INVESTIGATION OF IMBALANCE PROBLEM EFFECTS ON TEXT CATEGORIZATION

DENGESİZLİK PROBLEMİNİN METİN SINIFLAMA ÜZERİNDEKİ ETKİLERİNİN ARAŞTIRILMASI

Behzad NADERALVOJOUD

ASSOC. PROF. DR. Ebru AKÇAPINAR SEZER
Supervisor

Submitted to Institute of Sciences of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

January 2015

This work named **"INVESTIGATION OF IMBALANCE PROBLEM EFFECTS ON TEXT CATEGORIZATION"** by **Behzad NADERALVOJOUD** has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING** by the below mentioned Examining Committee Members.

Asst. Prof. Dr. Ahmet Burak CAN
Head ....

Assoc. Prof. Dr. Ebru AKÇAPINAR SEZER
Supervisor ....

Asst. Prof. Dr. Erhan MENGÜŞOĞLU
Member ....

Dr. Fuat AKAL
Member ....

Dr. Murat HACIÖMEROĞLU
Member ....

This thesis has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING** by Board of Directors of the Institute for Graduate Studies in Science and Engineering.

Prof. Dr. Fatma SEVİN DÜZ
Director of the Institute of
Graduate School of Science and Engineering

ETHICS

In this thesis study, prepared in accordance with the spelling rules of Institute of Graduate School of Science and Engineering of Hacettepe University,

I declare that

- all the information and documents have been obtained in the base of the academic rules.
- all audio-visual and written information and results have been presented according to the rules of scientific ethics
- in case of using others works, related studies have been cited in accordance with the scientific standards
- all cited studies have been fully referenced
- I did not do any distortion in the data set
- and any part of this thesis has not been presented as another thesis study at this or any other university.

19/01/2015

BEHZAD NADERALVOJOUD

ABSTRACT

INVESTIGATION OF IMBALANCE PROBLEM EFFECTS ON TEXT CATEGORIZATION

Behzad NADERALVOJOUD
Master of Science,Computer Engineering Department
Supervisor: Assoc. Prof. Dr. Ebru AKÇAPINAR SEZER
January 2015, 105 pages

Text classification is a task of assigning a document into one or more predefined categories based on an inductive model. In general, machine learning algorithms assume that datasets consist of almost homogeneous class distribution. However, learning methods can be tended to the classification which has poorly performance over the minor categories while using imbalanced datasets. In multi-class classification, major categories correspond to the classes with the most number of documents and also minor ones correspond to the classes with the lowest number of documents. As a result, text classification is the process which can be highly affected from the class imbalance problem.

In this study, we tackle this problem using category based term weighting approach in combination with an adaptive framework and machine learning algorithms. This study first investigates two different types of feature selection metrics (one-sided and two-sided) as a global component of term weighting scheme (called as *tffs*) in scenarios where different complexities and imbalance ratios are available. *tfidf* as a traditional term weighting scheme is employed to evaluate the effects of *tffs* term weighting approach. In fact, the goal is to determine which kind of weighting schemes are appropriate for which machine learning algorithms on different imbalanced cases. Hence, four popular classification algorithms (SVM, kNN, MultiNB and C4.5) are used in the experiments. According to our achieved results, regardless of *tfidf*, term weighting methods based on one-sided feature selection metrics are more suitable approaches for SVM and kNN algorithms while two-sided based term weighting schemes are the best choice for MultiNB and C4.5 algorithms on the imbalanced texts. Moreover, *tfidf* weighting method can be more recommended for kNN algorithm in imbalanced text classification.

Furthermore, Two category based functions named as PNF and PNF^2 are proposed as a global component of term weighting scheme. To better evaluate the proposed approaches with the existing methods, an adaptive learning process is proposed. In fact, this algorithm learns a model which intensively depends on the term weighting schemes and can obviously show the performance of different weighting methods in classification of imbalanced texts. According to the experiments which were carried out on the two benchmarks (Reuters-21578 and WebKB), the proposed methods yield the best results.

Keywords: Text Classification, Term Weighting Approach, Class Imbalance Problem, Machine Learning.

ÖZET

DENGESİZLİK PROBLEMİNİN METİN SINIFLAMA ÜZERİNDEKİ ETKİLERİNİN ARAŞTIRILMASI

Behzad NADERALVOJOUD
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Danışman: Doç. Dr. Ebru AKÇAPINAR SEZER
Ocak 2015, 105 sayfa

Metin sınıflandırma, dökümanı önceden tanımlanmış bir veya daha fazla kategori içerisinden birine atama işlemidir. Genel olarak, makine öğrenmesi algoritmaları veri kümelerinin neredeyse homojen olduğunu varsaymaktadır. Bununla birlikte, öğrenme yöntemleri dengesiz veri kümelerini kullanarak küçük kategoriler üzerinde kötü performansa sahip sınıflandırma eğilimi gösterebilmektedir. Çoklu sınıflandırmada, ana kategoriler çok sayıda dökümanı içeren sınıflara karşılık gelirken, küçük kategoriler ise dökümanı sayısı küçük olan sınıflara karşılık gelmektedir. Bunun sonucu olarak, metin sınıflandırma dengesiz sınıf probleminden oldukça etkilenen bir süreçtir.

Bu çalışma içerisinde, makine öğrenmesi algoritmaları ve uyarlanabilir çerçeveyle birlikte kategori tabanlı vadeli ağırlıklandırma yaklaşımı kullanılarak bu sorun ele alınmaktadır. İlk olarak bu çalışmada, farklı karmaşıklık ve dengesizlik oranları olan senaryolar içerisinde $tffs$ olarak adlandırılan özellik seçim metriklerinin iki farklı türü incelenmektedir. Geleneksel terim ağırlıklandırma olarak $tfidf$, $tffs$ terim ağırlıklandırma yaklaşımının etkilerini değerlendirmek için kullanılır. Aslında amaç farklı dengesizlik durumlarında makine öğrenmesi algoritmaları için uygun olan ağırlık şemalarının belirlenmesidir. Bu nedenden dolayı, deneylerde SVM, kNN, MultiNB ve C4.5 gibi popüler olan sınıflandırma algoritmaları kullanılmıştır. $Tfidf$ dikkate alınmadan, elde edilen sonuçlara göre; tek taraflı özellik seçim ölçümlerine dayalı terim ağırlıklandırma yöntemleri SVM ve kNN algoritmaları için daha uygun yaklaşımlar iken dengesiz metinler üzerinde iki taraflı terim ağırlıklandırma şemaları için ise MultiNB ve C4.5 algoritmaları en iyi seçimdir. Bununla birlikte, $tfidf$ ağırlıklandırma yöntemi kNN algoritması için dengesiz metin sınıflandırma üzerinde daha fazla önerilebilir.

Çalışma içerisinde ayrıca PNF ve PNF^2 olarak adlandırılan fonksiyon tabanlı iki kategori, terim ağırlık şemasının global bileşeni olarak önerilmektedir. Mevcut yöntemlerle birlikte önerilen yaklaşımların değerlendirilmesi için, uyarlanabilir öğren-me süreci önerilmektedir. Aslında bu algoritma terim ağırlık şemalarına bağlı bir model öğrenir ve dengesiz metinlerin sınıflandırmasında farklı ağırlıklandırma yöntemlerinin performansını açıkça göstermektedir. Reuters-21578 ve WebKB üzerinde yapılan deneylere göre, önerilen yöntemler iyi sonuçlar vermektedir.

Anahtar Kelimeler: Metin Sınıflandırma, Terim Ağırlık Yaklaşımı, Sınıf Dengesizlik Sorunu, Makine Öğrenme.

ACKNOWLEDGEMENTS

This thesis could not have been finished without the help and contributions of my advisor Assoc. Prof. Dr. Ebru AKÇAPINAR SEZER. First and foremost I would like to thank her for her advice, encouragement, guidance and endless supports during my MSc. education. Besides my advisor, I would like to thank my thesis committee for revising it and providing useful help and feedbacks.

My deepest gratitude goes to my mom and dad for all of their sage advice and especially my lovely wife whose supports and what she have been doing for me cannot be described by any words. I could not have achieved this success without her unlimited support, patience and encouragements.

I would also like to thank the members of Multimedia Information Retrieval (MIR) Laboratory. Particularly, I would like to acknowledge Ahmet Selman Bozkır, Alaettin Uçan, Rahem Abri for useful suggestions that helped me advance my research area better.

I must also thank many others who have helped form the ideas presented here and write the content of this thesis but whose names are too long to fit on a single page.

Finally, thank God for all that He has done and all that He will do and for allowing me to accomplish this thesis right here.

CONTENTS

Page

ABSTRACT i

ÖZET iii

ACKNOWLEDGEMENTS v

CONTENTS vi

FIGURES viii

TABLES x

1. INTRODUCTION 1

1.1. Overview 1

1.2. Class Imbalance Problem 2

1.3. Motivation 3

1.4. Major Contributions of the Thesis 5

1.5. Scope and Organization of the Thesis 7

2. RELATED WORKS 8

2.1. Overview 8

2.2. Data Level Approaches 8

2.3. Algorithmic Level Approaches 13

3. CLASSIFICATION 17

3.1. Overview 17

3.2. Basic Concept 18

3.3. Underfitting and Overfitting 19

3.4. Model Complexity and Generalization Error 20

3.5. Classification Model Evaluation 21

4. TERM WEIGHTING AND FEATURE SELECTION APPROACHES 27

4.1. Term Weighting Scheme 28

4.2. Supervised Term Weighting Scheme 29

4.3. Feature Selection Scheme 30

4.4. Investigation of Term Weighting Schemes in Classification of Imbalanced Texts .. 39

5. IMBALANCED TEXT CATEGORIZATION BASED ON POSITIVE AND NEGATIVE TERM WEIGHTING APPROACH USING ADAPTIVE FRAMEWORK .. 51

5.1. Proposed Positive and Negative Based Term Weighting Scheme 51

5.2. Empirical Observation of Term Weighting and Feature Selection Approaches 53

5.3. The Properties of Term Weighting Approach Associated with Imbalance Problem 54

5.4. Adaptive Framework 59

5.5. Experiments 60

6. CONCLUSION 71

A APPENDIX, CHI-SQUARE TABLE 73

B APPENDIX, Z-TABLE 74

C APPENDIX, PUBLISHED COPY OF PROCEEDING FROM THE THESIS 76

REFERENCES 85

FIGURES

Page

1.1. The class distribution of training data in two different complexities 3

1.2. Learning process 4

3.1. Supervised classification process 19

3.2. Training and test error rates while overfitting occurs 20

4.1. The variety of impurity level for two categories. The set indicated on the left side is a good training set for learning, while the right set cannot be known as a good training set 36

4.2. The term weighting process for binary text classification 43

4.3. The F_1-values of five weighting schemes tested over R8 dataset with balanced setting using four different classifies 47

4.4. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with balanced setting using four different classifies 47

4.5. The F_1-values of five weighting schemes tested over R8 dataset with imbalanced setting using four different classifies 48

4.6. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with imbalanced setting using four different classifies 49

5.1. The distributions of documents for a given term in two different situations for binary classification 56

5.2. Box-whisker-plot description 68

5.3. Box-whisker-plot types 68

5.4. Box-whisker-plot for F-measure values gained by the term weighting methods on the Reuters dataset 69

5.5. Box-whisker-plot for F-measure values gained by the term weighting methods on the WebKB dataset 70

TABLES

3.1. Confusion matrix for binary classification task TP: true positive, TN: true negative, FP: false positive, FN: false negative 22

3.2. The evaluation metrics by confusion matrix values 23

4.1. All metrics used in the experiments as the global component of term weighting scheme for binary classification 30

4.2. General notation for a binary contingency table 32

4.3. Chi-square distribution table 33

4.4. Binary contingency table for occurrence of term t_i in category c_j 34

4.5. Properties of Ruters-21578 dataset (R8) with imbalance ratio of each class while considering as target class in training set. $Imbalance\ ratio = \frac{N_{C_j}}{N-N_{C_j}}$ .. 45

4.6. Properties of 20 Newsgroups dataset with imbalance ratio of each class while considering as target class in training set. $Imbalance\ ratio = \frac{N_{C_j}}{N-N_{C_j}}$ 46

4.7. ANOVA test for F_1-values obtained from 5 weighting methods for each algorithm for imbalanced cases 50

5.1. Fundamental information elements which are used in feature selection functions 52

5.2. The characteristics of top 4 terms selected by different manners for *grain* category in Reuters-21578 dataset 54

5.3. The most negative, most positive and neutral cases based on document frequency elements for *grain* class of Reuters-21578 dataset 55

5.4. The more relevant terms to *grain* class in terms of semantic in conjunction with their evaluation elements 57

5.5. The less relevant terms to *grain* class in terms of semantic in conjunction with their evaluation elements 59

5.6. Properties of Ruters-21578 dataset (R8) with its class distributions 61

5.7. Properties of WebKB dataset with its class distributions 61

5.8. The F-measure values of different term weighting schemes for Reuters-21578 dataset 63
5.9. The F-measure values of different term weighting schemes for WebKB dataset 64
5.10. The macro average of precision and recall for different term weighting schemes 65
5.11. ANOVA test on the F-measure values obtained from the term weighting methods for Reuters-21578 dataset with P-Value = 0.0000 and 95.0 percent LSD intervals 66
5.12. ANOVA test on the F-measure values obtained from the term weighting methods for WebKB dataset with P-Value = 0.0028 and 95.0 percent LSD intervals 66
5.13. Kruskal-Wallis Test for F-measure values gained by 7 term weighting methods on Reuters dataset. Test statistic = 23.6714 P-Value = 0.0006 69
5.14. Kruskal-Wallis Test for F-measure values gained by 7 term weighting methods on the WebKB dataset. Test statistic = 19.1133 P-Value = 0.0039........ 70

1. INTRODUCTION

1.1. Overview

In many diverse applications of the virtual environments large amount of data is produced with imbalanced distribution. Virtual bank systems, social websites, medical diagnostic systems, satellite environmental monitoring systems, virtual shopping systems and other real applications daily generate huge number of data. On the other hand, the rapid improvement in internet and data transfer rates lead to increase the size of data in the area of technology. In order to analyze data from different perspectives and transform it into useful information, data mining methods are applied on the data. Technically, data mining is used to extract patterns or correlations among the fields of data and consequently convert data into information. One of the important tasks of data mining is classification. Classification is known as a predictive task which is used to assign objects to one or more predefined categories. In fact, the objective is to predict the value of target attribute based on the values of other attributes. For example, predicting whether an email will be known as a spam is a classification task because the target attribute has two possible values. The aim of this task is to learn a model that minimize the error between the predicted and actual values of the target attribute.

In the recent decade, classification was a problem that embraced many real applications, e.g. cancer detection based on MRI scans or network intrusion detection. However, generating skewed (imbalanced) data from real applications may aggravate this problem . Moreover, classification of data can go to the critical point when the unbounded size of data is available. Actually, this problem was emerged when machine learning techniques were used to classify data. On the other hand, these techniques matured from academic discipline to functional technology used in industry, business and scientific applications.

The imbalanced data sets affected the performance of machine learning methods and caused suboptimal classification results. The importance of this issue may appear as more and more in medical diagnosis, to detect if a patient has a specific disease or not. The objective is presence of the disease. In such case, the relative proportion of different types of errors is taken into consideration. It means that a false positive (detecting a disease when it is not present) is observed differently from a false negative (not detecting a disease when it is present). Similar situations can be observed in the other domains such as detection of oil spills upon satellite images, risk management, geographical monitoring, text classification, Information Retrieval, and there are many others.

1.2. Class Imbalance Problem

In machine learning, text classification is a task of assigning unlabelled documents into thematic predefined categories based on an inductive model. In text classification, class imbalance problem typically occurs where the number of documents of some classes are more than the others. In the imbalanced datasets, classes containing more number of instances are known as major classes while the ones having relatively less number of instances are called as minor classes. At this point, most of standard classifiers tend towards major classes and consequently show poorly performance on the minor classes. It means that they may classify everything as major class and ignore the minor ones. This situation can be seen in many different domains such as credit card fraud detection, network intrusions detection, medical monitoring and more much [1–6]. The domain of information retrieval (IR) is so good to observe effects of class imbalance problem because making a decision about a page or document which should belong to the result set of a search or not can be assumed as binary classification. In other words, the aim of classification in IR is to find more relevant results (minor class) to the user's query while the huge number of irrelevant documents (major class) exists. In some domains, imbalanced distribution of classes is intrinsic to the domain. For example, there are typically very few relevant pages while compared to the large number of irrelevant pages against the user's query. On the other hand, this problem may also occur in the domains that do not have intrinsic imbalance property, because there may be a limitation in collection of data due to privacy, economic or many other reasons [7]. Thus, use of imbalanced datasets becomes inevitable for many application areas, if the problem is classification.

There are other domain characteristics that aggravate the problem such as (1) class complexity (2) size of training set and (3) subclusters [8]. In typical binary classification, where there are 2 classes, class complexity refers to scattering of samples in the space. If two classes are linearly separable, least level of complexity can be obtained. At this point, it is possible that each class consists of samples which belong to one object and involves uniform scattering (as shown in Figure 1.1.a). By increasing samples in the negative class, two situations can occur; 1) increment in imbalance ratio, 2) increment in complexity level. In most cases, positive class belongs to one object while the negative class may belong to more than one. For example, when a user searches a subject in the internet, all of documents that do not belong to the subject are considered as irrelevant (or negative) documents. So it may not have uniform scattering since it involves a lot of documents with different subjects (as shown in Figure 1.1.b). Thus, increment in the number of negative class samples may lead to growth of class complexity. In this case, the positive class can be formed as a cluster while the negative class cannot. Therefore, raising the degree of imbalance by incrementing the negative samples causes aggravation of class distribution and growing the number of subclusters. On the other hand, in order to generate a classification model with low generalization error, error on the previously unseen samples, for positive (minor) class, existence of adequate number of samples in the training set is crucial. Therefore, in the datasets which have insufficient number of positive samples, classification is biased towards major class and ignored the minor one. Because standard classification algorithms aim to build models which only cover the global quantity. In fact, when only small number of positive samples are available, a classier generates hypotheses which are less general and may lead to the overfitting drawback. Thus, generated model cannot show reasonable performance on the positive unseen samples.

The mentioned characteristics of imbalanced data will be discussed in chapter 4., and classification task is carried out when training data with different imbalance characteristics are available.

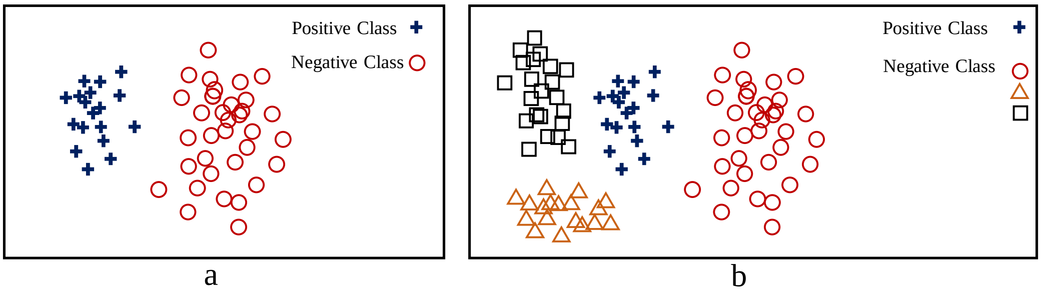


Figure 1.1. The class distribution of training data in two different complexities

1.3. Motivation

In machine learning, text classification model is typically generated through the three transformation steps, from data preprocessing to post processing as shown in Figure 1.2.

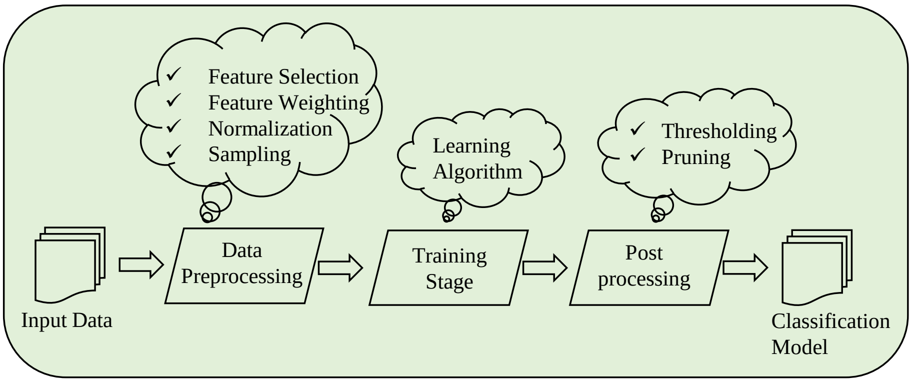


Figure 1.2. Learning process

The aim of *preprocessing step* is to transform the textual input documents into an appropriate format for any learning algorithm. The steps contained in preprocessing include selecting features that are more useful to classification task, weighting features to distinguish documents better, normalizing data to provide a limited range of values within a norm, sampling data either to remove the noise or selection of instances that are more relevant to classification.

As mentioned previously, because of the advances in data collection and existing privacy limitation, imbalanced data sets with huge sizes are becoming common. Such these data impose a huge constraint on the learning algorithms in which they cannot achieve reasonable results. Hence, preprocessing is known as an inevitable stage in the learning process and assists to generate an efficient classification model.

Machine learning algorithms may not always provide expectable results because of occurring model-overfitting (see chapter 3.) over the training process. In other words, the obtained model just fits the training data and cannot perform well on the data that it has never seen before. On the basis of the type of learning algorithms, the related *post processing* is employed on the model. In decision tree algorithm after building the decision tree, a tree-pruning step can be performed to reduce the size of the tree. Because a decision tree with too large size is susceptible to overfitting. In SVM classifier, finding a decision threshold can be useful to balance the performance of classification.

Classification by machine learning algorithms is usually performed based on a fundamental assumption that the distribution of classes should be close to each other. In other words, there should be as many examples belonging to major classes as examples belonging to minor ones [7, 8]. As mentioned before, this fundamental requirement cannot be always met and standard applications of machine learning algorithms may not provide satisfactory results for such cases. Because, the aim of these classifiers is to generate a model that best fits the training data with minimum error rate. Furthermore, they consider the global quantities in generating the model.

One of the effective approaches that was proposed to resolve this problem and also used in information retrieval and text mining, is *instance weighting strategy* via *tfidf* term weighting method [9]. *Tfidf* weighting is used to express how much a term can be important while documents are represented in the vector space model (VSM). In the text classification, VSM is used to represent documents in the form of term vectors. In fact, *tfidf* weighting by multiplying term frequency (*tf*) as a local weight by inverse document frequency (*idf*) as a global weight can show the significance of a term in a specific document. On the other hand, *idf* value of any term is the same for all classes which include this term, because this type of term weighting does not consider category membership in documents and takes into account the whole collection. Thus, category based term weighting schemes were proposed for text classification task [10–13]. This approach has various influences on the learning process of different algorithms. Actually, some algorithms are more affected by these weighting methods and others less (see chapter 4.). For example, standard SVM often learns the best decision boundary and applying term weighting strategy may not make a significant progress on the performance of classification [14]. Moreover, the performance of a classifier may be improved by using a term weighting method, whereas the same weighting method may not have more impact on the performance of another classifier. For example, over imbalanced datasets, $tf.or$ or $tf.rf$ weighting methods cannot improve the classification performance of C4.5 or multinomial naive Bayesian classifiers as much as they can grow the performance of k-NN and SVM (see chapter 4.). As the techniques employed in the learning process of algorithms are different, each weighing method may achieve different results.

