amount of outlier and rare examples, `ecoli` has borderline characteristics, and `new thyroid` has primarily safe examples. The basic information about these datasets is presented in Table 4.2.

We computed safe levels with two configurations of similarity degrees. In the first configuration, all similarity degrees $\mu_{i,j}$ are set to 0, except similarity degrees to the class itself that by definition are equal to 1. This configuration gives the same results as the baseline solution, i.e., applying the definition of safe level for binary data directly to multi-class data, hence we will refer to it as the binary safe level. The second configuration assumes that the similarity degree between minority classes is equal to $\mu_{min1,min2} = 0.8$ and no similarity between minority and majority classes ($\mu_{min,maj} = 0$). This configuration will be referred to as the multi-class safe level.

The average safe levels for each class are presented in Table 4.4. Additionally, the recalls of minority classes obtained by decision tree (C4.5), decision rules (PART), Naive Bayes, and 3-nearest neighbors classifier are given in Table 4.5. All classifiers were used with default values of hyperparameters as implemented in WEKA framework [48]. The results were obtained through 5-fold stratified cross-validation.

For the `new thyroid` dataset, we can observe that the safe levels for both similarity configurations are the same, which indicates that minority classes do not overlap. Effectively, looking at the multidimensional scaling (MDS) [80] visualization of this dataset in Figure 4.1, we can confirm this observation. High values of the safe level also correspond to high recall values for all minority classes, which are among the highest recalls obtained in this study for all classifiers.

On the visualization of `ecoli` dataset, we can observe that its class overlapping structure is more complicated. The minority classes Min1 and Min3 overlap with the majority

Table 4.2: Total number of examples in considered real datasets and the cardinalities of minority classes. For simplicity, all majority classes were merged into one majority class in this experiment.

		Min1		Min2		Min3	
dataset	Size	name	size	name	size	name	size
new thyroid	215	2	35	3	30		
ecoli	336	imU	35	om	20	pp	52
cleveland	303	2	36	3	35	4	13

Table 4.3: Class similarity configurations used in the experiment

Configuration name	$\mu_{min1,min2}$	$\mu_{min,maj}$	$\mu_{maj1,maj2}$
binary safe level	0	0	0
multi-class safe level	0.8	0	0

Table 4.4: Average safe levels for studied datasets and similarity configurations.

	binary safe levels			multi-class safe levels		
dataset	Min1	Min2	Min3	Min1	Min2	Min3
new thyroid	0.77	0.78		0.77	0.78	
ecoli	0.57	0.74	0.82	0.57	0.91	0.86
cleveland	0.14	0.13	0.08	0.29	0.32	0.34

class, and additionally Min3 also overlaps with another minority class Min2. We can deduce these relations also from the results of safe level values. The safe levels for Min1 are again the same for both similarity configurations as it only overlaps with the majority class. The binary safe level for Min2 is much lower than the multi-class safe level, which indicates that it overlaps only with another minority class that causes smaller drops in predictive performance. Similarly, the multi-class safe level is slightly higher than the binary one for Min3 because it also partially overlaps with another minority class. This dataset is also more difficult to classify than `new thyroid`, with class Min1 being the most challenging class for all classifiers (except Naive Bayes).

According to average safe levels, `cleveland` dataset is the most difficult one, which is also reflected in the reported low recall values for all classifiers. Also, in Figure 4.1 one can see that the overlapping is very high, with the examples from minority classes having neighborhoods with other minority and majority classes.

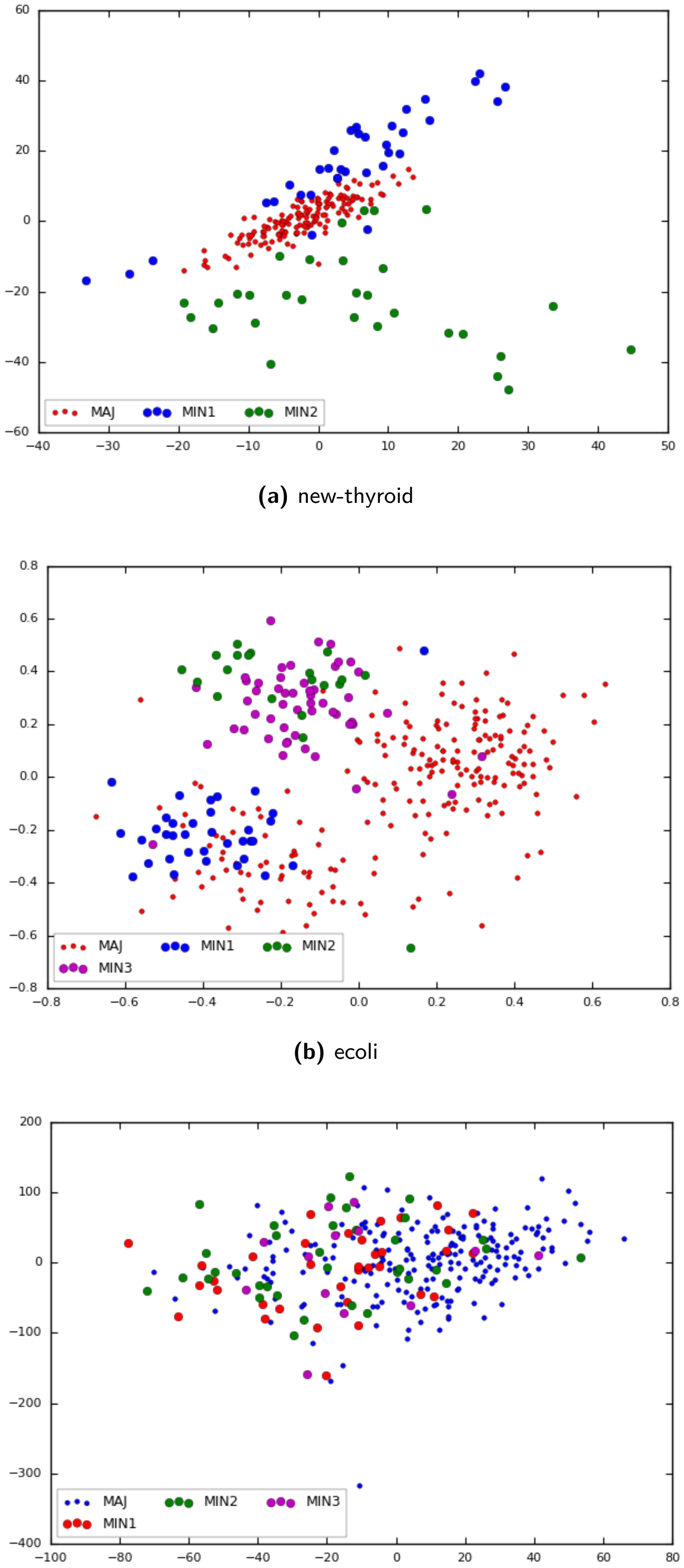
4.3 Proposed Method

To further show the usefulness of the proposed method for detecting local data difficulty factors in multi-class imbalanced datasets, we propose a new resampling algorithm called *Similarity Oversampling and Undersampling Preprocessing* (SOUP) that apart from global difficulty factors also exploits local information provided by safe levels.

The method constructs a balanced dataset by resampling all classes to the same size that is equal to the average taken from the sizes of the biggest minority and the smallest majority class. Such a selected post-resampling class size means that the method needs to undersample majority classes while oversampling minority ones. Such method design

Table 4.5: Minority class recalls for studied datasets and classifiers.

	CART			Naive Bayes			3NN			PART		
dataset	Min1	Min2	Min3	Min1	Min2	Min3	Min1	Min2	Min3	Min1	Min2	Min3
new thyroid	0.94	0.83		0.94	0.86		0.71	0.80		0.94	0.83	
ecoli	0.60	0.85	0.78	0.68	0.30	0.90	0.48	0.75	0.84	0.46	0.80	0.79
cleveland	0.28	0.11	0.07	0.14	0.25	0.15	0.08	0.00	0.00	0.20	0.11	0.08



(a) new-thyroid

(b) ecoli

(c) cleveland

Figure 4.1: MDS visualizations of studied imbalanced datasets, from left to right: `new-thyroid`, `ecoli` and `cleveland`. All MDS visualizations have stress value below 0.1 which is usually considered as a good fit [141].

is partially inspired by SCUT method (see Sec. 2.4.2.2) and comes from the observation that both under- and oversampling have their drawbacks. Undersampling can remove important information about the majority classes distribution from the dataset, whereas oversampling can make noise and outlier examples too strongly pronounced. These drawbacks are much more harmful to classifier performance when the differences between class sizes are large, i.e., the number of removed or oversampled examples is very high. Therefore, by resampling the classes to the aforementioned average value, we partially limit their negative impact while constructing a fully balanced dataset.

SOUP under- and oversamples training examples in a principled way: it removes the most harmful majority examples and strengths the representation of clear minority regions. First, the majority classes are processed from the largest to the smallest. The examples with the lowest safe levels are removed from each class until the required number of examples is reached. Therefore, the algorithm concentrates on removing examples that are close to minority ones, cleaning the overlapping region from majority examples.

Then, the method oversamples minority classes starting from the smallest class to the biggest by duplicating the instances with the highest safe level. All examples of a minority class are stored in a list sorted according to the safe level. The algorithm iterates over the list, duplicating currently processed examples, until the class size reaches the required number of examples. If the method reaches the last element in the list, which means that the number of examples to generate is bigger than the class size, the method continues its operation by starting from the beginning of the list.

The pseudocode of SOUP can be found in Algorithm 1. The computational complexity of the method is $O(n^2 + n|C|)$ where n is the total number of instances and $|C|$ is the number of classes. The computational complexity can be reduced by using special data structures for nearest neighbors search known from the machine learning and big data literature. For instance, while using kd-trees [12] the computational complexity in the average case is of $O(n \log n + n|C|)$.

4.4 Experiments

4.4.1 Setup

The experiments described in this section have the following goals:

- to check how the similarity degree values used in safe levels computation impact the SOUP performance,
- to compare the classification performance of the most relevant approaches for multi-class imbalanced data,
- to compare selected most related approaches with SOUP in terms of classification performance.

The methods were tested with three popular classifiers: C4.5 decision tree, PART decision rules, and k-nearest neighbors. These methods were selected as they are sensitive to the class imbalance and can be conveniently used with both preprocessing methods and decomposition ensembles. They are also typically used in related experimental studies

Algorithm 1 Similarity Oversampling and Undersampling Preprocessing (SOUP)

Input: D: original training set of $|D|$ examples with $|C|$ classes; C_{min}: indexes of minority classes; C_{maj}: indexes of majority classes; μ_{ij} similarities between classes
Output: D': balanced training set

```
Split dataset $D$ into $|C|$ homogeneous parts $D_1, D_2, ..., D_{|C|}$. Each $D_i$ contains all examples from $i$ class
$D' \leftarrow \emptyset$
$m \leftarrow mean(\min_{i \in C_{maj}} |D_i|, \max_{j \in C_{min}} |D_j|)$
for all $i \in C_{maj}$ do
    for all $x \in D_i$ do
        find $k$ nearest neighbors of $x$
        calculate safe level of $x$, according to Eq. 4.3
    end for
    sort examples in $D_i$ by the decreasing value of safe level
    remove $|D_i| - m$ examples with the lowest safe level values from $D_i$
    $D' \leftarrow D' \cup D_i$
end for
for all $j \in C_{min}$ do
    for all $x \in D_j$ do
        find $k$ nearest neighbors of $x$
        calculate safe level of $x$, according to Eq. 4.3
    end for
    sort examples in $D_i$ by the decreasing value of safe level
    duplicate $m - |D_i|$ examples with the highest safe level values in $D_j$
    $D' \leftarrow D' \cup D_j$
end for
return $D'$
```

with imbalanced data, e.g. [40, 38]. The hyperparameters of the classifiers were selected to default values in the WEKA framework, except for $k = 3$ in kNN and pruning deactivation in decision trees since we have found in our earlier studies that they generally offer better performance.

SOUP was used with similarity configurations (SIM1-6) described in Table 4.6 and with similarity degrees obtained by earlier presented heuristic (SIM Heur, see Eq. 4.2). Selected similarity configurations contain configurations from our earlier study (SIM1-2) [87], baseline configuration (i.e., all similarities are equal to 0, SIM3), and configurations presenting extreme cases (SIM4-6).

Table 4.6: Different configurations of similarity degrees

Similarity	$\mu_{min1\ min2}$	$\mu_{min\ maj}$	$\mu_{maj1\ maj2}$
SIM1	0.8	0	0.1
SIM2	0.7	0.15	0.2
SIM3	0	0	0
SIM4	1.0	0	1.0
SIM5	0	0.5	0
SIM6	1.0	0	0

The classification performance will be evaluated on 19 datasets, presented in Table 4.7. Our testbed includes fifteen real-world datasets selected from popular UCI and KEEL repositories, as they are often used in related studies on imbalanced data. They come from a broad spectrum of application areas and present different imbalanced ratios and numbers of features. The datasets used in the experiment also contain four artificial datasets, that present different difficulties for learning classifiers: `art1` has two minority classes and one majority class with no overlapping, `art2` is the same dataset but with additional overlap between minority and majority classes, `art3` introduces even stronger overlap between minority class, and `art4` additionally has rare and outlier examples belonging to both minority classes. Visualizations and further descriptions of these artificial datasets can be found in [87], the description of the used data generator can be found in [160].

As indicated by our analysis in Sec. 3.5, this is also a diversified collection that contains datasets with studied data difficulty factors, including different levels and types of class overlapping as well as various class configurations: multi-majority, multi-minority, and gradual with intermediate classes. Diversified difficulty of this data can be well seen when one calculates average safe levels separately in minority and majority classes. These averaged safe levels for all datasets are presented in Table 4.8.

Due to the significant amount of numerical results obtained during the experiments, only selected results on G-mean measure are presented in this work. The extended version of the results (including measurements of average accuracy and F-score) can be found in the supplementary material[1]. G-mean is selected for more detailed discussion as it has some important theoretical advantages [19], is often used in practice, and has an intuitive interpretation.

4.4.2 Classification performance of SOUP with various similarity setting

Table 4.9 shows results of decision tree classifier on G-mean measure for SOUP used with different similarity configurations described in Table 4.6. The result of post-hoc Nemenyi analysis of Friedman test ranks is presented in Figure 4.2.

First, we can observe that SOUP in all similarity configurations improves over the baseline, i.e. lack of preprocessing. We also notice significant differences in classification performance on individual datasets while using different similarity configurations, particularly on more challenging datasets. The most significant difference in SOUP performance is on `cleveland2` dataset, where changing similarity configuration can lead to over 30% of improvement for the decision tree. On this dataset, the difference for kNN is 10% and 35% for PART. Different similarity configurations works best for different datasets, for instance SIM6 lead to superior results for `wine quality` or `cleveland2`, SIM5 for `glass`, `cmc` or `flare` etc.

