FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 44 (2019) No. 2
ISSN 0867-6356
sciendo DOI: 10.2478/fcds-2019-0009 e-ISSN 2300-3405

Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study

Mateusz Lango *

Abstract. Sentiment classification is an important task which gained extensive attention both in academia and in industry. Many issues related to this task such as handling of negation or of sarcastic utterances were analyzed and accordingly addressed in previous works. However, the issue of class imbalance which often compromises the prediction capabilities of learning algorithms was scarcely studied. In this work, we aim to bridge the gap between imbalanced learning and sentiment analysis. An experimental study including twelve imbalanced learning preprocessing methods, four feature representations, and a dozen of datasets, is carried out in order to analyze the usefulness of imbalanced learning methods for sentiment classification. Moreover, the data difficulty factors — commonly studied in imbalanced learning — are investigated on sentiment corpora to evaluate the impact of class imbalance.

Keywords: sentiment analysis, imbalanced data, multi-class learning, data difficulty factors, text classification

1. Introduction

Over 500 million tweets are posted online every day, many of which are emotionally marked. This fact, together with millions of reviews which are available online, creates a great opportunity for businesses to assess subjective, difficult to measure, yet critical quality factors of their products and services. Simultaneously, this poses an important research challenge to create systems which extract emotions from unstructured data.

Sentiment classification, which is an essential part of such systems, aims to automatically detect the sentiment polarity of a given text by assigning it to an appropriate category (e.g. positive, negative, or neutral). Due to the practical importance of sentiment classification, several research works has been conducted to understand the main

*Institute of Computing Sciences, Poznan University of Technology, Poznań, Poland, mateusz.lango@cs.put.poznan.pl

challenges posed by this problem [41, 69, 28]. Negation handling, sarcastic utterances, adverbial sentiment modifiers and words changing polarity over time are only a few of the adversities that were noticed and handled by the proposed solutions. However, the fact that the datasets in sentiment analysis are regularly imbalanced, with positive opinions overwhelming the negative ones, was noticed only recently [8, 38, 40].

The problem of class imbalance is an essential difficulty in the construction of learning systems which gained extensive attention in the machine learning community [9, 12, 21, 22]. Nonetheless, the works on tackling class imbalance in the area of sentiment classification are rather scarce and often limited to the comparison of random resampling methods on a handful of datasets. Similarly, existing works usually treat this classification problem as a binary one (positive/negative), whereas frequently the user would like to get more refined information in terms of 5-star ordinal scale or at least by adding the neutral category [28]. Since research on sentiment analysis is somewhat orthogonal to imbalance learning, there is a lack of experimental comparison of imbalanced techniques, including recent ones, applied to the problem of sentiment classification.

Moreover, in the imbalance learning community datasets are frequently analyzed not only in terms of their global imbalance ratio but also taking into account local data difficulty factors [5, 50, 35], which are more influential to the overall classifier performance than the global class imbalance itself [50]. Such analysis of local data difficulty factors was never performed for sentiment classification datasets and yet it could bring additional insight into the class-imbalance issue in sentiment analysis, and provide some guidance in the construction of new feature representations for the problem.

To address the above-mentioned issues, in this paper we perform an experimental study on dataset and classifier characteristics that are crucial in tackling multi-class imbalanced sentiment classification problems. The main goals of this study are:

1. to compare different feature representations used in sentiment classification in terms of local data difficulty factors;

2. to provide an experimental evaluation of imbalanced learning techniques for multi-class sentiment classification which include different feature representations, learning algorithms, and a diverse selection of datasets ranging from collections of full-length reviews to short tweets.

The remainder of the paper is organized as follows. Section 2 provides a review of related works both in sentiment analysis and imbalanced learning, focusing on a description of the methods used in the experimental study. Section 3 describes the feature representations being under investigation together with the selected datasets. In Section 4, we discuss the results of the analysis of different representations with respect to local data difficulty factors, whereas Section 5 contains the results of our experimental comparison of various imbalanced learning methods for sentiment classification task. Finally, Section 6 provides conclusions and draws lines of future research.

2. Related Works

In the following subsections, we present the necessary background in sentiment analysis and imbalanced learning. We also survey the current state of the research which involves both imbalance learning and sentiment classification.

2.1. Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a relatively young branch of data mining research which aims at constructing systems which can understand users' emotions about certain entities, e.g. products or services [41]. The research in this area gained wide attention in recent years due to the emergence of new opinion data sources, such as social media, and the increasing interest from the industry which uses such methods to assess marketing campaigns and to improve product designs. The analysis of the expressed sentiment proved to be useful in many other areas such as recommendation systems [51] or stock market analysis [11].

An essential task in sentiment analysis is the polarity detection of a given text, to which we refer as sentiment classification. Such analysis can be performed at different levels of granularity: document, sentence, and aspect/entity[1] levels are usually distinguished. In this work, we focus on presumably the simplest and most popular document-level sentiment classification methods. However, such defined classification task is still of considerable difficulty. For instance, Pang et al. [54] demonstrated that this task is still significantly more difficult for machine learning methods that standard text classification problems whose goal is to assign texts to a certain set of topics. Although the majority of related works deals with binary positive/negative polarity classification, it has been shown that the implicit modeling of the neutral class can be beneficial [28]. Additionally, many modern review data sources operate on 5-star, 10-star or other multi-level ordinal scales, which naturally shifts our attention towards multi-class methods.

Even a humble overview of machine learning techniques used for sentiment classification would go beyond the scope of this paper. A wide range of methods for this task include supervised [40], unsupervised [64], semi-supervised [38] and active [37] learning techniques. Besides typical classification methods, methods based on matrix factorization have been investigated [39] and significant research attention gained the problem of constructing more appropriate feature representations for sentiment anlaysis [1]. For the details and review of sentiment classification methods, please refer to [41].

2.2. Imbalanced Learning

A dataset is called class imbalanced when the numbers of examples representing each class are not equal [21]. However, when talking about the problem of imbalanced

[1] assigning a polarity to several different aspects of a product/service assessed in a text

learning we refer to the situation in which the difference between class cardinalities in a dataset is somewhat extreme. The class with more examples is typically denoted as the majority class, whereas the underrepresented class is called the minority class. To measure class imbalance, the imbalance ratio is often defined as

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

where N_- and N_+ are cardinalities of the majority and the minority classes respectively. There is no clearly defined threshold on the imbalance ratio to consider a dataset as an imbalanced one, however, sometimes as a rule of thumb $IR > 1.5$ is used [14].

Recently, the problem of imbalanced learning gained significant attention from the research community [9, 21], which resulted in the development of many dedicated methods and algorithms. The approaches for imbalanced data are often categorized into algorithmic, data-level, and cost-sensitive ones. While algorithmic methods modify particular learning algorithms and cost-sensitive methods pose a costs-handling requirement on the classifier, the data-level methods are more universal since they modify the underlying data distribution and can be applied with virtually any classification method. Moreover, data-level methods often play a key role in the construction of ensemble approaches for imbalanced data which are considered as one of the most effective for imbalanced classification [16].

The data-level approaches for binary imbalanced problems can be roughly divided into under-sampling and over-sampling approaches (and combinations of both). Both groups of approaches modify the class distribution in a dataset to create a more suitable data distribution for a further applied learning algorithm. Usually, this means the creation of a more balanced dataset which may be constructed by either removing majority examples, artificially adding minority examples, or combining these two approaches. The first idea is explored by under-sampling methods whereas the second one is examined by over-sampling methods.

The most popular under-sampling method is Random Under-Sampling (RUS), which randomly deletes majority examples until the numbers of majority and minority examples are equal. Other under-sampling methods try to use some data characteristics to guide the sampling process. For instance, Tomek links [63] were proposed to steer the sampling process towards majority examples which lie close to the decision boundary between classes. A Tomek link exists between two data points if they belong to different classes and are the nearest neighbors of each other. The resampling method based on Tomek links (TL) detects all of such point pairs in a dataset and subsequently removes the majority example from each link. On the other hand, the Edited Nearest Neighbors method (ENN) [70] tries to clear the class overlapping region in a different way. For each example from the majority class, k nearest neighbors are found. If all of them belong to the majority class, then the example is retained in the dataset and removed otherwise. One-Sided Selection (OSS) [30] is another popular under-sampling method that, contrary to previously described approaches, starts with a dataset containing one majority example and incorporates selected majority instances during its operation. The method starts with a dataset

containing all minority instances and a single, randomly selected majority example. Then it re-classifies the original training set with 1-nearest neighbors rule and all incorrectly classified instances are added to the resampled training set. Finally, it applies Tomek links to clear the resampled dataset from the borderline and noisy examples.