1.4. Major Contributions of the Thesis

In this study, we focus on the category based term weighting approach since this approach has more potential to solve the class imbalance problem according to us. Two term weighting schemes based on two probabilities of relevant documents frequency are proposed. These weighting schemes present the relevancy power of terms with respect to categories. In this approach, we combine the idea existed in the feature selection process with the traditional term weighting approach for multi-class classification task. This is the fact, there is a difference between feature selection and term weighting approaches and their privileges. In other words, the common feature selection functions may not likely assign appropriate weights to terms, though they are suitable for feature selection purposes. This issue will be further discussed in chapter 5. As a result, the directly use of feature selection metric values may not always achieve expected results. This observation can be seen in the study which was accomplished by Debole and Sebastiani in [10]. This situation encouraged us to include more adaptive idea for using feature selection approach into the term weighting scheme. This idea will be discussed in chapter 5., as positive and negative based term weighting scheme.

In this approach, we tackle the class imbalance problem using a probability based weighting scheme for better multi-class classification task. Actually, two category based functions named as PNF and PNF^2 are proposed as a global component of term weighting scheme. These functions are based on two probabilities of relevant documents distribution. PNF^2 is designed as a symmetric function in which it assigns a positive or negative weight to terms. In this way, it can indicate either the type of term relevancy or the strength of relevancy (or not relevancy) with respect to a specific category. Conversely, PNF is known as asymmetric version of PNF^2 which can determine the power of relevancy. In fact, we can distinguish documents better either in minor or major categories by replacing idf with the proposed category based functions. In the experiments, we compare the proposed weighting scheme with five methods employed in [11] and demonstrate its superiority to others.

Aa another contribution, a framework is proposed to better evaluate the quality of different term weighting schemes in classification of imbalanced texts. In fact, the objective is to find an independent way to investigate the strength of term weighing methods in discrimination of documents. We estimate that the best weighting method identified by this framework likely present more suitable classification results than the others in combination with standard machine learning algorithms. In this framework, a classification model is built by using a statistical approach which is sharply affected by the weighting methods. Thus, the strength of weighting methods obviously appear over the classification task. This approach will be further discussed as adaptive framework to evaluate the weighting methods in chapter 5. It means that it provides an adaptive learning process to reflect the influence of various weighting methods on classification of imbalanced texts.

1.5. Scope and Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2. reviews the strategies proposed to resolve the class imbalanced problem in literature. Basic concepts of classification is presented in Chapter 3. Chapter 4. introduces the traditional term weighting approach and also describes the feature selection metrics which are used in the term weighting scheme. After investigating the traditional and supervised term weighting schemes on the classification of imbalanced texts, in Chapter 4., we propose a probability based term weighting scheme by relying on the positive and negative features, in chapter 5. The adaptive framework and experimental results are also proposed in the continuation of this chapter. Finally, in chapter 6., we conclude the thesis and present a brief view of our future research.

2. RELATED WORKS

2.1. Overview

Class imbalance problem (or imbalanced learning problem) is one of the challenging problems for machine learning algorithms. In order to handle this problem, a variety of strategies were proposed both at the data and algorithmic levels [7, 15]. While data level strategy addresses the preprocessing stage of learning process, the algorithmic level takes into account training and post processing stages (see Fig. 1.2.). As mentioned in previous chapter, both preprocessing and post processing stages of leaning process have a major influence on the training model. Therefore, the most number of strategies focus on the methods associated with these stages and try to integrate them for handling the class imbalance problem. At data level, the proposed approaches include the different forms of resampling methods [7, 8, 16], feature selection approach [17, 18] and term weighting approach [10–12, 19]. At algorithmic level, the employed strategies include recognition and cost-sensitive based learning methods [15, 20–22], determining the decision threshold [23–25] and kernel-based methods [26, 27], adjusting the probabilistic estimate at the information gain and Bayesian based methods such as decision tree and naive Bayes respectively [28, 29], etc.

2.2. Data Level Approaches

2.2.1. Resampling Approach

Resampling approach is known as a significant state-of-the-art solutions in the class imbalance problem context. In general, resampling techniques are realized by two methods: over-sampling and under-sampling. The over-sampling approach consists of increasing the minor class documents, at random or heuristic, until it contains as many samples as the major one. Conversely, under-sampling method moderates the majority class by eliminating its documents until it reaches the size of the minor class.

Random under-sampling is known as the simplest method to resampling. In this approach the documents belong to majority class is randomly eliminated form the training data. Thus, it can significantly reduce the training time as well as consumed memory space where high dimensional input data is available. As there is no control on eliminating documents, valuable information to build an accurate decision boundary may be thrown away from the majority class. Despite this, empirical studies [8, 16, 30] indicated that it can provide favorable results for the imbalanced data sets. Japkowicz and Stephen in [8] discussed the effect of random over-sampling (ROS) and random under-sampling (RUS) approaches on the imbalanced datasets. They indicated that both the resampling methods were effective in dealing with the imbalance problem, though the under-sampling approach performs better than the over-sampling in the large domains.

Under-sampling method is also employed by selecting a subset of negative samples (samples belong to majority class) using sophisticated methods. Four different methods were proposed in [30] to choose the negative training samples. These metrics were compared with random under-sampling method. Empirical results indicated that only one of the four proposed methods can perform better than random under-sampling. In the other similar study, [16] proposed two under-sampling methods based on the work in [30] for text classification. It performed more experiments with different level of resampling and stated that random under-sampling often outperforms sophisticated under-sampling techniques. In the both studies [30] and [16] the proposed methods for under-sampling were based on the distance between the majority and minority class samples. For example, one of the methods belonging to [16] (called as 'Distant2') selects the majority class samples whose average distance to the three farthest samples of minority class is largest and 'NearMiss1' method which belong to [30] selects the members from majority class whose average distance to the three closest minority class samples is smallest.

Alternatively, cluster-based under-sampling approach was proposed in [31] to select the representative samples from majority class to improve the classification performance for minority class. This approach was compared with other under-sampling methods and proved not only can achieve high stability and classification accuracy on predicting the minority class samples but also can perform faster than other methods.

In some state-of-the-art solutions, feature selection approach was employed to select majority class samples which are more representative with respect to the target class. A good way of thinking about this is to ask which negative samples are less representative and should be eliminated from training set. In the imbalanced data sets which negative documents outnumber the positive ones, the negative documents (i.e. the support vectors) play critical role to define the hyperplane in support vector machines [15, 25]. In fact, the negative documents which are close to the positive ones are used to define the decision boundary and others which are far away from positive documents cannot have any useful contribution. These negative documents are known as less representative samples in building the classification model.

In [32] a generic algorithm known as FISA was proposed to select a subset of negative training documents for SVM classifier where the negative documents significantly outnumber the positive ones. It claimed the proposed instance selection approach which was inspired by feature selection methods, was useful for efficient text classification. On the other hand, in the under-sampling method, eliminating the negative samples may lead to overfitting drawback. [25] indicated that the common strategies of under-sampling cannot be a best choice for SVM. In SVM classifier, the support vectors of the negative class are more than the positive one where an imbalanced training data is available. In this case, selecting appropriate negative samples (i.e. negative support vectors) will be important to define the direction of the decision boundary. In this approach, considering the negative samples which are close to positive ones is more critical since overlapping often occurs in this region.

Over-sampling technique is realized by duplicating the positive samples (samples belong to minority class) or generating new ones. Unlike the under-sampling, in this approach no information is lost from the original training data, since all members of majority and minority classes are preserved. However, duplicating positive samples lead to increase in the size of training set. Therefore, we face with a high training time as well as high required memory space for holding training data. By regarding the time and memory complexity, under-sampling performs better than over-sampling in high dimensional data sets. But if the classification performance is considered, over-sampling may be performed better than under-sampling. A variety of empirical studies were carried out with regard to which resampling method is the best in terms of classification performance [7, 8, 16, 25, 33]. Since different data sets are combined with different classification algorithms, achieved results are more likely inconsistent with each other.

In [16], the several resampling methods were investigated in the realm of imbalanced text classification. It presented several sophisticated over-sampling techniques and compared them with under-sampling methods. According to its experiments which were carried out in combination with different classifiers (e.g. SVM and k-NN), over-sampling techniques outperformed the under-sampling methods. The 'generating oversampling' method which was proposed in [16], achieved best results in the most of test cases. This method which was inspired by multinomial naive Bayes classifier, generates completely new documents based on the probability that a word will appear in a minority class document. In this approach, new positive samples are generated instead of duplicating them.

Similarly, [33] combined the over-sampling and under-sampling techniques and stated that it can yield better performance than only under-sampling. Moreover, It provided a synthetic technique for over sampling the minority class named as SMOTE. This approach generates any random point from the hyperplane or hypercube between two neighbor positive samples.

In the recent study [34], a novel over-sampling method was proposed based on document content to deal with the class imbalance problem. In this approach, an HMM which is a document generator, produces synthetic instances based on what it was trained with the corpus. According to its finding, the proposed method presented the greater performance than the SMOTE in the most of experiments.

2.2.2. Term Weighting and Feature Selection Approaches

Term weighting and feature selection approaches will explained completely in chapter 4. In this subsection we briefly review the state-of-the-art studies which contain major contributions for class imbalance problem.

The term weighting approach is a strategy which is used to improve the efficiency of text classification by assigning appropriate weights to terms. $Tfidf$ as a traditional term weighting scheme provided an influential solution for classification of imbalanced texts in common studies [17, 18]. Debole and Sebastiani in [10] proposed a number of supervised variant of $tfidf$ weighting by replacing idf with feature selection metrics and provided a category based weighting scheme for classification task. They demonstrated that supervised weighting can provide an effective solution for classification of imbalanced texts.

In the other study [11], the supervised term weighting, $tf.rf$, was proposed based on distribution of relevant documents. The rf metric indicates the relevance level of a term associated with a category. In fact, for a given term, it gained the discriminating power of the term by only imposing the number of relevant documents which contain this term. It evaluated $tf.rf$ weighting scheme using SVM and kNN algorithms over different corpus and showed it consistently preforms well in comparison with the other weighting methods (e.g. $tf.or$ and $tf.idf$). On the other hand, [12] introduced a probability based term weighting scheme which can better distinguish documents in minor categories. Its proposed method provided an effective solution to increase the performance of classification on the imbalanced texts.

While many experiments have been conducted by using document indexing based term weighting, [35] proposed a class-indexing based term wighting to improve the automatic text classification in different circumstances. This approach addressed the inverse class frequency (ICF) and inverse class space density frequency ($ICS_\sigma F$) in the term weighting scheme. It first incorporated ICF into indexing based term weighting scheme i.e. $tf.idf$ and presented a class-indexing based term weighting scheme as $tf.idf.icf$. Subsequently, ICf function was revised and replaced with $ICS_\sigma F$ which measures the class density of a certain term. This new function in associated with $tf.idf$ generated a new weighting method that provides a positive discrimination on both frequent and infrequent terms. These two approaches were compared with traditional weighting methods on either balanced or imbalanced datasets and achieved a good performance in combination with SVM classifier.

Alternatively, [17] addressed the feature selection process for solving the class imbalance problem and took into consideration the abilities and characteristics of various metrics for feature selection. It asserted that the negative features (the features with respect to majority class) makes a positive influence on the classification performance. Because in the imbalanced circumstance, symmetric feature selection metrics (e.g. information gain or chi square) tend to select more positive features. In other words, the number of negative features is noticeably reduced in the selected features set. This fact leads to reach a lower classification performance than the situation which there are either positive or negative features simultaneously. Therefore, a novel feature selection metric was proposed in [17] that was implicitly selecting an optimal combination of positive and negative features in imbalanced circumstances. This was the while a feature selection framework had been introduced in [36] to create an optimal combination of positive and negative features.

Feature selection approach was differently addressed in [37] for high-dimensional imbalanced data. In one hand the samples of minor classes are essential in the performance of learning process associated with minor classes. On the other hand samples belonging to majority classes have a dominant influence on the feature selection methods. Regarding these observations, a new feature selection framework was proposed in [37] based on class decomposition. It means that major classes are decomposed into pseudo-subclasses with relatively balanced sizes and then feature evaluation is applied to decomposed data. Experimental results demonstrated that this approach outperforms the traditional feature selection methods in terms of F-measure, ROC and AUC.

In a recent study [18] the feature selection policies were explored in text categorization by using SVM classifier. For imbalanced circumstances, it also proposed a novel feature selection framework called as AKS to select terms for each class in which the number of selected terms depends on the size of classes. Its experimental observations proved that this framework can make a significant progress on the performance of imbalanced text classification.

2.3. Algorithmic Level Approaches

2.3.1. One-Class Approach

Algorithmic approach takes into account optimizing the classification performance under a lack of homogeneous class distribution. One-class approach which is known as recognition based learning, try to recognize only the samples that belong to the target class. In fact, the classification model is generated based on the samples of the target class. Because when a data set is extremely imbalanced, the discriminative (two-class) based classifiers such as decision tree and neural networks biased towards the overfitting. When the major and minor classes are separately considered, this approach can be useful in solving the class imbalance problem. Moreover, according to [38] under data sets with high feature space dimensionality, one class approach performs better than discriminative approach.

One of the related works in this area include the one-class SVMs [38–42]. Manevitz et al. in [39], implemented a variety of SVM, appropriate for one-class classification and compared it with the performance of one-class versions of Rocchio, nearest neighbor, naive Bayes, and neural network algorithms. Their SVM approach consistently performed better than the other methods except the neural network, where it provided a comparable results. In other related study [43], a simple feed-forward neural network was proposed to efficient classification and retrieval of ’interests’ on the internet when only positive information is available.

2.3.2. Cost-Sensitive Learning

As another alternative to resolve the class imbalance problem, cost-sensitive learning (CSL) was introduced in literature [44, 45]. This approach takes into account the costs associated with misclassified samples where there are a variety of costs for different misclassification cases (false positives and false negatives). Actually, cost-sensitive learning methods by using different cost matrices that assesses the costs of misclassified samples, try to deal with the class imbalance problem. At this point, the goal of cost-sensitive classifiers is to minimize the cost of misclassification. In last decade, fundamental theories of cost-sensitive learning were applied to imbalanced data problem and proved this approach can be superior to other methods such as resampling [21, 46].

According to [20], imbalanced distribution of classes have a major impact on the performance of cost-sensitive classifiers. [20] presented a empirical study on the influence of class imbalance on cost-sensitive learning and stated that when the misclassification costs are almost equal, cost-sensitive classifiers generally favor a natural class distribution. Conversely, when misclassification costs are seriously different, a balanced class distribution is more favorable. To rebalance the class distribution in cost-sensitive learning, a popular approach is to use various weights associated with training samples of different classes in proportion of their corresponding misclassification costs [20, 47–49]. This approach is known as instance weighting strategy in the state-of-the-art. In fact, this approach by assigning different error-classification costs to negative and positive instances, try to deal with the class imbalance problem.

As another approach, resampling techniques were combined with cost-sensitive learning to reduce the total misclassification costs of the model [25, 50]. In [21], two empirical methods were proposed to deal with the class imbalance problem by using both resampling and cost sensitive learning methods. While its first method was combining several sampling techniques with CSL, the second method was locally optimizing the cost ratio to apply it to learning process. In other study [50], a series of modifications were designed to support vector machines. It combined cost-sensitive learning and resampling techniques (over and under sampling) in SVM training process and proposed four SVM modeling techniques. These four algorithms were extensively compared with state-of-the-art approaches on highly imbalanced data sets in terms of G-mean, AUC-ROC, F-measure, and AUC-PR metrics.

In another similar work, [14] performed a comparative study on the effectiveness of resampling and instance weighting strategies in the classification of imbalanced texts using SVM classifier. It evaluated 10 different methods including from the both of strategies on various data sets. According to its experimental results, standard SVM often learns the best decision boundary on the less imbalanced circumstance. For high ratio of imbalance, finding appropriate threshold can be more critical than applying any resampling or instance weighting strategies.

2.3.3. Kernel-Based Methods

Although cost-sensitive learning methods provided effective solutions to handle the imbalanced data problem, numerous other approaches have been followed in the community. One of the most important approaches that can provide state-of-the-art techniques for many real applications, is known as kernel-based learning methods. The basis of this approach is relying on theories of statistical learning and Vapnik-Chervonenkis (VC) dimensions [51]. Support vector machines (SVMs) as a representative kernel-based learning model, can perform relatively well on the imbalanced date sets [8]. A training process in SVMs is carried out for binary classification by using particular samples of two classes near the decision hyperplane (support vectors). The objective is to maximize the separation margin between the support vectors and hyperplane as well as minimizing the total classification error.

As the decision boundary is defined based on support vectors, it is expected that SVM has less suffering from imbalanced class distribution [14]. However, in the imbalanced data sets SVM is biased towards the majority class since it tries to minimize the total error rate. On the other hand, the hypothesized hyperplane may be affected by the negative support vectors more than positive ones since there is an imbalanced ratio between the support vectors. It means that, the support vectors representing the minority class may be far away from the 'ideal' hyperplane, and consequently, will contribute less in building the final model [15]. Therefore, it can be concluded that SVMs can suffer from high incidences of false negative errors in the imbalanced data sets [27].

The same characteristics can also occur in a linearly non-separable space. At this point, a kernel function is employed to transform the linearly non-separable space into a higher dimensional space which may be separable. In either cases, the hypothesized hyperplane tends towards the majority class and consequently achieves the more false negatives. To handle this problem, a variety of kernel based approaches were proposed in the class imbalance community [26, 27, 52]. In fact the prior known information can be incorporated with the appropriate kernel function [53]. In [27] a kernel-boundary-alignment algorithm was introduced to augment SVMs to improve classification accuracy. It used imbalanced data distribution as a prior information and applied it to adjust the class boundary by modifying the kernel matrix. A kernel classifier construction algorithm was proposed in [52] using orthogonal forward selection (OFS) to optimize the model generalization for imbalanced binary classification. In some other state-of-the-art solutions, resampling and weighting techniques were inserted into the SVM training process [14, 25, 54].

3. CLASSIFICATION

3.1. Overview

Text classification is a task of assigning a document to one or more predefined categories. This is a problem in library, information and computer sciences that can be realized manually or algorithmically. In information and computer sciences that use algorithmic approach, documents may be classified based on their subjects. Each subject is considered as a *category*. In this thesis single subject classification is considered, i.e. each document can belong to only one category. In the cases that there are only two possible categories, classification places in the binary classification domain but, for more than two categories multiclass (or multinomial) classification is taken into account. Binary classification classify samples of a data set into two groups according to a classification rule. As a example of binary classification task, we can mention to *information retrieval*, in detecting whether a page or document should belong to the result set of a search or not. The aim of classification is finding more relevant results to the user's query.

Opinion mining can be considered as multiclass classification task while three *positive*, *negative* and *neutral* categories are addressed. In machine learning, while some classification algorithms naturally classifies documents into more than two classes, others accomplish this using nature binary algorithms. In fact, they turn the binary algorithms into the multinomial classifiers by using various strategies.

The *one vs. all* is known as a popular strategy to reduce the problem of multiclass classification to multiple binary classification problems. In this strategy, a single binary classifier is generated for each class in which the samples of the class is considered as positive and all others is assumed as negatives. At the prediction time, all classifiers are applied to an unseen sample and predicting is realized by the way that corresponding classifier achieves the highest confidence score.

As another alternative, *one vs. one* reduction method trains $k(k+1)/2$ classifiers for k multiclass problem. Each classifier uses the samples of a pair classes to build a model that distinguishes these two classes. Finally, a decision scheme is applied to identify the class label of an unseen sample. Decision scheme means after applying all $k(k+1)/2$ classifiers to an unseen sample, the class that gained the highest number of predictions to itself is known as the target class.

In machine learning, algorithmic or automatic classification is realized according to three approaches: 1) *supervised classification* where prior known information in category membership (as an external feedback) are available for correct classification; 2) *Unsupervised classification* where it is done without any feedback information and known as document clustering, 3) *semi-supervised classification* where is applied by some documents labeled by the external agents.

This chapter addresses supervised classification task and presents its basic concepts. Over the chapter, various metrics and methods are also introduced either to evaluate the performance of classification models or make a correct comparison between them.

3.2. Basic Concept

In machine learning, classification is identifying category of a new document on the basis of a training set containing documents whose category memberships are known. Actually, the aim of classification is to find mapping Φ, from a set of documents as a training set $X : \{x_1, x_2, \cdots, x_k\}$ to a set of categories as a target set $Y : \{y_1, y_2, \cdots, y_j\}$, i.e. $\Phi : X \rightarrow Y$ in order to predict class labels of previously unseen documents. In classification, the input data is represented as a collection of pairs (x, y) where x is known as a feature set that is used for representation of documents and y is the output feature that identifies the class label. In the training stage, input data is first represented by the vectors of features, then they are given to a learning algorithm to induct the classification model (shown in Figure 3.1.). At the prediction stage, this model is used to predict the class label of unknown documents.

Machine learning techniques including decision tree, support vector machines, k-nearest neighbors, neural networks, Bayesian are used to classification of documents [12, 16, 24, 55, 56]. Each technique employs a learning algorithm to generate a model that best fits the relationship between the feature set and class label of input data. Moreover, the generated model should perform well on the data that has never seen before. In fact, the objective of learning algorithms is to generate models with high accuracy in predicting the class label of previously unseen documents as well as known ones. To evaluate the performance of a classification model, while the number of input documents are considered as a test set, evaluation is carried out based on correctly and incorrectly predictions of their class labels. The evaluation of classification models will be discussed in section 3.5.

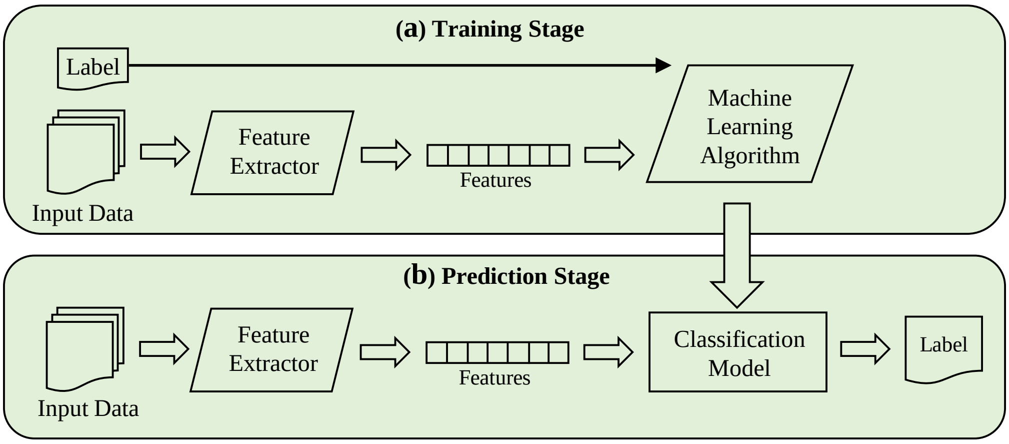


Figure 3.1. Supervised classification process

3.3. Underfitting and Overfitting

Generally, two types of errors can occur over the classification process called as *training error* and *generalization error*. Training error is the number of misclassification cases occurred on the training set, while generalization error is a estimation of the number of misclassification cases that can be observed over unseen instances. In order to evaluate the accuracy of a classification model, data set is divided into two groups. First group consists of labeled instances which are used in training stage for building classification model (called as training set) and second one is employed to evaluate the generated model (called as test set). A good classification model must accurately fit the training data as well as instances it has never seen before. It means that a model with low training error may not be always known as a good classification model. This is crucial on which models may not fit the training data well but can accurately classify unseen instances. In other words, a model with a low training error may generate a high generalization error; this situation is known as *model overfitting*.

