Performance of Classifiers on Oversampled Data obtained from SMOTE to Address Class Imbalance

Anirudh Madhusudan
Department of Industrial and Systems Engineering
University of Illinois, Urbana-Champaign
Urbana, IL 61801
anirudh2@illinois.edu

Abstract

Effective techniques using oversampling methods to build classifier for an unbalanced data set is described in this literature survey . A dataset is imbalanced if there are uneven representation of instances from a certain class within the data set. Typically instances are oversampled or undersampled to improve the sensitivity of a classifier while predicting the majority class. Synthetic Minority Oversampling Technique tries to blend under sampling of majority class and oversampling synthetically generated minority samples.This paper explore the application of SMOTE and other algorithms which are modified version of SMOTE, such are SMOTE borderline and ADASYN. By experimenting on a imbalanced dataset the performance of combination of oversampling techniques and classifiers are studied in this paper. The python code and relevant data set can be found **here**.

1 Introduction

When a dataset is said to be imbalanced, the classes are not represented equally. For instance, medical screening for a certain health condition is carried out on a large population of people without the condition, to find the small minority within it (only 0.5% of USA's population has HIV). When there is such an imbalance in the dataset, the algorithm chosen for classification is bound to be biased towards the majority class since the loss function attempts to optimize quantities like error rate while not necessarily taking the distribution of the training data into consideration. In most classification problems involving imbalanced data it is more important to correctly identify these rare instances (minority class) correctly.

The performance of various algorithms are evaluated using their predictive accuracy of an unseen dataset. Typically a Receiver Operating Characteristic curve (ROC) is used in understanding the performance of the classifier given a range of tradeoffs. The Area Under the Curve (AUC) is used as the performance metric given an ROC curve.

There have been several studies including a a recent review [1] of various state of the art solutions for obtaining and evaluating models based on classification and regression tasks. From this study it is evident that the following methods are used to address imbalanced datasets:-

- Balance training set by oversampling minority class, undersampling majority class or synthesizing new minority classes
- Eliminate minority class and switch to anomaly detection framework
- Modify the algorithm to be more sensitive to minority class by tuning the misclassification cost or decision threshold
- Conceptualize new algorithms to improve performance on imbalanced data

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

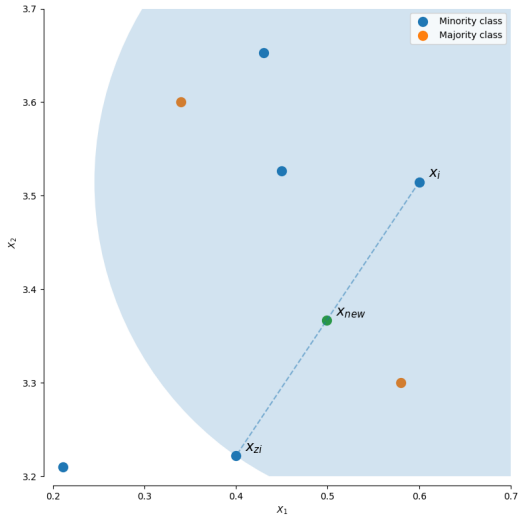


Figure 1: Demonstration of Synthetic Minority Oversampling Technique
Image Source: `contrib.scikit-learn.org/imbalanced-learn`

Chawla et.al [2] introduced Synthetic Minority Oversampling Technique (SMOTE) in 2002 as a method to blend undersampling of majority class and oversampling synthetically generated minority samples. The paper discusses the improvement in performance of various classifiers such as Decision Trees, Ripper and Naive Bayes.

2 Literature Overview

2.1 SMOTE

Given a sample x_i, a new sample x_{new} shall be generated considering its k-nearest neighbors. For example, the 3 neighbors are included in the blue circle is illustrated in the figure 1. One of these nearest neighbors x_{zi} will be chosen and a new synthetic x_{new} will be created, given as

$$x_{new} = x_i + \beta(x_{zi} - x_i) \tag{1}$$

The β value can lie anywhere in range [0,1]. the algorithm randomly picks a number in this range and interpolates a sample between the two points x_i and x_{zi} Since SMOTE operates by interpolating between minority instances, it can only generate instance within the body of available instance but never outside its space i.e SMOTE can only fill in the convex hull of existing minority examples, but not create new exterior regions of minority examples. SMOTE has been generally successful has led to many variants, extensions, and adaptations to different concept learning algorithms.There are a few popular variants of SMOTE in the recent years, such as SMOTE Borderline1, SMOTE Borderline2 and ADASYN.