Random Over-Sampling (ROS) which iteratively duplicates minority examples in order to fully balance class cardinalities is the simplest, yet effective, oversampling method. Another very popular method is the Synthetic Minority Over-sampling Technique (SMOTE) [10], which generates artificial minority examples. For a given minority example x, SMOTE finds the example's k-nearest minority neighbors and randomly selects one of them y. Then, it generates an artificial example which belongs to the minority class with features given by

$$x_{new} = x + r \cdot (y - x)$$

where r is a randomly generated number from 0 to 1. In other words, the new example is created as a random linear interpolation of a given minority example with its randomly selected minority neighbor. While SMOTE generates minority examples from each minority example iteratively, ADASYN [20] is an adaptive sampling method that focuses on difficult minority examples. The artificial examples are introduced in the same way as SMOTE does but the number of generated examples is proportional to the number of majority examples in the neighborhood. Another idea for guiding the over-sampling process is incorporated in Borderline-SMOTE [19]. This method generates minority examples only from those examples which: 1) have more majority neighbors than minority ones (potentially error-prone region) and 2) at the same time do not have all the neighbors from the majority class (presumably noisy example). Many other preprocessing methods were proposed in the literature for binary imbalanced data, including further extensions of SMOTE [13] and combinations of undersampling with oversampling e.g. SPIDER [62].

As mentioned earlier, data-level approaches are often incorporated into ensemble approaches such as bagging and boosting to construct ensemble models which are able to handle imbalanced data distributions. Most notably, there exist both theoretical recommendations [65] as well as an empirical evidence [5, 35] which favor bagging-based ensembles for imbalanced data. One particularly effective example of an under-sampling, bagging-based ensemble is Roughly Balanced Bagging (RBB) [23] which, as the name suggests, creates roughly balanced bootstrap samples. In each iteration of ensemble construction, it samples with replacement N_+ examples from the minority class, and later randomly samples a comparable number of majority examples. The exact number of majority instances is determined by taking a sample from the negative binomial distribution with the probability of success $p = 0.5$ and the number of successes set to N_+. The bootstrap samples created in this way are balanced on average. Furthermore, for high dimensional data which often occurs in text classification problems, an extension of RBB denoted as RBB+RSM (RBB with Random Subspace Method) was proposed in [35]. In every iteration, this method, aside from constructing an under-sampled bootstrap, selects a random subset of features on which the classifier is trained. This operation not only results in the construction of

less error-prone ensembles, but also increases ensemble diversity. On the other hand, SMOTEBagging [67] is a notable example of over-sampling bagging. This ensemble resulted from employing the SMOTE sampling procedure to construct bootstraps. An extensive comparison of bagging and boosting ensembles can be found in [16].

For a more detailed review of imbalanced learning methods refer to [21].

Multi-class imbalanced data Multi-class imbalanced data received limited attention from the research community which started to change in recent years [12, 72]. In general, multi-class imbalanced problems are considered much more difficult than their binary counterparts [66], for various reasons. For instance, the class imbalance in a multi-class dataset can occur in different configurations, i.e., it can contain one minority class and several majority ones or one majority and several minority classes etc. Likewise, the methods should try to capture and exploit the relations between multiple imbalanced classes as well as to be prepared to work with more extreme imbalances since they are more likely to occur in multi-class datasets.

The most prominent approaches for multi-class imbalanced data are those based on the decomposition of multi-class problems into a collection of binary ones. The one-against-rest (OAR) strategy creates an ensemble of binary classifiers in which every classifier is responsible for detecting one selected class only. For each component classifier, a binary training set is constructed by copying the original multi-class data and replacing all but one class labels with one common value. After training each component classifier on such a dataset, at prediction time the final decision of the ensemble is exercised by, e.g., selecting the class indicated by the most confident classifier. Analogously, one-against-one (OAO) decomposition creates a component classifier to distinguish between each possible pair of classes. The training sets for the component classifiers contain examples from two selected classes only and the final decision is given, e.g., by majority voting. In the context of imbalanced data, OAO and OAR ensembles are often combined with binary resampling methods described in the earlier part of this section. An extensive comparison of various combinations of decomposition approaches with data-level methods for imbalanced data can be found in [14]. Generally, OAO ensembles are usually preferred for imbalanced data since the binary problems constructed by OAR are even more extremely imbalanced than the original problem. On the other hand, OAO constructs a quadratic number of component classifiers which could be problematic and also costly for datasets with a high number of classes.

Recently, an extension of Roughly Balanced Bagging for multi-class imbalanced data has been proposed. Multi-class Roughly Balanced Bagging (MRBB) [35] establishes the cardinality of each class in the bootstrap by taking a sample from the multinomial distribution with uniform event probabilities. After establishing the number of examples which must be drawn from each class, random sampling with replacement is performed for each class separately. Contrary to its binary counterpart, which is an under-sampling bagging technique, MRBB can be parametrized by the bootstrap size to perform oversampling as well as under-sampling. The authors of [35] investigated two parametrizations of the ensemble, they set the bootstrap size to 1) the size of the smallest minority class which results in under-sampling (uMRBB) and 2) to the size

of the whole training set which results in oversampling minority classes and undersampling majority ones (oMRBB). Their experimental study suggests that uMRBB works better than oMRBB and a straightforward extension of RBB to multi-class data.

2.3. Imbalanced Sentiment Classification

The problem of imbalanced data in the context of sentiment classification was initially overlooked by the research community because typically studied datasets were purposely prepared to contain the same number of positive and negative reviews (see datasets [4, 53, 54]). However, the balanced data assumption does not hold in practice where habitually positive reviews overwhelm negative ones. This phenomenon is sometimes attributed to marketing actions of product manufacturers and vendors, as well as to the fact that products with good reviews are more often selected by customers, hence, receiving even more, mostly positive, reviews [38]. Burns et al. [8] were among the first who noticed the problem and showed that using realistic unbalanced datasets results in the constructions of classifiers that work much better in practice.

Since then, a limited number of works on applying or adopting imbalance learning methods for sentiment classification has been carried out. In [38] random undersampling, over-sampling, one-class learning and a cost-sensitive method were experimentally investigated on four binary datasets. Li et al. [40] proposed a clustering-based under-sampling approach which performs clustering on majority class instances and then selects the representatives of the majority class from each detected cluster. More recently, SMOTE was used to over-sample text representations constructed by a recursive neural tensor network for English and Chinese sentiment data [74] and Roughly Balanced Bagging was successfully applied for Twitter sentiment analysis [33]. Furthermore, a whole pipeline consisting of Multiple Correspondence Analysis, SMOTE, and OAO decomposition was proposed for multi-class sentiment classification [29]. Additionally, in [59] resampling was used in the context of matrix decomposition methods for sentiment analysis which are out of the scope of this study.

All the above-mentioned works evaluate imbalance learning methods for sentiment classification on a very limited number of datasets. Especially, the works involving multi-class imbalanced data [29, 74] use one multi-class dataset each. Another related empirical study [46] investigates only three under-sampling methods by evaluating it on three binary datasets. For the above reasons, we believe that there is a need for a more extensive evaluation of imbalanced learning methods for sentiment classification, both in terms of investigated algorithms and evaluated datasets. This need is particularly evident in the context of multi-class problems, since earlier studies of imbalance learning methods usually do not take into consideration the multi-class data which is of special focus of this paper. It is also worth highlighting that the lack of comparison between methods has resulted the recommendations that are sometimes contradictory. For instance, some researchers claim that under-sampling is the most suitable methods for the sentiment classification [38], whereas others claim that it is

Table 1. Basic characteristics of datasets under study.

Dataset	IR	# classes	# examples	# features			
				base	bow	mixed	cbow
books	2.72	4	2000	30	5305	149594	300
dvd	2.5	4	2000	30	5097	153284	300
electronics	2.12	4	2000	30	2888	101635	300
housewares	3.07	4	2000	30	2585	88828	300
movie_aut1	10.08	8	1027	30	3788	102592	300
movie_aut2	26.08	10	1306	30	7354	184168	300
movie_aut3	11.87	10	902	30	4684	116257	300
movie_aut4	52.75	9	1768	30	6833	198032	300
restaurants	3.17	4	1325	30	1004	35530	300
tripadvisor	6.37	5	20491	30	14231	435810	300
tweeter5point	33.31	5	9090	30	2913	104938	300
tweeter3point	3.13	3	9090	30	2913	104938	300

inappropriate and prefer over-sampling [74]. Furthermore, earlier works frequently investigate the methods' performance on a single set of features whereas different feature engineering procedures can result in extremely different text representations and could change the method recommendation.