Despite differences observed on individual datasets, the global Friedman rank test does not detect statistically significant differences averaged over all datasets. The best performing configurations among all datasets are configurations obtained by the proposed heuristic, inspired by our earlier observations from Chapter 3. The worst performing

[1] http://www.cs.put.poznan.pl/mlango/publications/soup.html

Table 4.7: Characteristics of multi-class imbalanced datasets. Names of classes are given in the first row, while their cardinalities in the second row.

dataset	Minority classes					Majority classes				
balance scale	B					L	R			
	49					288	288			
car	good	vgood				unacc	acc			
	69	65				1210	384			
cleveland_1	1	2	3	4		0				
	55	36	35	13		164				
cleveland_2	2	3	4			0+1				
	36	35	13			219				
cmc	2					1	3			
	333					629	511			
dermatology	6					1	2	3	4	5
	20					112	61	72	49	52
ecoli	pp	imUimS	omomL			cpimL	im			
	52	37	25			145	77			
flare	4	5				1	2	3	6	
	116	51				212	287	327	396	
glass	vwf	con	tab			bwf	bwnf	head		
	17	13	9			70	76	29		
hayes roth	3					1	2			
	31					65	64			
led7digit	5	10				1	2	3	6	
	52	49				98	94	108	99	
new thyroid	2	3				1				
	35	30				150				
vehicle	bus	van				opel_saab				
	218	199				429				
yeast	2	3	5	6	7	1	8	9	10	
	20	30	35	44	51	463	168	244	429	
wine quality	7	8				5	6			
	199	81				681	638			
art1	MIN1	MIN2				MAJ				
	120	240				840				
art2	MIN1	MIN2				MAJ				
	120	240				840				
art3	MIN1	MIN2				MAJ				
	120	240				840				
art4	MIN1	MIN2				MAJ				
	120	240				840				

Table 4.8: Average safe levels for all Minority and all Majority classes calculated for SIM2 class similarities.

dataset	Minority	Majority
balance scale	0.16388	0.88009
car	0.90716	0.96745
cleveland_1	0.62374	0.83104
cleveland_2	0.51762	0.89210
cmc	0.45580	0.59982
dermatology	0.96488	0.97110
ecoli	0.66316	0.83252
flare	0.48246	0.78493
glass	0.51795	0.74309
hayes roth	0.45710	0.66891
led7digit	0.76267	0.77206
new thyroid	0.85092	0.98073
vehicle	0.89434	0.89142
wine quality	0.46754	0.64299
yeast	0.56089	0.60316
art1	0.94994	0.96924
art2	0.77986	0.92512
art3	0.61383	0.84882
art4	0.79836	0.90873

configurations are SIM1, SIM2, and SIM4, the only ones in our study that assume some level of similarity between majority classes. Note that even the worst-performing SOUP configuration has a much higher rank than the baseline.

4.4.3 Comparison of related multi-class imbalanced learning methods

To compare the classification performance of SOUP with the best of the related methods, we decided to first perform an experimental evaluation of both preprocessing and decomposition methods in order to select the best performing methods for further comparison. We compared the results of Global CS, Static-SMOTE, and decomposition methods: OVA and OVO used together with random oversampling (ROS), random undersampling (RUS),

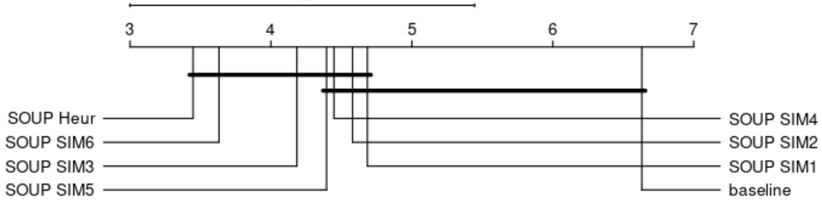


Figure 4.2: Visualization of Nemenyi post-hoc analysis results for SOUP with different similarity degree configurations on G-mean measure and decision tree classifier.

Table 4.9: G-mean values obtained by a decision tree learned on datasets preprocessed with SOUP using different similarity configurations.

	SOUP						
dataset	SIM1	SIM2	SIM3	SIM4	SIM5	SIM6	Heur
balance scale	0.598	0.614	0.598	0.576	0.598	0.598	0.585
car	0.938	0.938	0.938	0.932	0.938	0.940	0.941
cleveland_1	0.272	0.285	0.285	0.208	0.275	0.208	0.266
cleveland_2	0.256	0.256	0.000	0.305	0.000	0.305	0.303
cmc	0.520	0.520	0.528	0.518	0.531	0.528	0.535
dermatology	0.960	0.960	0.960	0.960	0.960	0.960	0.962
ecoli	0.717	0.721	0.762	0.745	0.710	0.727	0.735
flare	0.586	0.575	0.573	0.560	0.568	0.571	0.566
glass	0.662	0.667	0.672	0.715	0.671	0.672	0.667
hayes roth	0.818	0.817	0.828	0.825	0.833	0.828	0.835
led7digit	0.780	0.790	0.772	0.772	0.774	0.777	0.778
thyroid	0.922	0.922	0.922	0.922	0.922	0.922	0.922
vehicle	0.909	0.909	0.909	0.910	0.909	0.910	0.915
wine quality	0.449	0.448	0.465	0.454	0.469	0.482	0.472
yeast	0.371	0.398	0.407	0.464	0.434	0.410	0.451
art1	0.960	0.960	0.960	0.960	0.960	0.960	0.960
art2	0.777	0.777	0.777	0.793	0.779	0.793	0.777
art3	0.609	0.594	0.605	0.605	0.602	0.605	0.608
art4	0.899	0.899	0.899	0.899	0.899	0.899	0.899

and neighborhood cleaning rule (NCR)[2]. The results for C4.5 classifier are presented in Table 4.10. The Friedman rank test indicated statistically significant differences ($p < 0.001$). The performed post-hoc Nemenyi analysis is summarized in Figure 4.3.

According to the average rank on G-mean measure, the best-performing methods were ex aequo OVO ensemble with random oversampling and OVO with random undersampling. In general, approaches based on OVO decomposition were much more effective than those based on OVA, which were even worse than the baseline. Among prepossessing methods, Global CS was the most effective (4.11), which results were followed by Static SMOTE (5.65). According to the post-hoc analysis, the results of both these methods were not statistically different from the best OVO approaches. Therefore, in further comparisons, all these approaches will be taken into account.

4.4.4 Comparing the classification performance of SOUP with other approaches

The evaluation of SOUP performance will be performed twofold. First, we will compare SOUP with other pre-processing methods: Global-CS and Static-SMOTE. Then, the performance of SOUP will be compared with the best ensemble approaches identified in the previous section: OVO with ROS, OVO with RUS, and OVO with NCR.

Furthermore, we investigate the possibility of using SOUP-inspired resampling in OVO ensembles. To this end, we developed two binary resampling algorithms: Similarity Oversampling (SO, see Alg. 2) and Similarity Undersampling (SU, Alg. 3). These algorithms were constructed by extracting from SOUP the respective under- and over-sampling parts. SOUP is used with similarity configuration calculated by the proposed heuristic since it provided the best results and was also inspired by our earlier analysis of data difficulty factors. Since SU and SO methods are for binary classification, they use binary safe levels (Eq. 4.1), i.e., without class similarities.

The left part of Table 4.11 shows G-mean values obtained by the C4.5 classifier learned on datasets preprocessed with various methods. P-values of performed paired Wilcoxon rank tests between SOUP and other preprocessing methods can be seen in Table 4.12 for C4.5, PART, and kNN classifiers.

The results demonstrate that the best-performing preprocessing method is SOUP. The difference between SOUP and other resampling methods is statistically significant for all classifiers with $\alpha = 5\%$ (except Global-CS and PART classifier). On the C4.5 decision

[2] see Chapter 2 for references and descriptions of these methods

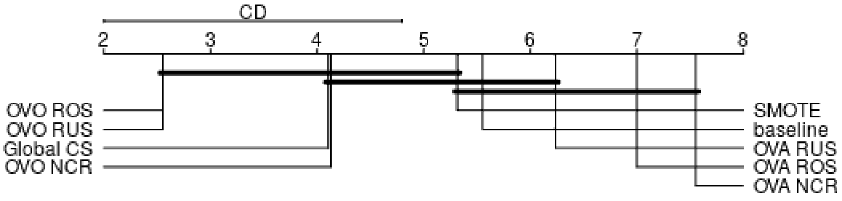


Figure 4.3: Visualization of Nemenyi post-hoc analysis results for preprocessing and decomposition methods.

Table 4.10: Comparison of G-mean for decomposition methods and Global-CS for using decision trees as a basic classifier.

dataset	baseline	Global CS	Static SMOTE	OVO ROS	OVO RUS	OVO NCR	OVA ROS	OVA RUS	OVA NCR
balance scale	0.000	0.340	0.080	0.526	0.602	0.474	0.302	0.297	0.000
car	0.847	0.940	0.897	0.939	0.876	0.919	0.112	0.184	0.130
cleveland_1	0.227	0.000	0.052	0.255	0.287	0.262	0.254	0.259	0.000
cleveland_2	0.000	0.000	0.037	0.288	0.285	0.000	0.280	0.287	0.000
cmc	0.483	0.478	0.452	0.509	0.514	0.526	0.510	0.511	0.529
dermatology	0.945	0.952	0.927	0.921	0.929	0.948	0.000	0.000	0.000
ecoli	0.728	0.710	0.738	0.805	0.767	0.000	0.000	0.000	0.000
flare	0.446	0.570	0.421	0.544	0.568	0.522	0.000	0.000	0.000
glass	0.625	0.715	0.322	0.699	0.697	0.691	0.000	0.000	0.000
hayes roth	0.843	0.832	0.835	0.843	0.843	0.838	0.000	0.000	0.000
led7digit	0.786	0.770	0.756	0.771	0.779	0.722	0.120	0.162	0.156
thyroid	0.889	0.922	0.879	0.922	0.886	0.913	0.904	0.927	0.898
vehicle	0.912	0.912	0.915	0.916	0.923	0.915	0.133	0.141	0.164
wine quality	0.432	0.464	0.356	0.492	0.476	0.434	0.459	0.489	0.356
yeast	0.000	0.406	0.184	0.442	0.479	0.000	0.000	0.000	0.000
art1	0.945	0.961	0.947	0.958	0.949	0.949	0.039	0.000	0.039
art2	0.686	0.734	0.741	0.758	0.777	0.762	0.250	0.253	0.244
art3	0.410	0.534	0.535	0.615	0.612	0.559	0.307	0.304	0.236
art4	0.785	0.829	0.856	0.840	0.872	0.839	0.000	0.000	0.000

Algorithm 2 Similarity Oversampling (SO)

Input: D: original training set of $|D|$ examples with two classes;
Output: D': balanced training set

```
Split dataset $D$ in two parts containing examples from a single class, denoted $D_{min}$ and $D_{maj}$
$D' \leftarrow \emptyset$
$diff \leftarrow |D_{maj}| - |D_{min}|$
if $diff > 0$ then
    for all $x \in D_{min}$ do
        find $k$ nearest neighbors of $x$
        calculate safe level of $x$
    end for
    duplicate $diff$ examples with the highest safe level values from $D_{min}$
end if
$D' = D_{maj} \cup D_{min}$
return $D'$
```

Algorithm 3 Similarity Undersampling (SU)

```
Input: $D$: original training set of $|D|$ examples with two classes;
Output: $D'$: balanced training set
Split dataset $D$ in two parts containing examples from a single class, denoted $D_{min}$ and $D_{maj}$
$D' \leftarrow \emptyset$
$diff \leftarrow |D_{maj}| - |D_{min}|$
if $diff > 0$ then
    for all $x \in D_{maj}$ do
        find $k$ nearest neighbors of $x$
        calculate safe level of $x$
    end for
    remove $diff$ examples with the lowest safe level values from $D_{maj}$
end if
$D' = D_{maj} \cup D_{min}$
return $D'$
```

tree classifier, SOUP obtains on average 5.4% improvement with respect to the second-best performing method, i.e., Global-CS in terms of G-mean. It also yields better results on the kNN classifier, where the improvement over Global-CS is on average 4.9% with the median of 1.3%. Similarly, on PART classifier (where the result was not statistically significant), SOUP offered an improvement of 0.8% (both median and average).

We proceed with the comparison of SOUP with the best decomposition methods as well as decompositions with similarity resampling (SU, SO). The right part of Table 4.11 presents the results of G-mean obtained by the decision tree classifier. The Friedman rank test indicated statistically significant differences with $\alpha = 5\%$. The result of post-hoc Nemenyi analysis is presented in Figure 4.4. The averaged ranks obtained by the compared methods on C4.5, PART and kNN are presented in Table 4.12.

SOUP obtained the lowest average rank for all classifiers. For decision tree it was followed by OVO RUS and three methods of quite similar ranks: OVO SO, OVO SU, and OVO ROS. OVO with NCR obtained the highest (i.e. the worst) rank. While using kNN, the second-best performing method was OVO with SU, followed by OVO NCR ex aqueo with OVO RUS. Nevertheless, the difference in average rank between SOUP and OVO SU is relatively high (2.5 vs. 3.45). Finally, on PART classifier the best result of SOUP is followed by OVO RUS and OVO SO. Quite comparative results of OVO ensembles with random resampling and resampling based on safe levels is an indicator that the high performance of SOUP comes from exploiting class interrelations in the proposed safe levels with class similarities and not merely from incorporating standard safety information.