To depict the model overfitting, the training and test error rates of decision tree are schematically represented in Figure 3.2. At first, the both training and test error rates of the model are large, because the model has not yet learned a true structure of data. This situation is known as *model underfitting*. At this point, the model has not fitted the training data well. By growing the model, the both error rates degrade and model are biased to fit the training data. But after a while the test error rate begins to increase though the training error+++ continues its own reduction tendency. After this point, the model overfitting occurs (as shown in Figure 3.2. by a black line) and growth of the model reduces the efficiency of learning task.

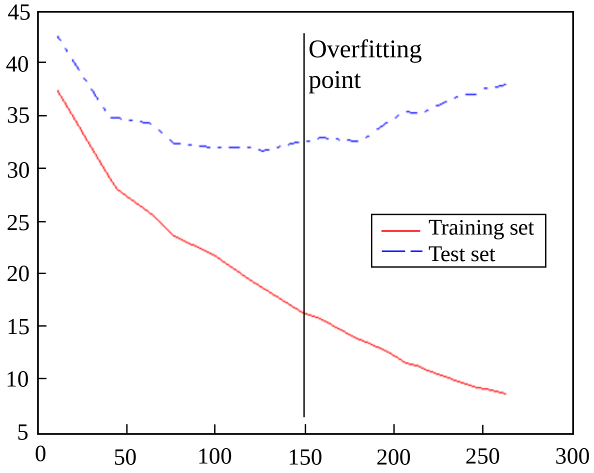


Figure 3.2. Training and test error rates while overfitting occurs

By going on the training procedure, the complexity of model may increase and model persists in fitting the data. Therefore, the training error rate can be reduced while the test error rate may be still large.

Excessive fitting of training data may lead to bring noise into the learning process. At this point, the model try to fit the faulty data and consequently, its generality may be reduced on the test instances. The shortage of representative samples can be known as another cause of model overfitting. In fact, learning algorithm cannot refine its own model when few samples are available in the training set.

3.4. Model Complexity and Generalization Error

Since model overfitting is related to model complexity, it is important to know when the right complexity of model is achieved. A model with the lowest generalization error propose an ideal complexity. However, just training data is available to learning algorithms over the model building. Thus, it is not possible to know how well a model performs on the samples which have been never seen before. At this point, an estimation of generalization error can be computed for the model. Several methods is used to estimate the generalization error, but in this section two methods are presented as follows:

1. *Resubstitution Estimate*

   In this approach training error is considered as an optimistic estimate for the generalization error. In fact, resubstitution estimate assumes that training set can provide a good representation of data. Nevertheless, it can not be considered as a strong estimation of the generalization error.

2. *Use of Validation Set*

   In this approach, to estimate generalization error, the original training set is divided into two subsets. One of them is used to build a model and other one that is known as validation set is used to estimate the generalization error. The ratio of training set to validation set is typically assumed as 2 to 1. In parametric algorithms which different level of complexity might be obtained (e.g. neural networks), this approach can estimate the best complexity. In fact, by adjusting the parameters of learning algorithm, model with the lowest error rate on the validation set is introduced as the best. Overall by this way, an estimation of model complexity can be produced during the learning process. As stated in chapter 1., this procedure is included in the post processing stage of the learning process.

3.5. Classification Model Evaluation

3.5.1. Metrics for Performance Evaluation

As mentioned earlier, the estimate of generalization error assists the learning algorithm in finding a model with the right complexity which is not sensitive to overfitting. After building classification model via training data, it is applied to the test set to evaluate the performance of the model on previously unseen data. In general, the performance of a classification model is estimated based on the number of test samples correctly and incorrectly predicted by the model. *Accuracy* is the ratio of correctly predicted samples to all test samples and subsequently *error rate* is known as the ratio of incorrectly predicted samples to all samples. These metrics provide a general perspective of the model performance. They calculated based on confusion matrix values indicated in Table 3.1. for binary classification task. In this table, each entry shows the number of samples with respect to the corresponding condition. For example, if two *positive* and *negative* classes are supposed, true positive (TP) denotes the number of correctly predicted samples as positive and false negative (FN) is known as the number of incorrectly predicted samples as negative. The sum of true positive and false negative cases ($TP + FN$) yields the number of total samples belongs to positive class. Although the confusion matrix values present the information about how well a model performs, they cannot provide a single circumstance to compare the performance of different classifiers. To do this, the different evaluation metrics were proposed by using different combination of the confusion matrix values. These metrics are used to compare the performance of different classifiers on the same domain. The most widely used metrics are summarized in Table 3.2. as well as their formulas.

Accuracy and error rate are widely used as evaluation metrics to determine the performance of classifiers. Most of classifiers are reluctant to achieve the highest accuracy as well as the lowest error rate. However, these metrics may not always attain a right view of classification performance because these metrics only take the global quantities into consideration. This issue would be crucial if data sets had imbalanced class distribution. In such datasets, accuracy and error rate metrics are affected by the major class and ignore the results obtained from minor class. To better evaluate the performance of classifiers on both minor and major classes, a variety of metrics were used in some domains. For example in medical diagnosis, *sensitivity* and *specificity* tests are considered as evaluation metrics. Test sensitivity which is called as *true positive rate* shows a capability of a model to correctly identify all people which have the disease, whereas the specificity which is known as *true negative rate* indicates the capability of a model to correctly detect all people without the disease.

In information retrieval context, *precision* and *recall* are used to evaluate the performance of retrieval task on the basis of retrieved documents which are produced for a query and the all

Table 3.1. Confusion matrix for binary classification task
TP: true positive, TN: true negative, FP: false positive, FN: false negative

		Predicted Value	
		Class=Positive	Class=Negative
Measured Value	Class=Positive	TP	FN
	Class=Negative	FP	TN

Table 3.2. The evaluation metrics by confusion matrix values

The metric name	Formula
Accuracy	$Acc = \frac{TP+TN}{TP+TN+FP+FN}$
Error rate	$ER = \frac{FP+FN}{TP+TN+FP+FN}$
True positive rate (recall or sensitivity)	$TPR = \frac{TP}{Actualpositives} = \frac{TP}{TP+FN}$
True negative rate (specificity)	$TNR = \frac{TN}{Actualnegatives} = \frac{TN}{TN+FP}$
Positive predictive value (precision)	$PPV = \frac{TP}{Predictedaspositive} = \frac{TP}{TP+FP}$
Negative predictive value	$NPV = \frac{TN}{Predictedasnegative} = \frac{TN}{TN+FN}$
F_1-measure	$F_1 = \frac{2\times Precision\times Recall}{Precision+Recall}$

relevant documents which are related to the certain topic in large domain. Precision takes all retrieved documents into account and is defined as the ratio of retrieved relevant documents to all retrieved documents, while recall is the number of relevant documents which were retrieved divided by all relevant documents.

In imbalanced classification task, precision and recall can be separately computed for each class. Thus, they can provide an independent perspective of classification performance without any consideration to class distribution. In text classification, precision is the ratio of documents correctly labeled as positive to all documents which are classified as positive. Recall is the ratio of documents correctly classified as positive to all existing positive documents. In fact, precision estimates a local accuracy for a class and recall a global accuracy. It is worth to note that recall can be known as the probability that a positive document is classified as positive by the model. However, to achieve recall of 100% by classifying all documents as positive is not significance. Therefore, recall alone cannot be enough to represent the goodness of model. Consequently the number of misclassified documents should be considered by another metric such as precision. As a result, precision and recall together can represent a right estimation of classification model independent of class distribution.

As categorization systems want to maximize either precision or recall, their harmonic combination called as *F-measure* (or *F-score*) is generally used in many research [11, 12, 14, 57]. The popular and balanced form of that known as F_1-measure is shown in Table 3.2. F_1 means that recall and precision contains equal weights. For other domains in which the weights of precision and recall are different, the general form of F-measure is computed as follow [58]:

$$F_\beta = (1+\beta^2)\frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{1}$$

where β is a non-negative real value that shows the significance of recall than the precision. For $\beta > 1$ recall is more significant than the precision e.g. F_2 and for $0 < \beta < 1$ precision is more emphasized than recall e.g. $F_{0.5}$.

3.5.2. Methods for Estimation of Classifier Performance

Previous section presented the methods to estimate the generalization error during training. This is useful to find a model with the right complexity which avoid overfitting. However, after constructing the classification model, it should be employed to predict the class labels of samples it has never seen before. To achieve this, a test set is provided to measure the performance of the model. It is clear that the class labels of test samples must be known to measure the performance. In this subsection, a variety of methods are presented to produce test data to estimate the performance of a classifier.

1. *Holdout Method*

   In this method, the original labeled data is divided into two disjoint sets called as training and test sets (e.g. two-third for training set and one-third for test set). The training set is used to build the classification model and subsequently test set is employed to evaluate its performance.

2. *Random Subsampling*

   This method performs the holdout method for several times to improve the performance of a classifier. As training and test sets are randomly selected, there is not any control on the number of times each sample occurs in training and test sets. The overall performance is computed by taking an average of the performance of iterations.

3. *Cross Validation*

   Since there is no control over the number of times each sample is used for training and testing in the random sampling method, cross validation approach is proposed to resolve this problem. In this approach the original data is divided into k equal-sized partitions. In the k iterations, each partition is used once for testing and the union of k-1 partitions are used for training. This approach is called as k-fold cross validation. Thus, it guarantees that each sample is used the same number of times for training and exactly once for testing.

4. *Stratified Sampling*

   In the cross validation approach, since partitioning is done randomly, there is no control over the non-uniformity of data distribution in each partition. Stratification is the process that divides samples of a dataset into homogeneous subsets before sampling. Each subset should be mutually exclusive. It means that every sample in the dataset must be assigned to only one subset. Each subset should be also comprehensive. It means that no sample in the dataset can be excluded. Finally, simple random sampling or cross validation is applied within each subset. This method can improve the performance of the sampling and consequently leads to better evaluation.

5. *Bootstrap Sampling*

   The previous methods were carried out without any replacement, i.e. there are no duplicate samples in the training and test sets. In the bootstrap approach, when a sample is chosen for training, it does not eliminate from the original set, so the already chosen sample may be reselected for training set. If the data set is sufficiently large, it can be shown by probabilistic theory, on average, a *bootstrap sample* contains about 63.2% of the samples in the original data. This estimation comes from the fact that the probability a instance is chosen by bootstrap sample is $1 - (1 - 1/N)^2$ where N is the number of instances in original data. For large Ns, the probability close to $1 - e^{-1} = 0.632$. Samples that are not included in the bootstrap sample, constitute the test set. The model induced from the training set is then applied to the 36.8% remaining samples as a test set to estimate the performance of the model. The sampling procedure is iterated n times to generate the n bootstrap samples.

   In this approach, to compute the overall accuracy for a classifier, several methods are used. However, *0.632 bootstrap* is a method which is widely used in computing the overall accuracy as follows:

$$acc = \frac{1}{n}\sum_{i=1}^{n}(0.632 \times bacc_i + 0.328 \times tacc) \tag{2}$$

where $bacc$ is the accuracy of each bootstrap sample and $tacc$ denotes the accuracy computed from a training set which includes all labeled samples in the original data.

3.5.3. Confidence Interval for Accuracy

When the performance of two classifiers are evaluated on two test sets that vary in the number of samples, the observed difference in accuracy between two classifiers may not be statistically significant. In order to know how much confidence we can place on the accuracy of a model, the confidence interval of the model is estimated based on normal distribution of the accuracy. In other words, for large test sets, accuracy has a normal distribution with mean p and variance $p(1-p)/N$. So true accuracy (p) of the model is computed as follows:

$$p = \frac{2 \times N \times acc + Z^2_{\alpha/2} \pm Z_{\alpha/2}\sqrt{Z^2_{\alpha/2} + 4Nacc - 4Nacc^2}}{2(N + Z^2_{\alpha/2})} \tag{3}$$

Where N is the number of samples in the test set and acc is the empirical accuracy is obtained from Z Table of normal distribution.

4. TERM WEIGHTING AND FEATURE SELECTION APPROACHES

In machine learning, text classification is a supervised learning task to categorize unlabelled documents into thematic predefined categories based on an inductive model. A text classifier typically consists of the following phases:

1. *term selection (or feature selection) phase*:
   In this phase a subset of the most relevant terms are selected for classification task. This phase leads to faster computation as well as more effective representation for classification task.

2. *document representation phase*:
   This phase provides a numeric representation of documents in which each document is represented as a set of words without any regarding to grammatical points and word order. The objective is to transform textual documents into a realizable form for any classifier. As a well-known method, vector space model (VSM) is known as a text representation model which makes a transformation from content of the natural language texts into a vector of term space [9].

3. *training phase*:
   In this phase, represented documents are given to a classifier to train the classification model for predicting class labels of previously unseen documents.

While the first two phases are known as preprocessing tasks, they can consistently affect the performance of classifiers. As the text data sets with large size, noisy samples, high dimension and imbalanced class distributions are available, the preprocessing always becomes as a challenge task in text classification domain. Term weighting approach as an effective preprocessing task is widely used in text classification process. Depending on its capabilities, it can be made a progress on the performance of classifiers when the data sets contain noisy samples or imbalanced class distributions. Feature selection as another alternative can be useful either in the same circumstances or in the dimensionality reduction. This chapter introduces the term wighting approach and subsequently explains feature selection methods and their characteristics. Overall, the aim of the chapter is to present two preprocessing tasks and show their positions in text classification.

4.1. Term Weighting Scheme

To better distinguish documents in the vector space model, the term weighting approach is inserted into the document representation phase to improve the performance of text classification. At first, traditional methods inspired by information retrieval are used for the purpose of term weighting. Their basic assumptions are listed as follows [10]:

- "multiple appearances of a term in a document are no less important than single appearance" (*tf* assumption)
- "rare terms are no less important than frequent terms" (*idf* assumption)
- "for the same quantity of term matching, long documents are no more important than short documents " (*normalization* assumption).

The *tfidf* as a standard weighting scheme has been used in many studies [11, 12, 17, 35]. Because, it provides an effective solution for the classification of imbalanced texts by relying on these assumptions. It has been formulated in form of multiplying term frequency (*tf*) by inverse document frequency (*idf*). The common and normalized form of *tfidf* weighting are shown in Eqs. 4 and 6 respectively [9, 55]:

$$tfidf(t_i, d_j) = tf(t_i, d_j) \times idf(t_i) \tag{4}$$

$$idf(t_i) = \log(\frac{N}{N_{t_i}}) \tag{5}$$

$$w_{i,j} = \frac{tfidf(t_i, d_j)}{\sqrt{\sum_{k=1}^{|T|} tfidf(t_k, d_j)^2}} \tag{6}$$

where $tf(t_i, d_j)$ denotes the number of times that term t_i occurs in document d_j, N is the number of all documents in the training set, N_{t_i} denotes the number of documents which term t_i occurs at least once and $|T|$ denotes the number of unique terms.

The *tfidf* method is constituted from local and global principles. The frequency of a term within a specific document (tf) provides the local principle in the term weighting scheme and inverse document frequency (idf) supplies the global principle. In other words, tf specifies the weight of term t_i within a particular document and idf determines the contribution level of the term t_i in a global perspective.

According to previous research [9, 11, 55], term frequency (tf) is known as a fundamental element for local component of the term weighting scheme because, even if tf is used as a term weighting scheme alone, it can yield good performance [9, 11, 55]. On the other hand, *idf* is considered as an unsupervised function since it does not take into account the category membership in documents. By idf function, a term with less document frequency possesses a higher degree of importance than the others. Thus, the terms belong to minor class likely include the higher values of idf than the major one due to the shortage of documents in the minor class. Consequently, the term weighting process predominates the minor class and causes the classification are not biased towards major class. Hence, $tfidf$ weighting method has provided better representation for imbalanced data sets in many studies [10, 17, 18].

4.2. Supervised Term Weighting Scheme

In information retrieval, regarding the lack of prior known information on the category membership in documents, inverse document frequency (idf) was used as a global component of term weighting scheme. This factor takes into account the distribution of documents in the whole collection. In text classification, if labelled documents are available, finding a global component can be expanded from *idf* to other more accurate metrics. Thus, the term weighting approach which uses the prior known information, has been introduced as supervised term weighting in the literature [10]. In this approach, metrics used in the term selection phase are replaced by the *idf* function, because the aim of the term selection phase is to associate important terms with each category. In fact, the ability of feature selection to capture the more relevant information for each category by selecting significant terms bring up a motivating factor for supervised approach. Therefore, the supervised approach used category based term selection metrics as the global component of term weighting scheme, since the main purpose of text classification is to identify whether document belongs to a particular category.

Table 4.1. All metrics used in the experiments as the global component of term weighting scheme for binary classification

Metric name	Abbreviation	Formula
Chi square	X^2	$N\frac{(ad-bc)^2}{(a+c)(b+d)(a+b)(c+d)}$
Information gain	ig	$\frac{a}{N}log\frac{aN}{(a+c)(a+b)} +$ $\frac{b}{N}log\frac{bN}{(b+d)(a+b)} +$ $\frac{c}{N}log\frac{cN}{(a+c)(c+d)} +$ $\frac{d}{N}log\frac{dN}{(b+d)(c+d)}$
Odds ratio	or	$log\frac{ad}{bc}$
Relevance frequency	rf	$log(2 + \frac{a}{max(1,c)})$

Notation:

a denotes the number of documents belong to positive class which contains term t_i
b denotes the number of documents belong to positive class which does not contain term t_i
c denotes the number of documents belong to negative class which contains term t_i
d denotes the number of documents belong to negative class which does not contain term t_i
N denotes the number of all documents in the training set

4.3. Feature Selection Scheme

In text classification, term selection (or feature selection) is employed to reduce the dimension of input data by selecting more relevant terms to categories. Moreover, it can improve the classification accuracy by preventing the model overfitting [18]. Feature selection metrics by using probability and information theories compute the relevancy (or not relevancy) power of terms associated with each category. In this thesis, we use the popular term selection metrics employed in [11] for supervised term weighting scheme. These metrics are represented by information elements in Table 4.1.

In general, feature selection is realized on the basis of feature ranking. First, a feature selection metric evaluates all features and estimates their importance level by assigning a score for each one. Then, the scores are sorted in descending order to select the features which possess the highest scores. Two major approaches is basically used to rank and assess the features, i.e. 'local' and 'global' approaches. In local approach different sets of features are selected for each category, whereas in global approach a feature set is generated globally for all categories. In other words, local approach identifies the most relevant terms to each category, while global approach selects the features possessing high discriminating power across all categories.

In this section, we present four major feature selection metrics that are widely used in text classification domain. Table 4.1. summarizes these metrics. It is worth to note that the shown metrics calculate local scores. It means that they indicate the relevance power of a term with respect to a specific category $f(t_i, c_j)$.

In order to evaluate the score of term t_i in the global sense $f_{global}(t_i)$, a *globalization* technique is applied to its local feature selection score $f(t_i, c_j)$. The most common globalization techniques are listed as follows [10]:

$$f_{sum}(t_i) = \sum_{j=1}^{|c|} f(t_i, c_j) \tag{7}$$

$$f_{wsum}(t_i) = \sum_{j=1}^{|c|} P(c_j) f(t_i, c_j) \tag{8}$$

$$f_{max}(t_i) = \max_{j=1}^{|c|} f(t_i, c_j) \tag{9}$$

The globalization techniques $f_{sum}(t_i)$, $f_{wsum}(t_i)$ and $f_{max}(t_i)$ are known as the global version of their category-specific values $f(t_i, c_j)$.

4.3.1. Chi-Square

In statistic, the chi-square test is used to determine whether there is a significant association between two categorical variables. Let A and B denote our variables. Suppose the both variables have two categorical values. The null and alternative hypotheses are represented as follows:

- H_0: Variable A and Variable B are independent.
- H_a: Variable A and Variable B are not independent.

While the null hypothesis states that the variable A is not associated with the variable B, alternative hypothesis indicates the variables are related. It means that one variable causes the other.

At this point, by using a sample data we can make a decision to accept or reject the null hypothesis. Several types of chi-square test are employed in different domains. The simplest one is accomplished by a binary contingency table, shown in Table 4.2., when only two nominal values are available for each variable.

Table 4.2. General notation for a binary contingency table

		Variable B		Total
		value 1	value 2	
Variable A	value 1	a	b	a+b
	value 2	c	d	c+d
Total		a+c	b+d	a+b+c+d=N

The letters a, b, c and d denote the number of instances contained in the sample data corresponding to their variable values. For binary contingency table, the *chi-square statistic* is calculated through the Eqs. 10 to 12:

$$X^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{10}$$

where o_i corresponds to the observed frequency in the i^{th} cell of the contingency table and E_i is the expected frequency in the i^{th} cell of the table. The expected frequency corresponding to each cell is generally estimated as the following equation:

$$E_i = \frac{RowTotal * ColTotal}{N} \tag{11}$$

According to Eq. 10, the chi-square statistic compares the observed frequency in each table cell to the frequency which would be expected under the assumption of no association between variable A (table rows) and variable B (table columns). By combining two Eqs. 10 and 11, the chi-square statistic formula is transformed to Eq. 12.

$$X^2 = \frac{N(ad - bc)^2}{(a+c)(b+d)(a+b)(c+d)} \tag{12}$$

The degree of freedom can be calculated for this case as follow:

$$df = (n_A - 1) \times (n_B - 1) \tag{13}$$

where n_A denotes the number of possible values for variable A and n_B is the number of possible values for variable B. For binary contingency table, the value of df equals to 1. Finally, by using chi-square statistic, significance level which is mostly chosen as 0.05, degree of freedom and chi-square distribution table (see Table 4.3.), we can make a decision to accept or reject the null hypothesis. For this purpose, a comparison is made between chi-square statistic and the value obtained from chi-square distribution table for a specific df and *alpha level*. If the chi-square statistic is greater than the table value, the null hypothesis is rejected in the sense that there is a significant relationship between two variables. At this point, if two variables A and B are dependent, it can be say that the occurrence of variable A more likely makes the occurrence of the variable B.

Table 4.3. Chi-square distribution table

	significance level (alpha)					
df value	0.5	0.1	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.21	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.07	13.388	15.086	20.517

This sense can be used in feature selection process to specify the relevance power of a particular term to a category. In this case, variables A and B correspond to the occurrence of term t_i and category c_j respectively, and the chi-square statistic measures the dependence between term t_i and category c_j. Therefore, the contingency table and chi-square statistic are transformed to the following equation and table.

$$X^2(t_i, c_j) = \frac{N[P(t_i, c_j)P(\bar{t_i}, \bar{c_j}) - P(t_i, \bar{c_j})P(\bar{t_i}, c_j)]^2}{P(t_i)P(\bar{t_i})P(c_j)P(\bar{c_j})} \tag{14}$$

In Table 4.4., t_i and c_j are two random categorical variables which correspond to i^{th} term in the vocabulary set and j^{th} category among all categories. According to values of each

Table 4.4. Binary contingency table for occurrence of term t_i in category c_j

	Containing term t_i	Not containing term t_i
Belonging to class c_j	a	b
Not belonging to class c_j	c	d

variable, four letters a, b, c and d denote their corresponding document frequencies and N is the number of total documents.

