Class Weight technique for Handling Class Imbalance

Technical Report · July 2022

CITATIONS
0

READS
441

2 authors, including:

Karmanya Kumar
BMS Institute of Technology
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Class Weight technique for Handling Class Imbalance

Ravindra V Asundi
Assistant Professor
Deptt. of Electronics and Communication Engineering
BMS Institute of Technology and Management
Bengaluru, India
ravindra_v_asundi@bmsit.in

Ravi Prakash
Final year student
Dept. of Electronics and Communication Engineering
BMS Institute of Technology and Management
Bengaluru, India
1by18ec131@bmsit.in

Karmanya Kumar
Final year student
Deptt. of Electronics and Communicaion Engineering
BMS Institute of Technology and Management
Bengaluru. India
1by18ec087@bmsit.in

***Abstract*— Data generation is continually expanding in today's internet era. Medical, e-commerce, social networking, and other data are all extremely important. However, a large number of these datasets are skewed. When one of the classes heavily outnumbers the others in a data set, it is referred to as skewed or imbalanced, which causes a classifier to perform poorly. We'll look at various strategies for balancing the datasets that are skewed. This work looked at 52 research papers to see whether data mining techniques may be used to balance the dataset that was unbalanced.**

***Keywords* – Imbalanced dataset, Balancing data, Data mining techniques.**

Introduction

The majority of real-world classification problems exhibit some level of class imbalance, which occurs when each class does not account for an equal share of the data set. Most machine learning techniques assume that data is evenly distributed in the domains of data mining and machine learning. When some types of data distributions considerably dominate the instance space compared to other data distributions, imbalanced learning develops. When data is unbalanced, dominant classes outnumber minority classes. Machine learning classifiers become more biased towards majority classes as a result of this. As a result, minority classes are poorly classified. There are two sorts of learning problems caused by imbalance :

i. **Between class imbalances:**
It refers to the imbalance that arises between the samples of two classes.

ii. **Within class imbalances:**
It occurs when the majority and minority samples are greater or lower than others.

To improve the performance of the classifier, this study will identify and describe alternative strategies for dealing with imbalanced data.

I. Literature Review

1. [Sotiris Kotsiantis, Dimitris Kanellopoulos and Panayiotis Pintelas] This paper discusses numerous strategies for dealing with dataset imbalances. Undersampling, oversampling, threshold value, cost sensitive learning, and other techniques are discussed. It will provide an overview of the many ways for pre-processing data that are accessible.

2. [Nitesh V. Chawla] They discovered a new approach called SMOTE in this paper (Synthetic Minority Oversampling Technique). This algorithm combines the techniques of oversampling and undersampling. They used several datasets and classifiers such as C4.5 decision tree, Naive Bayes, and Ripper to test the efficacy of this technique.

3. [Puja Dwivedi, Udaya Kumar] They presented an Imbalanced Ensemble Feature Selection technique using sampled data using SMOTE in this study. When classifications are performed on an imbalanced class dataset, this will be developed to address the poor accuracy problem of minority classes. They selected five datasets from the machine learning library at UC Irvine. They compare the IEFS classification algorithm to three other classification algorithms. The inclusion of attentions into the diversity measure improves the prediction accuracy of the minority class, according to the results of the studies.

4. [H. Yin, K. Gai] This study examines different balancing procedures and comes up with four conclusions:

i. Feature extraction is a marginal improvement over sampling.
ii. Undersampling is more effective when the dataset is significantly unbalanced.
iii. They don't recommend pre-processing when the dataset is less unbalanced.
iv. In wrapper-based feature selection, a more intricate searching strategy, such as genetic searching, may not yield better results than best-first searching.

5. [S. Babu, N.R. Ananthanarayanan] They suggested a novel technique called Enhanced Minority Oversampling Technique (EMOTE) to overcome the problem of unequal class distributions on the dataset in this paper. The trials were carried out on nine different data sets with the help of the machine learning algorithm C4.5 and other tools. The classifiers' output is calculated in terms of effective metrics

like F-Measure, G-Mean, and AUC, and then compared to a variety of commonly established methodologies. The results reveal that the proposed EMOTE generates a balanced dataset with minimal information loss and a smaller number of instances.