4.5 Discussion

In this chapter, we have proposed a new resampling method called Similarity Oversampling and Undersampling Preprocessing. Contrary to previous approaches for multi-class imbalanced data, this method exploits the observations about data difficulty factors in multi-class imbalanced data observed in the previous chapter. More concretely, it uses the

Table 4.11: Comparison of best methods and SOUP with tree J48 algorithm and G-mean.

dataset	Global	Static	SOUP	OVO	OVO	OVO	OVO	OVO
	CS	SMOTE		ROS	RUS	NCR	SO	SU
balance scale	0.340	0.080	0.585	0.526	0.602	0.474	0.542	0.547
car	0.940	0.897	0.941	0.939	0.876	0.919	0.940	0.794
cleveland_1	0.000	0.052	0.266	0.255	0.287	0.262	0.268	0.302
cleveland_2	0.000	0.037	0.303	0.288	0.285	0.000	0.284	0.312
cmc	0.478	0.452	0.535	0.509	0.514	0.526	0.522	0.524
dermatology	0.952	0.927	0.962	0.921	0.929	0.948	0.925	0.939
ecoli	0.710	0.738	0.735	0.805	0.767	0.000	0.791	0.739
flare	0.570	0.421	0.566	0.544	0.568	0.522	0.582	0.506
glass	0.715	0.322	0.667	0.699	0.697	0.691	0.701	0.697
hayes roth	0.832	0.835	0.835	0.843	0.843	0.838	0.843	0.775
led7digit	0.770	0.756	0.778	0.771	0.779	0.722	0.765	0.704
thyroid	0.922	0.879	0.922	0.922	0.886	0.913	0.897	0.896
vehicle	0.912	0.915	0.915	0.916	0.923	0.915	0.904	0.880
wine quality	0.464	0.356	0.471	0.492	0.476	0.434	0.524	0.490
yeast	0.406	0.184	0.451	0.442	0.479	0.000	0.000	0.484
art1	0.961	0.947	0.960	0.958	0.949	0.949	0.959	0.951
art2	0.734	0.741	0.777	0.758	0.777	0.762	0.754	0.804
art3	0.534	0.535	0.608	0.615	0.612	0.559	0.627	0.634
art4	0.829	0.856	0.899	0.840	0.872	0.839	0.831	0.878

Table 4.12: p-values of the paired Wilcoxon signed rank test between SOUP and other preprocessing methods on G-mean measure for various classifiers.

classifier	baseline	Global-CS	Static-SMOTE
C4.5	< 0.001	0.036	< 0.001
PART	< 0.001	0.153	< 0.001
kNN	< 0.001	0.005	0.002

Table 4.13: The average rank of compared algorithms (the lower, the better) from Friedman tests on G-mean measure for various classifiers.

classifier	SOUP	OVO RUS	OVO SO	OVO ROS	OVO SU	OVO NCR
C4.5	2.87	3.18	3.39	3.42	3.45	4.68
PART	3.25	3.35	3.35	3.7	3.6	3.75
kNN	2.5	3.55	3.95	4.0	3.45	3.55

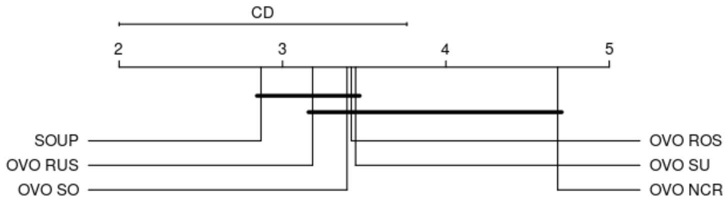


Figure 4.4: Visualization of Nemenyi post-hoc analysis results for SOUP and best ensemble methods.

information about imbalanced ratio, class overlapping as well as class sizes configuration. The usage of the last one was enabled by extending an existing method for examples' safe level computation with the class similarity degrees. The proposed extension allows modeling class interrelations and, similarly to its predecessor, can be used to detect and measure class overlapping or prevalence of outliers. The provided safe levels for multi-class imbalanced data can also be further discretized into example types, enabling similar analysis as in the case of binary imbalanced problems.

The performed experimental analysis demonstrated that exploiting information about data difficulty factors through SOUP leads to very good classification performance. The results indicated that the proposed method obtains better G-mean scores than other preprocessing methods. It also has higher classification performance than the best decomposition ensembles on three popular classifiers.

The performance gains in classification performance come at the price of higher computational complexity in comparison with random resampling methods such as Global-CS. However, this complexity is on par with other informed resampling approaches like Static-SMOTE. The method requires additional parameters, namely class similarity degrees $\mu_{i,j}$ that, in practice, can cause an additional effort of fine-tuning. Nevertheless, this issue was mitigated as a simple heuristic that allows automatic computation of these parameters was presented. This heuristic was successfully used in our final comparisons with related methods, obtaining superior results. Therefore, fine-tuning similarity degrees for a particular dataset can lead to even better results but is not required to obtain satisfactory results with SOUP.

Chapter 5

Bagging-based Ensemble Methods For Multi-class Imbalanced Data

5.1 Motivation

Classification ensembles are very effective learning methods that exploit the possibility of improving classification performance by constructing several classifiers that work together. As mentioned in Chapter 2, ensembles are also very useful methods for dealing with class imbalance, obtaining superior classification results.

Most of the works on ensembles for imbalanced data, mainly concerning binary classification problems, are extensions of bagging and boosting. Such ensembles are usually constructed by coupling one of these ensembling techniques with data resampling methods. Even though there are not many detailed comparison studies of ensembles for imbalanced data, the existing experimental studies [43, 69] demonstrated that the generalizations of bagging work better than boosting-based approaches. It was also noticed by Khoshgoftaar et al. [69] that bagging gains a particularly strong advantage over boosting on more difficult imbalanced data with noise. Other studies [15, 137] have shown that undersampling strategies used within the bagging framework are more effective than oversampling ones. According to these studies, one of the most effective ensemble methods for binary imbalanced data is Roughly Balanced Bagging (RBBag) [56] that incorporates a particular random undersampling to bagging ensemble.

In this chapter, we will propose two new bagging-based ensembles for multi-class imbalanced data. First, we will extend the very effective RBBag for binary imbalanced classification to the multi-class setting. The proposed reformulation of the RBBag sampling strategy allows its usage both as under- and oversampling method. Then, we will propose a bagging-based extension that incorporates our Similarity Oversampling and Undersampling Preprocessing (SOUP), verifying if it can be used to construct effective ensembles for multi-class imbalanced data.

5.2 Proposed Methods

5.2.1 Multi-class Roughly Balanced Bagging

Classical bagging [17] constructs an ensemble by running the same learning algorithm on many different training sets. These training sets are a result of resampling with replacement of the original dataset. Typically, the resampled datasets have the same size as the original dataset, therefore each contains on average $1-(1-\frac{1}{N})^N$ percent of the unique instances. For a large dataset, this quantity can be approximated by $1-\frac{1}{e}\approx 63.2\%$. While classifying a test instance, the final decision is made by averaging the outputs of all base (also called: component) classifiers.

Roughly Balanced Bagging (RBBag) [56] adjusted the original bagging to better handle binary imbalanced problems by replacing standard resampling with replacement with a form of undersampling. Each resampled dataset is constructed in the following way. First, N_+ minority instances are sampled with replacement from the set of minority class instances C_+. Then, majority instances are added to the dataset also with random resampling with replacement, but their number k is determined by a random draw from the negative binomial distribution. The probability mass function of this distribution is given by

$$P(X=k)=\binom{k+r-1}{r-1}(1-p)^k p^r$$

with the probability of success set to $p=0.5$ and number of failures fixed to $r=N_+$. This means that the training sets have different proportions of minority and majority examples, but on average they are balanced. Hence, the name of "roughly balanced" bagging.

Hido et al. [56] claimed that this approach better reflects the philosophy of bagging than most of the related bagging extensions for imbalanced data since in the original bagging the class proportions in samples are also varying. Nevertheless, we note that the classical bagging constructs samples of equal size, but RBBag uses resampling that outputs datasets of different sizes. Similarly, the number of minority class examples is constant in RBBag, whereas in the standard bagging it varies. These issues will be solved by our proposal of Multi-class Roughly Balanced Bagging that additionally extends RBBag to the multi-class setting.

Recall that in the standard bagging, the component classifiers' training sets are constructed by sampling instances from the uniform joint distribution $P(x,c)$ where x is an example of a class c. Using the chain rule, this distribution can be rewritten as $P(x,c)=P(c)P(x|c)$, suggesting a two-step sampling procedure that leads to the same results. More concretely, while sampling an instance, one can first select a class c from the distribution $P(c)$ i.e. selecting the class according to its frequency. Then, randomly select an element x only from the subset of selected class c instances. Note that the first step of the resampling procedure, i.e., selecting a class according to $P(c)$, is the only step that is sensitive to class imbalance. In Multi-class Roughly Balanced Bagging (MRBBag), we modify this step by selecting the class c from the uniform distribution.

The number of instances from each class in such resampled training set follows the multinomial distribution. The probability mass function of the multinomial distribution is given by the following equation:

$$P(n_1, n_2, ..., n_{|C|}) = \frac{n!}{n_1!n_2! \cdots n_{|C|}!} p_1^{n_1} p_2^{n_1} \cdots p_{|C|}^{n_{|C|}}$$

where $n = \sum_{i=1}^{|C|} n_i$ and $p_1, p_2, ..., p_{|C|} \geq 0$; $\sum_{i=1}^{|C|} p_i = 1$ are the parameters of the distribution.Therefore, MRBBag will construct training datasets by first drawing a sample from the multinomial distribution to determine how many instances need to be drawn from each class. Subsequently, the indicated number of examples from each class will be sampled randomly (with replacement).

To conclude the description of the algorithm, we have to establish the parameter values of the multinomial distribution. As mentioned earlier, in MRBBag we assume that the classes are selected with uniform frequency (i.e. the dataset is balanced), therefore we set $p_1 = p_2 = \cdots = p_{|C|} = \frac{1}{|C|}$. The parameter n of the distribution, which determines the size of the constructed sample, can also be considered as the parameter of the MRBBag algorithm since different its settings lead to under- or oversampling procedures.

Setting n parameter to the size of the dataset will cause the construction of training sets of the same size as the original dataset, just like in standard bagging. It will also cause oversampling of smaller classes since all classes will have the same number of instances, on average. Even though such a setting of n goes in line with the classical bagging, related works on binary imbalanced data indicated that undersampling bagging ensembles proved to be more effective. Therefore, another considered possibility is to set n to the size of the smallest minority class multiplied by the number of classes, which will cause the undersampling of bigger classes. In such a setting, each class should have, on average, the same size as the smallest minority class. Later, we refer to the first setting as oversampling MRBBag (oMRBBag) and to the latter as undersampling MRBBag (uMRBBag). The pseudocode of the algorithm is given in Algorithm 4.

5.2.2 SOUP-Bagging

As seen in the previous chapter, Similarity Oversampling and Undersampling Procedure (SOUP) is a very effective method for handling multi-class imbalanced data. Taking into account that many other preprocessing methods for imbalanced data were often later used to construct classification ensembles that frequently obtained even better results, the question about the possibility of further improvements by proposing a SOUP-based ensemble seems natural. In particular, combining resampling methods like SOUP with bagging seems very promising since, as we have mentioned in Sec. 5.1, bagging-based ensembles obtain better performance than boosting-based ensembles while working on difficult imbalanced data [69, 15, 137]. Moreover, combining bagging with methods that undersample majority classes brings the advantage of limiting the information loss from potentially removing important majority instances since different ensemble components can learn - due to resampling - from different majority instances. In this way, the ensemble utilizes all majority instances in the dataset while having components learned from balanced datasets.

Therefore, we propose the SOUP-Bagging algorithm that incorporates SOUP into the bagging framework. The algorithm trains its base classifiers on datasets preprocessed

Algorithm 4 Multi-class Roughly Balanced Bagging

```
Input: $D = \cup_{j=1}^{c} D_j$: original training set with $c$ classes, $N$: size of each bootstrap sample,
$k$: number of bootstrap samples, $LA$: learning algorithm;
Output: $C^*$ bagging ensemble with $k$ component classifiers
Learning phase:
for $i = 1 \rightarrow k$ do
    $S_i \leftarrow \emptyset$
    $[n_1, n_2, ..., n_c] \leftarrow$ following multinomial distribution with $n = N$ and $p_i = 1/c$ for $i = 1, 2, ..., c$
    for $j = 1 \rightarrow c$ do
        $S_{i,j} \leftarrow n_j$-element bootstrap sample drawn with replacement from $D_j$
        $S_i \leftarrow S_i \cup S_{i,j}$
    end for
    $C_i \leftarrow LA(S_i)$
end for
Prediction phase:
```

$$C^*(x) = \arg\max_y \sum_{i=1}^{k} p_{C_i}(y|x)$$

with the SOUP algorithm. Since SOUP is mostly deterministic procedure[1], to introduce the diversity of base classifiers, the dataset is first resampled with a standard, stratified sampling with replacement. The final prediction of the ensemble is made by averaging the decisions of base classifiers. The pseudocode of SOUP-Bagging is presented in Alg. 5.

SOUP-Bagging from the computational perspective is much more complex than MRBBag since the latter requires only random resampling of each class' examples with a specific proportion. Contrary, SOUP-Bagging exploits in each iteration SOUP that requires calculating safe levels through nearest neighbors search, which results, as mentioned in the previous chapter, in the complexity of $O(n^2 + n|C|)$. This makes MRBBag a significantly faster method, especially for large datasets. In fact, in one of our studies [83] performed on 12 real datasets, training uMRBBag with 100 decision trees was on average 30% faster than training a single decision tree on the original dataset and 40% faster than the fastest tested decomposition ensemble.

5.3 Experiments

In this section, the classification performance of the proposed algorithms will be evaluated. First, we will evaluate the effectiveness of the proposed Multi-class Roughly Balanced Bagging and compare its performance with a simplistic, straightforward extension of Roughly Balanced Bagging. We will also verify the differences in performance of the two proposed versions of MRBBag, namely oversampling MRBBag and undersampling MRBBag. Then, we will compare the performance of the best version of MRBBag with SOUP-Bagging and other state-of-the-art approaches identified in the previous chapter: SOUP, OVO RUS, OVO ROS, and OVO NCR.

[1] SOUP is deterministic except handling ties in safe level sorting or determining the set of nearest neighbors within equally-distant neighbors.

Algorithm 5 SOUP-Bagging

```
Input: D: original training set of examples of size N, k: number of bootstrap samples,
LA: learning algorithm;
Output: C* bagging ensemble with k component classifiers
Learning phase:
for $i = 1 \to k$ do
    $S_i \leftarrow N$-element sample drawn with replacement from $D$
    $S_i \leftarrow$ SOUP $(S_i)$
    $C_i \leftarrow LA(S_i)$
end for
Prediction phase:
    $C^*(x) = \arg\max_y \sum_{i=1}^{k} p_{C_i}(y|x)$
```

5.3.1 The comparison of the performance of Multi-class Roughly Balanced Bagging with Roughly Balanced Bagging

We begin the evaluation of MRBBag performance with a preliminary experiment on artificial datasets with different difficulty factors. The datasets have 900 examples generated by a method proposed in [160] and have one majority and two minority classes. The imbalance ratio is equal to 6. The smallest minority class has 100 instances, the second minority class has 200, and the remaining instances belong to the majority class. Each minority class has a predefined spherical shape within which the instances are randomly sampled. In the area surrounding these two spheres are sampled majority instances.

Three configurations of interactions between minority classes are considered:

- lack of overlapping between classes - the spheres have empty intersection,
- small overlapping – there is a slight overlapping on the borders of the spheres, involving approx. 10% of the smallest class instances,
- overlapping – the spheres have a larger common part of about 30% for the smallest class.

The last two versions of the dataset are visualized in Figure 5.1.