In equation 14, probabilities are interpreted on a collection of documents in the sense that e.g. $P(t_i, \bar{c_j})$ indicates the probability that for a random document k, term t_i occurs in document k and k does not belong to category c_j. On the hand, the $P(t_i)$ denotes the probability term t_i can occur in a particular document and $P(c_j)$ is the probability that a particular document can belong to category c_j . The following equations estimate these probabilities based on contingency table values.

$$P(t_i, c_j) = \frac{a}{a+b+c+d} \tag{15}$$

$$P(\bar{t_i}, c_j) = \frac{b}{a+b+c+d} \tag{16}$$

$$P(t_i, \bar{c_j}) = \frac{c}{a+b+c+d} \tag{17}$$

$$P(\bar{t_i}, \bar{c_j}) = \frac{d}{a+b+c+d} \tag{18}$$

$$P(t_i) = \frac{a+c}{a+b+c+d} \tag{19}$$

$$P(\bar{t_i}) = \frac{b+d}{a+b+c+d} \tag{20}$$

$$P(c_i) = \frac{a+b}{a+b+c+d} \tag{21}$$

$$P(\bar{c_i}) = \frac{c+d}{a+b+c+d} \tag{22}$$

According to the definition of chi-square for text feature selection, the most valuable terms are those that are distributed differently in the two sets of documents that do or do not belong to category c_j. In fact, chi-square captures the terms for a particular category that help to identify membership or non-membership in this category [10, 36, 59, 60].

4.3.2. Information Gain

This metrics indicates the number of bits of information for category prediction when the presence or absence of a term in a document is given. suppose a set of independent random samples of document D in which D has four possible terms t_1 up to t_4. If the probability of term occurring in the document D are the same for each term (e.g. $P(D = t_1) = 1/4$, ... , $P(D = t_4) = 1/4$), the minimum number of bits needed to encode the all terms is 2. It means that document D can be transmitted to a binary form by using 2 bits for each term (e.g. $t_1 = 00,\ t_2 = 01,\ t_3 = 10$ and $t_4 = 11$). While the probabilities are not the same e.g. $P(D = t_1) = 1/2,\ P(D = t_2) = 1/4,\ P(D = t_3) = 1/8$ and $P(D = t_4) = 1/8$, the average number of bits to encode the document D will equals to 1.75 per term (e.g. $t_1 = 0$, $t_2 = 10,\ t_3 = 110$ and $t_4 = 111$). In general, it can be say that the smallest possible number of bits depends on the distribution of terms in documents. For n number of terms t_1, ... t_n, the smallest number of bits is calculated as the sum of entropies for each term probability p_i, i.e.

$$E(D) = -\sum_{i=1}^{n} p_i \log p_i \tag{23}$$

In equation 23, the $E(D)$ is known as the entropy of D. Based on obtained value for $E(D)$, the following statements are achieved:

- *High Entropy*, indicates the uniform distribution of terms in D.
- *Low Entropy*, indicates the erratic distribution of terms in D.

In general, entropy comes from information theory and measures the level of impurity for a group of samples. Impurity indicates the distribution of samples through the categories. Fig. 4.1. illustrates the various levels of impurity when there are two groups of samples. In machine learning, a data set with high level of impurity can provide more useful information for category prediction.

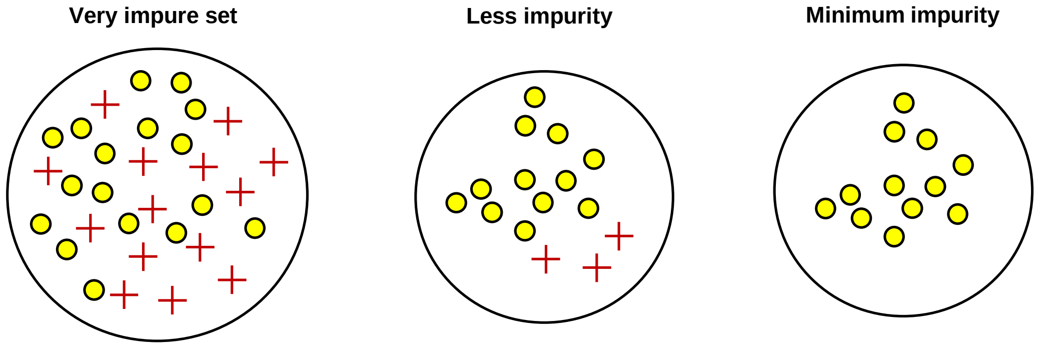


Figure 4.1. The variety of impurity level for two categories. The set indicated on the left side is a good training set for learning, while the right set cannot be known as a good training set

Let move to entropy Eq. 23 to infer a global sense. If an instance space includes n groups of samples in which the occurring probability of each one is p_i (i = 1, ..., n), Eq. 23 gives a sense how well the samples of different groups are distributed in the whole space. A good way of thinking this is to suppose that a higher entropy is equivalent to a more information content.

In feature selection, this sense is used to capture terms associated with a particular category by using four different cases of presence or absence of them i.e. (t_i, c), $(t_i, \bar{c})$, $(\bar{t_i}, c)$ and $(\bar{t_i}, \bar{c})$. At this point, the entropy $E(t_i, c)$ indicates the impurity level of the four conditions while their occurring probabilities are different. The main purpose is to capture terms that provide high impurity associated with the category c. It means that the terms which can provide union distribution in the presence or absence of category c, possess a high representation power associated with category c and consequently are useful to classification task.

The information gain of term t_i and category c_j is defined as follow [36, 59, 60]:

$$ig(t_i, c_j) = \sum_{c \in \{c_j, \bar{c_j}\}} \sum_{t \in \{t_i, \bar{t_i}\}} P(t, c) \log \frac{P(t, c)}{P(t)P(c)} \tag{24}$$

Information gain is widely used as a term goodness measuring in the filed of machine learning [61–63]. A high value of information gain for the term t_i and category c_j indicates a high association between term t_i and category c_j and consequently term t_i can be considered as an ideal indicator associated with the class c_j. The probabilities are interpreted in the same way that were explained in the previous subsection.

4.3.3. Odds Ratio

The odds ratio is used to measure the strength of association between the presence or absence of event A and the presence or absence of event B in a given sample space. It is worth to note that the odds ratio is known as the ratio of the odds, not the percentage and term 'odds' is often used to mean 'chance' or 'likelihood'. In statistics, the odds of occurrence of an event is calculated as the probability of the event divided by the probability of the event when it does not occur, i.e. $p/(1-p)$. In fact, the odds of an event indicates how likely the event occurs compared to how likely the event does not occur [64]. For example, in a coin flip if the probability of getting head is 80% and the probability of getting tail is 20%, then the odds of getting head is $80/20 = 4$.

We again come back to odds ratio. This is addressed when the odds of occurring of two different events are compared. At this point, the odds ratio is obtained by dividing the odds of the first event by the odds of the second event. i.e.

$$or = \frac{p/(1-p)}{q/(1-q)} = \frac{p(1-q)}{(1-p)q} \tag{25}$$

where p is the probability for the first event, and q is the probability for the second.

Let see the odds ratio while an event occur associated with an exposure. In this case, the odds ratio can measure the association between the exposure and the event. In other words, odds ratio indicates the odds that an event will occur given a particular exposure, in comparison with the odds of the event occurring in the absence of that exposure.

Moreover, the odds ratio can be used to specify whether a particular exposure can be known as a risk factor for a particular event. This is addressed when a variety of risk factors are considered for an event and their magnitudes should be compared [65, 66]. At this point, the following statements can be considered:

- $or = 1,$ it means that exposure does not make any impact on the odds of event
- $or > 1,$ it means that the higher odds of event associated with the exposure
- $or < 1,$ it means that the lower odds of event associated with the exposure

Similarly, in text feature selection odds ratio can measure the odds of the term occurring associated with the presence or absence of a particular category. Its extended version known as log odds ratio is calculated for term t_i and category c_j as follow [11, 36, 60]:

$$or(t_i, c_j) = \log \frac{P(t_i|c_j)[1 - P(t_i|\bar{c_j})]}{[1 - P(t_i|c_j)]P(t_i|\bar{c_j})} \tag{26}$$

The major idea is that the distribution of term t_i on the documents belong to category c_j is different from the distribution of documents belong to non-category c_j.

The conditional probabilities indicates the chance of occurring term t in a particular document which does or does not belong to category c. They are estimated by the contingency table elements as follows:

$$P(t_i|c_j) = \frac{a}{a+b} \tag{27}$$

$$P(t_i|\bar{c_j}) = \frac{c}{c+d} \tag{28}$$

$$1 - P(t_i|c_j) = \frac{b}{a+b} \tag{29}$$

$$1 - P(t_i|\bar{c_j}) = \frac{d}{c+d} \tag{30}$$

According to these equations, the value of odds ratio is estimated for two particular term and categoty as follow:

$$or = \log \frac{ad}{bc} \tag{31}$$

4.3.4. Relevance Frequency

Relevance frequency (rf) is another strong feature goodness evaluation metric that was proposed in [11]. This metric is inspired by odds ratio. The difference between these two metrics is relying on the attitude that slightly increasing or decreasing the value of 'b' and 'd' cannot not have any impact on the discriminating power of terms [11]. In other words, adding or deleting the documents which do not contain a particular term does not affect on its term's goodness. Actually, the tf metric is defined based on the distribution of only relevant documents which contain this term.

According to this sense of feature evaluation, tf metric is formulated as follow:

$$rf(t_i, c_j) = \log[2 + \frac{P(t_i, c_j)}{P(t_i, \bar{c_j})}] \tag{32}$$

Based on contingency table values, the rf value for a particular term and category is estimated as follows:

$$rf = \log[2 + \frac{a/N}{c/N}] = \log[2 + \frac{a}{c}] \tag{33}$$

Since the base of logarithm is 2, a constant 2 is assigned to the formula in order to generate non-zero weights. Moreover, to avoid zero divisor, a minimal denominator as 1 is considered. Thus, the rf formula is transformed as:

$$rf = \log[2 + \frac{a}{\max\{1, c\}}] \tag{34}$$

4.4. Investigation of Term Weighting Schemes in Classification of Imbalanced Texts

In machine learning, binary text classification is a supervised learning task which categorizes unlabeled documents into two categories based on an inductive model learned from labeled documents. The common machine learning algorithms which have been used for this purpose include support vector machine (SVM), k-nearest neighbor (k-NN), naïve Bayesian (NB), neural networks (NN), decision trees (C4.5) and Rocchio [12, 17, 33]. As mentioned in chapter 1., binary classification by machine learning algorithms usually needs a training data with homogeneous class distribution. In imbalanced data sets that this requirement is not observed, major classes make prominent effect on the learning process of classifiers. At this point, while the dominant classes are well classified, the samples belonging to the minority class tend to be misclassified. In this case, samples belong to majority class are assumed as negative samples and ones belong to minority class is known as positive samples.

Class imbalance problem occurs in binary text classification when the number of negative documents significantly outnumber the positive ones. As stated in chapter 1., imbalanced data sets can be different in terms of class complexity, size of training set and the number of subclusters. Most of learning algorithms are suffering from these characteristics. In this chapter, we address binary text classification where the imbalanced circumstance are achieved by applying *one vs. all* strategy explained in chapter 3. This strategy provides an imbalanced circumstance in which variety of complexity level can be available as well as subclusters.

In this condition, the positive class consists of documents that belong to one subject and negative class consists of all other remaining items. Thus, increment in the number of negative class samples leads to growth of class complexity. In this case, the positive class can be formed as a cluster while the negative class cannot. Therefore, raising the degree of imbalance by incrementing the negative documents with different subjects causes aggravation of class distribution and growing the number of subclusters.

While documents are represented in vector space model, term weighting approach can be have major influence on predicting the class label of positive documents where an imbalanced class distribution is available. According to techniques employed in different term weighting schemes, various observations are more likely obtained under imbalanced data with different complexities. Moreover, each term weighting scheme differently affect training stage of learning algorithms for dealing with the class imbalance problem. Therefore, in this section a comparison is made between three types of term weighting schemes in combination with four classification algorithms. In fact, it is a challenge between traditional and supervised term weighting approaches when two kinds of feature evaluation metrics i.e. one-sided and two-sided are considered as global component of term weighting scheme. One-sided metrics are those that take only the positive features (i.e. relevant features associated with a certain category) into consideration, whereas two-sided metrics consider the negative features (non-relevant features) as well as the positive ones.

To best of our knowledge, in most of the studies the proposed weighting methods for dispelling the class imbalance problem were evaluated by using one or two classifiers (especially by SVM). In this chapter, we try to survey the term weighting strategy in combination with four algorithms which work based on four different approaches. Thus, the following objectives will be addressed in this section:

- Investigation of the supervised and unsupervised weighting approaches on imbalanced data sets as well as compatibility of each weighting method associated with machine learning algorithms.
- Comparing the effect of two-sided feature selection metrics with one-sided ones at the term weighting perspective.

In fact, we try to discuss which kind of feature selection metrics, as a component of term weighting scheme, can be beneficial to represent imbalanced data and which term weighting schemes are suitable for which machine learning algorithms. For this purpose, four different classifiers (SVM, k-NN, NB and C4.5) are employed in the experiments. The main reason of this selection is that they are based on different approaches (i.e. perceptron based, instance based, probabilistic based and information gain based).

4.4.1. Supervised Term Weighting Methodology

Feature selection is often employed in text classification tasks in order to reduce dimensionality when documents are represented as a set of words without considering the grammar and order of the words. On the other hand, it has positive effects on improving the classification accuracy by reducing over fitting problem (see chapter 4.). In this chapter information gain with local policy is used as feature selection metric since it has introduced better performance on the imbalanced text classification [18].

Feature selection metrics can be used as a global factor of term weighting scheme since they evaluate the importance of a term with respect to a specific category. In this chapter, two approaches are used in the formula of different feature selection metrics; (1) one-sided and (2) two-sided metrics. One-sided metrics take only positive features (i.e. relevant terms) into consideration since they compute the relevancy power of terms associated with a category.

We test two common one-sided metrics represented in chapter 4. i.e. 'relevance frequency' and 'odds ratio' in the experiments. Two-sided metrics consider both positive and negative features implicitly. In other words, they can take into account either the relevancy or non-relevancy power of terms with respect to a category. We also investigate the effectiveness of two well-known two-sided feature selection metrics i.e. 'information gain' and 'chi square' at the term weighting perspective. The mentioned feature selection metrics and their formulas have been summarized in Table 4.1.

In text classification, term weighting is usually realized by methods taken from information retrieval and text search fields as represented in chapter 4. In this chapter, *tfidf* is used as a standard term weighting scheme throughout the experiments and the effectiveness of supervised term weighting approach is probed in handling the class imbalance problem where two kinds of metrics are used. The supervised approach is named as *tffs* in the rest of chapter.

Figure 4.2. shows how supervised term weighting approach is realized in the preprocessing stage of the learning algorithms. In this process, after extracting unique terms from training data, the information elements of document frequencies are calculated for each term (i.e. a, b, c and d values of contingency table represented in chapter 4.). According to the number of categories, 4 values are yielded for each category associated with a certain term. e.g. for 8 categories, 32 values are computed per term. By applying a local feature selection, a particular set of terms is selected to represent documents in the vector space model. (e.g. we used information gain to select top 25 features per category). Then, feature goodness evaluation is applied to selected terms by means of feature selection metrics represented in Table 4.1. It worth to note that the goodness of features are computed based on positive class since the objective is to augment positive class where negative documents outnumber positive ones. Finally, the training and test documents are represented by the selected terms in the form of normalized $tf.fs$ term weighting scheme (see chapter 4.) to hand out to a classifier.

4.4.2. Experiments

In this chapter, the effect of each feature selection metric is investigated over the imbalanced text classification by considering as a global component of the term weighting scheme. At the experiment stage, we have used a subset of Reuters-21578 called as R8 and 20Newsgroups datasets which are publicly available at [67] for single label text categorization. These two

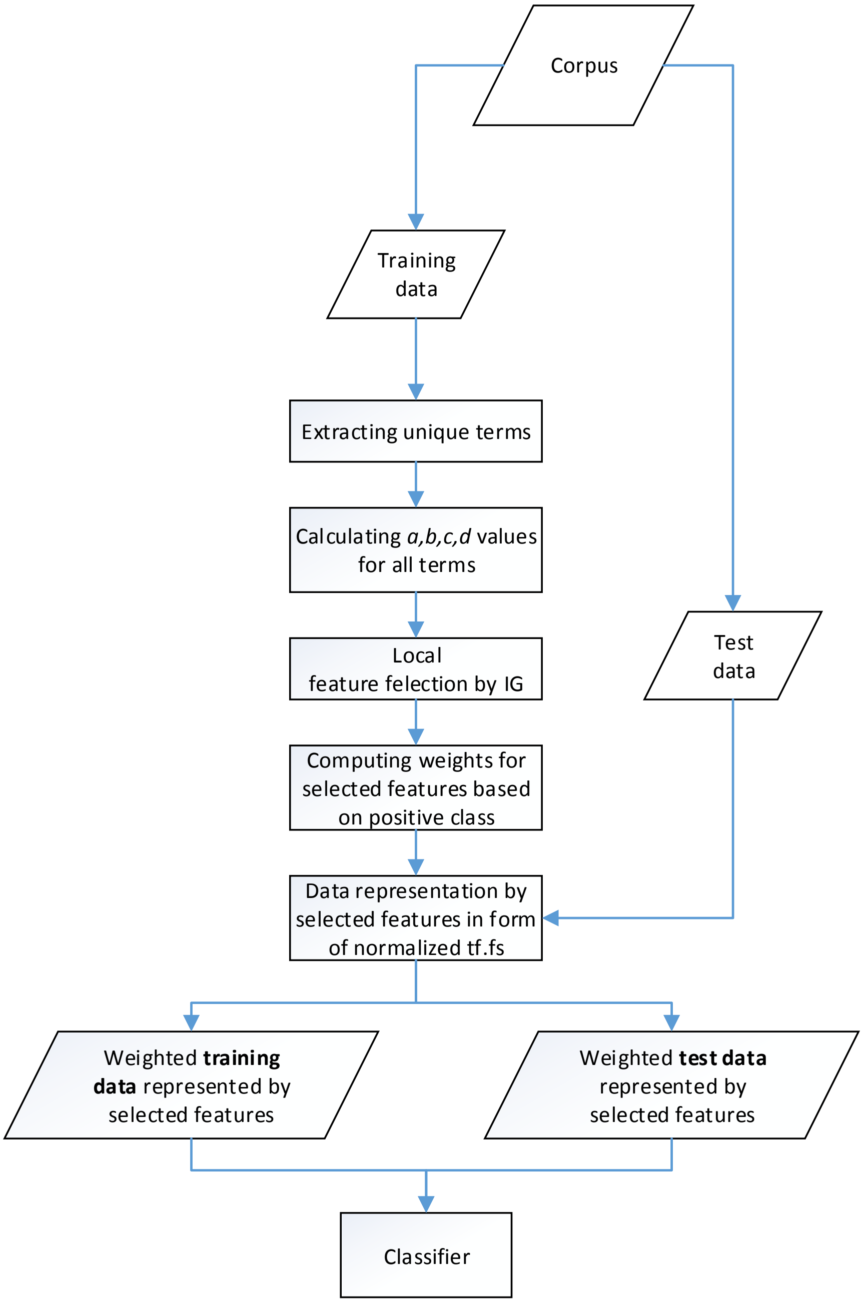

Corpus
Training data
Extracting unique terms
Calculating *a,b,c,d* values for all terms
Local feature felection by IG
Computing weights for selected features based on positive class
Data representation by selected features in form of normalized tf.fs
Test data
Weighted **training data** represented by selected features
Weighted **test data** represented by selected features
Classifier

Figure 4.2. The term weighting process for binary text classification

datasets have been widely used in text classification research [10, 14, 17]. Pre-processing steps have been applied on the datasets such as removing the 524 SMART stop words and applying Porter's Stemmer algorithm. We conducted two types of experiments for balanced and imbalanced cases.

4.4.3. Experimental Setup

In order to control the state of imbalance and degree of complexity, we selected one category as the positive class and the remaining portion as the negative one as [17] had done. The R8 dataset has eight categories with imbalanced number of documents for categories and consequently it has lower complexity than the 20Newsgroups dataset. In the 20Newsgroups dataset, there exist 20 categories with almost equal number of documents. Thus, with one vs. all configuration, we can make an imbalanced case with high complexity due to the abundance of different categories in the negative class.

First, we tried to make 1:1 configuration for R8 dataset by selecting the largest category among the others (i.e. *earn* category) as positive class, while the sum of remaining categories were considered as negative class. For 20Newsgroups dataset, *sci.space* was selected as positive class and *sci.electronics* was chosen as negative class. In the second stage, the imbalance situation was constituted on the R8 and 20Newsgroups datasets by selecting the *trade* and *sci.space* categories as positive class respectively with the consideration of the union of the other categories as the negative class. Thus, 1:20 imbalance ratio was approximately obtained for each dataset with different degrees of complexity. The experiments were performed on the original training and test sets for the both datasets as shown in Tables 4.6. and 4.5.

Four popular classification algorithms i.e. libSVM [68], Multinomial Naïve Bayes (MultiNB) [28], decision tree (C4.5) [33] and k-Nearest Neighbors (k-NN) [17] were used to evaluate the term weighting methods. In fact, we evaluated the compatibility of each classifier associated with each of the term weighting schemes. Furthermore, we used linear kernel with default parameters for libSVM and chose k=5, 15, 25 and 35 for k-NN algorithm. For k-NN, we computed the average of the results obtained from different values of k in the experiments. To evaluate the results, F_1-score metric obtained from *precision* and *recall* values is used as explained in chapter 3.

Table 4.5. Properties of Ruters-21578 dataset (R8) with imbalance ratio of each class while considering as target class in training set. $Imbalance\ ratio = \frac{N_{C_j}}{N-N_{C_j}}$

Class name	# of train docs	Imbalance ratio	# of test docs	Total # of docs
acq	1596	0.410	696	2292
crude	253	0.048	121	374
earn	2840	1.074	1083	3923
grain	41	0.008	10	51
interest	190	0.036	81	271
money-fx	206	0.039	87	293
ship	108	0.020	36	144
trade	251	0.048	75	326
Total	**5485**	**0.210**	**2189**	**7674**

4.4.4. Experimental Results and Discussion

4.4.4.1. Balanced Case

In the first stage of experiments, we took the 1:1 balanced situation into consideration combined with different complexity. Figure 4.3. shows the results of the supervised (*tffs*) and unsupervised (*tfidf*) term weighting schemes over the R8 dataset using the four different classifiers. It is observed that the SVM performs significantly better than the other classifiers. It also shows the compatibility of SVM with two-sided feature selection metrics when they are used in the term weighting scheme. According to obtained results, *tfidf* weighting gives better results than the supervised ones for k-NN, C4.5 and MultiNB. Among these classifiers, C4.5 and MultiNB are more sensitive to weighting schemes. Nonetheless, term weighting based on one-sided metrics are better approach for them in comparison with two-sided ones.

We compared the previous observation with the results obtained from 20Newsgroups dataset. Figure 4.4. indicates the performance of weighting schemes over the 20Newsgroups dataset using the same classifiers. As shown in Figure 4.4., both C4.5 and MultiNB methods perform better than the k-NN and SVM.