6. [Reshma C. Bhagat, Sachin S. Patil] The SMOTE (Synthetic Minority Oversampling Technique) data pre-processing technique for multi-class unbalanced data is provided in this paper. They also used the RF (Random Forest) algorithm as a foundation, which is a decision tree ensemble with an excellent track record. Because of the massive amount of data that is now being generated, big data is a point of attraction in today's environment. Data mining techniques that have been used in the past are unable to meet the demands imposed by big data. They assessed the suggested system's quality in terms of accuracy and F-measure. Various datasets from the UCI repository were used in the experiment. The results reveal that the SMOTE+OVA algorithm performs well in the case of unbalanced data.

7. [T.Deepa, M.Punithavalli] To overcome the misclassification and overfitting problem, this paper employed the EST (Evolutionary Sampling Technique) technique with a Genetic algorithm and SVM (Support Vector Machine) to extract features from a high dimensional imbalanced dataset. Two types of micro array datasets were used in this proposed work, both of which were inherently imbalanced, and the findings showed that the EST technique yielded more accuracy than the random sampling technique.

8. [Nadir Mustafa, Jian-Ping Li] In A new method for constructing an appropriate classification model is given in this study, with the goal of balancing the minority instances in the training data. It can be done with the maximum distance based SMOTE, and it can also be done with the combined fuzzy roughest and SMOTE to increase the accuracy of the testing data. Different classifiers based on the maximum distance SMOTE were employed in the qualitative and quantitative study, and it was discovered that the new approach improves the performance of Area Under the Curve (AUC) metrics and accuracy metrics in a variety of datasets. The results of this study suggest that the Fuzzy Rough Set in combination with MD SMOTE (Maximum Distance based SMOTE) is more effective than other combined approaches like a Rough Set combined with MD SMOTE(Maximum Distance based SMOTE) technique is more effective than other combined approaches such as a Rough Set Theory (RST) and support vector machine (SVM).

9. [Sachin Subhash Patil, Shefali Pratap Sonavane] Using clustering based oversampling approaches, this work presents superior data balancing techniques for two-class and multi-class imbalanced data. As foundation classifiers, a variety of classifiers are utilised. The accuracy, AOC area, G-Mean, and F-measure metrics can all be used to index the system quality testing standard. Various data sets from the **UCI/KEEL** collection are used in the experimental analysis. For chosen data sets, the suggested three approaches, **MEMMOT** (Mere Mean Minority Oversampling Technique), **MMMmOT** (Minority Majority Mix Mean Oversampling Technique), and **CMEOT** (Clustering Minority Examples Oversampling Technique), outperform existing methods. The results reveal that when compared to the basic techniques for classification, the proposed methods have a higher F-measure and ROC area score. The issues related to data set shift and changing oversampling rate needs to be further addressed in depth.

10. [Aamer hanif, Noor Azhar] To address the problem of class imbalance, three strategies were used in this study. In addition, three feature selection approaches were applied. To analyse and compare the strategies, a random forest classification model was created. A real-world dataset was used to investigate the impact of different strategies. In this study, random oversampling produced the best results for balancing the datasets. However, all three feature selection strategies performed equally well and only emphasised the call-related features as the most significant. This paper fairly gives a new approach towards the pre-existing basic /fundamental sampling techniques with the newer feature selection method.

II. OUR PROJECT

We have developed a machine – learning model, which primarily works on the **class – weight** approach for balancing an imbalanced dataset. This model was built using the **RandomForestClassifier[*]** in **python 3** language.

We not only implemented the weight method, but also the previously discovered methods like **Oversampling, Undersampling, SMOTE** etc. and then compared the results for each of them.

The entire project was carried out in **2 phases :**

i. **DATA CLEANING**
ii. **MODEL BUILDING**

This project revealed that out of the above listed methods, class-weight method yields us the best results considering the following aspects :

i. It gives us a **consistent accuracy** during both training and testing of the model
ii. It results in a **reasonable** cross – validate
iii. Unlike Over Sampling methods like **random** over-sampling and **SMOTE,** there is no need of wasting any extra device memory, creating/duplicating the minority class samples
iv. Nor we need to ignore any available majority class sample (as in case of Under Sampling methods), which might be a **potential** data.
v. Class weight method yields a reasonably valid **F1** score.