3. Datasets and feature representations

3.1. Data description

For this study, we gathered twelve sentiment classification datasets which contain opinions and reviews on a wide range of topics. We selected corpora of long professional reviews, as well as collections of brief tweets with informal opinions. The basic characteristics of each dataset, such as the number of examples, the number of classes and the imbalance ratio, can be found in Table 1. Given that the imbalance ratio is defined for a pair of classes and all the selected datasets contain multi-class problems, we report the highest imbalance ratio among all the classes, i.e., the ratio between the biggest and the smallest class. The table additionally presents the number of features generated by different feature engineering procedures which will be described later.

The Movie Review Dataset [53] contains four corpora of full-length movie reviews written by four different film critics. The reviews were collected from several Web pages, hence, rating evaluations of different critics are in different scales e.g. in 5-stars scale (with possible half-stars) or in a 100-point scheme. Besides that, the dataset's authors indicate that distinct critics can understand the same rating differently e.g. for one critic giving 2-stars can mean negative opinion, whereas for another it can be still a slightly positive review. This led us to evaluate the reviews for those four authors as separate datasets denoted as `movie_aut1`, `movie_aut2`, `movie_aut3` and

`movie_aut4` — similarly as it was done in earlier studies [53]. The classes with less than 10 training examples were merged to the closest class on the scale in order to have enough data for learning and testing. For the same reason, the ratings on the 100-point scale where rounded to a 10-point scale. Each of such prepared datasets has a highly imbalanced distribution with imbalance ratio ranging from 10.08 to 52.75.

Blitzer et al. [4] collected four review datasets from Amazon which are also used in our study. Each dataset consist of 2000 reviews from four product categories: books (`book` dataset), electronics (`electronics`), DVDs (`dvd`), and kitchen appliances (`housewares`). Although the original datasets contain many additional pieces of information such as review date or product name, in this study we used only rating evaluation as the target class and the review text to construct features. The review system on Amazon originally allows for 5-star evaluation, but the authors of the dataset excluded 3-star reviews due to their class binarization strategy, hence, the datasets contain four classes only.

Two datasets under study contain rather short reviews, which typically do not exceed ten sentences. The `tripadvisor` dataset [68] contains opinions about hotels expressed at the TripAdvisor website in a one-month period at the turn of February and March 2009. This dataset is already preprocessed with simple normalization and stemming, and has a target class containing 5-star user evaluations. On the other hand, the restaurants' reviews collected from the Citysearch webpage by Ganu at al. [17] were labeled manually. The `restaurants` dataset contains four classes: positive, negative, neutral, and conflict class with the last one meaning that the review contains both positive and negative utterances without a clear winner[2].

The last pair of datasets, `tweeter3point` and `tweeter5point`, was taken from the Semantic Evaluation competition [47]. These datasets contain utterances collected from Twitter expressing opinions about 200 different topics. To annotate the data, Mechanical Turk and CrowdFlower services were used, with the final class label being consolidated among several annotators. Since Tweeter's technical constraints limit the length of the utterance to 140 characters[3], the texts in these datasets are extremely short.

3.2. Feature representations

Basic feature representation Our basic feature representation (later denoted as base) consists of features constructed from handcrafted sentiment lexicons. Such lexicons basically provide a list of words with sentiments assigned to them e.g. "excellent" will be listed as a word with positive sentiment. Depending on the dictionary, the sentiment intensity is sometimes expressed numerically, also additional emotions such as joy or trust can be listed.

To construct our representation, we used five well-known dictionaries: the NRC emotion lexicon [45], the Opinion lexicon [24], the NRC Hashtag Affirmative/Negated

[2]An example of such review is: "The fish was adequate but inexpertly sliced".

[3]Twitter recently changed its policy and doubled the maximal length of a tweet, but it was after the collection of these datasets.

Context Sentiment lexicon [27], the Multi-perspective Question Answering corpus [69], and SentiWordNet [2]. From each dictionary, a feature was created for each listed emotion or sentiment. The value of such a feature was calculated by taking a sum over the polarity scores assigned to all words in the text. In the case of lexicons without numerical polarity scores, the count of words with particular sentiment was used instead of the sum. Additionally, from the Affirmative/Negated Context Sentiment Lexicon two more features were calculated by taking (besides the sum) the minimal and maximal value of word's sentiment.

Bag-of-words representation The bag-of-words representation (bow) is a classical feature set for text and its definition can be found in any text mining textbook e.g. in [58]. Prior to extracting features, we performed some standard preprocessing of text. The text was tokenized, lemmatized by NLTK Word-Net Lemmatizer [42], also a hand-crafted stopwords list was applied. Finally, certain symbols such as urls, numbers, dates etc. that occurred less than five times in the dataset were grouped according to their meaning using regular expressions. The rest of the tokens that occurred less than five times was removed from the representation.

Mixed feature representation We have also adopted a mixed text representation which was previously successfully used by us in SemEval competition [47]. In our study, this feature collection was intended to represent a typical feature set of an efficient sentiment classification system. This representation is denominated as mixed because it contains all the features from bow and base representations. Additionally, word $2, 3, 4, 5$-grams, negation unigrams and bigrams, part-of-speech unigrams, $3, 4, 5$-character-grams and words' representations from Brown clustering with 1000 clusters were used as features. For a more detailed description of this representation, please refer to [33].

Continuous bag-of-words representation Recently, word embeddings have revolutionized the construction of text features. In this study, we adopted vectors from a classic word2vec model [44] with 300 dimensions which was pre-trained on the Google News corpus. In order to create a feature representation of the whole text, the text is preprocessed (just like for bow representation) and, subsequently, arithmetic mean of vectors corresponding to all text's words is calculated and used as feature representation.

4. The impact of class imbalance on sentiment classification

Initially, high imbalanced ratio seemed to be the main indicator of difficulty in imbalanced learning [22]. However, systematic experimental studies performed on decision trees revealed that for simple, linearly separable problems any amount of class imbalance is virtually of no harm the final classifier's predictive performance [25]. It was also observed that with growing difficulty of the learned concept, the impact of

class imbalance rapidly increases and severely degrades the quality of induced classifiers [60]. This led to a conclusion that the real challenge is not the class imbalance itself but its combination with other data difficulty factors [25, 50, 34, 73].

Several different data difficulty factors were defined and studied in the imbalanced learning community. Jo and Japkowicz [26] studied the problem of small disjuncts which is a result of under-represented minority sub-concepts. The decomposition of the minority class into numerous sub-concepts is often closely related to the previous one but also to the problems arising from the unequal representation of different sub-concepts (within-class imbalance). Another important data difficulty factor is class overlapping [18, 61] which, similarly to other factors, always introduces complications to the induction of a correct decision boundary. However, in the presence of imbalanced data, it always hinders the decision boundary towards better recognition of the majority class [65]. Finally, it was also noticed that the impact of class imbalance is more harmful when the sample size is small [21].

Although difficulty factors play an important role in imbalanced learning and their analysis led to the development of several successful algorithms for class-imbalanced data [5, 49], to the best of our knowledge, they were never properly analyzed in the context of sentiment classification. Therefore, we decided to perform an analysis of data difficulty factors on real opinion datasets.

Measuring data difficulty factors in real datasets is not a trivial task even for binary datasets. Napierała and Stefanowski [50] proposed a method based on the analysis of minority examples' nearest neighbors to distinguish between four types of examples which are closely related to several difficulties. The authors of [32] designed a special clustering algorithm which detects these types of examples together with providing an estimation of the number of sub-concepts. Recently, an extension of Napierała's method has been proposed for measuring data difficulties in multi-class imbalanced data [34], which will be the method adopted to our multi-class study. Performing such analysis will give us an indication about the true impact of class-imbalance on sentiment classification.

The method consists of measuring the safety of each minority class example and, as an indicator of class difficulty, the average or median of those values is used. The safety of an example is given by[4]

$$safety(x) = \frac{|\{y : y \in NN_k(x) \wedge Class(y) = Class(x)\}|}{k}$$

where $NN_k(x)$ is a set of k-nearest neighbors of the example x and $Class()$ is assumed to return the class value of an example. This method, although quite simple, proved to provide a fairly good estimation of the class difficulty in multi-class imbalanced datasets in the experiments performed both on artificial and real datasets [34].

Although the method originally was used together with HVDM distance [71] to calculate example neighborhood, we decided to also run the method with cosine distance, which is usually considered a more appropriate measure for textual data [58].