Additionally, the majority and minority instances are generated in a way ensuring for each minority class the same distribution of example types, as introduced by Napierala [123]. Since the dataset is imbalanced, Napierala's algorithm to identify examples types uses the safe level definition for multi-class data introduced in the previous chapter. The degrees of similarity between minority classes were set to its maximal value $\mu_{min1,min2} = 1$, and the similarities between other classes are set to 0. We consider three possible distributions of minority examples' types:

- 50% of safe examples, 35% of borderline, 10% of rare and 5% of outliers (later denoted as 50-35-10-5 configuration);
- 50% of safe, 30% of borderline, 10% of rare and 10% of outliers (50-30-10-10);

- 70% of safe and 30% of borderline examples (70-30-0-0).

In total, we obtained nine artificial datasets that combine three possible interactions of minority classes with majority classes and three overlapping configurations between minority classes.

The performance of MRBBag will be compared against a single C4.5 decision tree, the classical bagging, and a more simple modification of original Roughly Balanced Bagging (denoted as RBBag*), adjusted to multi-class data. To construct bootstrap samples, RBBag divides the training set into a set of minority examples and a set of majority examples, which are later used to sample from both classes separately and to set negative binomial distribution parameters. In the multi-class scenario, such partitioning of the dataset is not possible since there are many classes. To enable RBBag to handle multiple classes, we a) determine the negative binomial distribution parameters as if there were two classes only, constructed by merging all minority and all majority instances, respectively b) perform stratified sampling of minority and majority classes.

The results of G-mean measure obtained by averaging over five runs of 10-fold cross-validation, as well as the average rank for each method (as in the Friedman test), are presented in Table 5.1. All the ensembles had 30 component classifiers and used C4.5 unpruned decision trees as base learners. This particular number of components was chosen based on our earlier work [88] that investigated the impact of the number of base classifiers on the classification performance of imbalanced data, showing that RBBag ensemble does not need a high number of components to obtain good results.

The best performing ensemble in our experiment is undersampling MRBBag, followed by oversampling MRBBag and RBBag*. All these methods offered improvements over the classical bagging and single decision tree, which obtained the lowest average rank. RBBag* obtained a better G-mean value than our MRBBag only for one dataset that happened to be the simplest one, as it contained neither minority class overlapping nor rare and outlier examples. The classification performance of oMRBBag is comparable to that of RBBag*, which is reflected in very similar average rank values. oMRBBag obtained better results than RBBag* on 4 out of 9 datasets, but some of the differences are negligible. Undersampling MRBBag, on the other hand, is almost always better than RBBag*, and even when it is not, the differences are very small. On more difficult datasets the disparities between uMRBBag and RBBag* are considerably bigger. In particular, on

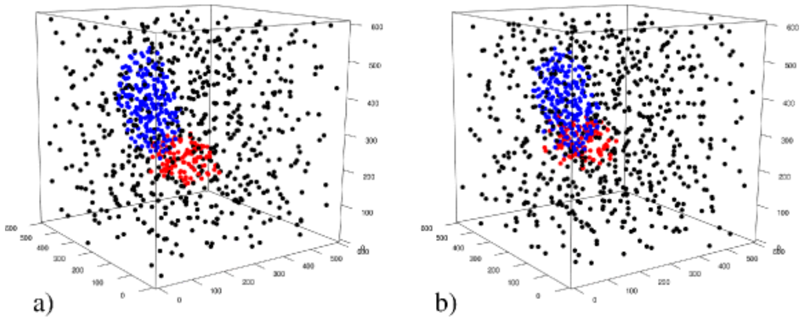


Figure 5.1: Visualization of two artificial datasets with different levels of overlapping: "small overlap" (a) and "overlap" (b). Minority classes in both datasets contains 70% of safe and 30% of borderline examples.

Table 5.1: G-mean and average ranks (the lower, the better) of MRBBag and related approaches on artificial datasets

Example types distribution	Minority class overlapping	Bagging	uMRBBag	oMRBBag	RBBag*	C4.5
50-30-10-10	no overlap	0,7863	0,8111	0,7980	0,8104	0,7853
50-30-10-10	small overlap	0,7735	0,7945	0,7989	0,7962	0,7666
50-30-10-10	overlap	0,6822	0,7293	0,6839	0,6955	0,6576
50-35-10-5	no overlap	0,8195	0,8523	0,8522	0,8444	0,8101
50-35-10-5	small overlap	0,8169	0,8438	0,8428	0,8345	0,8010
50-35-10-5	overlap	0,6686	0,7449	0,7193	0,7280	0,6168
70-30-0-0	no overlap	0,9401	0,9496	0,9525	0,9566	0,9340
70-30-0-0	small overlap	0,9154	0,9378	0,9402	0,9386	0,9045
70-30-0-0	overlap	0,8245	0,8707	0,8335	0,8439	0,8022
average rank		4,0000	1,6666	2,2222	2,1111	5,0000

the most difficult dataset (i.e., with the most prominent minority class overlapping and the highest number of hard examples - 50-30-10-10), uMRBBag outperforms RBBag* by more than 3%.

We have also performed an experiment to assess the performance of MRBBag on real datasets representing various levels of difficulty. We collected 15 datasets that represent three rough categories: easy i.e. mostly safe datasets with average safe level for minority class from 0.85 to 1.0 (see Chapter 4); intermediate with safe level from 0.6 to 0.85; and difficult with safe levels from 0.40 to 0.6. Some of the datasets were additionally preprocessed by removing classes with less than five examples (such datasets have added suffix `-sm` to its name) or merging examples into one common majority class, leaving only two selected minority classes (suffix `-3`). Finally, each category is represented by five datasets in our testbed. Other details of the experimental setup were left as in the previous experiment.

The results of G-mean obtained by bagging ensembles and a single decision tree are presented in Table 5.2. The Friedman rank test indicated significant differences between tested algorithms with a very small p-value ($p < 0.0001$). The post-hoc Nemenyi analysis confirms a significant difference between a single decision tree, the classical bagging, and specialized approaches. However, the obtained value of critical distance $CD = 1.575$ does not confirm statistically significant differences between RBBag extensions.

The method with the lowest (best) rank is undersampling MRBBag, followed by our oversampling MRBBag. The differences between uMRBBag and RBBag* are sometimes very significant e.g. over 10% for `glass-3` and `yeast-sm`, which belong to the dataset category with the lowest average safe levels. We observe that the differences between uMRBBag and RBBag* are higher for more difficult datasets. The average difference within the easy category is 0.005, indicating a rather negligible superiority of uMRBBag. However, the average difference for datasets of medium difficulty is 0.016 and 0.055 for difficult datasets, which can be significant in practice. The average improvement over all datasets is 0.025, the median is 0.015.

oMRBBag is competitive with uMRBBag on 6 datasets (the difference is smaller than

Table 5.2: G-mean and average ranks (the lower, the better) for multi-class real datasets. Horizontal lines separate three categories of datasets: easy, medium and difficult (top to bottom).

dataset	Bagging	uMRBBag	oMRBBag	RBBag*	C4.5
car	0,8603	0,9016	0,9516	0,8680	0,7890
dermatology	0,9542	0,9668	0,9658	0,9512	0,9436
dermatology-3	0,9494	0,9602	0,9569	0,9569	0,9275
thyroid	0,9425	0,9232	0,9455	0,9383	0,9420
vehicle	0,7162	0,7141	0,7181	0,7243	0,6825
cleveland	0,0000	0,0128	0,0034	0,0000	0,0037
cleveland-sm	0,0833	0,1746	0,1238	0,1910	0,0755
ecoli	0,6534	0,7800	0,7460	0,7108	0,6095
ecoli-3	0,6872	0,8474	0,8004	0,8272	0,6613
new thyroid	0,8937	0,9224	0,9215	0,9276	0,8778
glass	0,2819	0,4169	0,4424	0,4229	0,2591
glass-3	0,1386	0,6191	0,5236	0,5076	0,2885
yeast	0,0000	0,0336	0,0348	0,0000	0,0058
yeast-sm	0,0000	0,1307	0,0699	0,0109	0,0058
yeast-3	0,5602	0,8296	0,7104	0,8150	0,5874
average rank	4,0000	1,7142	2,0000	2,73331	4,5454

1%), better on 3 datasets (of which two belong to the easiest category), and worse on 6 datasets (of which three belong to the most difficult category and the remaining three to the medium category). The average differences between uMRBBag and oMRBag are 0.05 for difficult, 0.028 for medium, and -0.014 for easy datasets (indicating that for the last category oversampling obtained a better average). The superiority of uMRBBag is also reflected in a lower average rank. Therefore, we will use the undersampling version of Multi-class Roughly Balanced Bagging for further experiments.

5.3.2 Evaluation of the performance of proposed bagging ensembles with state-of-the-art methods

Finally, we performed an experiment comparing the classification performance of MRB-Bag, SOUP-Bagging, and other decomposition approaches, namely OVO RUS, OVO ROS, and OVO NCR. These ensembles were selected as, according to our earlier experiments, they offered the highest G-mean values. Additionally, to compare SOUP-Bagging performance with standard SOUP, we also included this preprocessing method into our study. All other experimental setup details were left as before (in particular, bagging ensembles used 30 component classifiers). The experiment uses the same collection of datasets as in Chapter 4.

The results are presented in Table 5.3. We have also performed the Friedman rank test that rejected the null hypothesis about the lack of significant differences between methods in the study ($p = 0.0032$). The results of the Nemenyi post-hoc analysis are presented in 5.2.

The best performing method was SOUP-Bagging that obtained the lowest average rank (2.61), followed by SOUP (3.13) and MRBBag (3.32). Although SOUP-Bagging obtained a better rank than SOUP, the Wilcoxon paired test does not detect significant differences ($p = 0.312$). In fact, the median of differences in G-mean between SOUP and SOUP-Bagging is 0.3%. Nevertheless, SOUP-Bagging offered improvements on 12 datasets, including several significant ones e.g. on `ecoli`, `led7digit` and `wine quality` (all above 2%).

5.4 Discussion

In this chapter, we introduced two ensemble methods for multi-class imbalanced data. One of these ensembles, SOUP-Bagging, builds upon our previous method for analyzing data difficulty factors in imbalanced datasets. The proposal of the second ensemble, Multi-class Roughly Balanced Bagging, is motivated by the good performance of previous bagging approaches on imbalanced datasets with more difficult characteristics e.g. with a higher number of rare and outlier example types. Nevertheless, this approach does not directly handle particular data difficulty factors nor use any form of its analysis.

The experimental results demonstrated that SOUP-Bagging ensemble, which uses our analysis of data difficulty factors, obtains better results than MRBBag with random undersampling. Interestingly, SOUP preprocessing that uses only one component classifier also obtained better results than MRBBag. This demonstrates the importance of properly handling the data difficulty factors in the design of bagging-based approaches for multi-class imbalanced data.

Even though MRBBag generally obtains weaker classification performance than SOUP and SOUP-Bagging, it outperforms classical decomposition ensembles. It is worth noting that MRBBag, contrary to SOUP and SOUP-Bagging, is simplistic and therefore does not require the calculation of nearest neighbors, resulting in lower computational cost. Additionally, undersampling MRBBag used in final experiments trains base classifiers on datasets that on average have $|C| \cdot \min_i |D_i|$ examples i.e. the size of the smallest minority class times the number of classes. This means that for datasets with a high imbalance ratio MRBBag constructs small datasets, which will result in even fast training of base classifiers.

Finally, one possible future research direction would be to extend the proposed methods by combining some elements of MRBBag with SOUP-Bagging. Note that SOUP-Bagging applies a rather complex informed resampling procedure that exploits data difficulty fac-

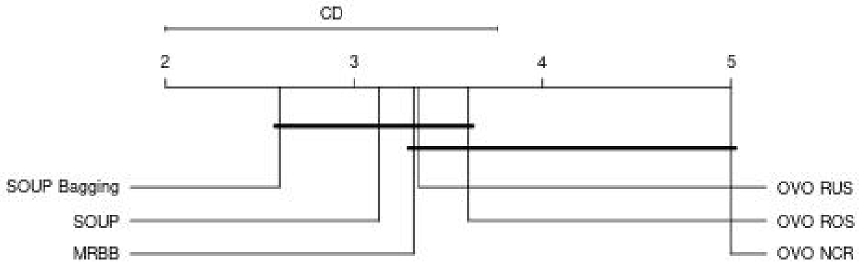


Figure 5.2: The visualization of post-hoc Nemenyi analysis for G-mean obtained by related ensemble methods

Table 5.3: Values of G-mean obtained by ensemble methods and SOUP on real datasets

dataset	OVO ROS	OVO RUS	OVO NCR	SOUP	MRBB	SOUP-Bagging
balance scale	0.526	0.602	0.474	0.585	0.683	0.591
car	0.939	0.876	0.919	0.941	0.907	0.938
cleveland_1	0.255	0.287	0.262	0.266	0.021	0.247
cleveland_2	0.288	0.285	0.000	0.303	0.055	0.000
cmc	0.509	0.514	0.526	0.535	0.517	0.537
dermatology	0.921	0.929	0.948	0.962	0.959	0.972
ecoli	0.805	0.767	0.000	0.735	0.768	0.771
flare	0.544	0.568	0.522	0.566	0.542	0.569
glass	0.699	0.697	0.691	0.667	0.400	0.655
hayes roth	0.843	0.843	0.838	0.835	0.823	0.828
led7digit	0.771	0.779	0.722	0.778	0.778	0.800
thyroid	0.922	0.886	0.913	0.922	0.932	0.940
vehicle	0.916	0.923	0.915	0.915	0.943	0.933
wine quality	0.492	0.476	0.434	0.471	0.525	0.499
yeast	0.442	0.479	0.000	0.451	0.201	0.439
art1	0.958	0.949	0.949	0.960	0.960	0.977
art2	0.758	0.777	0.762	0.777	0.808	0.773
art3	0.615	0.612	0.559	0.608	0.631	0.619
art4	0.840	0.872	0.839	0.899	0.893	0.899

tors. On the other hand, MRBBag has a particular formula for establishing the number of examples sampled from each class but uses random resampling. Therefore, it is viable to propose a bagging algorithm that uses roughly balanced training sets obtained with informed resampling, inspired by SOUP.

Chapter 6

Case Study: Multi-class Sentiment Classification

6.1 Motivation

Problems with imbalanced data arise in many different research domains and areas of application. The issue of imbalanced classes may be one of the main research considerations in such application area, but sometimes this issue does not receive sufficient research attention and goes somehow unnoticed while the research concentrates on other, often equally important, domain problems and complexities. Nevertheless, the proper treatment of the issue of imbalanced data can result in a considerable classification performance improvement. This can be illustrated with many different real-world problems, but we have selected the sentiment classification task for further consideration in this work. This problem was selected as, apart from the dissertation author's interests in natural language processing, it demonstrates well the specifics of imbalanced classification with several classes and can show the practical usefulness of the previously introduced methods.