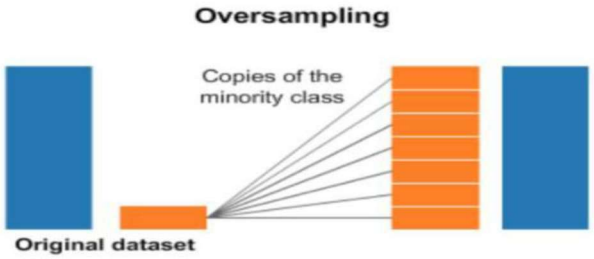
	Training	CV	Test
No class balancing	1.0	0.859044	0.863912
Class balancing (undersampling)	1.0	0.874650	0.212249
Class balancing (oversampling)	1.0	0.999930	0.849125
Class balancing (class weights)	1.0	0.845852	0.847381

Hence, the class-weight approach for balancing of datasets can be considered as the most **efficient** method.

III. PREVIOUS METHODS

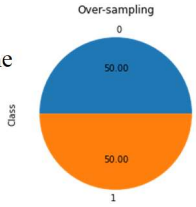
❖ **OVER SAMPLING:**

To balance the data set, additional samples are added to the minority class in this method. Random oversampling is the process of replicating existing minority samples in order to expand the size of a minority class.



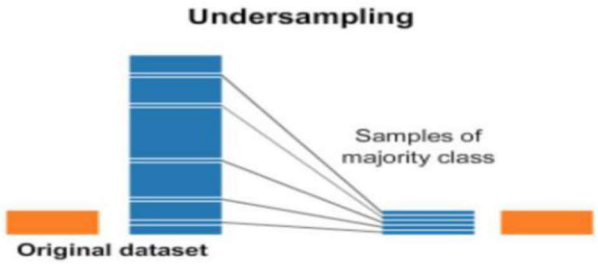
- ***Achievements :***

Oversampling successfully balances the input dataset and yields the **maximum** accuracy, **replicating** the minority class samples until both classes have equal count.



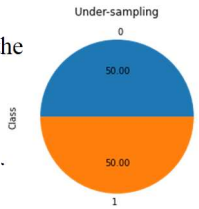
❖ **UNDER SAMPLING:**

In this method, the majority class samples are reduced to balance the data set. This decrease can be done at random, which is referred to as **random** undersampling, or it can be done with the use of statistical knowledge, which is referred to as **informed** undersampling.



➢ ***Achievements :***

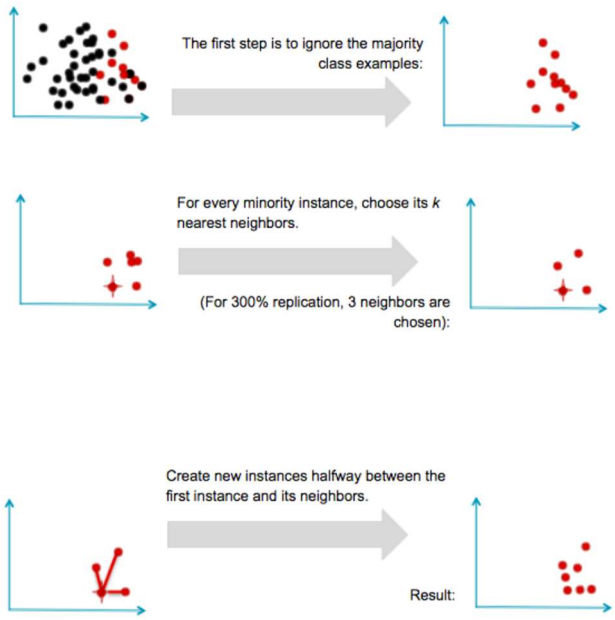
Undersampling successfully balances the input dataset, considering only certain samples from the majority class which meet the count equal to that of minority class, ignoring the rest samples available.



❖ **SMOTE**

Rather than over-sampling with replacement, the minority class is over-sampled by creating "synthetic" instances in this method. This is a statistical strategy for balancing the increase in the number of cases in a dataset. The module creates new instances based on existing minority cases that are provided as input. The number of majority cases does not change as a result of SMOTE adoption.

The next figure shows the process of SMOTE technique **:**



IV. PROPOSED METHOD

❖ **CLASS WEIGHT METHOD:**

The weighted class or class-weight method approaches in a very different way as compared to the pre-existing sampling methods. Instead of creating new or ignoring the existing samples, we here develop as model which would we fed composite inputs comprising of their actual times the **inverse** of their occurrence(frequency).