It is noted that the observation is different than the R8 dataset since its complexity is different from the 20Newsgroups. Also we selected two similar categories for 20Newsgroups dataset while the positive class in R8 dataset is less similar to negative class. This leads to increase in the error region between positive and negative classes in the training set and consequently

Table 4.6. Properties of 20 Newsgroups dataset with imbalance ratio of each class while considering as target class in training set. $Imbalance\ ratio = \frac{N_{C_j}}{N-N_{C_j}}$

Class name	# of train docs	Imbalance ratio	# of test docs	Total # of docs
alt.atheism	480	0.04	319	799
comp.graphics	584	0.05	389	973
comp.os.ms-windows.misc	572	0.05	394	966
comp.sys.ibm.pc.hardware	590	0.06	392	982
comp.sys.mac.hardware	578	0.05	385	963
comp.windows.x	593	0.06	392	985
misc.forsale	585	0.05	390	975
rec.autos	594	0.06	395	989
rec.motorcycles	598	0.06	398	996
rec.sport.baseball	597	0.06	397	994
rec.sport.hockey	600	0.06	399	999
sci.crypt	595	0.06	396	991
sci.electronics	591	0.06	393	984
sci.med	594	0.06	396	990
sci.space	593	0.06	394	987
soc.religion.christian	598	0.06	398	996
talk.politics.guns	545	0.05	364	909
talk.politics.mideast	564	0.05	376	940
talk.politics.misc	465	0.04	310	775
talk.religion.misc	377	0.03	251	628
Total/Average	**11293**	**0.05**	**7528**	**18821**

raises the generalization error for the model obtained from SVM. Hence the performance of SVM degrades in the 20Newsgroups dataset. According to both observations, we can conclude that the performance of one-sided metrics is better than the two-sided ones excluding SVM which can work well with two-sided based metrics, shown in Figure4.3.

4.4.4.2. Imbalanced Case

In the second stage of the experiments, we tested the behavior of term weighting schemes and classification algorithms over the 1:20 imbalanced case obtained form R8 dataset. First observation is that SVM performs well with one-sided term weighting methods and can even outperform *tf.idf*, while k-NN shows an adaptation with *tf.idf* and *tf.rf* term weighting schemes. On the contrary, MultiNB and C4.5 give better performance by two-sided methods and outperform *tfidf* (please see *tfidf* in the R8 dataset, shown in Figure 4.5.). In fact, Figure

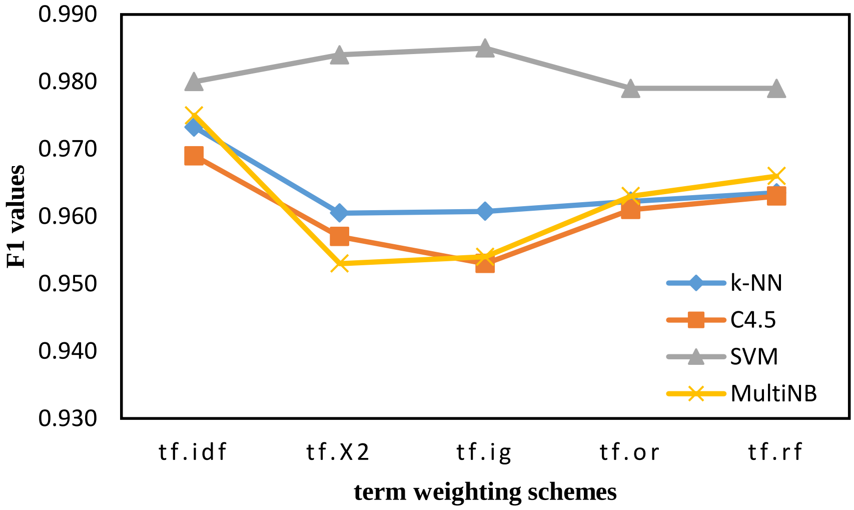


Figure 4.3. The F_1-values of five weighting schemes tested over R8 dataset with balanced setting using four different classifies

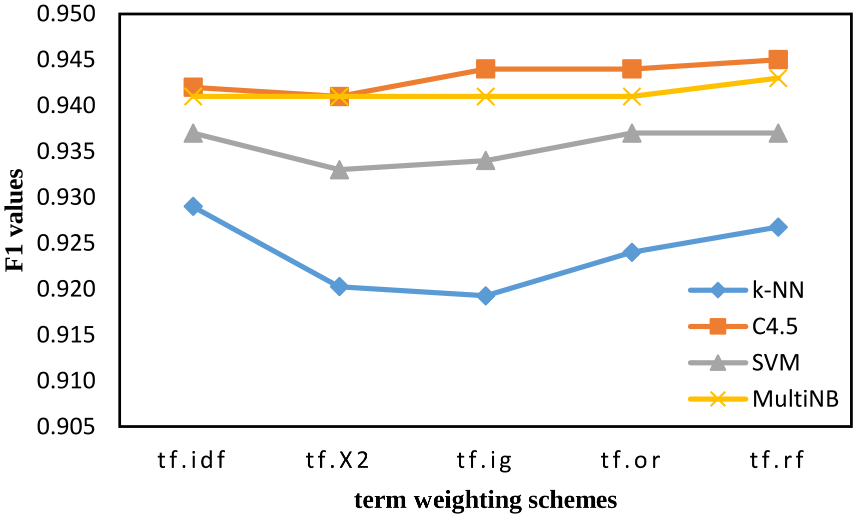


Figure 4.4. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with balanced setting using four different classifies

4.5. demonstrates the compatibility of one-sided methods with SVM, two-sided ones with MultiNB and C4.5, and both *tf.idf* and *tf.rf* with k-NN algorithm. It can be also observed that SVM and MultiNB effectively perform via supervised term weighting schemes on the imbalanced data. In order to expand the obtained results, we employed the same experiments on the 20Newsgroups dataset by using same imbalance ratio and more complexity configuration.

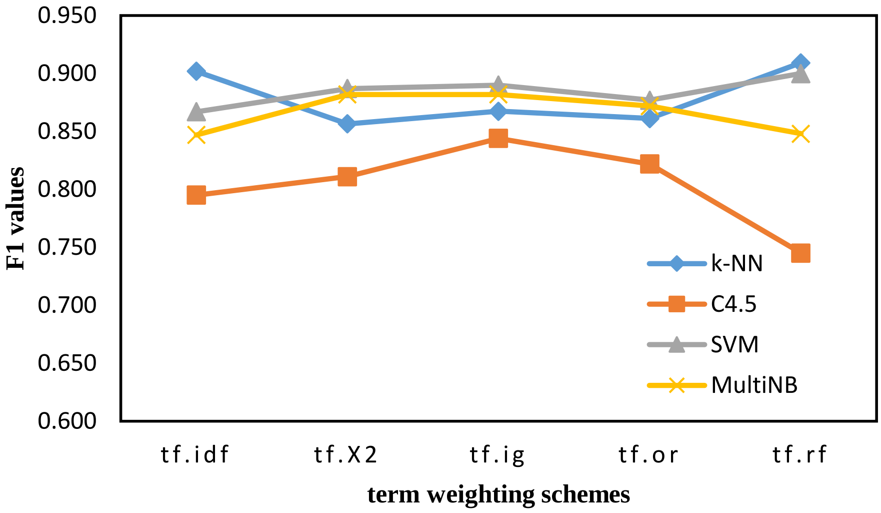


Figure 4.5. The F_1-values of five weighting schemes tested over R8 dataset with imbalanced setting using four different classifies

Figure 4.6. shows the classification performance of five term weighting schemes tested on the 20Newsgroups dataset using different classifiers. As shown in Figure 4.6., *tfidf* outperforms the supervised term weighting schemes in the 20Newsgroups dataset which has more complexity than the R8. In the 20Newsgroups dataset, it is observed that as the degree of class complexity raises, the number of subclusters increases. In this case, it can be concluded that category based metrics cannot clearly distinguish positive documents from the negative ones. Nonetheless, *tfidf* which has no attention to category labels creates a good contrast in the imbalanced case with high complexity. Among the supervised weighting schemes, SVM and k-NN perform well associated with one-sided metrics, whereas C4.5 and MultiNB are compatible with two-sided metrics to augment the minor class. This is similar to the previous observation which was obtained from R8 dataset. According to the both results in imbalanced cases, SVM in associated with the term weighting schemes based on one-sided metrics usually achieves good performance for minority class as shown in Figures 4.5. and 4.6.

According to our findings, we can conclude that supervised term weighting schemes usually provide better representation of data for the classifiers, with respect to minor class, on the imbalanced circumstance with less complexity (as shown in Figure 4.5.). Nonetheless, for high degree of complexity, *tfidf* seems a better term weighting scheme for the machine learning algorithms.

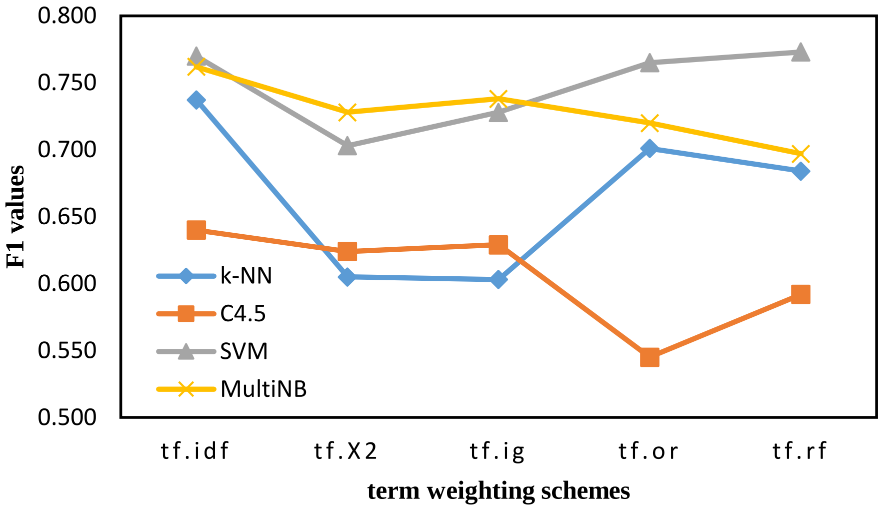


Figure 4.6. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with imbalanced setting using four different classifies

4.4.4.3. statistical Analysis

To determine the significance of the term weighting methods for each algorithm, we perform the ANOVA test on the F_1 values obtained from term weighting methods rather than t-test since it shows the significance of the results in more than 2 groups. As shown in Table 4.7., since the P-values of the tests are less than 0.05 for each case, there is a statistically significant difference between the mean F_1 values of methods at the 95.0% confidence level. Table 4.7. presents a multiple comparison of results to determine which algorithms differ significantly from others with respect to term weighting approaches. It can be observed that MultiNB and SVM significantly perform better than the others by using term weighting methods. At the Table 4.7., two and three homogenous groups are identified using columns of X's for R8 and 20Newsgroups datasets respectively. Within each column, the levels containing X's constitute groups which there are no statistically significant differences. To create a discrimination between F means, Fisher's least significant difference (LSD) procedure is employed here.

Table 4.7. ANOVA test for F_1-values obtained from 5 weighting methods for each algorithm for imbalanced cases

	R8 with P-Value = 0.0003			20 Newsgroup with P-Value = 0.0002			
Algorithms	F means	Homogeneous Groups		F means	Homogeneous Groups		
C4.5	0.8034	X		0.6060	X		
KNN	0.8793		X	0.6661		X	
MultiNB	0.8662		X	0.7290			X
SVM	0.8842		X	0.7478			X

5. IMBALANCED TEXT CATEGORIZATION BASED ON POSITIVE AND NEGATIVE TERM WEIGHTING APPROACH USING ADAPTIVE FRAMEWORK

In the previous chapter, the class imbalance problem was investigated on the binary classification domain where the diverse characteristics of class distribution are available. In addition, the efficiency and consistency of term wighting approach was observed on the learning process of various classifiers to deal with the class imbalance problem. In this chapter, the imbalanced data problem are surveyed on the domains in which more than two classes with imbalanced distributions exist (i.e. imbalanced multi-class classification). As mentioned before, in either cases the imbalanced distribution will affect the performance of machine learning algorithms and leads to weak results on the minority classes.

Multi-class classification presents a more general task of categorization and can be considered in divers applications of text mining community, e.g. opinion mining where three classes 'positive', 'negative' and 'neuter' are taken into account. The domain complexity is raised while the number of classes increase. In this case, the overlapping region grow up and consequently it leads to more misclassification errors. This situation can be worse when there is no balanced distribution of classes in the training set. In order to improve the classification performance, supervised term weighting approach explained in chapter 4., is used as a impressive strategy where data is represented in vector space model.

5.1. Proposed Positive and Negative Based Term Weighting Scheme

In general, the traditional function that was discussed before i.e. *idf* is known as asymmetric function in which it takes into account only the significance level of terms through the collection. In the supervised functions, asymmetric function only takes relevant terms (terms that appear mostly in the given category) into consideration such as rf and or functions, whereas symmetric function takes into account the irrelevant terms (terms that do not mostly appear in the given category) as well as relevant ones such as X^2 and ig functions. In this chapter, a symmetric function (Eq. 35) is proposed for global component of term weighting scheme based on two probabilities of relevant documents; i.e. $P(t_i|C_j)$ which is known as the probability of documents from category C_j where term t_i occurs at least once and $P(t_i|\bar{C}_j)$ which

Table 5.1. Fundamental information elements which are used in feature selection functions

	Containing term t_i	Not containing term t_i
Belonging to class C_j	$a_{i,j}$	$b_{i,j}$
Not belonging to class C_j	$c_{i,j}$	$d_{i,j}$

is considered as the probability of documents not form category C_j where term t_i occurs at least once. The main idea is to specify the degree of being relevant or non-relevant for a term with respect to each category where the negative documents outnumber the positive ones. To achieve this, the difference between two probabilities is computed as shown in Eq. 35. In fact, if $P(t_i|C_j)$ is bigger than $P(t_i|\bar{C}_j)$, which basically indicates that term t_i is relevant to category C_j, then the term is labelled as a positive term associated with category C_j and otherwise is assumed as negative. By dividing the difference into the summation of two probabilities, the normalized values of weights are obtained and the weights are transformed to [-1, 1] interval. It is worth to note that occurrences of each term in all categories are assumed as at least ξ times. ξ is a very low value and is used to be ensure that each of the probabilities can not be zero (Eqs. 36 and 37). In other words, if a term does not occur into a category, the document frequency of the term equals ξ. We named the proposed function as PNF^2 which is the abbreviation of *Positive Negative Features* and power of 2 symbolizes that equation is symmetric.

$$PNF^2(t_i, C_j) = \frac{P(t_i|C_j) - P(t_i|\bar{C}_j)}{P(t_i|C_j) + P(t_i|\bar{C}_j)} \tag{35}$$

To estimate the probabilities of Eq. 35, total 4 information elements shown in Table 5.1. are used. In Table 5.1., C_j denotes the class corresponding to the j^{th} category in the dataset; t_i is the i^{th} term in the vocabulary set; $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $d_{i,j}$ denote the document frequencies associated with the corresponding conditions. Therefore, the probabilities are calculated by using Eqs. 36 and 37:

$$P(t_i|C_j) = \frac{a_{i,j}}{a_{i,j} + b_{i,j}} \tag{36}$$

$$P(t_i|\bar{C}_j) = \frac{c_{i,j}}{c_{i,j} + d_{i,j}} \tag{37}$$

If PNF^2 is used as a global component of term weighting scheme, either positive or negative values are assigned to terms. When PNF^2 computes a negative value for a term, it shows not only the term is irrelevant for the given category but also it has a negative effect for that category as much as its absolute value. To eliminate the negative effect, the asymmetric form of the PNF^2 (Eq. 38) is defined as another alternative for the global component of term weighting scheme. In fact, we transform the PNF^2 to an asymmetric function abbreviated as PNF and compare it with the performance of PNF^2.

$$PNF = 1 + PNF^2 \tag{38}$$

PNF function does not produce any negative weights for terms and it assigns just low positive values to non-relevant terms instead of negative. For instance, if a term is assumed as the most irrelevant feature for a category, it has a value which is close to -1 by PNF^2 function while PNF assigns a value which is close to zero. Therefore, PNF function does not transform the trend of weighting to the negative space. This is plausible since the weighting scheme is employed for only training data. Test data is represented by *tf* values because it is assumed the category membership of test documents are not known.

5.2. Empirical Observation of Term Weighting and Feature Selection Approaches

Theoretical explanation of functions which have been used as a global component was presented previously and to demonstrate their behaviour, we try to apply them to a real example. First, the scores of the terms in the *grain* category of Reuters dataset are calculated by using two popular feature selection metrics i.e. ig, X^2 and proposed PNF metrics; then the scores of terms are sorted in descending order to select top 4 terms of each metric. Actually, *grain* is a minor category with 41 documents and Table 5.2. lists a, c and idf values of the selected top 4 terms. At this point, we want to emphasize the differences between feature selection and term weighting approaches. Feature selection means the identification of more representative terms to create low dimensional space. The selected features should represent the most number of documents in the data set. As a result, feature selection metrics do not take into account rare terms and assign low scores to them. This approach is highly different from the idf assumption presented in section 4.1. In fact, a term weighting scheme which uses idf as a global component gives higher score to terms with low document frequency. As can be

Table 5.2. The characteristics of top 4 terms selected by different manners for *grain* category in Reuters-21578 dataset

Terms	X^2			IG			PNF		
	a	c	idf	a	c	idf	a	c	idf
t_1	36	15	6.75	36	15	6.75	14	0	8.61
t_2	14	0	8.61	24	52	6.17	3	0	10.84
t_3	11	5	8.42	14	0	8.61	3	0	10.84
t_4	24	52	6.17	11	5	8.42	3	0	10.84

seen from Table 5.2., idf values of terms selected by PNF are higher than the idf values of other terms. The difference between the term weighting and feature selection approaches can be obviously proven with c values. Although, most of terms selected by ig and X^2 metrics have high document frequency in non-*grain* categories (i.e. high c values), terms selected by PNF metric have 0 values for the c parameter. Since use of feature selection metrics for category based weighting purposes has been preferred in the previous studies [10–12], we have to evaluate our proposed metrics by comparing with them. Therefore, idf, several feature selection metrics, and proposed PNF and PNF^2 metrics are employed as global component of term weighting scheme in the experiments. The last point is that, proposed PNF metric has closer approach to idf than the others but unlike idf, PNF is proposed for category based weighting.

5.3. The Properties of Term Weighting Approach Associated with Imbalance Problem

The previous section indicated the various viewpoints on the feature selection and term weighting approaches. Let's incorporate imbalance problem on the matter. We represent another empirical example to show what type of term weighting method can be more consistent in handling the class imbalance problem. To achieve this, we semantically choose 5 terms associated with *grain* class of Reuters dataset without any consideration about their document frequencies. In fact, we try to gather an eclectic group of the most relevant terms to *grain* class in terms of their meaning. These terms will most likely have influential role in distinguishing the *grain* class form the others.

At this point, we evaluate them by the 4 metrics explained in chapter 4., and PNF metric to see their evaluation values and document frequencies. As evaluation metrics yield different range of values, making comparison between them would not be accessible unless we applied

Table 5.3. The most negative, most positive and neutral cases based on document frequency elements for *grain* class of Reuters-21578 dataset

Information elements	Most negative	Most positive	Neutral
a	0	41	20.5
b	41	0	20.5
c	5444	0	2722
d	0	5444	2722

to them a normalization process. In order to normalize, the minimum and maximum values gained by each metric are calculated on the basis of the most negative, most positive and neutral cases. These cases are generated by certain group of document frequency elements. Table 5.3., shows these cases based on the fact that term t_i is known as the most positive term with respect to *grain* class while it occurs in all documents of the class and never be seen in documents of the other classes. The most negative and neutral cases are realized similarly as shown in Table 5.3.

The following equation is used to normalize the values of metrics:

$$norm = \frac{metric\ value - minimum\ value}{maximum\ -\ minimum} \tag{39}$$

It worth to note that the minimum values are computed for two-sided metrics (i.e. ig and X^2) by their neutral cases and for one-sided ones (i.e. or, rf, PNF) by the most negative cases due to their characteristics explained in previous chapter.

Table 5.4., shows the computed evaluation values for 5 most relevant terms to *grain* class. To compute the evaluation values, all metrics use the distribution of documents in associated with each of terms. In term weighting approach, the documents distribution are differently interpreted for each of balanced and imbalanced circumstances. Figure 5.1., depicts the difference between distribution of documents associated with a certain term in both of circumstances. To better understand the characteristics of term weighting approach in imbalanced circumstance, let first consider the balanced case.

According to the distribution of documents for a balanced case, term t_i is known as relevant (or positive) feature for target class when $a > c$. The growth in the difference between a and c leads to rise the relevancy power. On the other hand, when $a < c$ the term will be known as irrelevant (or negative) feature. It means that the term has occurred in irrelevant documents more than relevant ones, so it likely symbolizes the irrelevant documents better than positive ones. In fact, the occurrence of this term makes the test documents similar to negative training documents.

In conditions that the amounts of a and c are close to each other (i.e. $a \approx c$), the degree of relevancy will depend on b and d values. In this state, if $d > b$ the term will be likely relevant to the target class, and consequently for $d < b$, it may not be known as a good relevant term. However, based on the various distribution of documents, each metric will show different manner for generating weights [11].

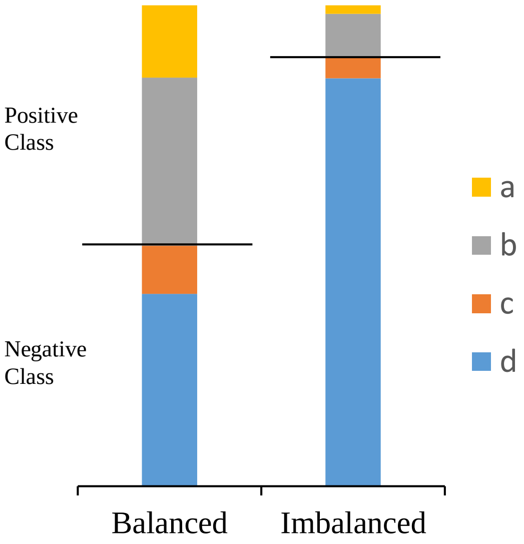


Figure 5.1. The distributions of documents for a given term in two different situations for binary classification

Let's return to our empirical example of imbalanced circumstance. In the most of imbalanced cases, the amount of a is not bigger than b and the amount of c is much less than d for a certain term with respect to the minor class.

From Table 5.4., the first point is that PNF generates more accumulated weights for the 5 terms than the others. It means that it produces weights which are either highest or close to each other since all 5 terms are the most relevant terms. To prove this, the standard deviation is calculated for each metric. The PNF by possessing 0.0061 and rf by 0.2249 standard deviations are the best and worst metric respectively.

According to the document frequency elements in Table 5.4., the following observations can be discussed in association with the term weighting methods:

Table 5.4. The more relevant terms to *grain* class in terms of semantic in conjunction with their evaluation elements

	Term evaluation metric values					Document frequency elements			
Term	PNF	or	rf	X^2	ig	a	b	c	d
crop	1.0000	0.9604	0.6778	0.3398	0.2950	14	27	0	5444
soil	1.0000	0.9024	0.1322	0.0242	0.0203	1	40	0	5444
harvest	0.9981	0.6242	0.3581	0.1546	0.1449	8	33	2	5442
feed	0.9925	0.5971	0.1322	0.0837	0.1045	7	34	7	5437
agriculture	0.9839	0.5954	0.0677	0.1800	0.3281	24	17	52	5392
Standard deviation	0.0061	0.1610	0.2249	0.1067	0.1162	-	-	-	-

- *The relationship between the low values of 'c' and metric values*

  As the number of documents that do not belong to minor class (e.g. *grain* class) are increasingly more than the belonging one, a term with low document frequency in majority class (i.e. small c values) can be taken into account by a term weighting method for distinguishing the minor class. Therefore, if rare terms in majority class are available, the discriminating point appears while we want to decide whether a high value of a is good or not for minor class. The PNF metric takes into account both viewpoints in which both frequent and infrequent terms make a same influence on the level of association with minor class (e.g. see *crop* and *soil* terms in Table 5.4.). At this point, feature selection and term weighting approaches contradict each other (see the *soil* term in Table 5.4.). By excluding the PNF, the odds ratio seems assign more reasonable weights for terms than the others. However, all 4 metrics tend to assign low weights to terms while the amount of a is weak.