[4]The original method proposed in [34] assumes that expert knowledge on class similarities is available. Since such information is not available for our datasets, we use this methodology assuming that the similarity between each pair of classes is equal to 0.

Table 2. The average safety of minority classes calculated by method described in [34] with HVDM and cosine distance.

	avg. safety (HVDM)				avg. safety (cosine)			
	base	bow	mixed	cbow	base	bow	mixed	cbow
books	0.223	0.144	0.143	0.283	0.197	0.234	0.271	0.234
dvd	0.258	0.161	0.153	0.269	0.227	0.209	0.275	0.253
electronics	0.224	0.115	0.181	0.278	0.203	0.213	0.264	0.231
housewares	0.150	0.219	0.163	0.203	0.157	0.196	0.216	0.182
movie_aut1	0.078	0.053	0.060	0.125	0.080	0.106	0.122	0.126
movie_aut2	0.078	0.035	0.015	0.143	0.083	0.115	0.102	0.165
movie_aut3	0.075	0.049	0.053	0.081	0.084	0.073	0.084	0.097
movie_aut4	0.134	0.070	0.057	0.174	0.130	0.136	0.179	0.152
restaurants	0.373	0.401	0.359	0.306	0.391	0.417	0.447	0.387
tripadvisor	0.266	0.136	0.119	0.225	0.298	0.181	0.222	0.307
tweeter5point	0.159	0.181	0.142	0.202	0.165	0.216	0.217	0.217
tweeter3point	0.227	0.178	0.102	0.298	0.254	0.266	0.269	0.277
average	0.187	0.145	0.129	0.216	0.189	0.197	0.223	0.219

Since a full report of average safety for each class in the datasets would take a considerable amount of space, Table 2 presents an average of all minority classes' safeties on the datasets under study[5]. We note that a class was treated as a minority one (and hence included into the average) when its imbalance ratio was higher than 1.5. Following earlier studies [35, 32], the method was run with the size of the analyzed neighborhood $k = 5$.

Regardless of the feature representation and the distance metric being used, all average safety values are rather low. Napierała and Stefanowski [48, 50] categorized binary imbalanced datasets into datasets with safe, borderline, rare and outlier characteristics (from the easiest to the hardest one). Following their labeling methodology, all the datasets under study fall into two of the most difficult categories. Only the `restaurants` dataset could be treated as a borderline one, but usually safety values for such datasets are higher. It is noteworthy that such low safety values are not a direct result of a multi-class nature of a dataset. The most difficult dataset reported in the earlier study on multi-class data [34] had the average of minority classes safeties equal to 0.11. Here, we observe safety values even several times smaller for some movie review data. This gives some intuition about the general hardness of multi-class sentiment classification and demonstrates that all considered datasets pose very difficult imbalanced learning tasks.

Low safety values are associated with datasets with the highest imbalance ratios, which demonstrates the impact of class imbalance on the data quality. These low values suggest that the application of data preprocessing methods possibly could improve data quality and, hence, lead to some improvements. They also advocate

[5]See the full safety reports on accompanying website: `http://www.cs.put.poznan.pl/mlango/publications/imbalanced-sentiment-class.html`

the construction of better feature representations for this problem since potentially one could create feature spaces with a clearer separation between classes [57]. Interestingly, contrary to a common belief that the classification of short utterances is more challanging [47, 33], there is no apparent correlation between safety values and the length of a review. Moreover, for datasets with full-length reviews, we report the smallest values of safety which indicate that the problem of class imbalance is important in sentiment classification regardless of review length.

The difference between results reported on HVDM and cosine distance is rather evident. The method with employed cosine measure to neighborhood estimation seems to generally return higher values for our data. On the other hand, the method with HVDM returns results which are rather counter-intuitive. For instance, the most advanced feature representation which contains many specifically designed features for sentiment classification and which usually achieves quite good results on this task [33] resulted to be the least safe. The same representation is the safest one according to the method with cosine distance. This result is also quite interesting since the methods which label examples based on safety were extensively experimentally studied. It has been shown that there is no significant impact of using different sizes of neighborhoods or even of replacing the nearest neighbors method with kernel-based estimation [6, 50]. Here, we found some evidence that the choice of proper distance measure can be critical for such methods in some domains.

Further analyzing the results for cosine measure which seems to be more appropriate, we see that the representation which creates the least safe feature spaces is our baseline representation. On the other hand, bag-of-words representation is only slightly more safe overall. However, there are several datasets on which lexicon-based representation is safer than their bag-of-words counterpart. There is also evidence that the representations created by continuous bag-of-words are safer than the discrete ones. Interestingly, especially on data with shorter texts, this representation has quite often comparable or better safety properties than the mixed representation which uses many additional pieces of information such as part-of-speech tags, hand-crafted lexicons etc.

Beside our analysis, we also acknowledge that for some domains the availability of labeled data is very limited. The collection of such data is quite expensive since its creation requires working with internal human experts [17] or using crowdsourcing platforms [47] which probably will force the majority of researchers and practitioners to work with rather small datasets. As a consequence, it further aggravates the impact of class imbalance on the predictive accuracy of sentiment classifiers.

5. Comparison of imbalanced learning methods for sentiment classification

5.1. Experimental setup

In order to evaluate the usefulness of imbalanced learning methods for sentiment classification, we performed computational experiments with four classifiers combined with twelve imbalanced learning methods. The experiments were performed in Python using the scikit-learn [55] and imbalanced-learn [36] libraries.

We measured the prediction capabilities of classifiers in terms of two popular measures in imbalanced learning. F-measure [33, 47] and G-mean [40, 74, 29] were studied in many related works for sentiment classification. Both of these measures were initially defined for binary classification tasks and require an adjustment for multi-class data. For F-measure macro-averaging is used, i.e., we compute an average over F-measures calculated for each class separately. F-measure for a given class is computed as unweighted harmonic mean of its precision and recall. On the other hand, G-mean is computed as a geometric mean of recall values over all classes. As it was earlier noticed [32], G-mean can resolve to zero when any of the classes is completely unrecognised. This can happen quite often while working with multi-class data with at least one class with an extremely small number of examples in the training set. To alleviate this issue, we used corrected recall defined as $Recall_{Corr}(c) = \max(Recall(c), 0.001)$ in our G-mean calculations.

As learning algorithms, we decided to use four very popular classifiers, namely Naive Bayes (NB), Multinomial Logistic Regression (MLR), CART decision trees (DT) and k-Nearest Neighbors (kNN). The algorithms were tuned by trying out all the combinations of the following parameters and selecting the best one (via grid search):

- Naive Bayes — since this classifier has virtually no parameters, we have not tuned it. However, discrete bag-of-words features were modelled with multinomial distribution whereas continuous values were modeled with Normal distribution which can be seen as an implicit form of tuning.
- Multinomial Logistic Regression classifiers were trained with the SAGA optimizer[6] because in our preliminary experiments this optimizer was achieving the best results. In our hyperparameter optimization procedure, we were selecting the amount of L2 regularization controlled by the C parameter. We experimented with five consecutive values of this parameter on the logarithmic scale, namely 0.01, 0.1, 1, 10 and 100.
- k-Nearest Neighbors was tested in two variants. The first one was weighting points by the inverse of their distance (closer neighbors having a greater influence on the classifier's decision) whereas the second one was treating all the neighbors equally. Additionally, we considered 1, 3, 5, 7 and 9 as possible values of the neighborhood size k. Cosine distance was used as a distance function.

[6] See scikit-learn documentation for a discussion of this optimizer as well as descriptions of other parameters mentioned later in the text.

- Decision Tree — for this classifier we controlled the maximum depth of the induced tree. The following depths were tested: 10, 20, 40, 80, plus the construction of a full tree.

All the other parameters that are not mentioned above were set to default values in the scikit-learn package.

The results were calculated using stratified 5-fold cross validation (CV) with the hyperparameter optimization performed in every iteration. During each iteration of CV, the subset of data intended for the training was additionally divided into training (90%) and validation (10%) parts. The whole tuning process was performed using only those training and validation parts, with the final evaluation performed on the previously unseen testing examples. Note that in each CV iteration, the training, validation, and testing sets were different and the whole hyperparameter optimization was run.