Hence, for a set containing 9 ones and 1 zero, we can develop a model which when takes input(both training and testing sets), would receive a one as **1** while a zero as **0,0,0,0,…** 9 times, i.e. a total of **18** samples comprising 9 ones and 9 zeros, without actually creating any extra sample.

➢ ***All the models are built using RandomForestClassifier available in the ensemble package of python 3.***

V. RANDOM FOREST CLASSIFIER

Random Forest Classifier is an ensemble learning method for classification, regression, and other problems that works by building a large number of decision trees during training.

Ensemble means to combine multiple models or multiple grouping of models. Here, we may have any kind of models with us, which may be added up sequentially or parallelly but the computation of each and every model is important.

There are 2 different techniques in Ensemble **:**

i. **Bagging :** This technique makes the use of Random Forest Classifier.
ii. **Boosting :** This technique makes the use of AdaBoost, XgBoost, Gradiant Boost, etc..

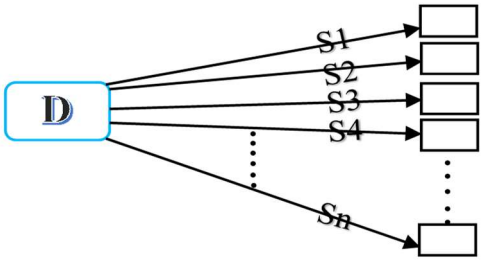
VI. BAGGING

This Ensemble technique basically uses **multiple-decision trees,** where each and every decision-tree plays its role.

Working :

i. Let us Suppose we have a dataset **"D".** Now, this dataset would be fed to multiple decision-trees.
ii. For each decision-tree, we feed some sample of our dataset, i.e. some part of our entire dataset.
iii. **All these samples are given with replacement to different decision-trees i.e. no parts of previous decision-tree input would be repeated in the current sample.**
iv. Once all the samples are fed into the separate decision-trees, that particular decision-tree would get trained with the sample provided to it :

Hence,
DT1 will get trained on S1
DT2 will get trained on S2
DT3 will get trained on S3
DT4 will get trained on S4, etc..



v. During the training of our model, different decision-trees would have shown **different accuracies**, *hence,* we take **mean** of all their separate accuracies and call it as **Average accuracy** of the model.

$$A_{avg} = \sum Ai \,/\, n$$

vi. After the training is done and our model actually created, whenever we give a new input data, i.e the **test-data,** different decision-trees would give us different outputs. In such cases, the module counts the **frequency** of different outputs and the output with **maximum** frequency would be selected as the model's final solution.

The step when we divide our database and train different models, is called as **Bootstrap** (with replacement), while the step when we finally combine different outputs of various decision-trees and finally come to a conclusion is called as **Aggregation.** Hence, Random Forest Classifier is also known as **Bootstrap-Aggregation.**

Random Forest Classifier deals with two kinds of problems :

i. Classification – based problems
ii. Regression – based problems

In case of **Classification-Based problems,** we come to a solution which has the maximum frequency, i.e. the **most probable solution** is chosen here.

In case of **Regression-Based problems,** we don't have binary (0/1) or Boolean(T/F) values, rather we have **continuous** values. Hence, in such cases, the **arithmetic mean** of all these continuous values would be taken and termed as the **final** output of the *test*.

The internal functioning of Random Forest Classifier can be correlated with the working of *Hard and Soft voting* classifiers.

Hard – Voting Classifier : It simply visits and observes the output of each and every decision-tree and then selects the **most probable solution.**

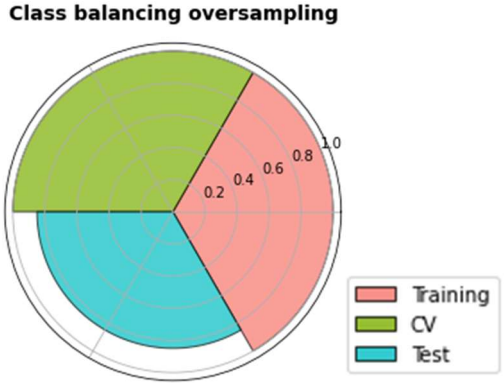
Soft – Voting Classifier : Instead of simply *repeatedly* selecting the maximum frequent solution, both at the model's level and decision-trees' level, this classifier would collect their respective **probabilities** at each and every step and find the mean probability. Then, the value with maximum mean probability would be selected as the final output.