- *The relationship between the high values of 'c' and metric values*

  By increasing the value of c the probability that a term is associated with the minor class is diminished. But, the degree of growth in c value cannot be equivalent to the degree of growth in a. Because the number and diversity of documents in the majority class outnumber the minor class in the imbalanced cases. It means that a rare term in minor class is no less important than a rare term in major class. Thus, a term weighting method should be able to generate a consistent weight according to degree of imbalance ratio. To clarify this, let consider the document frequency elements of terms *feed* and *agriculture* in Table 5.4. For term *feed* the number of documents belong to category *grain* which term *feed* occurs in them are similar to the ones which do not belong to category *grain* (i.e. $a = c$). In this case, the PNF metric assigns the highest weight to this term among the others; because 7 documents belonging to *grain* class are more significant than the other 7 documents which do not belong to *grain* class due to having 5444 documents. By increasing the c value from 7 to 52, the generated weight by PNF metric gradually dwindles to 0.9839, whereas this reduction is realized sharply in other metrics. Here is the place that we can perceive the influence of imbalance problem on the function of weighting metrics.

Another point that can be noted here is paying attention to the amount of elements b and d while imbalance problem is available. Actually, these two elements indicate the imbalance level of data for the weighting methods. Regardless of this, imbalance problem may impose a constraint on the function of various weighting methods and does not allow them to generate consistent weights.

Let's consider terms which may not semantically associate with minor class (i.e. *grain* class). For this case, we select 6 terms from different categories based on their meaning, so that there is no strong association with *grain* class. These terms are known as negative terms with respect to *grain* class. Table 5.5., shows the evaluation elements of the terms including their metric values and document frequency elements.

As mentioned earlier, increase in the amount of a leads to enhance the probability of being an association between the term and minor class. On the other hand, growth in c resists this probability. From Table 5.5., the value of PNF metric rises by growing the amount of a from 0 to 1 despite being growth in c value; because incrementing from 15 to 434 in c value cannot be significant through the 5444 documents. For term *sale*, a slightly reduction can be seen in the value of PNF, since the growth level of c relatively exceed the growth level of a. However, by dropping c to 400 in term *finance*, PNF value shoot up 0.4989. It means that the term *finance* is more likely close to *grain* class than the term *sale*, though it is still known as negative term for *grain* class. Because its value is still less than 0.5.

It worth to note that increase in the values of PNF from 0 to 0.5 is equivalent to move from most negative state to neutral state, and consequently rising the value to 1 indicates that the term begins to move from neutral state to most positive one. Therefore, the values which are close to 0.5 shows that a term possesses a neutral condition in distinguishing minor class from the major one. The same condition exists on the odds ratio metric.

Table 5.5. The less relevant terms to *grain* class in terms of semantic in conjunction with their evaluation elements

	Term evaluation metric values					Document frequency elements			
Term	PNF	or	rf	X^2	ig	a	b	c	d
sugar	0.0000	0.2333	0.0000	0.0000	0.0005	0	41	15	5429
oil	0.2342	0.4762	0.0004	0.0003	0.0047	1	40	434	5010
sale	0.2070	0.4713	0.0003	0.0009	0.0141	2	39	1017	4427
finance	0.4989	0.4999	0.0012	0.0000	0.0000	3	38	400	5044
profit	0.3625	0.4876	0.0007	0.0003	0.0037	4	37	934	4510
march	0.4595	0.4964	0.0010	0.0000	0.0003	5	36	781	4663

As another observation from Table 5.5., three other metrics i.e. rf, ig and X^2 cannot make significant discrimination between the six negative terms which have various documents frequencies.

5.4. Adaptive Framework

To best of our knowledge, when a weighting method is proposed, it is evaluated according to the classification performance of one or two machine learning algorithms. Each of the algorithms, which has the classification ability, use different techniques for learning process. However the size and characteristics of data may affect the learning process. For example, kNN algorithm may not be a suitable classifier for high dimensional space, or Multinomial Naive Bayes can not work with the negative values of term weights. At this point, to analyze the effect of different weighting methods on the imbalanced data classification problem, we need a learning algorithm which is associated with term weighting approach. In other words, we need a simple classifier which has the properties such as (1) term weighting scheme is the most effective factor in the learning process (2) it does not need huge effort for preprocessing and consequently performs fast in a high dimensional space, (3) it guarantees the creation of equivalent circumstances for different weighting schemes.

In the proposed algorithm, after representing documents in vector space model and applying a term weighting scheme, learning process is realized by combining training document vectors $\vec{d}$ into a vector $\vec{c}_j$ for each category. The vector $\vec{c}_j$ is computed for category C_j by dot dividing of two vectors as Eq. 40:

$$\vec{c}_j = \frac{1}{\vec{a}_j} \sum_{\vec{d} \in C_j} \vec{d} \tag{40}$$

In Eq. 40 the $\vec{a}_j$ is the vector yielded from the document frequency of terms with respect to category C_j (as shown in Table 5.1.) and $\sum_{\vec{d} \in C_j} \vec{d}$ yields the summation of document vectors which belong to category C_j. Consequently, the set of $\vec{c}_j$ vectors which are computed for each category, represent the learned model. This model is used to classify document d^t which has never seen before. This test document is represented by the vector $\vec{d}^t$ which has only tf values as weights. In order to classify the test document, cosine similarity is computed between two vectors such as $\vec{d}^t$ and each of $\vec{c}_j$. Finally, the vector $\vec{d}^t$ is assigned to the category which has the highest similarity with $\vec{d}^t$ as indicated in Eq. 41.

$$F(\vec{d}^t) = \arg\max_{c_j \in C} \frac{\vec{c}_j}{||\vec{c}_j||} . \frac{\vec{d}^t}{||\vec{d}^t||} \tag{41}$$

It is worth to note that the normalization of vector $\vec{d}^t$ by its length is not considered since it cannot have any impact on the $argmax$. We also omit the normalization of vector $\vec{c}_j$ because its values have been already computed based on normalized values by the length of documents in the term weighting phase.

5.5. Experiments

In this chapter, all experiments were conducted on two different benchmarks such as Reuters-21578 and WebKB, because they are popular and well-known datasets in the field of text classification. To make clear comparisons between the metrics and corpus, we have to control the imbalance ratio and observe the effects of the term weighting methods on both minor and major categories, separately. For this purpose, a subset of each corpus was selected and two sets of experiments were carried out on these subsets in combination with the adaptive framework. The performance of classifications are assessed with F-measure values and ANOVA test is employed on the results for an statistical evaluation.

5.5.1. Experimental Setup

The Reuters-21578 dataset has been widely used in text classification researches as an imbalanced collection [10, 12, 14, 35, 57]. The R8 version of Reuters dataset which was used in the experiments [67], consists of two major categories called as *earn* and *acq* with almost 52% and 30% class distributions respectively and 6 minor categories with almost 3% class distributions. The distribution value of each class means the percentage of documents belong to the specified class in the whole collection (Table 5.6.).

Table 5.6. Properties of Ruters-21578 dataset (R8) with its class distributions

Class name	# of train docs	Class distribution %	# of test docs	Total # of docs
acq	1596	29	696	2292
earn	2840	52	1083	3923
crude	253	5	121	374
grain	41	1	10	51
interest	190	3	81	271
money-fx	206	4	87	293
ship	108	2	36	144
trade	251	5	75	326
Total/Average	**5485**	**100**	**2189**	**7674**

WebKB dataset consists of 4 categories of web pages (*project, course, faculty* and *student*) collected from computer science departments of four universities by CMU text learning group [67]. This dataset contains two minor categories called as *project* and *course* with almost 10% and 20% class distributions respectively and two major categories with 30% and 40% class distributions (Table 5.7.).

Table 5.7. Properties of WebKB dataset with its class distributions

Class name	# of train docs	Class distribution %	# of test docs	Total # of docs
project	336	12	168	504
course	620	22	310	930
faculty	750	27	374	1124
student	1097	39	544	1641
Total/Average	**2803**	**100**	**1396**	**4199**

For both datasets, experiments were performed on the original training and test sets obtained from benchmarks [67]. Standard text preprocessing steps such as removing the 524 SMART stop-words, punctuation removal and Porter's Stemmer algorithm were applied on them. All experiments were carried out in the adaptive framework represented in section 5.4. and since we aimed to investigate the influence of only term weighting schemes, we did not select any terms or features, actually all features were used in classification.

Precision, Recall and F-measure were used to evaluate the performance of classification as explained in chapter 3. In multi-class classification, precision is the fraction of documents assigned to class C_j which are actually about class C_j and recall is the fraction of documents in class C_j classified correctly. While precision estimates local accuracy, recall estimates global one. As classification systems try to maximize both precision and recall values, F-measure is used as more conservative estimate of performance in text classification task [12, 57]. Its Popular form represented in chapter 3., is used in the experiments.

Macro and micro averages of F-measure values are also used to compare the overall performance of different methods. Macro-averaged F-measure value is obtained by averaging individual F-measure values of categories and micro-averaged F-measure value is computed by summing the values of TP, FP and FN obtained from individual categories [55].

5.5.2. Experimental Results and Discussion

Actually, F-measure value combines local and global accuracies of the classification and produces single value to make easy comparisons. Achieved F-measure values for the different weighting methods employed on Reuters-21578 benchmark are listed in Table 5.8. According to Table 5.8., $tf.PNF$ term weighting method consistently outperforms all other methods for all categories except one case (i.e. *trade* class) in which $tf.rf$ weighting scheme performs better than $tf.PNF$ with less than 1% difference. The results obtained from $tf.PNF^2$ can be competitive with the other methods. The $tf.PNF$ weighting scheme, which eliminates the negative impact existed in $tf.PNF^2$, significantly improves the performance of the classification. In fact, the proposed asymmetric function (PNF) provides better results than the symmetric one (PNF^2) as a global component of term weighting scheme.

The superiority of $tf.PNF$ scheme can be obviously seen by micro and macro averaged F-measure values. As macro-averaged value is computed based on individual categories, it has the same impact on all categories without any consideration over class distribution; whereas micro average tends to dominant categories which have more instances. Thus, macro average yields the results that better reflect the performance of methods on the imbalanced data. It can be also observed the $tf.PNF$ method performs well on the minor classes as well as the major ones, since it possesses the high micro and macro averaged F-measure values at the same time (as shown in Table 5.8.). Another point is that the $tfidf$ weighting scheme cannot provide a good distinction between categories and consequently performs weekly on the whole categories.

Table 5.8. The F-measure values of different term weighting schemes for Reuters-21578 dataset

Categories	The term weighting schemes						
	$tf.idf$	$tf.X^2$	$tf.ig$	$tf.or$	$tf.rf$	$tf.PNF^2$	$tf.PNF$
earn	0.771	0.512	0.845	0.945	0.981	0.950	**0.981**
acq	0.450	0.654	0.831	0.921	0.957	0.952	**0.961**
crude	0.698	0.896	0.887	0.867	0.902	0.835	**0.945**
trade	0.542	0.867	0.886	0.771	**0.906**	0.802	**0.898**
money-fix	0.646	0.789	0.781	0.798	0.719	0.834	**0.868**
interest	0.754	0.792	0.779	0.852	0.776	0.838	**0.881**
ship	0.539	0.831	0.679	0.781	0.806	0.794	**0.845**
grain	0.667	0.889	0.889	0.900	0.800	0.750	**0.900**
Macro average	0.633	0.779	0.822	0.854	0.856	0.844	**0.910**
Micro average	0.687	0.639	0.836	0.912	0.945	0.925	**0.958**

Table 5.9., lists the results of the same experiments on the WebKB benchmark. In this benchmark, the superiority of $tf.PNF^2$ and $tf.PNF$ can be observed among the other methods. Although the $tf.PNF$ is known as the best weighting scheme by possessing the highest micro and macro averaged F-measure values, $tf.PNF^2$ gives better results for minor categories. It can be also observed that the performance $tf.ig$, $tf.X^2$ and $tf.rf$ are degraded in contrast with their previous results on the Reuters benchmark and cannot keep their relative goodness. At this point, it can be said that they cannot perform well on different imbalanced circumstances and may not yield consistent results. Conversely, $tf.PNF$, $tf.PNF^2$ and $tf.or$ can provide more reliable results since they can make a relative minimum range of fluctuation in their results.

According to the achieved results from two benchmarks (Tables 5.8. and 5.9.), the proposed two functions as a global component of term weighting scheme yield better results than the others. Moreover, the category based term weighting schemes outperform the traditional $tfidf$ in most cases. In other words, $tfidf$ cannot make any clear distinction between documents of the different classes in multi-class classification task. As mentioned in section 5.2.,

Table 5.9. The F-measure values of different term weighting schemes for WebKB dataset

Categories	The term weighting schemes						
	$tf.idf$	$tf.X^2$	$tf.ig$	$tf.or$	$tf.rf$	$tf.PNF^2$	$tf.PNF$
student	0.636	0.587	0.588	0.636	0.735	0.705	**0.852**
faculty	0.372	0.236	0.224	0.688	0.673	0.750	**0.757**
course	0.608	0.014	0.006	0.859	0.662	**0.887**	0.860
project	0.088	0.403	0.424	0.649	0.443	**0.649**	0.617
Macro average	0.426	0.310	0.311	0.708	0.628	0.747	**0.772**
Micro average	0.549	0.452	0.454	0.703	0.683	0.749	**0.805**

ig and X^2 are successful for feature selection task [18] but they cannot consistently perform well as a global component of term weighting scheme in imbalanced text classification.

To clarify the reason of weak F-measure values gained by some weighting methods, we represent the macro average of precision and recall for each term weighting method. Table 5.10. shows these macro averaged values for the both benchmarks. As can be seen, despite the $tfidf$ yields good precision, it does not constitute a reasonable tradeoff between precision and recall. This issue happens on the $tf.ig$ and $tf.X^2$. Actually, they cannot achieve reasonable recall values as much as precision values. On the contrary, the asymmetric category based functions (i.e. or, rf and PNF) provide an appropriate tradeoff between precision and recall. This is robust for the PNF function by possessing the highest tradeoff among the others. As a result, although ig and X^2 are successful for feature selection task [18], they cannot perform well as global component of term weighting scheme in imbalanced text classification. As another observation, unlike ig and X^2 which cannot yield high recall values, PNF^2 as a symmetric function performs better and provides high global accuracy (recall) for categories, as shown in Table 5.10.

5.5.3. Statistical Analysis

To determine the statistical significance of the results, we performed ANOVA test on the F-measure values gained by the methods for the categories. ANOVA (analysis of variance) provides a statistical test to see whether or not the macro-averaged F-measure values belong to several groups (shown in Tables 5.8. and 5.9.) are equal, and therefore generalizes the t-test to more than two groups. Tables 5.11. and 5.12. represent the results of ANOVA test for Reuters-21578 and WebKB benchmarks respectively. As it is shown in Tables 5.11. and

Table 5.10. The macro average of precision and recall for different term weighting schemes

The term weighting schemes	Macro-averaged values of precision and recall			
	Reuters-21578		WebKB	
	Precision	Recall	Precision	Recall
$tf.idf$	0.870	0.551	0.671	0.429
$tf.ig$	0.858	0.764	0.489	0.353
$tf.X^2$	0.903	0.772	0.758	0.352
$tf.or$	0.835	0.890	0.749	0.749
$tf.rf$	0.873	0.853	0.776	0.592
$tf.PNF^2$	0.811	$\underline{0.897}$	0.763	**0.794**
$tf.PNF$	**0.918**	**0.903**	**0.823**	0.746

5.12., since the P-values of the tests are less than 0.05 for each case (P-value equals 0.0000 for Reuters and 0.0028 for WebKB), there are statistically significant differences between the macro-averaged F-measure values of the different schemes at the 95.0% confidence level. Tables 5.11. and 5.12. also provide a multiple comparison between the results to determine which term weighting schemes differ significantly from the others. For both benchmarks, 4 homogeneous groups were identified using columns of X's (Tables 5.11. and 5.12.). Within each column, the weighting methods containing X's constitute a group which there is no statistically significant difference between the means. To create discrimination between means, Fisher's least significant difference (LSD) procedure was employed. It can be observed that $tf.PNF$ alone performs significantly better than the others in the Reuters benchmark (as shown in Table 5.11.) while there is no statistically significant difference between macro averages of the other 4 term weighting schemes ($tf.PNF^2$, $tf.X^2$, $tf.or$ and $tf.rf$). In the WebKB benchmark, the three weighting schemes ($tf.PNF$, $tf.PNF^2$ and $tf.or$) are placed in a group that significantly provides the best performance in comparison with the others (as shown in Table 5.12.).

To indicate the reliability of each macro-averaged F-measure value, the confidence interval of each one was estimated at the 95.0% confidence level. The intervals displayed in Tables 5.11. and 5.12., are based on Fisher's least significant difference (LSD) procedure. They were computed in such a way that if two means are the same, their intervals will overlap 95.0% of the time. We also computed standard deviation for each mean in order to show how much dispersion from the mean exists. In fact, a macro-averaged F-measure value with low standard deviation is more reliable. In Reuters benchmark (Table 5.11.), $tf.PNF$ presents the highest confidence interval as well as the lowest standard deviation. It shows that

Table 5.11. ANOVA test on the F-measure values obtained from the term weighting methods for Reuters-21578 dataset with P-Value = 0.0000 and 95.0 percent LSD intervals

Term weighting scheme	Macro average	Lower limit	Upper limit	Standard deviation	Homogeneous group
$tf.idf$	0.633	0.5883	0.6784	0.1133	X
$tf.ig$	0.779	0.7337	0.8238	0.1328	X
$tf.or$	0.854	0.8093	0.8994	0.0659	X
$tf.rf$	0.856	0.8108	0.9009	0.0935	X
$tf.X^2$	0.822	0.7771	0.8672	0.0731	X
$tf.PNF^2$	0.844	0.7993	0.8894	0.0719	X
$tf.PNF$	**0.910**	**0.8647**	**0.9548**	**0.0475**	X

Table 5.12. ANOVA test on the F-measure values obtained from the term weighting methods for WebKB dataset with P-Value = 0.0028 and 95.0 percent LSD intervals

Term weighting scheme	Macro average	Lower limit	Upper limit	Standard deviation	Homogeneous group
$tf.ig$	0.310	0.1745	0.4455	0.2439	X
$tf.X^2$	0.311	0.1750	0.4460	0.2517	X
$tf.idf$	0.426	0.2905	0.5615	0.2545	X
$tf.rf$	0.628	0.4928	0.7637	0.1276	X
$tf.or$	0.708	0.5725	0.8435	0.1031	X
$tf.PNF^2$	0.747	0.6123	0.8832	**0.1016**	X
$tf.PNF$	**0.772**	**0.6360**	**0.9070**	0.1131	X

$tf.PNF$ is the most reliable method among the others. In the WebKB benchmark, $tf.PNF$ has the highest confidence interval and $tf.PNF^2$ contains the lowest standard deviation. It can be concluded that both proposed functions (i.e. PNF, PNF^2) as global components of term weighting scheme can provide confidant results as well as high F-measure values.

ANOVA test takes into account the average of F-measure values. However, average cannot always demonstrate a good perspective of the performance. Hence, we applied another statistical test to results to analyze the performance of term weighting methods from different perspective. The Kruskal-Wallis test check the null hypothesis that the medians of F-measure values within each of the 7 levels of weighting methods are the same. The data from all levels is first combined and ranked from smallest to largest. The average rank is then computed for the data at each level. Since the P-value is less than 0.05 for each of benchmarks, there is a statistically significant difference between the medians at the 95.0% confidence level. Tables

5.13. and 5.14. represent the results of the test for Reuters and WebKB datasets respectively. As shown in both tables, PNF possesses the highest average rank among the other methods. To determine which medians are significantly different from which others, we present box-and-whisker plots by Figures 5.4. and 5.5. for Reuters and WenKB datasets respectively. From both of box-whisker-plots, the following observation can be discussed:

1. *The Overall Range and Interquartile Range*

   The overall range is the distance between the smallest F-measure value and the largest including any outliers, whereas interquartile range goes from the lower quartile (first vertical line in the box) to the upper quartile (third vertical line). Actually, it shows the range of the 50% of data. Figure 5.2. depicts the different parts of the box-whisker-plot. According to the both types of range, $tf.PNF$ method has a smallest range among the other methods in the results of Reuters dataset (Figure 5.4.). This indicates that the consistency of F-measure values is the best in the $tf.PNF$ method .

   On the other hand, it can be observed that not only $tf.PNF$ possess the highest median but also the lower quartile of that exceeds the medians of all other methods in both of datasets (Figures 5.4. and 5.5.). It means that more than 75% of F-measure values in the $tf.PNF$ are greater than 50% of F-measure values in the other methods. Although the $tf.PNF^2$ is not more superior to $tf.or$ and $tf.rf$ methods in Reuters dataset, the consistency level of its F-measure values is greater than them due to having less range. Nonetheless, $tf.PNF^2$ succeeded to yield better results than $tf.or$ and $tf.rf$ in WebKB dataset with almost the same consistency level (Figure 5.5.).

2. *The skewness pattern of F-measure values*

   Box-whisker-plot also provide information about the distribution of data. In general three types of distribution can be considered in this plot, as shown in Figure 5.3. In the first one which is known as symmetric, the observations are evenly split at the median; it shows that data follows on from a normal distribution. If most of the observations are biased towards the vicinity of th minimum value, the distribution is skewed right; otherwise distribution is skewed left (as shown in Figure 5.3.).

   According to the results of Reuters dataset, Figure 5.4., all methods approximately produce a symmetric distribution of F-measure values except $tf.idf$, tfX^2, and $tf.ig$ methods which are extensively skewed left. It shows that there is no coherent distribution at least in 50% of F-measure vales, and consequently they cannot be known

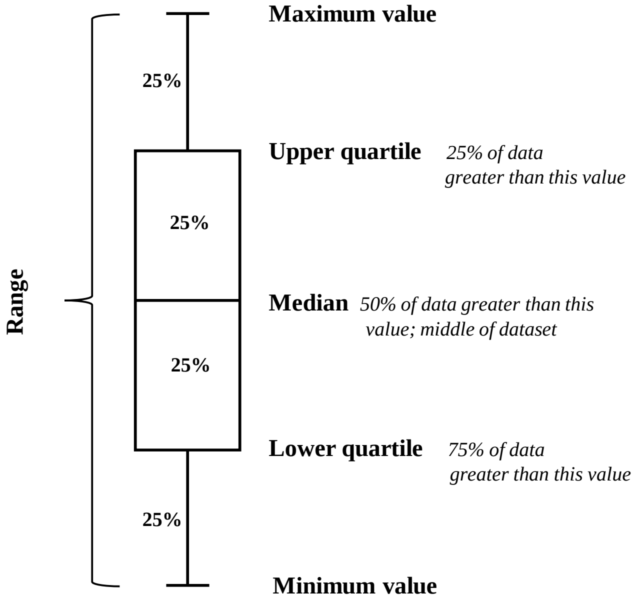

Maximum value
25%
Upper quartile
25% of data greater than this value
25%
Range
Median
50% of data greater than this value; middle of dataset
25%
Lower quartile
75% of data greater than this value
25%
Minimum value

Figure 5.2. Box-whisker-plot description

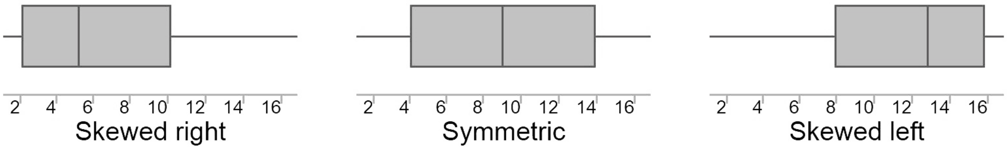


Figure 5.3. Box-whisker-plot types

as consistent results. In addition, it can be observed that the average of F-measure values which are indicated by red plus signs, nearly coincide with the medians in the four methods $tf.or$, $tf.rf$, $tf.PNF^2$, and $tf.PNF$. This conformity indicates the solidarity of their F-measure vales. The $tf.PNF^2$ and $tf.PNF$, however, are the best methods because they have smallest range of values (Figure 5.4.).