We incorporated to our study eight well-known preprocessing techniques for imbalanced data which are implemented in imbalanced-learn package [36]. Half of them being oversampling techniques: namely Random Over-Sampling (ROS), SMOTE (SMT), Borderline-SMOTE (BS), ADASYN (ADA), and another half being undersampling-based methods: Random Under-Sampling (RUS), Edited Nearest Neighbors (ENN), One-Sided Selection (OSS) and Tomek links (TL). Those methods were proposed in the literature for binary datasets, hence, their simple adjustments implemented in imbalanced-learn package [36] for multi-class data were used. For the simplest methods such as ROS or RUS, these adjustments consist of applying resampling on each class independently. For more advanced methods requiring neighborhood analysis, the amendments additionally consist of using a one-against-rest approach during k-nearest neighbors estimation (all the necessary statistics are computed by treating all the other classes as one).

There is also a very limited number of resampling methods designed especially for multi-class imbalanced data. Zhou and Liu [75] proposed a class weighting schema rooted in cost-sensitive learning framework which sometimes is referred in the literature as Global-CS method. The multi-class ROS which is included in our study is a realization of this weighting strategy. Fernández-Navarro et al. [15] proposed a generalization of SMOTE, called Static-SMOTE, which similarly to the SMOTE implementation used in our study also iterates over classes and applies one-against-rest strategy during neighborhood estimation. This similarity together with not particularly encouraging earlier experimental results [14] were the reasons why we have abandoned the direct implementation of Static-SMOTE.

Other, presumably much more popular, approaches for multi-class imbalanced data based on multi-class decomposition ensembles were incorporated to the study. According to an earlier experimental study [14], one-against-one ensembles outperform one-against-rest ensembles on imbalanced data. Hence, we restricted our study to two combinations of one-against-one ensembles with preprocessing methods: random under-sampling (OAO+RUS) and random over-sampling (OAO+ROS). We also experimented with Multi-class Roughly Balanced Bagging in its two versions: oMRBB and uMRBB which were described in Section 2, each of them using 100 component classifiers. Both versions were combined with the random subspace method, as rec-

ommended by MRBB authors for high-dimensional data [35].

5.2. Comparison of preprocessing methods

We start the discussion of experimental results by trying to identify the best classifier-preprocessing pair. Table 3 presents the best classifier-preprocessing combinations in our study[7]. The best approaches were selected by calculating its average rank across all dataset representations, an approach taken from the procedure of the Friedman test, with the lowest rank meaning the best result. For both measures, all the strongest approaches were based on the logistic regression classifier. The further analysis of rank values suggests that the choice of a classifier is much more important than the choice of a preprocessing method. The highest ranks were clearly assigned to logistic regression, followed by Naive Bayes. At the end of the ranking, kNN and decision trees take mixed positions, without a clear winner.

Table 3. The best combinations of classifiers and preprocessing methods for the datasets under study.

	F-measure			G-mean		
No.	Class.	Preprocessing method	Avg. Rank	Class.	Preprocessing method	Avg. Rank
1	MLR	RUS	5.08	MLR	ENN	9.83
2	MLR	ENN	5.66	MLR	ROS	10.16
3	MLR	One Sided Selection	5.75	MLR	One Sided Selection	10.41
4	MLR	Borderline-SMOTE	6.50	MLR	Borderline-SMOTE	11.33
5	MLR	Tomek Links	6.83	MLR	No preprocessing	11.50
6	MLR	SMOTE	7.08	MLR	Tomek links	12.16

For logistic regression, under-sampling approaches seem to be the best ones for the optimization of F-measure - especially RUS, ENN, and OSS. Also, all the imbalanced methods outperformed the baseline (no pre-processing). This did not happen for G-mean where the baseline was better than half of the preprocessing methods, including RUS which was the winner for F-measure. In general, G-mean seems to be more stable and difficult to optimize by the learners which can be observed by looking at the top ranks in Table 3. The best method for F-measure, on average, ranks around the fifth position for a considered dataset whereas the winner for G-mean ranks on the tenth position on average which indicates that it is more difficult to identify the outperforming method for that measure. Nevertheless, if we compare the results on the two measures, ENN and OSS under-sampling approaches seem to give a good trade-off between these two measures.

[7]Since our experiment includes four classifiers, eight preprocessing methods (plus lack of preprocessing) and four representations on twelve datasets taking two measures into account, the table with all the results would occupy several pages. Instead, we preferred to present several summaries of this data in the text. The reader is welcome to check out the full results on the accompanying website `http://www.cs.put.poznan.pl/mlango/publications/imbalanced-sentiment-class.html`.

In order to evaluate the overall performance of preprocessing methods for imbalanced data, we calculated the highest possible result for each method and dataset i.e. assuming that the best-performing classifier and dataset representation were selected a posteriori. Tables 4 and 5 presents the results of this computations for F-measure and G-mean, respectively.

Table 4. The best values of F-measure obtained for a given dataset and preprocessing method.

Dataset	Orig.	ROS	SMT	BS	ADA	RUS	ENN	OSS	TL
books	0.504	**0.510**	0.506	0.503	0.502	**0.510**	0.497	0.498	0.506
dvd	0.489	0.494	0.497	0.499	0.502	0.494	0.492	**0.507**	0.500
electronics	0.521	0.511	0.507	**0.529**	0.523	0.525	0.524	0.524	0.508
housewares	0.499	0.511	0.497	0.501	0.504	**0.514**	**0.514**	0.496	0.499
movie_aut1	**0.233**	0.223	0.214	0.225	0.220	0.223	0.223	0.224	0.225
movie_aut2	0.210	0.217	0.221	0.235	0.208	0.213	**0.239**	0.229	0.221
movie_aut3	0.126	0.124	0.152	0.137	0.135	0.142	0.142	**0.154**	0.135
movie_aut4	0.410	0.385	**0.412**	0.374	0.372	0.373	0.409	0.404	0.379
restaurants	0.682	0.675	0.677	0.681	0.674	**0.696**	0.676	0.692	0.680
tripadvisor	0.539	0.540	0.543	0.541	0.541	0.544	0.543	**0.545**	**0.545**
twitter3point	0.539	**0.544**	0.542	0.537	0.538	0.542	0.538	0.538	0.541
twitter5point	0.339	0.336	0.342	0.343	0.345	0.345	0.344	0.333	**0.346**

Table 5. The best values of G-mean obtained for a given dataset and preprocessing method.

Dataset	Orig.	ROS	SMT	BS	ADA	RUS	ENN	OSS	TL
books	0.435	0.435	0.431	**0.437**	0.432	0.430	0.427	0.423	0.423
dvd	0.424	0.429	0.435	0.437	0.426	0.429	0.441	**0.466**	0.423
electronics	0.456	0.453	0.464	**0.464**	0.457	0.461	0.461	0.456	0.451
housewares	0.398	0.404	0.405	0.400	**0.417**	0.412	0.399	0.393	0.384
movie_aut1	0.113	0.090	0.077	0.101	0.097	**0.125**	0.082	0.111	0.101
movie_aut2	0.080	**0.130**	0.074	0.078	0.096	0.100	0.093	0.093	0.114
movie_aut3	0.046	0.052	0.063	0.050	0.047	0.048	**0.064**	0.055	0.056
movie_aut4	0.216	0.205	**0.223**	0.193	0.190	0.182	0.221	0.219	0.188
restaurants	0.656	0.649	0.646	0.653	0.643	**0.670**	0.646	0.667	0.653
tripadvisor	0.502	0.506	0.507	0.505	0.506	**0.510**	0.508	0.509	**0.510**
twitter3point	0.513	**0.515**	0.514	0.507	0.509	**0.515**	0.508	0.507	0.510
twitter5point	0.372	0.365	0.369	0.358	0.351	0.353	0.366	0.360	**0.373**

A Friedman test performed on the results presented in Tables 4 and 5 does not indicate any statistically significant differences between the preprocessing methods. For both measures, the best performing method is Random Under-Sampling. Taking G-mean into account, RUS (rank equal to 4) is followed by SMOTE (4.42), ROS (4.75) and, ex aequo, OSS and ENN (4.83). ADASYN (6.09) oversampling is the

only method which performed worse than the baseline (5.42), which in turn was only slightly outperformed by Tomek links and the BS method (5.33). ADASYN was also the least effective method in terms of F-measure and, together with ROS, was one of the methods which did not perform better than the baseline. The top of the approaches' ranking with respect to this measure was dominated by under-sampling methods: RUS (3.75), OSS (4.5), Tomek links (4.58) with SMOTE in the second half of the ranking.