VII. Achievements Over Other Approaches

i. Over Sampling :

Though Over Sampling yields us the **maximum accuracy** both during training and testing of the model, but it wastes a lot of memory of the device as it unnecessarily duplicates the minority class sample until the classes become comparable in size, *hence,* leading to a **huge cross-validation.**

This can be understood better with the following results :



- Here as we can see, the test score is very close to **80%**, but at the same time cross-validation exceeds **99%,** which is much expensive to afford.
- Even, the **confusion-matrix** and other **precision parameters (obtained during testing of the model)** do

Actual	0	1
Predicted		
0	71082	25
1	7	88

	precision	recall	f1-score	support
0	1.00	1.00	1.00	71107
1	0.78	0.93	0.85	95
accuracy			1.00	71202
macro avg	0.89	0.96	0.92	71202
weighted avg	1.00	1.00	1.00	71202

not support oversampling as an ideal method for handling class imbalance.

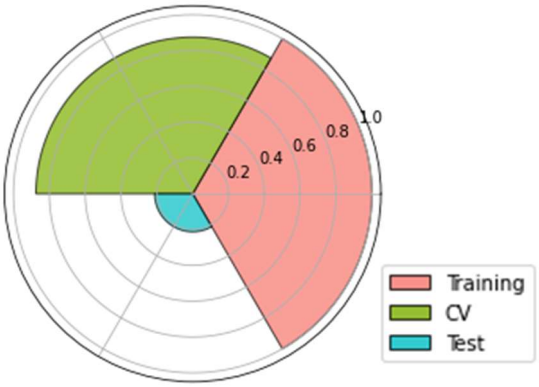
- Hence, Over Sampling model can not be accepted to solve the **Class-imbalance** problem.

ii. UNDER SAMPLING :

Under Sampling not only yields the **minimum** accuracy **(~21%)** during testing of the model, but also a not-so good cross-validate. As this method ignores a lot of available data during the training phase, hence it fails during the testing of our model; and again discarding such huge amount of data is not at all wise as that might be a potential data for us, which might get used somewhere.

We can understand these problems better having a look over the following results:

Class balancing undersampling



Actual	0	1
Predicted		
0	69137	9
1	1952	104

	precision	recall	f1-score	support
0	1.00	0.76	0.87	497
1	0.69	1.00	0.82	261
accuracy			0.84	758
macro avg	0.84	0.88	0.84	758
weighted avg	0.89	0.84	0.85	758

Hence, Under Sampling cannot be accepted as a good solution for our problem.

iii. CLASS WEIGHTS :

This study finds wieghted class method yields us a **reasonable** accuracy during the testing of the model as well as it results in a very reasonable cross-validate which makes this technique the best suitable solution for our problem statement.

The different type of inputs to **class-weight (**every classification algorithm has this parameter) allows to handle class imbalance using a different manner. By default, when no value is passed, the weight assigned to each class is equal e.g. 1. In case of class imbalance, here are different values representing different types of inputs:

- **balanced:** When passing **balanced** as class - weight, results in the values of **y (label)** automatically adjusting weights **inversely proportional to class frequencies** in the input data.

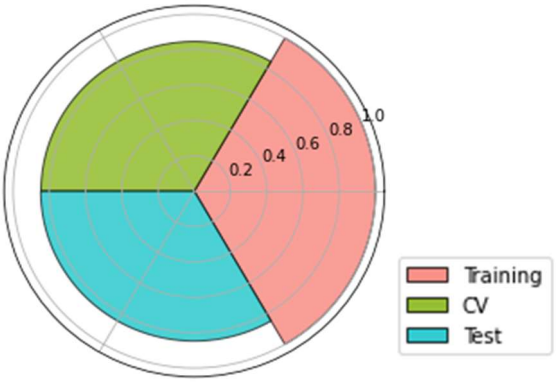
The same can be calculated as **:**

n-samples / (n_classes * np.bincount(y))

- **{class_label: weight}:** For two classes labelled as **0** and **1**. Passing input as **class-weight={0:2, 1:1}** means *class 0 has weight 2* and *class 1 has weight 1*.