The results obtained from WebKB dataset are not as symmetric as the results of Reuters dataset in most of methods. Nevertheless, the $tf.PNF^2$ and $tf.PNF$ again perform better than the others due to having the highest mean and median with small range of values.

According to the both of results shown in Figures 5.4. and 5.5., it can be concluded that the $tf.PNF$ in conjunction with $tf.PNF^2$ yield the most symmetric performance for imbalanced datasets as well as smallest range of F-measure values. Based on these two factors,

Table 5.13. Kruskal-Wallis Test for F-measure values gained by 7 term weighting methods on Reuters dataset. Test statistic = 23.6714 P-Value = 0.0006

Term weighting method	Category size	Average rank
$tf.idf$	8	6.8125
$tf.ig$	8	23.8125
$tf.or$	8	32.8125
$tf.PNF$	8	43.8125
$tf.PNF^2$	8	30.75
$tf.rf$	8	34.375
$tf.X^2$	8	27.125

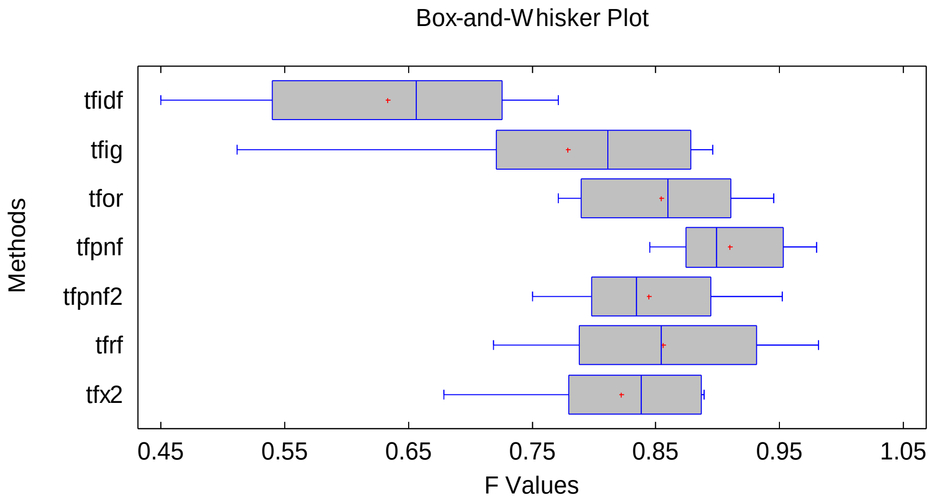


Figure 5.4. Box-whisker-plot for F-measure values gained by the term weighting methods on the Reuters dataset

we can state that the performance gained by $tf.PNF$ and $tf.PNF^2$ are the most consistent results through the others.

Table 5.14. Kruskal-Wallis Test for F-measure values gained by 7 term weighting methods on the WebKB dataset. Test statistic = 19.1133 P-Value = 0.0039

Term weighting method	Category size	Average rank
$tf.X^2$	4	6
$tf.idf$	4	8.875
$tf.ig$	4	6
$tf.or$	4	19.25
$t.fPNF$	4	22.25
$tf.PNF^2$	4	22.125
$tf.rf$	4	17

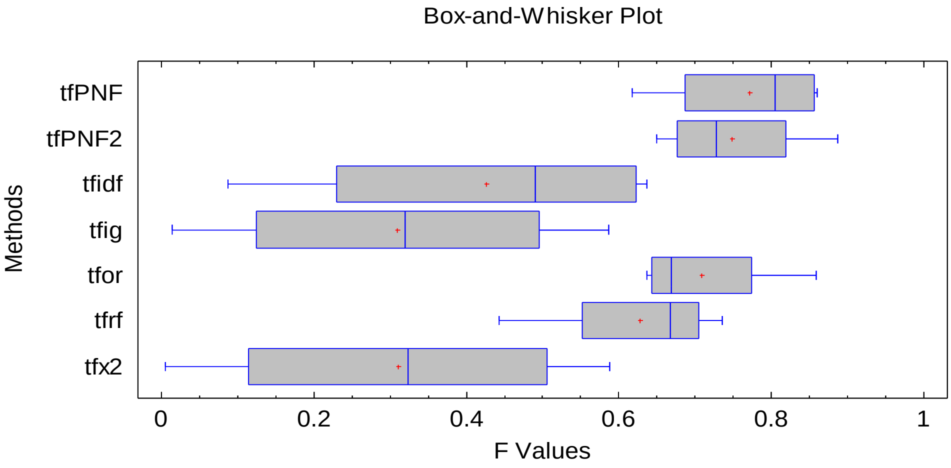


Figure 5.5. Box-whisker-plot for F-measure values gained by the term weighting methods on the WebKB dataset

6. CONCLUSION

In this thesis, we have attempted to resolve the class imbalanced problem effects on the performance of machine learning algorithms in the realm of text classification. After investigating term weighting methods on the classification of imbalance texts, we presented our own solutions to deal with this problem.

The contributions of this thesis can be summarized as follows:

- In this study, the effects of two kinds of supervised term weighting schemes (one-sided and two-sided based) were investigated on the balanced and imbalanced texts with different degrees of complexity. $Tfidf$ was used as a base line to evaluate the effect of supervised weighting methods on the imbalanced texts. We evaluated the performance of each weighting method by using four different machine learning algorithms (SVM, k-NN, MultiNB and C4.5). Actually, the appropriateness of weighting methods in associated with machine learning algorithms were studied here. To investigate the class imbalance problem, we generated datasets with two different complexity levels such as balanced and imbalanced cases. According to our findings, in the balanced cases, almost all weighting methods had a little impact on the performance of classifiers. Nonetheless, it can be seen that the supervised term weighting approach does not possess any effective superiority to *tfidf*. Furthermore, it was observed that one-sided based term weighting schemes outperform the two-sided based ones in the most of balanced cases.

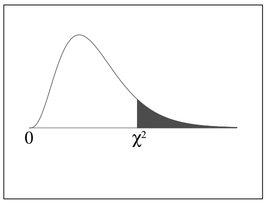
  In the imbalanced cases, it is realized that all four classifiers are sensible to the term weighting methods. Regardless of *tfidf*, one-sided term weighting methods are better approach for SVM and k-NN algorithms, while two-sided methods are the best choice for MultiNB and C4.5. According to our results, it can be concluded that supervised term weighting methods based on one-sided term selection metrics are the best choice for SVM in the imbalanced datasets and k-NN algorithm usually perform well with *tfidf*. It should be also noted that MultiNB classifier presents interesting results on the imbalanced cases. As another finding, although supervised methods cannot constantly retain their superiority to *tfidf* on the more complex imbalanced datasets, they can provide effective results for classification algorithms.

- In the second stage of this study, we tackled the class imbalance problem by category based term weighting approach in combination with an adaptive framework. Two functions named as PNF^2 and PNF were proposed as a global component of term weighting scheme based on two probabilities of the relevant documents frequency. Furthermore, an adaptive framework was proposed which can evaluate the strength of term weighting schemes over imbalanced texts. In fact, by a learning process associated with term weighting schemes, a simple model was generated for each category.

  For assessment of the proposed term weighting schemes, a comparison was made with several methods by using two benchmarks such as Reuters-21578 and WebKB. According to our findings, the $tf.PNF$ term weighting scheme performs the best in all experiments and can provide the best tradeoff between precision and recall. Despite the wide range of fluctuation in the results of $tf.ig$ and $tf.X^2$, $tf.PNF^2$ as a symmetric method achieves more expectable results with high F-measure values. Additionally, the asymmetric functions (i.e. or, rf and PNF) consistently perform better than the symmetric ones (i.e. ig and X^2), but PNF^2 presents competitive results in contrast with or, rf functions. As PNF^2 is not constantly superior to or and rf, the consistency of their results is the best in most of experiments. As a result, the PNF and PNF^2 functions as a global component of term weighting scheme are recommended for imbalanced classification task.

A APPENDIX, CHI-SQUARE TABLE

Chi-Square Distribution Table



The shaded area is equal to α for $\chi^2 = \chi^2_\alpha$.

df	$\chi^2_{.995}$	$\chi^2_{.990}$	$\chi^2_{.975}$	$\chi^2_{.950}$	$\chi^2_{.900}$	$\chi^2_{.100}$	$\chi^2_{.050}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

B APPENDIX, Z-TABLE

Standard Normal Probabilities

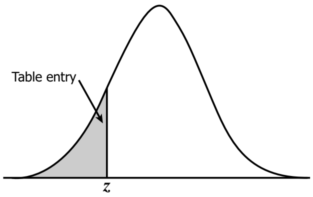


Table entry for z is the area under the standard normal curve to the left of z.

z	**.00**	**.01**	**.02**	**.03**	**.04**	**.05**	**.06**	**.07**	**.08**	**.09**
−3.4	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0002
−3.3	.0005	.0005	.0005	.0004	.0004	.0004	.0004	.0004	.0004	.0003
−3.2	.0007	.0007	.0006	.0006	.0006	.0006	.0006	.0005	.0005	.0005
−3.1	.0010	.0009	.0009	.0009	.0008	.0008	.0008	.0008	.0007	.0007
−3.0	.0013	.0013	.0013	.0012	.0012	.0011	.0011	.0011	.0010	.0010
−2.9	.0019	.0018	.0018	.0017	.0016	.0016	.0015	.0015	.0014	.0014
−2.8	.0026	.0025	.0024	.0023	.0023	.0022	.0021	.0021	.0020	.0019
−2.7	.0035	.0034	.0033	.0032	.0031	.0030	.0029	.0028	.0027	.0026
−2.6	.0047	.0045	.0044	.0043	.0041	.0040	.0039	.0038	.0037	.0036
−2.5	.0062	.0060	.0059	.0057	.0055	.0054	.0052	.0051	.0049	.0048
−2.4	.0082	.0080	.0078	.0075	.0073	.0071	.0069	.0068	.0066	.0064
−2.3	.0107	.0104	.0102	.0099	.0096	.0094	.0091	.0089	.0087	.0084
−2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
−2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
−2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
−1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
−1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
−1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
−1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
−1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
−1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
−1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
−1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
−1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
−1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
−0.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
−0.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
−0.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
−0.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
−0.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
−0.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
−0.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
−0.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
−0.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
−0.0	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641

Standard Normal Probabilities

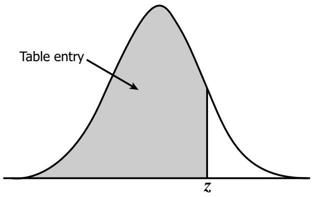


Table entry for z is the area under the standard normal curve to the left of z.

z	**.00**	**.01**	**.02**	**.03**	**.04**	**.05**	**.06**	**.07**	**.08**	**.09**
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990
3.1	.9990	.9991	.9991	.9991	.9992	.9992	.9992	.9992	.9993	.9993
3.2	.9993	.9993	.9994	.9994	.9994	.9994	.9994	.9995	.9995	.9995
3.3	.9995	.9995	.9995	.9996	.9996	.9996	.9996	.9996	.9996	.9997
3.4	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9998

C APPENDIX, PUBLISHED COPY OF PROCEEDING FROM THE THESIS

INVESTIGATION OF TERM WEIGHTING SCHEMES IN CLASSIFICATION OF IMBALANCED TEXTS

Behzad Naderalvojoud, Ahmet Selman Bozkir and Ebru Akcapinar Sezer
Hacettepe University, Computer Engineering Department, Ankara, Turkey

ABSTRACT

Class imbalance problem in data, plays a critical role in use of machine learning methods for text classification since feature selection methods expect homogeneous distribution as well as machine learning methods. This study investigates two different kinds of feature selection metrics (one-sided and two-sided) as a global component of term weighting schemes (called as *tffs*) in scenarios where different complexities and imbalance ratios are available. Traditional term weighting approach (*tfidf*) is employed as a base line to evaluate the effects of *tffs* weighting. In fact, this study aims to present which kind of weighting schemes are suitable for which machine learning algorithms on different imbalanced cases. Four classification algorithms are used to indicate the effects of term weighting schemes on the imbalanced datasets. According to our findings, regardless of *tfidf,* term weighting methods based on one-sided feature selection metrics are better approaches for SVM and k-NN algorithms while two-sided based term weighting methods are the best choice for MultiNB and C4.5 on the imbalanced texts. As a result, the use of term weighting methods based on one-sided feature selection metrics is recommended for SVM and *tfidf* is suitable weighting method for k-NN algorithm in text classification tasks.

KEYWORDS

Class imbalance problem, machine learning, text classification, term weighting, feature selection

1. INTRODUCTION

In machine learning, text classification is a supervised learning task which can predict the categories of unlabeled documents based on an inductive model learned from labeled documents. The common machine learning algorithms which have been used for this purpose include support vector machine (SVM), k-nearest neighbor (k-NN), naïve Bayesian (NB), neural networks (NN), decision trees (C4.5) and Rocchio (Ogura et al, 2011; Liu et al, 2009; Chawla et al, 2011). Binary classification by machine learning algorithms is usually performed based on a fundamental assumption that the distributions of two classes should be close to each other. In other words, there should be as many positive examples as negative ones (Chawla et al, 2004). This mentioned fundamental requirement cannot be always met since there are many imbalanced datasets relying on real world examples, (e.g. cancer detection, network intrusion detection, credit card fraud detection, oil-spill detection). At this point, classifiers generally present weak performance while the dominant class is well classified; the examples belonging to the minor class tend to be misclassified. Nonetheless, the aim of these classifiers is to generate a model that best fits the training data with minimum error rate. Furthermore, they consider the global quantities in generating the model.

Class imbalance problem occurs in text classification tasks when the numbers of positive samples are significantly lower than negative ones. There are other domain characteristics that aggravate the problem such as (1) class complexity (2) size of training set and (3) subclusters (Japkowicz and Stephen, 2002). In typical binary imbalanced text classification, the positive class consists of the documents that belong to one subject and negative class consists of all other remaining items. Thus, increment in the number of negative class samples leads to growth of class complexity. In this case, the positive class can be formed as a cluster while the negative class cannot. Therefore, raising the degree of imbalance by incrementing the negative documents with different subjects causes aggravation of class distribution and growing the number of subclusters. In order to generate a classification model with low generalization error for minor class, existence of adequate number of samples in the training data set is crucial. Therefore, the datasets which have insufficient number of positive samples tend to be misclassified since the classification algorithms aim to build models which have generalization capability.

Class imbalance problem also exists in the multi classification schemas when one class is assumed as a target category (positive class or minor class) and the union of the other classes are considered as negative class (majority class) (Ogura et al, 2011). In this case, most of the machine learning methods are often biased to the majority class and ignore the minor class since they attempt to minimize the global parameters such as total error rate and do not take the class distribution into consideration (Japkowicz and Stephen, 2002).

An inevitable stage in the text classification task is representing the textual documents in a realizable form for any classifier. As a well-known method, vector space model (VSM) is known as a text representation model which makes a transformation from content of the natural language texts into a vector of term space (Salton and Buckley, 1988). In this model, assigning a weight for each term is effective to represent data, since the importance of each term in different documents can vary. This issue can be taken into consideration in the imbalanced cases. Thus, *tfidf* as a basic term weighting scheme is used in text classification tasks. This method belongs to information retrieval field and does not need any prior information about the categories; hence it is called as unsupervised term weighting approach (Lan et al, 2009). In the text classification, since the labeled documents are available, this information can be used as a global parameter in the term weighting scheme. Thus, the term weighting approaches which use the prior known information, are called supervised approaches in the literature (Debole and Sebastiani, 2004).

The common strategies proposed in the class imbalance problem literature are addressed at data and algorithmic level. At algorithmic level, the employed strategies include determining the decision threshold (Chen et al, 2006), adjusting the probabilistic estimate at the information gain and Bayesian based methods such as decision tree and naïve Bayes respectively (Kibriya et al, 2005). At data level, the proposed approaches include the different forms of resampling methods (Chawla et al, 2004) and instance weighting schemes (Liu et al, 2009). In this study we focus on the data level approaches. The first approach is resampling data in via under sampling the majority class and over sampling the minority class. Moreover, Liu investigated several resampling techniques in the realm of imbalanced text classification (2004). Chawla et al. proposed a synthetic technique for over sampling the minority class samples named SMOTE (2011).

Another approach at the data level is using instance weighting methods in representation of data. In their study, Debole and Sebastiani, replaced the *idf* by category-based feature selection metrics (i.e. chi square, information gain and gain ratio) that had been used in the term selection phase (2004). They employed SVM as learning method with Reuters-21578 and showed supervised term weighting cannot be consistently superior to *tfidf*. In another study, (Lan et al, 2009) proposed a supervised term weighting method, *tf.rf*, based on distribution of relevant documents in the collection. Their proposed method was providing better performance than the other weighting schemes based on information theory and statistical metrics in combination with SVM and k-NN algorithms. On the other hand, a simple probability based term weighting scheme was proposed to better distinguish documents in minor categories (Liu et al, 2009). Moreover, Sun et al. provided a comparative study on the effectiveness of resampling and instance weighting strategies using SVM (2009).

To best of our knowledge, in most of the studies the proposed solutions for dispelling the class imbalance problem were evaluated by using one or two classifiers (especially by SVM). In this study, we try to survey the instance weighting strategy in combination with four algorithms which work based on four different approaches. Thus, the following objectives will be addressed in this study:

- Investigation of the supervised and unsupervised weighting approaches on imbalanced datasets as well as compatibility of each weighting method with machine learning algorithms.
- Comparing the effect of two-sided feature selection metrics (metrics that consider the negative, non-relevant, features as well as the positive, relevant, ones) with one-sided metrics (metrics that take only the positive features into consideration) at the term weighting perspective.

In fact, we try to discuss which kind of feature selection metrics (as a component of term weighting scheme) can be beneficial to represent imbalanced data and which term weighting schemes are suitable for which machine learning algorithms. For this purpose, four different classifiers (SVM, k-NN, NB and C4.5) are employed in the experiments. The main reason of this selection is that they are based on different approaches (i.e. perceptron based, instance based, probabilistic based and information gain based).

2. FEATURE SELECTION AND TERM WEIGHTING

Feature selection is often employed in text classification tasks in order to reduce dimensionality when documents are represented as a set of words without considering the grammar and order of the words. On the other hand, it has positive effects on improving the classification accuracy by reducing over fitting problem (Liu et al, 2009). In this study information gain with local policy is used as feature selection metric since it has introduced better performance on the imbalanced text classification (Tasci and Gungor, 2013).

Feature selection metrics can be used as a global factor of term weighting function since they evaluate the importance of a term for a specific category. In this study, two approaches are used in the formula of different feature selection metrics; (1) one-sided and (2) two-sided metrics. One-sided metrics take only positive features (i.e. relevant terms) into consideration since they compute the relevancy power of terms for a category. We test two common one-sided metrics i.e. *RF* and *Odds Ratio* (Lan et al, 2009) in the experiments. Two-sided metrics consider both positive and negative features implicitly. In other words, they can take into account either the relevancy or non-relevancy power of terms for a category. We also investigate the effect of two well-known two-sided feature selection metrics i.e. *Information Gain* and *Chi Square* which are based on probabilistic and information theories (Debole and Sebastiani, 2004). The mentioned feature selection metrics in the experiments and their formulas have been summarized in Table 1.

In text classification, term weighting is usually realized by methods taken from information retrieval and text search fields. There are three assumptions behind these traditional methods. They consider following points (1) multiple appearances of a term in a document are no less important than single appearance (*tf* assumption); (2) rare terms are no less important than frequent terms (*idf* assumption); (3) for the same quantity of term matching, long documents are no more important than short documents (*normalization* assumption) (Debole and Sebastiani, 2004).

Tfidf as a standard term weighing scheme is used in information retrieval and text classification tasks. It is formulated in form of multiplying term frequency (*tf*) by inverse document frequency (*idf*). The common and normalized form of that are shown in Equations 1 and 3 respectively (Salton and Buckley, 1988):

$$tfidf(t_i, d_j) = tf(t_i, d_j) \times idf(t_i) \quad (1)$$

$$idf(t_i) = log\left(\frac{N}{N_{t_i}}\right) \quad (2)$$

$$W_{i,j} = \frac{tfidf(t_i, d_j)}{\sqrt{\sum_{k=1}^{|T|} tfidf(t_k, d_j)^2}} \quad (3)$$

where $tf(t_i, d_j)$ denotes the number of times that term t_i occurs in document d_j, N is the number of all documents in the training set, N_{t_i} denotes the number of documents in the training set in which term t_i occurs at least once and $|T|$ denotes the number of unique terms which have been extracted from the training set. In this study, *tfidf* is used as a standard term weighting scheme throughout the experiments. At supervised term weighting, feature selection metrics are replaced instead of *idf* in the Equations 1 and 3. We named that as *tffs* in this study.

3. EXPERIMENTS

In this study, the effect of each feature selection metric is investigated over the imbalanced text classification by considering as a global component of the term weighting function. At the experiment stage, we have used R8 dataset which was extracted from Reuters-21578 and 20Newsgroups datasets which are publicly available at (Dataset for single-label text categorization, 2014) for single label text categorization. These two datasets have been widely used in text classification researches (i.e. Debole and Sebastiani, 2004; Sun et al, 2009; Ogura et al, 2011).Pre-processing steps have been applied on the datasets such as removing the 524 SMART stop words and applying Porter’s Stemmer algorithm. We conducted two types of experiments for balanced and imbalanced cases. In order to control the state of imbalance and degree of complexity, we selected one category as the positive class and the remaining portion as the negative one as (Ogura et al, 2011) had done. The R8 dataset has eight categories with imbalanced number of documents for categories and consequently it has lower complexity than the 20Newsgroups dataset. In the 20Newsgroups dataset, there exist 20 categories with almost equal number of documents. Thus, with one vs. all configuration, we can make an imbalanced case with high complexity due to the abundance of different categories in the negative class. First, we tried to make 1:1 configuration for R8 dataset by selecting the largest category among the others (i.e. *earn* category) as positive class and the sum of the other categories were considered as negative class. For 20Newsgroups dataset, *sci.space* was selected as positive class and *sci.electronics* was chosen as negative class. In the second stage, the imbalance situation was constituted on the R8 and 20Newsgroups datasets by selecting the *trade* and *sci.space* categories as positive class respectively with the consideration of the union of the other categories as the negative class. Thus, 1:20 imbalance ratio was approximately obtained for each dataset with different degree of complexities. The experiments were performed on the original training and test sets for the both datasets as shown in Table 2. By using information gain metric, the top 25 features were selected from each category for both datasets.