Sentiment analysis gained high research interest in recent years due to its possible high economic and social impact. First, companies want to seize the opportunity to gain information about their customers and the reception of their products through the sentiment of posts and reviews published on social media and other online platforms. The opinions expressed on the Internet can be very helpful in assessing or directing marketing campaigns, constructing recommendation systems, improving product designs, and analyzing the stock market [62, 31]. Furthermore, automatic sentiment analysis enables the construction of very useful tools to deal with hate speech and other negatively polarized messages that unfortunately are abundant on the web. It has been shown that such messages can cause a wide range of mental problems, including depression or anxiety [13, 119]. Moreover, even the exposure to hate speech towards another person on the Internet can have negative impacts such as emotional distress [161, 140].

Sentiment classification is the fundamental task of sentiment analysis, which aims to detect sentiment polarity (e.g. strongly negative, negative, neutral, etc.) of a given text. Even though the task can be seen just as a special case of text classification, it is usually treated as a separate task since it is much more difficult than the typically considered tasks of assigning texts to a certain set of topics [126]. The task also requires distinctive text preprocessing techniques and more advanced feature construction methods. For instance, the word "not" is often treated as a stop word and is removed in the standard topic-oriented text classification, but it is a key word that requires special treatment in sentiment classification ("not good" vs. "good"). Except for proper negation handling, other difficulties include sarcastic or ironic utterances, words changing polarity over time, or handling of adverbial sentiment modifiers.

Besides these domain-specific adversities that most of the research focuses on, the problem of sentiment classification is inherently imbalanced, with positive and neutral sentences overwhelming the negative ones. This phenomenon is sometimes ascribed to product makers' and vendors' marketing efforts, as well as the fact that buyers are more likely to choose products with positive evaluations, resulting in even more generally positive reviews [95]. Recently, there were increasing efforts to address this issue by using imbalanced learning techniques for sentiment classification [97, 96, 162, 85, 78, 136], however, most of these works were performed on binary sentiment classification problems. It is worth noting that the usage of 5-star or other multi-level ordinal scales is widespread in modern review platforms, so the usefulness of works on binary classification can be somewhat limited in practice. Furthermore, even if binary classification is needed, it has been demonstrated that adding a third, neutral class to binary sentiment classification problems can facilitate better recognition of positively and negatively polarized texts [72].

The works on multi-class imbalanced sentiment classification are rather limited. The usefulness of several binary-class SMOTE extensions applied iteratively to each class in the Chinese Emotion Corpus was tested in the experiments of Xu et al. [162]. A more complex pipeline consisting of OVO decomposition ensemble, standard SMOTE and Multiple Correspondence Analysis [1] was proposed in [118] for constructing classifiers for SemEval2016 Message Polarity dataset. Nevertheless, except our work [83] there are no other more comprehensive comparisons of multi-class imbalanced methods on multiple sentiment datasets and several types of data representation.

In this chapter, we will investigate the usefulness of the ensemble algorithms proposed in this thesis, namely SOUP-Bagging and MRBBag, to the practical task of sentiment classification. We will evaluate the performance of four standard classifiers and two types of data representation: one low-dimensional with handcrafted features and the other more high-dimensional with features extracted automatically through distributional semantic methods. The evaluation will be performed on 12 diversified datasets that range from short text from social media to fully-fledged professional reviews.

6.2 Experimental Setup

6.2.1 Datasets and feature representations

One of the important aspects of successful usage of classical machine learning algorithms is proper feature engineering. In our study, we will use two sets of features designed for sentiment detection.

The first representation (`low-dim`) consists of 30 handcrafted features constructed from special data resources, called sentiment lexicons. Sentiment lexicons basically contain a list of words with assigned sentiment polarity. Some of them contain just a list of positive and negative words, but some are more detailed and assign words to different emotions like "anger" or "happiness", sometimes even with a kind of intensity degree or polarity value. We use five popular lexicons: the Opinion lexicon [59], SentiWordNet [9], the NRC Hashtag Affirmative/Negated Context Sentiment lexicon [71], the NRC emotion lexicon [116], and the Multi-perspective Question Answering corpus [158]. A feature counting word concurrence in a given text and the dictionary was constructed for each sentiment or emotion listed in each dictionary. If a lexicon apart from a word list contained also concrete polarity values, the sum of these polarities were used instead of the word count.

The second representation (`high-dim`) has 300 features of continuous bag-of-word representation constructed from a popular word2vec model [115]. This representation is computed as an average of pre-trained vectors associated with each word in the text. These pre-trained vectors, usually called word embeddings, can model many semantic properties of words, including sentiment, since they come as a result of solving a simplified language modeling task. In our case, they were pretrained on the Google News corpus.

These two representations were constructed for each of the 12 datasets presented in Table 6.1. The datasets contain problems with a different number of classes, ranging from 3 to 10, and represent different types of text. Four datasets (`movie_aut1-4`) contain full-length movie reviews written by professional critics, another four datasets (`book`, `electronics`, `dvd`, `housewares`) are reviews written in an online shop, two datasets contain posts from Twitter and the last two contain users opinions from TripAdvisor and Citysearch. A more detailed description of these datasets can be found in our work [83].

Table 6.1 also contains average safe levels for instances of minority classes encoded in both representations. In this experiment, by a minority class we mean a class whose size is smaller than the average class size in the dataset. The similarity configuration used for safe level calculation was computed according to the earlier presented heuristic (Eq. 4.2).

In general, one can note that the safe level values are relatively low, indicating the difficulty of the sentiment classification task. Interestingly, the safe levels for datasets containing short utterances are not lower than for longer texts, even though many works indicated that extracting sentiment from short text is more challenging [85, 120]. We also find that the continuous bag of words representation is safer than the handcrafted features for most of the datasets.

6.2.2 Classifiers

The experiments were performed with four popular learning algorithms from sklearn python package: decision trees (DT), Naive Bayes (NB), Multinomial Logistic Regression (MLR), and k-Nearest Neighbors (kNN). For each dataset and feature representation, we selected the best hyperparameters values through grid search with a 10% validation set. We have tested 0.01, 0.1, 1, 10, and 100 as the values of L2 regularizer in MLR, kNN was tested with k ranging from 1 to 9 (only odd values), and the decision tree had tuned the maximum depth parameter (10, 20, 40, 80 and full tree). All reported results come from the 5-fold cross-validation, with the hyperparameter tuning performed within each fold.

Apart from methods proposed in this thesis, SOUP-Bagging and MRBBag, we will test the best performing ensemble methods from our previous experiments. Concretely, we will use one-versus-one decomposition with random oversampling and random undersampling.

Table 6.1: Basic characteristics of datasets under study.

				avg. safe level	
dataset	IR	# classes	# examples	low-dim	high-dim
books	2.72	4	2000	0.223	0.283
dvd	2.5	4	2000	0.258	0.269
electronics	2.12	4	2000	0.224	0.269
housewares	3.07	4	2000	0.150	0.203
movie_aut1	10.08	8	1027	0.078	0.125
movie_aut2	26.08	10	1306	0.078	0.143
movie_aut3	11.87	10	902	0.078	0.081
movie_aut4	52.75	9	1768	0.134	0.174
restaurants	3.17	4	1325	0.373	0.306
tripadvisor	6.37	5	20491	0.266	0.225
tweeter5point	33.31	5	9090	0.159	0.202
tweeter3point	3.13	3	9090	0.227	0.298

Moreover, MRBBag will be tested in its both versions proposed in the earlier chapter: undersampling MRBBag (uMRBB) and oversampling MRBBag (oMRBB).

As in our previous experiments, the classification performance is measured with G-mean measure and additionally, due to its high popularity in the text classification community, with F-score. Since the number of computed results is quite large, only the relevant results summaries are presented here. The interested reader can check the detailed results on the webpage [1].

6.3 Comparison of Imbalanced Learning Methods in the Domain of Sentiment Classification

We will begin the discussion of the results by selecting the best pair of a component classifier algorithm and an imbalance ensembling method for the sentiment classification task, i.e., such that obtains the highest results on F-score and G-mean measure. In this experiment, we measure the classification performance for both feature representation, i.e., treating datasets with different features as separate datasets. The five top-performing pairs and their ranks are presented in Table 6.2.

For F-score, the best performing method was the combination of SOUP-Bagging with multinomial logistic regression, followed by MLR with OVO ROS decomposition. For G-mean measure, the two best-performing methods were OVO RUS and SOUP-Bagging also coupled with MLR. In general, MLR used as a base classifier seems to be most suitable for sentiment classification data as ensembles with this algorithm fill almost all top positions in the ranking. Looking at lower positions in the ranking, Naive Bayes is generally obtaining the lowest results. The ensemble methods paired with Decision Tree and kNN fill the position in between NB and MLR.

[1] http://www.cs.put.poznan.pl/mlango/publications/imbalanced-sentiment.html

Table 6.2: The best combinations of classifiers and ensemble methods for the datasets under study.

	F-score			G-mean		
no.	classifier	ensemble method	rank	classifier	ensemble method	Rank
1	MLR	SOUP-Bagging	3.21	MLR	OVO RUS	2.83
2	MLR	OVO ROS	3.5	MLR	SOUP-Bagging	3.38
3	DT	SOUP-Bagging	5.88	MLR	OVO ROS	4.33
4	MLR	OVO RUS	6.08	MLR	oMRBB	5.58
5	MLR	Single classifier	7.58	MLR	uMRBB	5.79

Table 6.3: Average ranks of ensemble methods for the datasets under study, always selecting the base classifier with the highest result. SOUP-B is an abbreviation from SOUP-Bagging.

data rep.	measure	SOUP-B	uMRBB	OVO RUS	OVO ROS	oMRBB	baseline
low-dim	G-mean	1.67	2.25	3	3.92	4.17	6
	F-score	1.83	2.33	4.83	4.25	2.58	5.17
high-dim	G-mean	2.33	2	2.5	4.08	4.08	6
	F-score	2.42	2.33	5.17	3.67	2.75	4.67

Table 6.3 contains more detailed average ranks for each ensembling technique and data representation. The ranks are averaged over datasets where always the best result, i.e., the best base classifier for each ensembling method, was selected. Interestingly, for the continuous bag of word representation the lowest (best) rank for F-score and for G-mean was obtained by our undersampling Multi-class Roughly Balanced Bagging. For low dimensional representation, the best result for both F-score and G-mean obtained SOUP-Bagging, followed by uMRBB. The oversampling version of MRBBag always obtained higher average ranks than uMRBBag, confirming our earlier observation about the superiority of uMRBBag. Lack of any imbalanced learning ensembling technique (i.e. single classifier) was the worst-performing method on all datasets for G-mean measure. Its rank was also the lowest for F-score, however, for few datasets it offered higher results than some imbalanced learning techniques. For each presented ranking, we performed Friedman rank tests that indicated statistically significant differences (all with $p < 0.001$).

6.4 Discussion

The experiments presented in this chapter demonstrated that using imbalance learning methods can improve recognition in the sentiment classification task. The best-performing methods were those proposed in this thesis, confirming their usefulness in one important area of application.

More concretely, SOUP-Bagging with the multinomial logistic regression obtained the best results in terms of F-score across datasets. However, looking for the best method for a particular data representation, uMRBBag provided better performance for high-dimensional representation. Similarly, SOUP-Bagging was obtaining the best performance on G-mean for low-dimensional representation whereas it was outperformed for high-dimensional data by uMRBBag (although the difference in average ranks is minimal). This can suggest that SOUP-Bagging, which uses standard L2-distance to compute neighborhoods, can not be suitable to work with very high-dimensional data, suffering from the so-called curse of dimensionality.

The obtained results can be further improved in many ways. First, one can think of constructing even better feature representations, such as including negation n-grams or character grams. Such ideas are further explored in our recent work [83]. Another direction for constructing better representations is the utilization of deeper distributional semantics models, trained on even larger corpora, like BERT [33] and its extensions.

Secondly, the best-performing methods in our study are bagging-based methods which for multidimensional data can be extended with random subspace mechanism [90]. Extending bagging with this mechanism for binary imbalanced data was already proposed by us in work [88]. In the experiments presented there, we have shown significant performance improvement for multidimensional imbalanced data coming from this mechanism.

Next, one can consider using more advanced learning algorithms like neural networks or support vector machines with string kernels. For instance, in our solution for sentiment classification in very short text utterances, the gradient boosted trees proved to be very competitive with other state-of-the-art approaches [85].

Finally, some of the datasets used in this study come from social networks such as Twitter which, aside from text, also contain a certain time aspect. This opens a new line of research that can analyze this problem in the context of imbalanced data streams. Recently, a more general preliminary study [18] that included sentiment classification datasets from Twitter demonstrated the superior performance of incremental imbalanced bagging approaches. Moreover, a particular safe level analysis performed over time revealed interesting fluctuations of data difficulty factors.

Chapter 7

Conclusions and Future Work

The main goal of this thesis was to verify if new classification methods for multi-class imbalanced data can be proposed, particularly by taking inspiration from the analysis of local and global data difficulty factors. This involved analyzing the difficulty sources in multi-class imbalanced datasets and proposing a method for their detection in real datasets. Furthermore, new classification methods utilizing these detected difficulty factors have been proposed. Its effectiveness has been verified both on datasets from UCI and KEEL repositories, typically used in the related works, as well as on the datasets from a selected application area (sentiment classification). Therefore, in our opinion, the main goal of this dissertation has been accomplished. To support this claim, we summarize the main contributions of this thesis.

- A novel experimental analysis of sources of difficulties in multi-class imbalanced data has been presented. To this end, we have proposed and implemented an artificial data generator that allowed for investigating the impact of various data difficulty factors on standard classifiers as well as comparing it to a theoretically optimal solution. The performed analysis pointed out the considerable impact of class distribution overlapping, e.g., that increasing it has a more significant impact on classification performance than increasing the imbalance ratio. It was also showed that multi-minority data is more difficult than multi-majority and that overlap between minority and majority classes is more harmful than between minority classes. The experiments also highlighted the special role of intermediate classes, particularly the varying impact of its overlapping on the classification performance that depends on which class they overlap.
- We presented a method for detecting data difficulty factors through the analysis of examples' save levels in real datasets. The proposed definition of the save level incorporates the class similarity degrees that can model class interrelations. In particular, they can capture the varying role of intermediate classes. We have shown that the method, despite its simplicity, is helpful in assessing the difficulty of multi-class imbalanced datasets.
- A resampling method, called Similarity Oversampling and Undersampling Preprocessing (SOUP), that exploits the conclusions from our experimental analysis of difficulty factors has been presented. The method relies on the safe levels with similarity degrees to direct sampling towards the most difficult parts of the datasets,

constructing not only a balanced dataset but also a dataset from which it is easier to learn. The performed experimental evaluation on 15 real and 4 artificial datasets demonstrated that SOUP obtains statistically significantly better results than other resampling methods like Global-CS or Static-SMOTE. Moreover, a single classifier constructed on a dataset preprocessed with SOUP outperforms specialized decomposition ensembles such as one-vs-one with resampling methods for binary imbalanced data.