The following results support our study **:**

Class balancing class weights



Actual	0	1
Predicted		
0	71085	28
1	4	85

	precision	recall	f1-score	support
0	1.00	1.00	1.00	71113
1	0.75	0.96	0.84	89
accuracy			1.00	71202
macro avg	0.88	0.98	0.92	71202
weighted avg	1.00	1.00	1.00	71202

Hence, Class - weight technique can be accepted as the most-efficient method for our purpose.

VIII. CONCLUSIONS

In today's world, class disparity is a key issue in research. Unbalanced datasets can cause a research project to be misled. The many strategies for balancing data for efficient analysis are discussed in this work. This paper discusses a variety of strategies for balancing datasets, including sampling, cost sensitive learning, ensemble learning, and feature selection. 52 research papers are analysed and classified in this study based on the author's nation, year of publication, and methodologies utilised in those publications, which helps to determine the frequency of data balancing strategies. The SMOTE technique is the most regularly used technique in the publications analysed for this study, and feature selection is the second most commonly used technique. When compared to other strategies, the following two yield better results when it comes to balancing datasets. For better results, we recommend pre-processing the datasets before balancing. **Only 10 of the 52 papers assessed were written by Indian writers**; the rest were written by scholars from other countries. Still, there is a lot more I can contribute to this issue, therefore I'll be spending more time researching it.

IX. LIMITATIONS OF RESEARCH

Not all research journals are considered for evaluation in this study. However, only research publications that are trustworthy from every angle are examined. As a result, there are certain limits to this review.

For starters, the study looked at 52 journals from the last ten years. For research, more journals might be chosen.

Second, the journals were scoured using a strategy for balancing datasets from some real-world challenges. For future research, more challenges and techniques can be investigated and analyzed.

X. REFERENCES

[1]Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, Handling imbalanced datasets: A review, GESTS International Transactions on Computer Science and Engineering, Vol.30, 2006.

[2]N.V. Chawla, K.W. Bowyer and L.O. Hall, SMOTE: Synthetic Minority Over sampling Technique, 16 321–357, (2002).

[3]H. Yin and K. Gai. An empirical study on pre-processing high dimensional class-imbalanced data for classification. In 2015 IEEE 17th International Conference on High-Performance Computing and Communications; The IEEE International Symposium on Big Data Security on Cloud pages 1314–1319, New York, USA, 2015.

[4] Puja Dwivedi, Udaya Kumar, A Review on Classification Algorithm for Imbalanced-Class Datasets, International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882 Volume 6, Issue 5, May 2017

[5] S. Babu, N.R. Ananthanarayanan, EMOTE: Enhanced Minority Oversampling TEchnique, Journal of Intelligent & Fuzzy Systems, 33 67–78 (2017).

[6] Reshma C. Bhagat, Sachin S. Patil, Enhanced SMOTE Algorithm for Classification of Imbalanced Big-Data using Random Forest, IEEE, 2015.

[7] T.Deepa, M.Punithavalli, A New Sampling Technique and SVM classification for Feature selection in High-Dimensional Imbalanced dataset, IEEE, 2011.

[8] Nadir Mustafa, Jian-Ping Li, Medical Data Classification Scheme Based on Hybridized SMOTE Technique (HST) and Rough Set Technique (RST), IEEE International Conference on Cloud Computing and Big Data Analysis, 2017.

[9] Sachin Subhash Patil, Shefali Pratap Sonavane, Enriched Over_Sampling Techniques for Improving Classification of Imbalanced Big Data, IEEE Third International Conference on Big Data Computing Service and Applications, 2017.

[10] Aamer hanif, Noor Azhar, Resolving Class Imbalance and Feature Selection in Customer Churn Dataset, IEEE International Conference on Frontiers of Information Technology, 2017.

[11] Apurva Sonak, R.A.Patankar, A Survey on Methods to Handle Imbalance Dataset, IJCSMC, Vol. 4, Issue. 11, pg.338 – 343, November 2015.

[12] Kubat M, Matwin S, "Addressing the curse of imbalanced training sets: One-sided selection", In Douglas H. Fisher, editor, ICML, pages 179–186, 1997.

[13] Tomek Ivan, "An Experiment with the Edited Nearest-Neighbor Rule", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 6, No. 6, pp. 448-452, 1976.

[14] Zhu Jingbo, Hovy Eduard, "Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem"; Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 783–790, (2007).