Table 1. All metrics used in the experiments as the global factor of term weighting schemes

Metric name	Type	Formula
Chi square	Two-sided	$X^2 = N\dfrac{(ad-bc)^2}{(a+c)(b+d)(a+b)(c+d)}$
Information gain	Two-sided	$IG = \frac{a}{N}log\frac{aN}{(a+c)(a+b)} + \frac{b}{N}log\frac{bN}{(b+d)(a+b)} + \frac{c}{N}log\frac{cN}{(a+c)(c+d)} + \frac{d}{N}log\frac{dN}{(b+d)(c+d)}$
Odds ratio	One-sided	$OR = log\frac{ad}{bc}$
Relevance frequen	One-sided	$RF = \log\left(2 + \frac{a}{\max(1,c)}\right)$

Notation:
a denotes the number of documents belongs to positive class which contains term t_i
b denotes the number of documents belongs to positive class which does not contain term t_i
c denotes the number of documents belongs to negative class which contains term t_i
d denotes the number of documents belongs to negative class which does not contain term t_i
N denotes the number of all documents in the data training set

Four popular classification algorithms i.e. libSVM (Chang and Lin, 2011), Multinomial Naïve Bayes (MultiNB) (Kibriya et al., 2005), decision tree (C4.5) (Chawla et al., 2011) and k-Nearest Neighbors (k-NN) (Ogura et al., 2011) were used to evaluate the weighting methods. In fact, we evaluate the compatibility of each classifier with each of the term weighting functions. Furthermore, we used linear kernel with default parameters for libSVM and selected k=5, 15, 25 and 35 for k-NN algorithm. For k-NN, we computed the average of the results which are obtained from different values of k in the experiments. To evaluate the results, F_1-score metric obtained from Precision (P) and Recall (R) values is used via following formulas: (1) F_1 = 2PR/(P+R), (2) P = TP/ (TP+FP) and (3) R = TP/(TP+FN) where TP, FP and FN are true positives, false positives and false negatives, respectively.

Table 2. Properties of datasets

Dataset	# of training documents	# of test documents	# of classes
R8	5485	2189	8
20 Newsgroups	11293	7528	20

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Balanced Case

In the first stage of experiments, we took the 1:1 balanced situation into consideration combined with different complexity. Fig. 1 shows the results of the supervised (*tffs*) and unsupervised (*tfidf*) term weighting schemes over the R8 dataset using the four different classifiers. It is observed that the SVM performs significantly better than the other classifiers. It also shows the compatibility of SVM with two-sided feature selection metrics when they are used in the term weighting scheme. According to obtained results, *tfidf* weighting gives better results than the supervised ones for k-NN, C4.5 and MultiNB. Among these classifiers, C4.5 and MultiNB are more sensitive to weighting schemes. Nonetheless, term weighting based on one-sided metrics are better approach for them in comparison with two-sided ones.

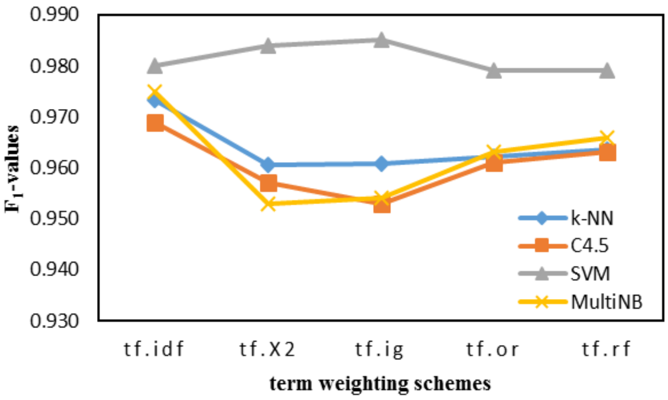


Figure 1. The F_1-values of five weighting schemes tested over R8 dataset with balanced setting using four different classifies.

We compared the previous observation with the results obtained from 20Newsgroups dataset. Fig. 2 indicates the performance of weighting schemes over the 20Newsgroups dataset using the same classifiers. As shown in Fig. 2, both C4.5 and MultiNB methods perform better than the k-NN and SVM.

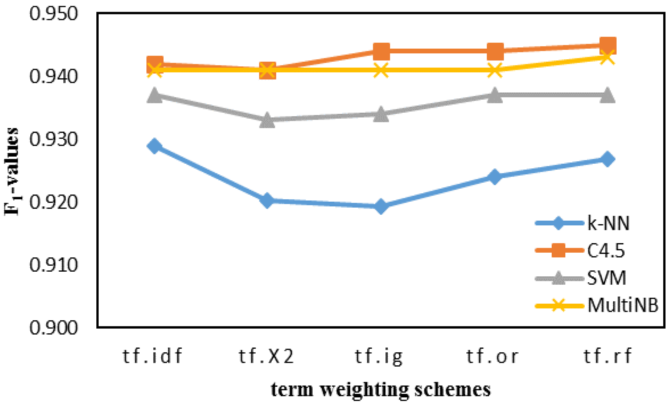


Figure 2. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with balanced setting using four different classifies.

It is noted that the observation is different than the R8 dataset since its complexity is different from the 20Newsgroups. Also we selected two similar categories for 20Newsgroups dataset while the positive class in R8 dataset is less similar to negative class. This leads to increase in the error region between positive and negative classes in the training set and consequently raises the generalization error for the model obtained from SVM. Hence the performance of SVM degrades in the 20Newsgroups dataset. According to both observations, we can conclude that the performance of one-sided metrics is better than the two-sided ones excluding SVM which can work well with two-sided based metrics, shown in Fig. 1.

4.2 Imbalanced Case

In the second stage of the experiments, we tested the behavior of term weighting schemes and classification algorithms over the 1:20 imbalanced case. First observation is that SVM performs well with one-sided term weighting methods and can even outperform *tf.idf*, while k-NN shows an adaptation with *tf.idf* and *tf.rf* term weighting schemes. On the contrary, MultiNB and C4.5 give better performance by two-sided methods and outperform *tfidf* (please see *tfidf* in the R8 dataset, shown in Fig. 3). In fact, Fig. 3 demonstrates the compatibility of one-sided methods with SVM, two-sided ones with MultiNB and C4.5, and both *tf.idf* and *tf.rf* with k-NN algorithm. It can be also observed that SVM and MultiNB effectively perform via supervised term weighting schemes on the imbalanced data. In order to expand the obtained results, we employed the same experiments on the 20Newsgroups dataset by using same imbalance ratio and more complexity configuration.

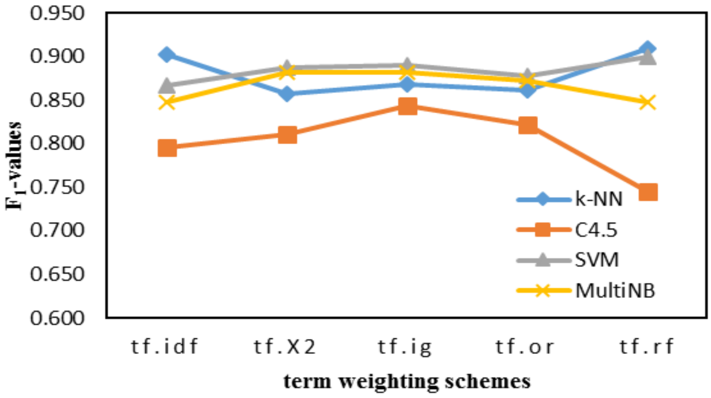


Figure 3. The F_1-values of five weighting schemes tested over R8 dataset with imbalanced setting using four different classifies.

Fig. 4 shows the classification performance of five term weighting schemes tested on the 20Newsgroups dataset using different classifiers. As shown in Fig. 4, *tfidf* outperforms the supervised term weighting schemes in the 20Newsgroups dataset which has more complexity than the R8. In the 20Newsgroups dataset, it is observed that as the degree of class complexity raises the number of subclusters increases. Therefore, it can be concluded that category based metrics cannot clearly make a contrast between documents of positive and negative classes. Nonetheless, *tfidf* which has no attention to category labels creates a good contrast in the imbalanced case with high complexity. Among the supervised weighting schemes, SVM and k-NN perform well with one-sided metrics, while C4.5 and MultiNB are compatible with two-sided metrics. This is similar to the previous observation which was obtained from R8 dataset. According to the both results in imbalanced cases, SVM with the term weighting schemes based on one-sided metrics usually performs well on the imbalanced datasets as shown in Figs. 3 and 4.

According to our findings, we can conclude that supervised term weighting schemes usually provide better representation of data for the classifiers on the imbalanced datasets with less complexity (as shown in Fig 3). Nonetheless, for high degree of complexity, *tfidf* seems a better term weighting scheme for the machine learning algorithms.

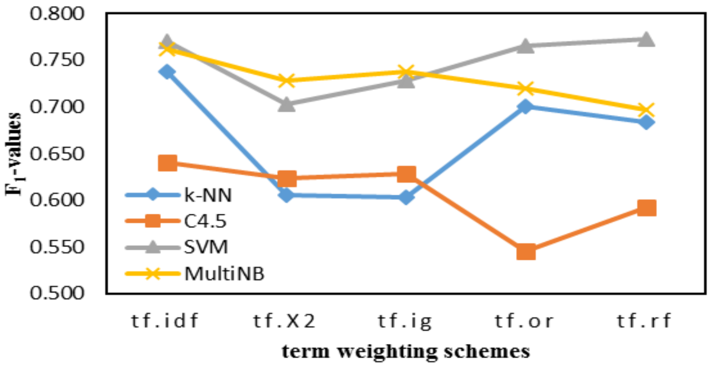


Figure 4. The F_1-values of five weighting schemes tested over 20Newsgroups dataset with imbalanced setting using four different classifies.

To determine the significance of the term weighting methods for each algorithm, we perform the ANOVA test on the F_1 values obtained from term weighting methods rather than t-test since it shows the significance of the results in more than 2 groups. As shown in Table 3, since the P-values of the tests are less than 0.05 for each case, there is a statistically significant difference between the mean F_1 values of levels at the 95.0% confidence level. Table 3 presents a multiple comparison of results to determine which algorithms differ significantly from others with respect to term weighting approaches. It can be observed that MultiNB and SVM significantly perform better than the others by using term weighting methods. At the Table 3, two and three homogenous groups are identified using columns of X's for R8 and 20Newsgroups datasets respectively. Within each column, the levels containing X's constitute groups which there are no statistically significant differences. To create a discrimination between means, Fisher's least significant difference (LSD) procedure is employed here.

Table 3. ANOVA test for F_1values obtained from 5 weighting methods for each algorithm for imbalanced cases

	R8 with P-Value = 0.0003			20 Newsgroup with P-Value = 0.0002			
Algorithms	F means	Homogeneous Groups		F means	Homogeneous Groups		
C4.5	0.8034	X		0.6060	X		
KNN	0.8793		X	0.6661		X	
MultiNB	0.8662		X	0.7290			X
SVM	0.8842		X	0.7478			X

5. CONCLUSION

In this study, the effects of two kinds of supervised term weighting schemes (one-sided and two-sided term selection metrics) were investigated on the balanced and imbalanced texts with different degrees of complexity. *Tfidf* was used as a base line to evaluate the effect of supervised weighting methods on the imbalanced texts. We evaluated the performance of each weighting method by using four different machine learning algorithms (SVM, k-NN, MultiNB and C4.5). Actually, the appropriateness of weighting methods and machine learning algorithms were studied here and, to investigate this problem we generated datasets with two different complexity level such as balanced and imbalanced cases. According to our findings, in the balanced cases, almost all classifiers had a little impact on the weighting methods. Nonetheless, it can be seen that the supervised term weighting approach does not possess any effective superiority to *tfidf*. Furthermore, it was observed that one-sided based term weighting schemes outperform the two-sided based ones in the most balanced cases.

In the imbalanced cases, it is realized that all four classifiers were susceptible to the term weighting methods. Regardless of *tfidf*, one-sided term weighting methods are better approach for SVM and k-NN algorithms while two-sided methods are the best choice for MultiNB and C4.5. According to our results, it can be concluded that supervised term weighting methods based on one-sided term selection metrics are the best choice for SVM in the imbalanced datasets and k-NN algorithm usually perform well with *tfidf*. It should be also noted that MultiNB classifier presents interesting results on the imbalanced cases. As another finding, although supervised methods cannot constantly retain their superiority to *tfidf* on the more complex imbalanced datasets, they can provide effective results for classification algorithms.

REFERENCES

Chang, C. C. and Lin, C. J., 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 2, No. 3, pp 27.

Chawla, N. V. et al, 2004. Editorial: special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, Vol. 6, No. 1, pp 1-6.

Chawla, N. V. et al, 2011. SMOTE: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*.

Chen, J. J. et al, 2006. Decision threshold adjustment in class prediction. *SAR and QSAR in Environmental Research*, Vol. 17, No. 3, pp 337-352.

Debole, F. and Sebastiani, F., 2004. Supervised term weighting for automated text categorization. In *Text mining and its applications*. Springer Berlin Heidelberg, pp. 81-97.

Japkowicz, N. and Stephen, S., 2002. The class imbalance problem: A systematic study. *Intelligent data analysis*, Vol. 6, No. 5, pp 429-449.

Kibriya, A. M. et al, 2005. Multinomial naive Bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, pp. 488-499.

Dataset for single-label text categorization, http://web.ist.utl.pt/acardoso/datasets/. (25.3.2014)

Lan, M. et al, 2009. Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 31, No. 4, pp 721-735.

Liu, A. Y. C., 2004. *The effect of oversampling and undersampling on classifying imbalanced text datasets* (Doctoral dissertation, The University of Texas at Austin).

Liu, Y. et al, 2009. Imbalanced text classification: A term weighting approach. *Expert systems with Applications*, Vol. 36, No. 1, pp 690-701.

Ogura, H. et al, 2011. Comparison of metrics for feature selection in imbalanced text classification. *Expert Systems with Applications*, Vol. 38, No. 5, pp 4978-4989.

Salton, G. and Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, Vol. 24, No. 5, pp 513-523.

Sun, Y. et al, 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, Vol. 40, No. 12, pp 3358-3378.

Sun, A. et al, 2009. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, Vol. 48, No. 1, pp 191-201.

Taşcı, Ş. and Güngör, T., 2013. Comparison of text feature selection policies and using an adaptive framework. *Expert Systems with Applications*, Vol. 40, No. 12, pp 4871-4886.

REFERENCES

[1] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, **2004**.

[2] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2):427–436, **2008**.

[3] Jan Luts, Fabian Ojeda, Raf Van de Plas, Bart De Moor, Sabine Van Huffel, and Johan AK Suykens. A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta*, 665(2):129–145, **2010**.

[4] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, **2011**.

[5] Abdul Majid, Safdar Ali, Mubashar Iqbal, and Nabeela Kausar. Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines. *Computer methods and programs in biomedicine*, 113(3):792–808, **2014**.

[6] Kok-Chin Khor, Choo-Yee Ting, and Somnuk Phon-Amnuaisuk. The effectiveness of sampling methods for the imbalanced network intrusion detection data set. In *Recent Advances on Soft Computing and Data Mining*, pages 613–622. Springer, **2014**.

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

[8] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, **2002**.

[9] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, **1988**.

[10] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, **2004**.

[11] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):721–735, **2009**.

[12] Ying Liu, Han Tong Loh, and Aixin Sun. Imbalanced text classification: A term weighting approach. *Expert systems with Applications*, 36(1):690–701, **2009**.

[13] Hakan Altınçay and Zafer Erenel. Analytical evaluation of term weighting schemes for text categorization. *Pattern Recognition Letters*, 31(11):1310–1323, **2010**.

[14] Aixin Sun, Ee-Peng Lim, and Ying Liu. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201, **2009**.

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

[16] Alexander Yun-chung Liu. *The effect of oversampling and undersampling on classifying imbalanced text datasets*. Ph.D. thesis, Citeseer, **2004**.

[17] Hiroshi Ogura, Hiromi Amano, and Masato Kondo. Comparison of metrics for feature selection in imbalanced text classification. *Expert Systems with Applications*, 38(5):4978–4989, **2011**.

[18] Şerafettin Taşcı and Tunga Güngör. Comparison of text feature selection policies and using an adaptive framework. *Expert Systems with Applications*, 40(12):4871–4886, **2013**.

[19] Ying Liu, Han Tong Loh, Youcef-Toumi Kamal, and Shu Beng Tor. Handling of imbalanced data in text classification: Category-based term weights. In *Natural Language Processing and Text Mining*, pages 171–192. Springer, **2007**.

[20] Xu-Ying Liu and Zhi-Hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 970–974. IEEE, **2006**.

[21] Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, **2010**.

[22] Claudio Persello, Abdeslam Boularias, Michele Dalponte, Terje Gobakken, Erik Naesset, and Bernhard Schoelkopf. Cost-sensitive active learning with lookahead: optimizing field surveys for remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6652–6664, **2014**.

[23] Yiming Yang. A study of thresholding strategies for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 137–145. ACM, **2001**.

[24] JJ Chen, C-A Tsai, H Moon, H Ahn, JJ Young, and C-H Chen. Decision threshold adjustment in class prediction. *SAR and QSAR in Environmental Research*, 17(3):337–352, **2006**.

[25] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004*, pages 39–50. Springer, **2004**.

[26] Gang Wu and Edward Y Chang. Aligning boundary in kernel space for learning imbalanced dataset. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 265–272. IEEE, **2004**.

[27] Gang Wu and Edward Y Chang. Kba: Kernel boundary alignment considering imbalanced data distribution. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):786–795, **2005**.

[28] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*, pages 488–499. Springer, **2005**.

[29] Yubin Park and Joydeep Ghosh. Ensembles of $(\{alpha\})$-trees for imbalanced classification problems. *Knowledge and Data Engineering, IEEE Transactions on*, 26(1):131–143, **2014**.

[30] 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**.

[31] Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, **2009**.

[32] Aixin Sun, Ee-Peng Lim, Boualem Benatallah, and Mahbub Hassan. Fisa: feature-based instance selection for imbalanced text classification. In *Advances in Knowledge Discovery and Data Mining*, pages 250–254. Springer, **2006**.

[33] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, **2011**.

[34] Eva Lorenzo Iglesias, A Seara Vieira, and Lourdes Borrajo. An hmm-based over-sampling technique to improve text classification. *Expert Systems with Applications*, 40(18):7184–7192, **2013**.

[35] Fuji Ren and Mohammad Golam Sohrab. Class-indexing-based term weighting for automatic text classification. *Information Sciences*, 236:109–125, **2013**.

[36] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 6(1):80–89, **2004**.

[37] Liuzhi Yin, Yong Ge, Keli Xiao, Xuehua Wang, and Xiaojun Quan. Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3–11, **2013**.

[38] Bhavani Raskutti and Adam Kowalczyk. Extreme re-balancing for svms: a case study. *ACM Sigkdd Explorations Newsletter*, 6(1):60–69, **2004**.

[39] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *the Journal of machine Learning research*, 2:139–154, **2002**.

[40] Ling Zhuang and Honghua Dai. Parameter estimation of one-class svm on imbalance text classification. In *Advances in Artificial Intelligence*, pages 538–549. Springer, **2006**.

[41] Ling Zhuang and Honghua Dai. Parameter optimization of kernel-based one-class classifier on imbalance text learning. In *PRICAI 2006: Trends in Artificial Intelligence*, pages 434–443. Springer, **2006**.

[42] Sebastián Maldonado and Claudio Montecinos. Robust classification of imbalanced data using one-class and two-class svm-based multiclassifiers. *Intelligent Data Analysis*, 18(1):95–112, **2014**.

[43] Larry Manevitz and Malik Yousef. One-class document classification via neural networks. *Neurocomputing*, 70(7):1466–1481, **2007**.

[44] Wei Fan, Salvatore J Stolfo, Junxin Zhang, and Philip K Chan. Adacost: misclassification cost-sensitive boosting. In *ICML*, pages 97–105. Citeseer, **1999**.

[45] Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, **2000**.

[46] Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st international workshop on Utility-based data mining*, pages 69–77. ACM, **2005**.

[47] Marcus A Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II*, volume 2, pages 2–1. **2003**.

[48] Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 435–442. IEEE, **2003**.

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

[50] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):281–288, **2009**.

[51] Vladimir Vapnik. *The nature of statistical learning theory*. springer, **2000**.

[52] Xia Hong, Sheng Chen, and Chris J Harris. A kernel-based two-class classifier for imbalanced data sets. *Neural Networks, IEEE Transactions on*, 18(1):28–41, **2007**.

[53] Bernhard Schölkopf, Patrice Simard, Alex J Smola, and Vladimir Vapnik. Prior knowledge in support vector kernels. *Advances in neural information processing systems*, pages 640–646, **1998**.

[54] Xiaoyun Wu and Rohini Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333. ACM, **2004**.

[55] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, **2002**.

[56] Behzad Naderalvojoud, Ahmet S Bozkir, and Ebru A Sezer. Investigation of term weighting schemes in classification of imbalanced texts. In *European Conference on Data Mining (ECDM)*. Lisbon, Portugal, **2014**. Accepted.

[57] Zafer Erenel and Hakan Altınçay. Nonlinear transformation of term frequencies for term weighting in text categorization. *Engineering Applications of Artificial Intelligence*, 25(7):1505–1514, **2012**.

[58] CJ Rijsbergen. v.(1979). *Information retrieval*, 2, **1979**.

[59] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420. **1997**.

[60] George Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, **2003**.

[61] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, **1986**.

[62] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and L Zadeh. Feature extraction. *Foundations and applications*, **2006**.

[63] Danny Roobaert, Grigoris Karakoulas, and Nitesh V Chawla. Information gain, correlation and support vector machines. In *Feature Extraction*, pages 463–470. Springer, **2006**.

[64] J Martin Bland and Douglas G Altman. The odds ratio. *Bmj*, 320(7247):1468, **2000**.

[65] Anthony S Robbins, Susan Y Chao, and Vincent P Fonseca. What is the relative risk? a method to directly estimate risk ratios in cohort studies of common outcomes. *Annals of epidemiology*, 12(7):452–454, **2002**.

[66] Anthony J Viera. Odds ratios and risk ratios: what is the difference and why does it matter? *Southern medical journal*, 101(7):730–734, **2008**.

[67] Cardoso Cachopo, Ana. Datasets for single-label text categorization, **2014**.

[68] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, **2011**.

CURRICULUM VITAE

Credentials

Name,Surname:	Behzad NADERALVOJOUD
Place of Birth:	Tabriz,Iran
Marital Status:	Married
E-mail:	b.naderalvojoud@gmail.com
Address:	Computer Engineering Dept., Hacettepe University Beytepe-ANKARA

Education

High School:	Ferdosi High School, Tabriz, Iran
BSc. :	Computer Engineering Dept., Islamic Azad University, Iran
MSc. :	Computer Engineering Dept., Hacettepe University, Turkey

Foreign Languages

English, Turkish

Work Experience

The Member of Multimedia Information Retrieval Laboratory in Computer Engineering Department at Hacettepe University

Areas of Experiences

Machine Learning, Information Retrieval, Text Mining

Project and Budgets

-

Publications

Behzad Naderalvojoud, Ahmet Selman Bozkir and Ebru Akcapinar Sezer, ”Investigation of term weighting schemes in classification of imbalanced texts”, *European Conference on Data Mining (ECDM), IADIS*, pp 39-46, Lisbon-PORTUGAL, **2014**.

Oral and Poster Presentations