2.2 SMOTE Borderline

Han et.al[3] explores minority instances and carries out experiments that have presented that SMOTE Borderline approaches achieve better TP rate and F-value than SMOTE and random over-sampling methods.SMOTE Borderline gets its name from the fact that the algorithm synthesizes minority instance near the borderline only. The algorithm for SMOTE Borderline can be summarized as below.Given a sample x_i, SMOTE Borderline will classify each x_i into 3 groups.

- Noise: all nearest neighbors are from a different class than that of x_i
- Danger: half or less of nearest neighbors are from a different class than x_i
- Safe: all of the nearest neighbors are from a same class than x_i

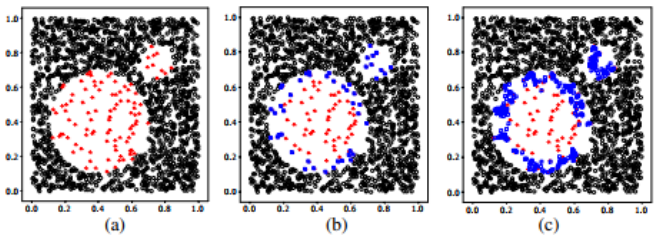


Figure 2: Demonstration of SMOTE Borderline: blue samples indicate oversampled data at the border
Image Source: `Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning`

Borderline SMOTE will choose the x_{zi} sample from the danger zone to generate new synthetic data. There are two approaches suggested in the Borderline SMOTE paper [3]

In Borderline-1 SMOTE, the x_{zi} is selected from a different class than x_i.

In Borderline-2 SMOTE, the x_{zi} is selected with no bias toward any class.

The borderline instances seem to be more easily misclassified than those ones far from it. Thus the borderline method strategically over-samples the borderline instances only of the minority class. SMOTE on the other hand augments the minority class throughout all instance hence cause possible overfitting.

2.3 ADASYN

ADASYN or Adaptive Synthetic Sampling Approach works similarly to the regular SMOTE algorithm. Haibo et al. [4] describes that ADASYN can autonomously shift the classifier decision boundary to be more focused on those difficult to learn instances, therefore improving learning performance. These objectives are accomplished by a dynamic adjustment of weights and an adaptive learning procedure according to data distributions. The pseudo algorithm is given as under -

Number of samples generated from each x_i depends on the proportion of number of samples that are not from the same class as x_i in the neighborhood. Therefore there will be more samples generated where there are lesser instances of the same class in the neighborhood of x_i

2.4 Choice of Classifer

Different studies have been done in the past to evaluate the performance of classifiers on re sampled dataset. Azra et.al[4] found that Decision Trees and Neural Networks gave good performances as long as the re sampled instances were balanced. Increasing the oversampling rate favored the minority class. Naive Bayes in contrast showed no clear changes to oversampling. This could probably be explained in the innate nature of the Naive Bayes classifier which is a probabilistic classifier with the independence assumption among variables.

3 Experimental Procedure

3.1 Dataset

The dataset used in this experimental effort is the **Wine Quality Data Set**from the UCI Machine Learning repository. The data sample consists of 4898 instances of 11 continuous attributes and the quality of the wine as the output (scored between 0 to 10). To simplify the evaluation, the scores with 6 or more where classified as "good" wine (1) and a score of 5 or less was classified as "bad" wine (0) thus making it a binary classification problem. Although the given sample is not imbalanced in itself, necessary data splits were made to create only 50 instance of the minority class ("bad" wine) and 2450 instances of majority class ("good" wine) for the training data. For the testing data, 500 good wine and 500 bad wine instances were used.