Looking at Table 4 we observe that there are no substantial differences between different preprocessing methods and the baseline. Even though for one dataset (`movie_aut1`) all the preprocessing methods were worse than the baseline, the improvements were usually pretty small. The highest observed improvements were those for longer texts, i.e.: `movie_aut2` and `movie_aut3` (around 3%) and for `dvd`, `restaurants`, `housewares` (around 1.5%). Similarly, the highest improvements for G-mean were on a similar set of datasets: `movie_aut4` and `dvd` (approx. 4%) as well as `restaurants`, `housewares`, `movie_aut1` and `movie_aut3` (above 1%). Notably, two of the datasets (`dvd` and `restaurant`) on which we get improvements on both measures, were the ones with the highest safety levels calculated in Sec. 4. Another such dataset (`movie_aut3`) seems to suffer from the difficulty of small sample size (it is the smallest one in our study).

Table 6. Comparison of the best results achieved for different text representations.

	F-measure				G-mean			
Dataset	base	bow	mixed	cbow	base	bow	mixed	cbow
books	0.371	0.491	0.510	0.485	0.348	0.437	0.430	0.435
dvd	0.390	0.507	0.507	0.474	0.367	0.466	0.442	0.441
electronics	0.390	0.504	0.529	0.481	0.352	0.464	0.464	0.442
housewares	0.376	0.491	0.514	0.460	0.331	0.417	0.412	0.369
movie_aut1	0.172	0.223	0.214	0.233	0.081	0.125	0.076	0.113
movie_aut2	0.148	0.208	0.186	0.239	0.059	0.059	0.056	0.130
movie_aut3	0.126	0.135	0.135	0.154	0.044	0.040	0.040	0.064
movie_aut4	0.211	0.323	0.412	0.286	0.111	0.151	0.223	0.150
restaurants	0.571	0.642	0.696	0.609	0.495	0.617	0.670	0.588
tripadvisor	0.433	0.522	0.545	0.533	0.355	0.485	0.510	0.494
twitter3point	0.443	0.539	0.544	0.514	0.416	0.515	0.515	0.491
twitter5point	0.252	0.344	0.346	0.334	0.184	0.247	0.202	0.373

Finally, we compared the performance of four text representations. Table 6 presents the values of F-measure and G-mean for each dataset and representation, assuming that the best classifier and preprocessing method were selected.

The performed Friedman test indicates that there are some statistically significant differences between the representations for both measures ($p-value < 0.01$). The Nemenyi post-hoc analysis indicated that bow, cbow and mixed representations are better than base representation, however, they are not distinguishable between each other for F-measure. A similar analysis for G-mean indicates that only the bow

representation is distinguishable from the base representation.

5.3. Comparison of ensemble methods

We start the discussion of the performance of ensemble methods dedicated to imbalanced data for sentiment classification by looking at the best pairs of a component classifier and ensemble strategies. Table 7 presents the top-ranked pairs on both classification measures used in this study. Similarly to the ranking of preprocessing methods, this ranking is also dominated by logistic regression. MLR and OAO with Random Under-sampling is clearly the best combination for both measures with the rank values being close to 1, which means that it was the best method for the vast majority of datasets and feature representations. The bagging ensemble combined with logistic regression did not outperform a single classifier with no preprocessing on F-measure, however, it definitely outperformed it on G-mean (the rank of a single MLR is 9.91).

Table 7. The best combinations of classifiers and ensemble methods for the datasets under study.

	F-measure			G-mean		
No.	Class.	Ensemble method	Avg. Rank	Class.	Ensemble method	Avg. Rank
1	MLR	OAO+RUS	1.33	MLR	OAO+RUS	1.08
2	MLR	OAO+ROS	3.25	MLR	OAO+ROS	3.41
3	MLR	Single classifier	4.08	MLR	oMRBB	3.91
4	MLR	oMRBB	5.33	NB	OAO+RUS	4.50
5	MLR	uMRBB	6.58	MLR	uMRBB	4.66

The results of ensemble methods on different datasets are shown in Table 8. By using an ensemble we can improve the result on any dataset by at least 1% on F-measure and by at least 3% on G-mean. By picking the best OAO approach for a dataset, we are able to improve F-measure by 2.3% and G-mean by 8.8% on average. Bagging ensembles again show not so good results in this comparison. Always taking the best variant of MRBB we get an average improvement of G-mean by 4% and degradation of F-measure (approx. 1%).

Comparing OAO approaches, the variant with Random Over-Sampling is always giving the same or better value of F-measure than its under-sampling counterpart. Particularly significant differences can be found on datasets such as `twitter5point` (6.7% improvement in comparison with OAO+RUS), `movie_aut1` (4.4%) and `movie_aut3` (3.4%). In the contrast, OAO with under-sampling works better in terms of G-mean, obtaining even higher increases e.g. 10% on `movie_aut2`, 8% on `housewares`, 7% on `movie_aut4`. A similar observation can be made on MRBB, where oMRBB works better on F-measure and uMRBB provides better results on G-mean.

This interesting relationship can be also observed in Table 9, which presents the average ranks of F-measure and G-mean values computed in the same way as previously

Table 8. The best values of F-measure and G-mean obtained for a given dataset and ensemble method. oMR and uMR are abbreviations from Over-sampling / Under-sampling Multi-class Roughly Balanced Bagging. RUS and ROS denotes OAO approaches combined with Random Under-Sampling and Random Over-Sampling.

	F-measure					G-mean				
Dataset	Single	oMR	uMR	ROS	RUS	Single	oMR	uMR	ROS	RUS
books	0.504	0.477	0.480	**0.515**	0.499	0.435	0.470	0.474	0.463	**0.506**
dvd	0.489	0.496	0.479	**0.516**	**0.516**	0.424	0.495	0.476	0.475	**0.521**
electro.	0.521	0.480	0.490	0.538	**0.544**	0.456	0.459	0.449	0.491	**0.548**
housewa.	0.499	0.487	0.476	**0.520**	0.512	0.398	0.435	0.437	0.445	**0.528**
movie_1	0.233	0.229	0.216	**0.265**	0.220	0.113	0.242	0.177	0.233	**0.248**
movie_2	0.210	0.232	0.198	**0.256**	**0.256**	0.080	0.205	0.209	0.156	**0.264**
movie_3	0.126	0.148	0.152	**0.171**	0.136	0.046	0.094	**0.144**	0.122	0.134
movie_4	**0.410**	0.322	0.220	0.375	0.360	0.216	0.187	0.201	0.184	**0.255**
restau.	0.682	0.646	0.649	**0.701**	0.699	0.656	0.626	0.632	0.688	**0.713**
tripadv.	0.539	0.469	0.470	**0.561**	0.537	0.502	0.476	0.479	0.549	**0.552**
twitter3.	0.539	0.526	0.526	**0.559**	0.552	0.513	0.531	0.518	0.552	**0.572**
twitter5.	0.339	0.335	0.302	**0.384**	0.317	0.372	0.397	0.387	0.393	**0.427**

Table 9. Average rank (as in Friedman test) of ensemble methods with different component classifiers. oMR and uMR are abbreviations from Oversampling / Under-sampling Multi-class Roughly Balanced Bagging. RUS and ROS denotes OAO approaches combined with Random Under-Sampling and Random Over-Sampling.

	F-measure					G-mean				
Comp.	Single	oMR	uMR	ROS	RUS	Single	oMR	uMR	ROS	RUS
DT	4.00	1.75	1.75	3.25	4.25	3.83	3.25	1.67	3.92	2.33
kNN	3.67	1.25	2.67	3.25	4.17	4.92	2.67	1.92	2.75	2.75
MLR	2.92	3.92	4.42	1.25	2.50	4.58	3.08	3.42	2.83	1.08
NB	2.83	3.50	4.17	1.75	2.75	3.75	3.50	3.42	3.25	1.08

but treating each learning algorithm separately. Clearly, for F-measure oversampling approaches have lower ranks where the opposite is true for G-mean.

Additionally in Table 9, one can observe two groups of component learners with similar ensemble methods rankings. Decision Trees and kNN obtain the highest results while working with MRBB, whereas with the weakest decomposition approach they obtain results that are worse than a single classifier (for both measures). On the other hand, for Logistic Regression and Naive Bayes, the decomposition approaches are most beneficial. Such a result can be explained by bagging's ability to reduce the prediction variance [31], hence, they are most beneficial while using unstable classifiers like decision trees.