- We proposed two new ensemble algorithms that extend the bagging framework for multi-class imbalanced datasets. The first algorithm, Multi-class Roughly Balanced Bagging (MRBBag), comes from extending Roughly Balanced Bagging for binary imbalanced problems that proved to obtain outstanding results on complex imbalanced problems, despite the lack of direct handling of data difficulty factors. The second algorithm integrates SOUP resampling into a bagging framework that uses the proposed method to better handle data difficulty factors. The experiments performed on a variety of datasets demonstrated that both methods offer better classification performance in terms of G-mean than the decomposition methods classically used for multi-class imbalanced problems. Furthermore, even a single classifier trained on a dataset preprocessed with SOUP outperformed the MRBBag ensemble.
- Finally, we have shown the usefulness of the proposed methods in one selected challenging area of application, namely sentiment analysis. SOUP-Bagging offered the best results among the studied methods on a low-dimensional representation of hand-crafted features specialized for sentiment classification, whereas MRBBag gave slightly better results on the high-dimensional representation of automatically learned features.
- It is worth mentioning that also some efforts to popularize the proposed methods have been undertaken. One result of these efforts is an open-source python library, compatible with the most popular sklearn machine learning library for that language[1] [47].

Naturally, the presented research can be extended in several ways. First, the difficulties related to the minority class decomposition into several subconcepts have not been investigated. Note that class decomposition into several subconcepts is not only characterized by the number of subconcepts, but also involves various possible configurations of theirs overlapping with different classes, the intensity of these overlappings, different configurations of subconcepts cardinalities, etc.

Nevertheless, we already undertook some efforts to construct methods that simultaneously detect class decomposition and other data difficulty factors in the binary imbalanced data setting through a specialized supervised clustering algorithm [84]. The result of such clustering was further used to propose a preprocessing method that during training considers particularly computed example's weights, which results in better classification performance [86]. We believe that such works can also be extended to multi-class imbalanced data and offer improvements by handling this additional difficulty factor.

Furthermore, the proposed SOUP preprocessing selects examples to resample in a non-

[1] http://www.cs.put.poznan.pl/mlango/publications/multiimbalance/index.html

trivial way through safe level calculations but oversamples them just by duplication. Many well-performing methods for binary imbalanced data take inspirations from SMOTE [24] and, instead of duplicating existing minority examples, constructs new ones, for instance, by taking a linear interpolation of two selected examples. Therefore, it seems viable to further improve SOUP results by using a more sophisticated method for oversampling examples. In particular, such a method could take into account the detected safe levels and construct new instances in a way depending on the detected difficulties in order to handle them better.

Finally, there are some related problems for which the analysis of data difficulty factors could be beneficial. For instance, in the computer vision community, there is recently a growing interest in solving the so-called problem of long-tailed class distribution [108]. It occurs when learning a classifier from a dataset that contains several classes with an abundance of examples and many classes with only a handful of examples. This problem differs from the multi-class imbalanced learning since usually there is no particular focus on minority classes recognition (e.g. the practitioners are not willing to sacrifice the recognition of majority classes to get much better recognition of minority) and the number of classes is usually much higher than in the typically considered datasets in the imbalanced data research. This problem is sometimes mitigated by transferring knowledge from large classes to small classes, e.g. though feature representation [104]. One can hypothesize that taking into account data difficulty factors while constructing feature representations could lead to some improvements, e.g., by generating easier-to-learn representations of minority classes.

To sum up, we hope that our experimental analysis and proposed methods can contribute to better understanding of the difficulties in multi-class imbalanced data and to inspire the further development of new learning algorithms dedicated for the multi-class imbalanced problems.

Appendix A

List of publications

- Pluciński K., Lango M., Stefanowski J., Prototypical Convolutional Neural Network for a phrase-based explanation of sentiment classification, ECML PKDD International Workshop and Tutorial on eXplainable Knowledge Discovery in Data Mining (XKDD), 2021
- Janiszewski P., Lango M., Stefanowski J., Time aspect in making an actionable prediction of a conversation breakdown, European Conference on Machine Learning (ECML-PKDD), 2021 (140 points)
- Marynowicz J., Lango M., Horna D., Kikut K., Andrzejewski M., Predicting ratings of perceived exertion in youth soccer using decision tree models, Biology of Sport, 2021 (70 points)
- Marynowicz J., Kikut K., Lango M., Horna D., Andrzejewski M., Relationship between the Session-RPE and external measures of training load in youth soccer training, Journal of Strength and Conditioning Research, 2020 (100 points)
- Grycza J., Horna D., Klimczak H., Lango M., Pluciński K, Stefanowski J., multi-imbalance: open source Python toolbox for multi-class imbalanced classification, European Conference on Machine Learning (ECML-PKDD), Ghent, Belgium, 2020 (140 points)
- Marynowicz J., Kikut K., Lango M., Horna D., Andrzejewski M., Relationship between the Session-RPE and external measures of training load in youth soccer training, European College of Sport Science (ECSS) congress, Sevilla, Spain, 2020
- Pluciński K., Lango M., Zimniewicz M., A Closer Look on Unsupervised Cross-lingual Word Embeddings Mapping, Proceedings of the 12th International Conference on Language Resources and Evaluation, Marseille, France, 2020 (20 points)
- Lango M., Žabokrtský Z., Ševčíková M. Semi-Automatic Construction of Word-Formation Networks, Language Resources and Evaluation, 2020 (70 points)
- Lango M., Stefanowski J., SOUP-Bagging: a new approach for multi-class imbalanced data classification, PP-RAI '19: Polskie Porozumienie na Rzecz Sztucznej Inteligencji, 2019
- Janicka M., Lango M., Stefanowski J., Using information on class interrelations to improve classification of multi-class imbalanced data: a new re-sampling algorithm,

International Journal of Applied Mathematics and Computer Science, 2019 (100 points)

- Lango M., Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study, Foundations of Computing and Decision Sciences, 2019 (70 points)
- Lango M., Brzeziński D., Stefanowski J., ImWeights: Classifying Imbalanced Data Using Local and Neighborhood Information, JMLR Proceedings of the 2nd International Workshop on Learning with Imbalanced Domains co-located with ECML/P-KDD, Dublin, Ireland, 2018
- Lango M., Ševčíková M., Žabokrtský Z., Semi-Automatic Construction of Word-Formation Networks (for Polish and Spanish), Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, 2018
- Lango M., Brzeziński D., Firlik S., Stefanowski J., Discovering Minority Sub-clusters and Local Difficulty Factors from Imbalanced Data, Proceedings of the 20th International Conference on Discovery Science, Kyoto, Japan, 2017 (15 old[1] points)
- Lango M., Napierala K., Stefanowski J., Evaluating difficulty of multi-class imbalanced data, Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence, Springer, 2017 (15 old points)
- Lango M., Stefanowski J., Multi-class and Feature Selection Extensions of Roughly Balanced Bagging for Imbalanced Data, Journal of Intelligent Information Systems, 2017 (20 old points)
- Lango M., Brzeziński D., Stefanowski J., PUT at SemEval-2016 Task 4: The ABC of Twitter Sentiment Analysis, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, US , 2016
- Błaszczyński J., Lango M., Diversity Analysis on Imbalanced Data Using Neighbourhood and Roughly Balanced Bagging Ensembles, Artificial Intelligence and Soft Computing, Lecture Notes in Artificial Intelligence, Springer , 2016 (15 old points)
- Lango M., Stefanowski J., The Usefulness of Roughly Balanced Bagging for Complex and High-dimensional Imbalanced Data, In New Frontiers in Mining Complex Patterns, Lecture Notes in Computer Science, Springer , 2016 (15 old points)
- Lango M., Stefanowski J., Applicability of Roughly Balanced Bagging for Complex Imbalanced Data, Proceedings of the 4th Workshop on New Frontiers in Mining Complex Patterns join together with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2015

In total: 710 ministerial points and 80 old ministerial points.

[1] Ministerial points before higher education reform in 2018.

Appendix B
Streszczenie

Uczenie klasyfikatorów jest podstawowym obszarem badawczym w nadzorowanym uczeniu maszynowym zajmującym się automatyczną konstrukcją systemów zdolnych do przypisywania instancji testowych do predefiniowanych klas na podstawie zbioru danych. Tak ogólnie postawiony problem doprowadził do opracowania licznych algorytmów, które są szeroko wykorzystywane w wielu obszarach, takich jak rekomendacja produktów, identyfikacja autorstwa czy filtrowanie wiadomości. Pomimo znaczących sukcesów w licznych zastosowaniach, pewne problemy wciąż pozostają otwarte i utrudniają powszechne wykorzystanie metod uczenia maszynowego w specyficznych, a zarazem ważnych dziedzinach zastosowań. Jednym z takich problemów jest problem uczenia z danych niezbalansowanych.

Zbiór danych nazywamy niezbalansowanym jeśli zawiera on klasy o różnych licznościach, a co najmniej jedna z klas nie jest dostatecznie dobrze reprezentowana. Te niedoreprezentowane klasy nazywamy klasami mniejszościowymi, a ich skuteczne rozpoznawanie jest kluczowe w wielu praktycznych problemach takich jak analiza wydźwięku tekstu, automatyczna konstrukcja grafów czy analiza danych medycznych. Przeciwnie do oczekiwań praktyków preferujących wysoką trafność rozpoznawania klas mniejszościowych, klasyczne metody uczenia maszynowego konstruują klasyfikatory dobrze rozpoznające przede wszystkim klasy większościowe. W ekstremalnych przypadkach skonstruowany klasyfikator całkowicie ignoruje klasy mniejszościowe, nie będąc w stanie zaklasyfikować do nich żadnego przykładu testowego.

Początkowo sądzono, że to niepożądane zachowanie systemów uczących się wynika jedynie z poziomu niezbalansowania (ang. *imbalance ratio*), czyli ze znacznej rozbieżności pomiędzy wielkościami klas w zbiorze treningowym. Jednakże późniejsze analizy eksperymentalne wykazały, że dla niektórych prostych problemów klasyfikacyjnych poziom niezbalansowania klas nie ma prawie żadnego wpływu na ostatecznie skonstruowany klasyfikator. Zaobserwowano, że poziom niezbalansowania wpływa na konstrukcję klasyfikatorów i znacząco pogarsza rozpoznawanie klas mniejszościowych jedynie wtedy, gdy występuje razem z innymi czynnikami trudności danych (ang. *data difficulty factors*). Czynniki trudności danych można podzielić na czynniki globalne, które wpływają na wszystkie przykłady w zbiorze danych oraz czynniki lokalne, które dotyczą tylko pewnego podzbioru instancji. Przykładem globalnego czynnika trudności danych jest omawiany poziom niezbalansowania klas, natomiast nakładanie się klas, dekompozycja klasy na podkoncepty czy znaczna liczba obserwacji odstających to typowe przykłady czynników lokalnych.

Ze względu na duże znaczenie praktyczne uczenia z danych niezbalansowanych, do jego rozwiązania zaproponowano wiele specjalistycznych metod, które można podzielić na metody algorytmiczne, metody modyfikujące rozkład danych oraz metody kosztowe. Metody algorytmiczne próbują rozwiązać trudności związane z konstrukcją klasyfikatorów z danych niezbalansowanych poprzez dostosowanie do nich istniejących systemów uczących się. W przeciwieństwie do tych metod, które skupiają się na modyfikacji konkretnych algorytmów, metody modyfikujące rozkład danych są uniwersalne i niezależne od wybranego algorytmu uczącego. Metody te modyfikują niezbalansowany zbiór danych w taki sposób, aby uruchomiony na nim algorytm uczący osiągał lepsze wyniki na klasie mniejszościowej. Ostatnia grupa metod wykorzystuje algorytmy uczenia się z kosztami (ang. *cost-sensitive learning*) koncentrując konstrukcję systemu klasyfikacyjnego na klasach mniejszościowych poprzez przypisanie ich przykładom wyższych kosztów błędnej klasyfikacji.

Prawie wszystkie metody wchodzące w skład wyżej wymienionych kategorii zostały zaprojektowane jedynie dla niezbalansowanych problemów klasyfikacji binarnej. Niemniej jednak zjawisko niezbalansowania klas występuje również w wieloklasowych zbiorach danych. Na przykład w zastosowaniach medycznych możemy tworzyć klasyfikator, który automatycznie wyróżnia nagłe przypadki wśród pacjentów zgłaszających się do Szpitalnego Oddziału Ratunkowego. Jednakże szpital może być zainteresowany nie tylko identyfikacją pilnych przypadków, ale także wykrywaniem przypadków które powinny być leczone przez lekarzy pierwszego kontaktu lub pacjentów których leczenie powinno być prowadzone w innych szpitalach (np. z powodu braku wyspecjalizowanego oddziału). Taki problem byłby niezbalansowany, ponieważ, jak pokazują badania przeprowadzone w USA, pacjenci którzy powinni być leczeni przez lekarzy pierwszego kontaktu stanowią ponad 80% wszystkich przypadków na tego typu oddziałach. Przedstawiony problem klasyfikacji ma dwie klasy mniejszościowe, ale w praktyce występują również problemy z kilkoma klasami większościowymi lub zarówno z wieloma klasami większościowymi jak i mniejszościowymi.

Nieliczne zaproponowane metody dla problemów wieloklasowych ograniczają się do dekompozycji problemów na problemy binarne oraz inne wyspecjalizowane metody, w zdecydowanej większości adaptacje binarnych metod przetwarzania wstępnego. Te proste modyfikacje metod binarnych nie uwzględniają jednak bardziej złożonych relacji, jakie pojawiają się między klasami w wieloklasowych problemach niezrównoważonych i nie radzą sobie z dodatkowymi źródłami trudności zidentyfikowanymi przez praktyków. Co więcej, dotychczasowo przeprowadzone teoretyczne i eksperymentalne analizy czynników trudności uczenia z danych niezrównoważonych, z nielicznymi wyjątkami, również dotyczą problemów binarnych, a ich wyników nie można bezpośrednio odnieść do danych wieloklasowych.

Na podstawie powyższych przesłanek sformułowano następującą hipotezę niniejszej rozprawy:

> Można zaproponować nowe metody konstrukcji klasyfikatorów dla wieloklasowych danych niezbalansowanych, które będą uwzględniać informacje o źródłach trudności związanych z rozkładem danych zarówno na poziomie lokalnym jak i bardziej globalnym.