3.2 Methodology and Observations

4 classifiers are selected in this experiment namely - Decision Trees, k-Nearest Neighbors, Naive Bayes Classifier and Logistic Regression in their default arguments from the sk-learn python module (i.e `DecisionTreeClassifier()`,`kNeighorsClassifier()`,`LogisiticRegression()` and `GaussianNB()`). These classifier are used to predict the labels from the test data, and the results were as follows:-

Prediction Accuracy with only training data (no oversampling)		
Classifier	Majority Class	Minority Class
Decision Tree	0.978	0.108
Naive Bayes	0.984	0.152
k-NN	0.996	0.002
Logistic Regression	1.00	0.000

Following this SMOTE,SMOTE Borderline1,SMOTE Borderline2 and ADASYN are implemented for varying values of neighbors ranging from [2,49]. The python module name imbalanced-learn was installed and the function SMOTE() and ADASYN were used. Borderline1 and Borderline2 were implemented using the kind argument in the SMOTE() function.The number of samples generated were about 4900 which was almost double the number of samples in the training data.

The accuracy of the majority and minority class is computed along with the overall accuracy for each classifier upon using the four oversampling techniques.

SMOTE Data Max Accuracy Performance				
Accuracy Type	Decision Tree	Naive Bayes	Logistic Regression	kNN
Overall	0.605	0.654	0.749	0.615
Majority Class	0.286	0.56	0.668	0.476
Minority Class	0.964	0.764	0.848	0.924

SMOTE Borderline1 Data Max Accuracy Performance				
Accuracy Type	Decision Tree	Naive Bayes	Logistic Regression	kNN
Overall	0.577	0.621	0.723	0.574
Majority Class	0.186	0.47	0.608	0.228
Minority Class	0.984	0.86	0.892	0.956

SMOTE Borderline2 Data Max Accuracy Performance				
Accuracy Type	Decision Tree	Naive Bayes	Logistic Regression	kNN
Overall	0.578	0.619	0.739	0.564
Majority Class	0.212	0.484	0.684	0.334
Minority Class	0.952	0.822	0.848	0.878

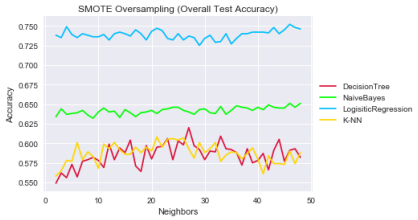
ADASYN Data Max Accuracy Performance				
Accuracy Type	Decision Tree	Naive Bayes	Logistic Regression	kNN
Overall	0.619	0.651	0.752	0.62
Majority Class	0.298	0.566	0.676	0.462
Minority Class	0.97	0.762	0.84	0.922

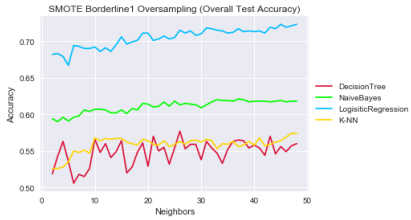
3.3 Results

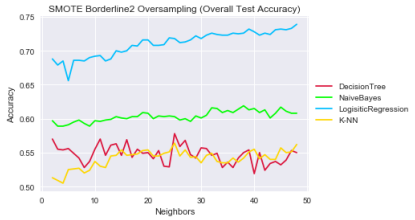
3.3.1 Classifier + Oversampling Technique

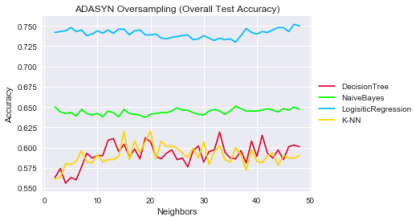
Given that the algorithms for all the oversampling methods were tuned in the arguments to only oversampling minority class, the classifiers have shown a better accuracy at predicting the minority class. However, the challenge is the effectiveness of the combination of oversampling technique and classifier that has worked to make accurate predictions. The combinations of Classifier and Oversampling Technique that has given a good minority class prediction are:-