We have also performed a comparison of the performance of different dataset representations, similar to that performed for preprocessing methods. However, we

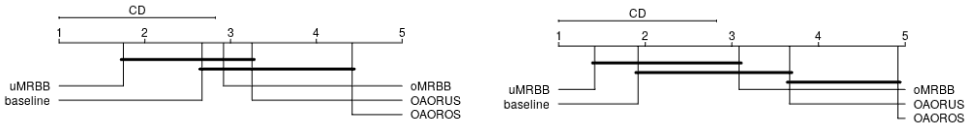

Figure 1. The visualizations of Nemenyi post-hoc test for the ensemble's training time (left) and model size (right).

skip its discussion here since the results and conclusions were quite analogous.

Dedicated ensembles for imbalanced data visibly improved the predictive capabilities of induced classifiers for sentiment classification, thus, it can be purposeful to use them in practice to advance current systems. For this reason, we believe that an additional analysis taking into account two important practical considerations, namely the training time and the model size, was necessary. The model size was measured by serializing a fitted model to a file using pickle library, which is a standard, general-purpose serialization library in Python. Such type of measurement should not favor any type of models, at the same time taking into account all the necessary factors. Both time and memory measurements were averaged over all models constructed during cross-validation without taking into account models induced during hyperparameter optimization. The time of hyperparameter optimization was also omitted since in our setup it is rather a property of a component learner algorithm than of the ensemble method.

The gathered measurements were analyzed by Friedman test which rejected the hypothesis about the lack of significant differences for both training time and model size ($p\text{–}value < 0.01$). The results of Nemenyi post-hoc analysis are presented on Figure 1. uMRBB is better than other algorithms both in terms of time and model size. Interestingly, on average it constructs smaller models and is faster even than a single model trained on the original dataset. Even though uMRBB constructs 100 component classifiers during training, each of them is trained on a small, under-sampled dataset which permits fast model construction. Taking into account this property of random under-sampling, it is not surprising that OAO with under-sampling also allows for faster training than its counterpart with over-sampling. The additional difference in time and memory usage between OAO and MRBB approaches can be a consequence of the random subspace method used by the latter. On average, training a uMRBB classifier took 2.8 seconds whereas the training of a single classifier took 4.2 s, OAORUS and OAOROS was constructed in 4.7 s and 10.2 s. In terms of model size, a single model was on average 1.85 bigger than that constructed by uMRBB and models of OAORUS and OAOROS were 6 and 12 times bigger, respectively.

6. Conclusions

In this work, we discussed the class imbalance issue in sentiment classification. We carried out an analysis of data difficulty factors typically studied in the imbalance

learning community. The analysis of example safety [34] demonstrated that all the datasets under study, ranging from professionally written reviews to informal opinions expressed in tweets, pose a substantial challenge for standard learning algorithms. According to this analysis, all the datasets were identified to have rare and outlier characteristics, both of which are considered the most difficult in imbalanced learning [50, 61].

Moreover, this analysis was also performed to compare different feature representations in the context of imbalanced learning. Not surprisingly, the mixed representation consisting of many specialized features for sentiment detection such as negated n-grams or polarity values from handcrafted lexicons resulted to be most safe with respect to safe leveles [34, 50]. Furthermore, the continuous bag-of-words representation resulted to be safer than the classic bag-of-words representation. Nevertheless, all the representations happen to obtain low safety levels which encourages further research in the construction of better, more suitable features representations.

In a quite extensive experiment, eight popular preprocessing methods for imbalanced data were studied. None of the preprocessing methods achieved results significantly better then the baseline. We see two possible reasons which could explain such a result.

Firstly, the datasets typically studied in imbalanced learning have a very limited number of features. For instance, in one of the most extensive experimental studies of ensemble methods for imbalanced learning [16], among the 44 datasets under study none of them had more than 20 features. Even the most simple feature representation considered in this work has one and a half times more features. The high-dimensional imbalanced data are still not sufficiently well studied as, to the best of our knowledge, this is one of the first studies concerning imbalanced learning problems with hundreds of thousands of sparse features[8].

Secondly, the preprocessing methods for multi-class imbalanced data are also insufficiently studied. The methods used in this work, similarly to the majority of the methods proposed in literature [12], are extensions of binary preprocessing methods which can fail to exploit complex class interrelations in multi-class datasets as suggested in [34]. For instance, they fail to capture that a class can be a minority one with respect to majority classes but at the same time act as a majority one in the context of other minority classes.

On the other hand, the ensemble methods specifically designed for multi-class imbalanced data proved to be useful in our experiment. Those methods allowed for achieving significantly better results in terms of both G-mean and F-measure. Multi-class Roughly Balanced Bagging provided the best results for decision trees and k-nearest neighbors as component learning algorithms, whereas one-against-one decomposition with resampling worked best with multinomial logistic regression and naive Bayes. Additionally, the under-sampling version of Multi-class Roughly Balanced Bagging constructs the smallest models and within the shortest training time.

In conclusion, we think that at the intersection of sentiment analysis and imbal-

[8]Several works on high-dimensional imbalanced data were performed in the area of bioinformatics. However, the representative work [3] still considers datasets with a smaller number of dimensions than ours with mixed representation.

anced learning there is plenty of research problems which should be studied more deeply. In addition to those mentioned in the previous paragraphs, sentiment classification of the data from social media could be additionally treated as a stream classification task [7]. Taking the aspect of time and order fully into account could be profitable, though dealing with the class imbalance in non-stationary data streams is still an open research problem.

Acknowledgment

This work was supported by Institute of Computing Science Statutory Funds. The author would also like to express his gratitude to Dr. Dariusz Brzeziński for his invaluable help in the design of experiments and for proofreading the manuscript.

References

[1] Abbasi, A., France, S., Zhang, Z., Chen, H.: Selecting Attributes for Sentiment Classification Using Feature Relation Networks. IEEE Transactions on Knowledge and Data Engineering, 23 (3), 447-462 (2011).

[2] Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Proc. of the Int. Conference on Language Resources and Evaluation (2010).

[3] Blagus, R., Lusa, L.: SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics, 14 (1), 1471–2105 (2013).

[4] Blitzer, M. D., Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL-2007), 440-447 (2007).

[5] Błaszczyński, J., Stefanowski, J.: Neighbourhood sampling in bagging for imbalanced data. Neurocomputing, 150 A, 184–203 (2015).

[6] Błaszczyński, J., Stefanowski, J.: Local data characteristics in learning classifiers from imbalanced data. In Advances in Data Analysis with Computational Intelligence Methods, 51–85, Springer (2018).

[7] Brzezinski, D. and Stefanowski, J.: Stream Classification. Encyclopedia of Machine Learning and Data Mining, Springer (2017).

[8] Burns N., Bi Y., Wang H., Anderson T.: Sentiment Analysis of Customer Reviews: Balanced versus Unbalanced Datasets. In: König A., Dengel A., Hinkelmann K., Kise K., Howlett R.J., Jain L.C. (eds) Knowledge-Based and Intelligent Information and Engineering Systems, LNCS, 6881, 161-170 (2011).

[9] Chawla, N.: Data mining for imbalanced datasets: An overview. In Maimon O., Rokach L. (eds): The Data Mining and Knowledge Discovery Handbook, Springer, 853–867 (2005).

[10] Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. J. of Artificial Intelligence Research, 16, 341-378 (2002).

[11] Das, S. R., Chen, M. Y.: Yahoo! for Amazon: Sentiment extraction from small talk on the web. Management Science, 53(9), 1375-1388 (2007).

[12] Fernández A., García S., Galar M., Prati R., Krawczyk B., Herrera H.: Learning from Imbalanced Data Sets. Springer (2018).

[13] Fernández, A., Garcia, S., Herrera, F., Chawla, N.V.: SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61, 863–905 (2018).

[14] Fernandez, A., Lopez, V., Galar, M., Jesus M., Herrera, F.: Analysing the classification of imbalanced data sets with multiple classes, binarization techniques and ad-hoc approaches. Knowledge-Based Systems, 42, 97–110 (2013).

[15] Fernández-Navarro, F., Hervás-Martínez, C., Gutiérrez, P.A.: A dynamic oversampling procedure based on sensitivity for multi-class problems. Pattern Recognition, 44, 1821–1833 (2011).

[16] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 463-484 (2012).

[17] Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In Proc. of 12th Int. Workshop on the Web and Databases, 9, 1–6 (2009).

[18] Garcia, V., Sanchez, J.S., Mollineda, R.A.: 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, 4756, 397–406 (2007).

[19] Han, H., Wen-Yuan, W., Bing-Huan, M.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in intelligent computing, 878-887 (2005).

[20] He, H., Yang, B., Garcia, E.A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE Int. Joint Conference on Neural Networks, 1322-1328 (2008).

[21] He H., Garcia E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering, 21 (9), 1263–1284 (2009).