Powyższą hipotezę zweryfikowano poprzez przeprowadzenie eksperymentalnej analizy źródeł trudności w wieloklasowych niezbalansowanych zbiorach danych oraz zaproponowaniu metody ich identyfikacji w rzeczywistych zbiorach danych. Ponadto, zaproponowano nowe metody klasyfikacji wykorzystujące wykryte czynniki trudności, których skuteczność została zbadana zarówno na typowo wykorzystywanych w pracach badawczych zbiorach danych z repozytoriów UCI i KEEL, jak i na zbiorach danych z wybranego obszaru zastosowań (klasyfikacja wydźwięku). W szczególności zrealizowano następujące zadania:

- W celu wykonania eksperymentalnej analizy źródeł trudności w wieloklasowych danych niezbalansowanych, zaproponowano i zaimplementowano generator sztucznych danych, który pozwolił na zbadanie wpływu różnych czynników trudności na jakość predykcji standardowych algorytmów uczących oraz porównanie ich z wynikiem klasyfikatora optymalnego (tzw. klasyfikator Bayesa).

  Przeprowadzona analiza wskazała na znaczący wpływ nakładania się klas, a w szczególności, że zwiększanie nakładania się klas ma bardziej znaczący wpływ na jakość klasyfikacji niż zwiększanie poziomu niezbalansowania. Wykazano również, że dane z wieloma klasami mniejszościowymi są trudniejsze niż z wieloma większościowymi oraz że nakładanie się klas mniejszościowych i większościowych bardziej obniża jakość predykcji niż nakładanie się klas mniejszościowych. Eksperymenty uwypukliły również szczególną rolę klas pośrednich, dotychczas rzadko omawianych w literaturze, m.in. pokazano zróżnicowany wpływ nakładania się klasy pośredniej na wydajność klasyfikacji, który zależy od tego, na jakiego typu klasę się nakłada.

- Zaprezentowano metodę wykrywania czynników trudności danych poprzez analizę poziomów bezpieczeństwa przykładów w rzeczywistych zbiorach danych. Proponowana definicja poziomu bezpieczeństwa zawiera współczynniki podobieństwa klas, które mogą modelować wzajemne relacje między klasami. W szczególności mogą one uchwycić zmienną rolę klas pośrednich. Pokazano również, że metoda ta jest pomocna w ocenie trudności rzeczywistych, wieloklasowych zbiorów niezbalansowanych.

- Zaprojektowano metodę wstępnego przetwarzania danych, Similarity Oversampling and Undersampling Preprocessing (SOUP), która wykorzystuje wnioski z przeprowadzonej wcześniej eksperymentalnej analizy czynników trudności. Metoda ta na podstawie obliczonych wcześniej stopni bezpieczeństwa przykładów, ukierunkowuje swoje działanie na konkretne części zbioru danych. SOUP poprzez nadlosowanie bezpiecznych przykładów mniejszościowych i odlosowanie przykładów większościowych ze strefy nakładania się klas, konstruuje nie tylko zbalansowany zbiór danych, ale również zbiór, z którego łatwiej można nauczyć klasyfikatory o wysokiej jakości rozpoznawania klas mniejszościowych.

  Przeprowadzona ocena eksperymentalna na 15 rzeczywistych i 4 sztucznych zbiorach danych wykazała, że SOUP uzyskuje na metryce G-mean lepsze wyniki niż inne metody wstępnego przetwarzania, takie jak Global-CS czy Static-SMOTE. Uzyskane różnice były statystycznie istotnie wg. sparowanych testów rangowych Wilcoxona ze standardowym progiem istotności tj. $\alpha = 0.05$. Co więcej, pojedynczy klasyfikator skonstruowany na przetworzonym przez SOUP zbiorze danych uzyskał niższą (tj. lepszą) średnią rangę w teście Friedmana niż popularne, wyspecjalizowane zespoły klasyfikatorów, które dekomponują problem wieloklasowy do serii problemów binarnych i stosują w ramach nich binarne techniki nad- i od-losowania.

- Zaproponowano dwa nowe algorytmy konstrukcji zespołów klasyfikatorów, które rozszerzają algorytm bagging dla wieloklasowych niezbalansowanych zbiorów danych. Pierwszy algorytm, Multi-class Roughly Balanced Bagging (MRBBag), jest rozszerzeniem algorytmu Roughly Balanced Bagging dla binarnych danych niezbalansowanych, który okazał się uzyskiwać bardzo dobre wyniki na złożonych binarnych problemach niezbalansowanych, pomimo braku bezpośredniego brania pod uwagę czynników trudności danych. Drugi zaproponowany algorytm wykorzystuje z kolei metodę obliczania poziomów bezpieczeństwa do identyfikacji czynników trudności danych, stosując technikę SOUP i integrując ją z zespołem typu bagging (SOUP-Bagging).

  Eksperymenty przeprowadzone na kilkunastu zbiorach danych wykazały, że obie metody oferują lepszą jakość klasyfikacji pod względem miary G-mean niż metody dekompozycji, klasycznie stosowane dla wieloklasowych problemów niezbalansowanych. Jednakże metoda wykorzystująca informacje o czynnikach trudności tj. SOUP-Bagging uzyskała lepsze wyniki niż MRBBag. Co więcej, nawet pojedynczy klasyfikator wytrenowany na zbiorze danych wstępnie przetworzonym za pomocą SOUP również przewyższył zespół MRBBag w sensie średniej rangi w teście Friedmana.

- Wreszcie, pokazano przydatność proponowanych metod w jednym z wymagających obszarów zastosowań, jakim jest analiza wydźwięku. Przeprowadzono eksperymenty na 12 zróżnicowanych zbiorach tekstów, wśród których były zarówno krótkie opinie z sieci społecznościowych jak i profesjonalne recenzje, oraz na dwóch typach reprezentacji tekstu. Na niskowymiarowej reprezentacji ręcznie stworzonych wyspecjalizowanych cech, SOUP-Bagging uzyskał najlepsze wyniki spośród badanych metod. Z kolei MRBBag uzyskał nieco lepsze wyniki na wielowymiarowej reprezentacji automatycznie uczonych cech metodami semantyki dystrybucyjnej.

- Warto nadmienić, że podjęto również pewne wysiłki mające na celu popularyzację zaproponowanych metod. Jednym z rezultatów tych starań jest biblioteka opensource dla języka Python, kompatybilna z biblioteką sklearn.

Bibliography

[1] Hervé Abdi and Dominique Valentin. Multiple correspondence analysis. In *Encyclopedia of Measurement and Statistics*. Sage, 2007.

[2] Lida Abdi and Sattar Hashemi. To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. *Soft Computing*, 19(12):3369–3385, 2015.

[3] Lida Abdi and Sattar Hashemi. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):238–251, 2015.

[4] Astha Agrawal, Herna L Viktor, and Eric Paquet. Scut: Multi-class imbalanced data classification using smote and cluster-based undersampling. In *7th International joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3k)*, volume 1, pages 226–234. IEEE, 2015.

[5] Roberto Alejo, Vincente García, Jose Sotoca, Ramon Mollineda, and Jose Sánchez. Improving the performance of the rbf neural networks trained with imbalanced samples. In *Proceedings of the 9th International Work Conference on Artificial Neural Networks*, IWANN'07, page 162–169, Berlin, Heidelberg, 2007. Springer-Verlag.

[6] Roberto Alejo, Jose Sotoca, and Gustavo Casañ. An empirical study for the multi-class imbalance problem with neural networks. In José Ruiz-Shulcloper and Walter G. Kropatsch, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 479–486, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1(Dec):113–141, 2000.

[8] Adnan Amin, Sajid Anwar, Awais Adnan, Muhammad Nawaz, Newton Howard, Junaid Qadir, Ahmad Hawalah, and Amir Hussain. Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access*, 4:7940–7957, 2016.

[9] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[10] Sukarna Barua, Md Monirul Islam, Xin Yao, and Kazuyuki Murase. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):405–425, 2012.

[11] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[12] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[13] Tanya Beran and Qing Li. Cyber-harassment: A study of a new method for an old behavior. *Journal of Educational Computing Research*, 32(3):265, 2005.

[14] Jerzy Błaszczyński and Jerzy Stefanowski. Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing*, 150:529–542, 2015.

[15] Jerzy Błaszczyński, Jerzy Stefanowski, and Łukasz Idkowiak. Extending bagging for imbalanced data. In Robert Burduk, Konrad Jackowski, Marek Kurzynski, Michał Woźniak, and Andrzej Zolnierek, editors, *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 269–278, Heidelberg, 2013. Springer International Publishing.

[16] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2), August 2016.

[17] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[18] Dariusz Brzezinski, Leandro L. Minku, Tomasz Pewinski, Jerzy Stefanowski, and Artur Szumaczuk. The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. *Knowl. Inf. Syst.*, 63(6):1429–1469, 2021.

[19] Dariusz Brzezinski, Jerzy Stefanowski, Robert Susmaga, and Izabela Szczech. Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences*, 462:242–261, 2018.

[20] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[21] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 475–482. Springer, 2009.

[22] Nicola Burns, Yaxin Bi, Hui Wang, and Terry Anderson. Sentiment analysis of customer reviews: Balanced versus unbalanced datasets. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 161–170. Springer, 2011.

[23] Alberto Cano, Amelia Zafra, and Sebastián Ventura. Weighted data gravitation classification for standard and imbalanced data. *IEEE Transactions on Cybernetics*, 43(6):1672–1687, 2013.

[24] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.

[25] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[26] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 107–119. Springer, 2003.

[27] Chao Chen, Andy Liaw, and Leo Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12):24, 2004.

[28] David A Cieslak, T Ryan Hoens, Nitesh V Chawla, and W Philip Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1):136–158, 2012.

[29] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

[30] Florin Cutzu. Polychotomous classification with pairwise classifiers: A new voting principle. In Terry Windeatt and Fabio Roli, editors, *Multiple Classifier Systems*, pages 115–124, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[31] Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.

[32] Misha Denil and Thomas Trappenberg. A characterization of the combined effects of overlap and imbalance on the SVM classifier. *CoRR*, 2011.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[34] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(1):263–286, 1995.

[35] Tom Dietterich, Michael Kearns, and Yishay Mansour. Applying the weak learning framework to understand and improve c4.5. In *In Proceedings of the Thirteenth International Conference on Machine Learning*, pages 96–104. Morgan Kaufmann, 1996.

[36] Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 155–164, New York, NY, USA, 1999. ACM.

[37] Wei Fan and Salvatore J Stolfo. Adacost: misclassification cost-sensitive boosting. In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

[38] Alberto Fernández, Victoria López, Mikel Galar, MaríA José Del Jesus, and Francisco Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-based systems*, 42:97–110, 2013.

[39] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets.* Springer, 2018.

[40] Francisco Fernández-Navarro, César Hervás-Martínez, and Pedro Antonio Gutiérrez. A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition*, 44(8):1821 – 1833, 2011.

[41] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[42] Jerome H Friedman. Another approach to polychotomous classification. *Technical Report, Statistics Department, Stanford University*, 1996.

[43] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 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*, 42(4):463–484, 2012.

[44] Mikel Galar, Alberto Fernández, Edurne Barrenechea, and Francisco Herrera. Empowering difficult classes with a similarity-based aggregation in multi-class classification problems. *Information Sciences*, 264:135–157, 2014. Serious Games.

[45] Vicente Garcia, Jose Sanchez, and Ramon Mollineda. An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets. In *Proc. of Progress in Pattern Recognition, Image Analysis and Applications , LNCS*, volume 4756, pages 397–406. Springer, 2007.

[46] Salvador García, Zhong-Liang Zhang, Abdulrahman Altalhi, Saleh Alshomrani, and Francisco Herrera. Dynamic ensemble selection for multi-class imbalanced datasets. *Information Sciences*, 445-446:22–37, 2018.

[47] Jacek Grycza, Damian Horna, Hanna Klimczak, Mateusz Lango, Kamil Plucinski, and Jerzy Stefanowski. multi-imbalance: Open source python toolbox for multi-class imbalanced classification. In *ECML/PKDD*, 2020.

[48] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[49] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer Berlin Heidelberg, 2005.

[50] Peter Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[51] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *Annals of statistics*, 26(2):451–471, 1998.

[52] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[53] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328. IEEE, 2008.

[54] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[55] Haibo He and Ma Yungian. *Imbalanced Learning. Foundations, Algorithms and Applications.* IEEE - Wiley, 2013.

[56] Shohei Hido, Hisashi Kashima, and Yutaka Takahashi. Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):412–426, 2009.

[57] Robert C. Holte, Liane Acker, and Bruce W. Porter. Concept learning and the problem of small disjuncts. In *International Joint Conference on Artificial Intelligence*, volume 89, pages 813–818, 1989.

[58] John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In *International Conference on Artificial Intelligence: Methodology, systems, and applications*, pages 77–86. Springer, 2006.

[59] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, 2004.

[60] Jimison Iavindrasana, Adrien Depeursinge, Gilles Cohen, Antoine Geissbuhler, and Henning Müller. Asymmetric-margin support vector machines for lung tissue classification. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[61] Tasadduq Imam, Kai Ming Ting, and Joarder Kamruzzaman. z-SVM: An SVM for improved classification of imbalanced data. In *Australasian Joint Conference on Artificial Intelligence*, pages 264–273. Springer, 2006.

[62] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 57–64, 2009.

[63] Małgorzata Janicka, Mateusz Lango, and Jerzy Stefanowski. Using information on class interrelations to improve classification of multiclass imbalanced data: A new resampling algorithm. *International Journal of Applied Mathematics and Computer Science*, 29(4):769–781, 2019.

[64] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedingsroc. of the International Conference on Artificial Intelligence*, volume 56, 2000.

[65] Nathalie Japkowicz. Class imbalance: Are we focusing on the right issue? In *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets*, ICML '03, 2003.

[66] Nathalie Japkowicz and Shaju Stephen. Class imbalance problem: a systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450, 2002.

[67] Jacek Jelonek and Jerzy Stefanowski. Experiments on solving multiclass learning problems by n 2-classifier. In *European Conference on Machine Learning*, pages 172–177. Springer, 1998.

[68] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40–49, 2004.

[69] Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(3):552–568, 2011.

[70] Hyun-Jung Kim, Nam-Ok Jo, and Kyung-Shik Shin. Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Systems with Applications*, 59:226–234, 2016.

[71] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

[72] Moshe Koppel and Jonathan Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109, 2006.

[73] György Kovács. smote-variants: a python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352–354, 2019.

[74] Michał Koziarski, Michał Woźniak, and Bartosz Krawczyk. Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise. *Knowledge-Based Systems*, 204:106223, 2020.

[75] Bartosz Krawczyk. Cost-sensitive one-vs-one ensemble for multi-class imbalanced data. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2447–2452, 2016.

[76] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[77] Bartosz Krawczyk, Michał Koziarski, and Michal Woźniak. Radial-based oversampling for multiclass imbalanced data classification. *IEEE Transactions on Neural Networks and Learning Systems*, 31:2818 – 2831, 08 2020.

[78] Bartosz Krawczyk, Bridget T McInnes, and Alberto Cano. Sentiment classification from multi-class imbalanced twitter data using binarization. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 26–37. Springer, 2017.

[79] Bartosz Krawczyk, Michał Woźniak, and Francisco Herrera. On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. *Pattern Recognition*, 48(12):3969–3982, December 2015.

[80] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[81] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *International Conference on Machine Learning*, volume 97, pages 179–186, 1997.

[82] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[83] Mateusz Lango. Tackling the problem of class imbalance in multi-class sentiment classification: An experimental study. *Foundations of Computing and Decision Sciences*, 44(2):151–178, 2019.

[84] Mateusz Lango, Dariusz Brzezinski, Sebastian Firlik, and Jerzy Stefanowski. Discovering minority sub-clusters and local difficulty factors from imbalanced data. In *Proceedings of the 20th International Conference on Discovery Science*, pages 324–339, 2017.

[85] Mateusz Lango, Dariusz Brzezinski, and Jerzy Stefanowski. Put at semeval-2016 task 4: The abc of twitter sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 126–132, 2016.

[86] Mateusz Lango, Dariusz Brzezinski, and Jerzy Stefanowski. Imweights: Classifying imbalanced data using local and neighborhood information. In Luís Torgo, Stan Matwin, Nathalie Japkowicz, Bartosz Krawczyk, Nuno Moniz, and Paula Branco, editors, *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 94 of *Proceedings of Machine Learning Research*, pages 95–109, ECML-PKDD, Dublin, Ireland, 10 Sep 2018. PMLR.

[87] Mateusz Lango, Krystyna Napierala, and Jerzy Stefanowski. Evaluating difficulty of multi-class imbalanced data. In Marzena Kryszkiewicz, Annalisa Appice, Dominik Ślęzak, Henryk Rybinski, Andrzej Skowron, and Zbigniew W. Raś, editors, *Foundations of Intelligent Systems*, pages 312–322, Cham, 2017. Springer International Publishing.

[88] Mateusz Lango and Jerzy Stefanowski. Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data. *Journal of Intelligent Information Systems*, 50(1):97–127, 2018.

[89] Mateusz Lango, Zdeněk Žabokrtský, and Magda Ševčíková. Semi-automatic construction of word-formation networks. *Language Resources and Evaluation*, pages 1–30, 2020.

[90] Patrice Latinne, Olivier Debeir, and Christine Decaestecke. Mixing bagging and multiple feature subsets to improve classification accuracy of decision tree combination. In *Proceedings of the 10th Belgian-Dutch Conference on Machine Learning*, 2000.

[91] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, pages 63–66, 2001.

[92] Sauchi Stephen Lee. Noisy replication in skewed binary classification. *Computational statistics & data analysis*, 34(2):165–191, 2000.

[93] Supatcha Lertampaiporn, Chinae Thammarongtham, Chakarida Nukoolkit, Boonserm Kaewkamnerdpong, and Marasri Ruengjitchatchawalya. Heterogeneous ensemble approach with discriminative features and modified-smotebagging for pre-mirna classification. *Nucleic acids research*, 41(1):e21–e21, 2013.

[94] Qianmu Li, Yanjun Song, Jing Zhang, and Victor S. Sheng. Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering. *Expert Systems with Applications*, 147:113152, 2020.

[95] Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[96] Shoushan Li, Guodong Zhou, Zhongqing Wang, Sophia Yat Mei Lee, and Rangyang Wang. Imbalanced sentiment classification. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2469–2472, 2011.

[97] Tao Li, Yi Zhang, and Vikas Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 244–252, 2009.

[98] Yuxuan Li and Xiuzhen Zhang. Improving k nearest neighbor with exemplar generalization for imbalanced classification. In *Advances in knowledge discovery and data mining*, pages 321–332. Springer, 2011.

[99] Guohua Liang and Anthony G. Cohn. An effective approach for imbalanced classification: Unevenly balanced bagging. In *Association for the Advancement of Artificial Intelligence*, 2013.

[100] T. Warren Liao. Classification of weld flaws with imbalanced class data. *Expert Systems with Applications*, 35(3):1041–1052, 2008.

[101] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 13(2):464–471, 2002.

[102] Minlong Lin, Ke Tang, and Xin Yao. Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24:647–660, 04 2013.

[103] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[104] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[105] Wei Liu, Sanjay Chawla, David A Cieslak, and Nitesh V Chawla. A robust decision tree algorithm for imbalanced data sets. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 766–777. SIAM, 2010.

[106] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, 2016.

[107] Xu-Ying Liu, Qian-Qian Li, and Zhi-Hua Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In *2013 IEEE 13th International Conference on Data Mining*, pages 478–487, 2013.

[108] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[109] Victoria López, Alberto Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.*, 250:113–141, 2013.

[110] Satyam Maheshwari, Jitendra Agrawal, and Sanjeev Sharma. A new approach for classification of highly imbalanced datasets using evolutionary algorithms. *J. Sci. Eng. Res*, 2:1—5, 2011.

[111] Inderjeet Mani and I. Zhang. kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of Workshop on Learning from Imbalanced Datasets*, 2003.

[112] Antonio Maratea, Alfredo Petrosino, and Mario Manzo. Adjusted f-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257:331–341, 2014.

[113] Eddy Mayoraz and Ethem Alpaydin. Support vector machines for multi-class classification. In *International Work-Conference on Artificial Neural Networks*, pages 833–842. Springer, 1999.

[114] Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014.

[115] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[116] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational intelligence*, 29(3):436–465, 2013.

[117] Aysşe Rumeysa Mohammed, Shady A Mohammed, and Shervin Shirmohammadi. Machine learning and deep learning based traffic classification and prediction in software defined networking. In *2019 IEEE International Symposium on Measurements & Networking (M&N)*, pages 1–6. IEEE, 2019.

[118] Asmaa Mountassir, Houda Benbrahim, and Ilham Berrada. An empirical study to address the problem of unbalanced data sets in sentiment classification. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 3298–3303. IEEE, 2012.

[119] Emily R Munro. The protection of children online: a brief scoping review to identify vulnerable groups. *Childhood Wellbeing Research Centre*, 2011.

[120] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, San Diego, California, 2016. Association for Computational Linguistics.

[121] Krystyna Napierała and Jerzy Stefanowski. BRACID: a comprehensive approach to learning rules from imbalanced data. *Journal of Intelligent Information Systems*, 39:335–373, 2012.

[122] Krystyna Napierala and Jerzy Stefanowski. Identification of different types of minority class examples in imbalanced data. In *7th International Conference on Hybrid artificial intelligent systems*, Lecture Notes in Computer Science, pages 139–150. Springer, 2012.

[123] Krystyna Napierala and Jerzy Stefanowski. Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46(3):563–597, 2016.

[124] Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *Rough Sets and Current Trends in Computing*, pages 158–167. Springer, 2010.

[125] Krystyna Napierała. *Improving Rule Classifiers For Imbalanced Data.* PhD thesis, Poznan University of Technology, Poznań, Poland, 2013.

[126] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*, 2004.

[127] Ronaldo C. Prati, Gustavo Enrique de Almeida Prado Alves Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence*, pages 312–321. Springer, 2004.

[128] Paulo VW Radtke, Eric Granger, Robert Sabourin, and Dmitry O Gorodnichy. Skew-sensitive boolean combination for adaptive ensembles–an application to face recognition in video surveillance. *Information Fusion*, 20:31–48, 2014.

[129] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141, 2004.

[130] José A Sáez, Bartosz Krawczyk, and Michał Woźniak. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178, 2016.

[131] Mojtaba Salehi and Isa Nakhai Kamalabadi. A hybrid recommendation approach based on attributes of products using genetic algorithm and naive bayes classifier. *International Journal of Business Information Systems*, 13(4):381–399, 2013.

[132] José Antonio Sanz, Dario Bernardo, Francisco Herrera, Humberto Bustince, and Hani Hagras. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data. *IEEE Transactions on Fuzzy Systems*, 23(4):973–990, 2014.

[133] Jürgen Schürmann. *Pattern classification: a unified view of statistical and neural approaches.* John Wiley & Sons, Inc., 1996.

[134] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

[135] Swati Shilaskar and A. Ghatol. Diagnosis system for imbalanced multi-minority medical dataset. *Soft Computing*, 23:4789–4799, 2019.

[136] Kaisong Song, Shi Feng, Wei Gao, Daling Wang, Ge Yu, and Kam-Fai Wong. Personalized sentiment classification based on latent individuality of microblog users. In *Social Media Content Analysis: Natural Language Processing and Beyond*, pages 197–214. World Scientific, 2018.

[137] Jerzy Stefanowski. On properties of undersampling bagging and its extensions for imbalanced data. In Robert Burduk, Konrad Jackowski, Marek Kurzynski, Michał Woźniak, and Andrzej Zolnierek, editors, *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, Advances in Intelligent Systems and Computing, pages 407–417. Springer International Publishing, 2015.

[138] Jerzy Stefanowski. Dealing with data difficulty factors while learning from imbalanced data. In Stan Matwin and Jan Mielniczuk, editors, *Challenges in Computational Statistics and Data Mining*, pages 333–363. Springer International Publishing, 2016.

[139] Jerzy Stefanowski and Szymon Wilk. Selective pre-processing of imbalanced data for improving classification performance. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 283–292. Springer, 2008.

[140] Wessel Stoop, Florian Kunneman, Antal van den Bosch, and Ben Miller. Detecting harassment in real-time as conversations develop. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 19–24, Florence, Italy, August 2019. Association for Computational Linguistics.

[141] Kenneth Sturrock and Jorge Rocha. A multidimensional scaling stress evaluation table. *Field methods*, 12(1):49–60, 2000.

[142] Nan Sun, Jun Zhang, Paul Rimba, Shang Gao, Leo Yu Zhang, and Yang Xiang. Data-driven cybersecurity incident prediction: A survey. *IEEE Communications: surveys & tutorials*, 21(2):1744–1772, 2018.

[143] Y. Sun, M. S. Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 592–602, 2006.

[144] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[145] José A. Sáez, Julián Luengo, Jerzy Stefanowski, and Francisco Herrera. Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.

[146] Mingzhu Tang, Steven X Ding, Chunhua Yang, Fanyong Cheng, Yuri AW Shardt, Wen Long, and Daifei Liu. Cost-sensitive large margin distribution machine for fault detection of wind turbines. *Cluster Computing*, 22(3):7525–7537, 2019.

[147] Tin Kam Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.

[148] Nikolaj Tollenaar and Peter van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013.

[149] Ivan Tomek. Two modifications of cnn. *IEEE Transactions on Systems Man and Communications*, 6:769–772, 1976.

[150] Sofie Verbaeten and Anneleen Van Assche. Ensemble methods for noise elimination in classification problems. In *International Workshop on Multiple Classifier Systems*, pages 317–325. Springer, 2003.

[151] Sarah Vluymans, Alberto Fernández, Yvan Saeys, Chris Cornelis, and Francisco Herrera. Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: A fuzzy rough set approach. *Knowl. Inf. Syst.*, 56(1):55–84, 2018.

[152] Carolin Wagner, Philipp Saalmann, and Bernd Hellingrath. Machine condition monitoring and fault diagnostics with imbalanced data sets based on the kdd process. *IFAC-PapersOnLine*, 49(30):296–301, 2016. 4th IFAC Symposium on Telematics Applications TA 2016.

[153] Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. Class imbalance, redux. In *11th International Conference on Data Mining (ICDM)*, pages 754–763. IEEE, 2011.

[154] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1):1–12, 2020.

[155] Shuo Wang and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pages 324–331, 2009.

[156] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, 2012.

[157] Garry M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

[158] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210, 2005.

[159] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:408–421, 1972.

[160] Wojciechowski, S. and Wilk, Sz. The generator of synthetic multi-dimensional data. Technical report, Poznan University of Technology Report RB-16/14, 2014.

[161] Ellery Wulczyn, Dario Taraborelli, Nithum Thain, and Lucas Dixon. Algorithms and insults: Scaling up our understanding of harassment on wikipedia. 2017.

[162] Ruifeng Xu, Tao Chen, Yunqing Xia, Qin Lu, Bin Liu, and Xuan Wang. Word embedding composition for data imbalances in sentiment and emotion classification. *Cognitive Computation*, 7(2):226–240, 2015.

[163] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[164] Show-Jane Yen and Yue-Shi Lee. Cluster-based sampling approaches to imbalanced data distributions. In AMin Tjoa and Juan Trujillo, editors, *Data Warehousing and Knowledge Discovery*, volume 4081 of *Lecture Notes in Computer Science*, pages 427–436. Springer Berlin Heidelberg, 2006.

[165] Kihoon Yoon and Stephen Kwek. An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pages 6–pp. IEEE, 2005.

[166] Hualong Yu, Jun Ni, and Jing Zhao. Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced dna microarray data. *Neurocomputing*, 101:309–318, 2013.

[167] Hualong Yu, Changyin Sun, Xibei Yang, Wankou Yang, Jifeng Shen, and Yunsong Qi. Odoc-elm: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. *Knowledge-Based Systems*, 92:55–70, 2016.

[168] Zhong-Liang Zhang, Xing-Gang Luo, Salvador García, and Francisco Herrera. Cost-sensitive back-propagation neural networks with binarization techniques in addressing multi-class problems and non-competent classifiers. *Applied Soft Computing*, 56:357–367, 2017.

[169] Zhongliang Zhang, Bartosz Krawczyk, Salvador Garcìa, Alejandro Rosales-Pérez, and Francisco Herrera. Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. *Knowledge-Based Systems*, 106:251–263, 2016.

[170] Xing-Ming Zhao, Xin Li, Luonan Chen, and Kazuyuki Aihara. Protein classification with imbalanced data. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1125–1132, 2008.

[171] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.

[172] Zhi-Hua Zhou and Xu-Ying Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010.

[173] Honghao Zhu, Guanjun Liu, Mengchu Zhou, Yu Xie, Abdullah Abusorrah, and Qi Kang. Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection. *Neurocomputing*, 407:50–62, 2020.

[174] Lipeng Zhu, Chao Lu, Zhao Yang Dong, and Chao Hong. Imbalance learning machine-based power system short-term voltage stability assessment. *IEEE Transactions on Industrial Informatics*, 13(5):2533–2543, 2017.

[175] Tuanfei Zhu, Yaping Lin, and Yonghe Liu. Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition*, 72:327–340, 2017.

Institute of Computer Science, Faculty of Computing and Telecommunications
Poznan University of Technology

Skład przy użyciu systemu LaTeX.

BibTeX:

```
@mastersthesis{ LangoMSc2015,
   author = "Mateusz Lango",
   title = "{Analysis of data difficulty factors for multi-class imbalanced problems and their
application in classification methods}",
   school = "Poznan University of Technology",
   address = "Pozna{\’n}, Poland",
   year = "2021",
}
```