[22] He, H. and Ma, Y.: Imbalanced learning: foundations, algorithms, and applications, Wiley (2013).

[23] Hido, S., Kashima, H.: Roughly balanced bagging for imbalance data. Statistical Analysis and Data Mining, 2 (5-6), 412–426 (2009).

[24] Hu, M., Liu, B.: Mining and summarizing customer reviews. In Proc. of the 10th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, 168–177 (2004).

[25] Japkowicz, N., Stephen, S.: Class imbalance problem: a systematic study. Intelligent Data Analysis Journal, 6 (5), 429–450 (2002).

[26] Jo, T., Japkowicz, N.: Class Imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter, 6 (1), 40–49 (2004).

[27] Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 50, 723–762 (2014).

[28] Koppel, M, Schler, J.: The Importance of Neutral Examples for Learning Sentiment. Computational Intelligence, 22, 100–109 (2006).

[29] Krawczyk B., McInnes B.T., Cano A.: Sentiment Classification from Multi-class Imbalanced Twitter Data Using Binarization. In: Martínez de Pisón F., Urraca R., Quintián H., Corchado E. (eds) Hybrid Artificial Intelligent Systems, LNCS, 10334, 26–37 (2017).

[30] Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-side selection. In Proc. of the 14th Int. Conf. on Machine Learning ICML-97, 179-186 (1997).

[31] Kuncheva, L. I.: Combining Pattern Classifiers: Methods and Algorithms: Methods and Algorithms. Wiley (2004).

[32] Lango M., Brzeziński D., Firlik S., Stefanowski J.: Discovering Minority Subclusters and Local Difficulty Factors from Imbalanced Data. In Proc. of the 20th Int. Conference on Discovery Science (2017).

[33] Lango M., Brzeziński D., Stefanowski J.: PUT at SemEval-2016 Task 4: The ABC of Twitter Sentiment Analysis, In Proc. of the 10th Int. Workshop on Semantic Evaluation (2016).

[34] Lango, M., Napierala, K., Stefanowski, J.: Evaluating Difficulty of Multi-class Imbalanced Data. In Proc. of 23rd Int. Symposium on Methodologies for Intelligent Systems, 312–322 (2017).

[35] Lango M., Stefanowski J.: Multi-class and Feature Selection Extensions of Roughly Balanced Bagging for Imbalanced Data. Journal of Intelligent Information Systems (2018).

[36] Lemaître G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 18 (17), 1–5 (2017).

[37] Li, S., Ju, S., Zhou, G., Li, X.: Active learning for imbalanced sentiment classification. In Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 139-148 (2012).

[38] Li, S., Wang, Z., Zhou, G., Lee, S. Y. M.: Semi-supervised learning for imbalanced sentiment classification. In Proc. of Int. Joint Conference on Artificial Intelligenc, 22 (3), 1826–1831 (2011).

[39] Li, T., Zhang, Y., Sindhwani, V.: A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th Int. Joint Conference on Natural Language Processing of the AFNLP, 1, 244-252 (2009).

[40] Li, S., Zhou, G., Wang, Z., Lee, S. Y. M., Wang, R.: Imbalanced sentiment classification. In Proc. of the 20th ACM Int. Conference on Information and Knowledge Management, 2469-2472 (2011).

[41] Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, Morgan & Claypool (2012).

[42] Loper, E., Bird, S.: NLTK: The natural language toolkit. In Proc. of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, 1, 63–70 (2002).

[43] Mathioudakis, M., Koudas, N.: Twitter-monitor: Trend detection over the twitter stream. In Proc. of the 2010 ACM SIGMOD Int. Conference on Management of Data, 1155–1158 (2010).

[44] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In Proc. of Neural Information Systems Processing (2013).

[45] Mohammad, S., Turney, P.D.: Crowd-sourcing a word-emotion association lexicon. Computational Intelligence, 29 (3), 436–465 (2013).

[46] Mountassir, A., Benbrahim, H., Berrada, I.: An empirical study to address the problem of Unbalanced Data Sets in sentiment classification. IEEE Int. Conference on Systems, Man, and Cybernetics (SMC), 3298-3303 (2012).

[47] Nakov, P., Ritter, A., Rosenthal, S., Stoy-anov, V., Sebastiani, F.: SemEval-2016 task 4: Sentiment analysis in Twitter. In Proc. of the 10th Int. Workshop on Semantic Evaluation (2016).

[48] Napierala, K., Stefanowski, J.: The influence of minority class distribution on learning from imbalance data. In Proc. of the 7th Int. Conference on Hybrid Artificial Intelligent Systems, LNAI, 7209, 139–150 (2012).

[49] Napierala, K., Stefanowski, J.: BRACID: a comprehensive approach to learning rules from imbalanced data. Journal of Intelligent Information Systems, 39 (2), 335–373 (2012).

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

[51] Niklas, J., Weber, S.H., Müller, M.C., Gurevych, I.: Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In Proc. of the 1st Int. Workshop on Topic-sentiment analysis for mass opinion (2009).

[52] Ohana, B., Tierney, B., Delany, S. J.: Domain independent sentiment classification with many lexicons. In 4th Int. Symposium on Mining and Web at 25th Int. Conference on Advanced Information Networking and Applications (AINA), 632–637 (2011).

[53] Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis using subjectivity summarization based on minimum cuts. In: 42nd Annual Meeting on Association for Computational Linguistics, 271–278 (2004).

[54] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Conference on Empirical Methods in Natural Language Processing, 10, 79–86 (2002).

[55] Pedregosa et al.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830 (2011).

[56] Prati, R., Batista, G., Monard, M.: Class imbalance versus class overlapping: an analysis of a learning system behavior. In Proc. of 3rd Mexican Int. Conf. on Artificial Intelligence, 312–321 (2004).

[57] Remus, R.: Modeling and representing negation in data-driven machine learning-based sentiment analysis. In Proc. of 1st Int. Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM 2013), 22–33 (2013).

[58] Schütze, H., Manning, C.D.: Foundations of Statistical Natural Language Processing. MIT Press (1999).

[59] Song, K., Feng, S., Gao, W., Wang, D., Yu, G., Wong, K. F.: Personalized Sentiment Classification Based on Latent Individuality of Microblog Users. In Proc. of Int. Joint Conferences on Artificial Intelligence, 2277-2283 (2015).

[60] Stefanowski, J.: Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In Ramanna, L.C.J.S. and Howlett, R.J. (eds), Emerging Paradigms in Machine Learning, 277–306 (2013).

[61] Stefanowski, J.: Dealing with Data Difficulty Factors while Learning from Imbalanced Data. In S. Matwin and J. Mielniczuk (eds), Challenges in Computational Statistics and Data Mining, Studies in Computational Intelligence, 605, 333–363 (2016).

[62] Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In Song, I.-Y., Eder, J., Nguyen, T.M. (eds) Data Warehousing and Knowledge Discovery, LNCS, 5182, 283–292 (2008).

[63] Tomek, I.: Two modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, 6, 769-772 (2010).

[64] Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL-2002) (2002).

[65] Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Class Imbalance, Redux. In Proc. of IEEE 11th Int. Conference on Data Mining, 754-763 (2011).

[66] Wang, S., Yao, X.: Mutliclass imbalance problems: analysis and potential solutions. IEEE Trans. System Man Cybern., Part B. 42 (4), 1119–1130 (2012).

[67] Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. IEEE Symp. Comput. Intell. Data Mining, 324–331 (2009).

[68] Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In Proc. of the 16th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, 783–792 (2010).

[69] Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation, 39 (2-3), 165–210 (2005).

[70] Wilson, D.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetrics, 2 (3), 408-421 (1972).

[71] Wilson D.R., Martinez T.R.: Improved heterogeneous distance functions. J. Artificial Intelligence Research, 6, 1–34 (1997).

[72] Wojciechowski, S., Wilk, S., Stefanowski, J.: An algorithm for selective preprocessing of multi-class imbalanced data. In Proc. of Int. Conference on Computer Recognition Systems, CORES 2017, 238–247 (2017).

[73] Wojciechowski, S., Wilk, S.: Difficulty Factors and Preprocessing in Imbalanced Data Sets: An Experimental Study on Artificial Data, Foundations of Computing and Decision Sciences, 42(2), 149-176 (2017).

[74] Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., Wang, X.: Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification. Cogn Comput, 7, 226 (2015).

[75] Zhou, Z. H., Liu, X.Y.: On multi-class cost sensitive learning. Computational Intelligence, 26 (3), 232–257 (2010).

Received 18.01.2019, Accepted 24.02.2019