Overall Accuracy

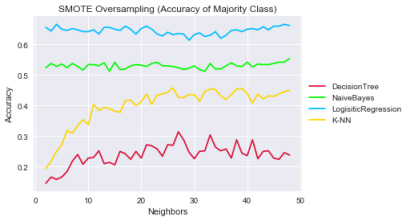


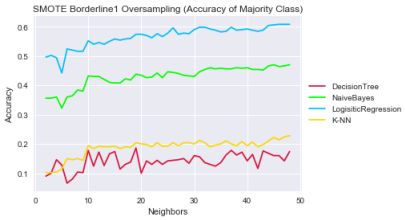


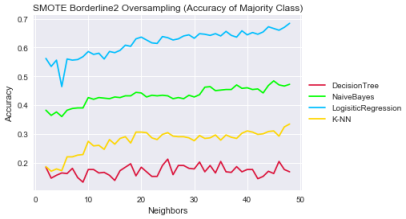


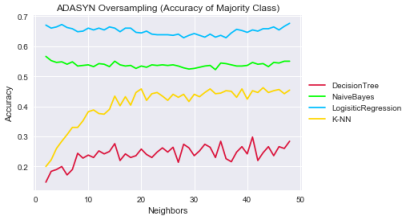


Majority Accuracy

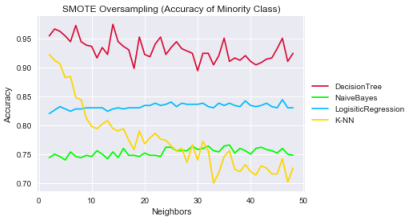


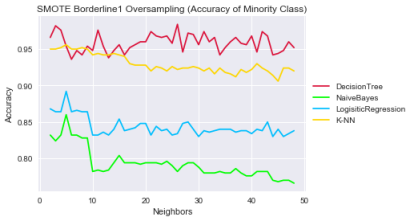


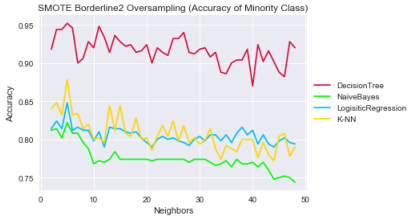


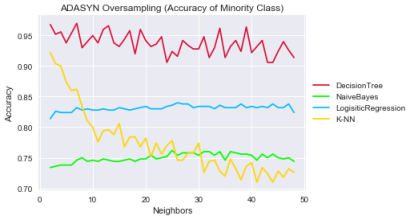


Minority Accuracy









- SMOTEBorderline1 + Decision Tree (98.4%)
- ADASYN + Decision Tree (97%)
- Regular SMOTE + Decision Tree (96.4%)
- SMOTE Borderline1 + kNN (96.4%)

Although Decision Tree has adopted the Oversampled data very well and accurately predicted the minority labels, it consistently fails to do will in the majority class (which now is a minority class after oversampling the minority class). **This proves that Decision Tree and KNN have actually over fitted with respect to the minority class.**

Logistic Regression is able address this balance and give a good overall prediction i.e fairly high prediction of both majority and minority class. In conclusion, the results show that Logistic Regression is able to generalize better in a binary classification problem given a moderate imbalance. The best overall performance is found in the following combinations of Classifier and Oversampling Techniques:-

- ADASYN + Logit (75.2%)
- Regular SMOTE + Logit (74.9%)
- Borderline2 SMOTE + Logit (73.9%)
- Borderline1 SMOTE + Logit (72.3%)

As described in the Chawla et.al[2] SMOTE followed by undersampling of minority class would perform better since it seems like the resulting resampled data set in heavily biased towards the old minority class. In addition as discussed in Chawla et.al [2] all the classifiers more or less perform the same. Sometimes one or more classifier might be more biased depending on the data and the bias vs. variance tradeoff within the classifier chosen.

Through this experiment it is clear that SMOTE provides more related minority class samples for the classifier to learn from, so that it learns broader decision regions, leading to more accuracy of predicting the minority class.

4 Conclusion

The SMOTE approach seems to have improved the accuracy of the classifier for the minority instances. SMOTE and its varients offer helpful methods to address imbalance in dataset. Through the elaborate experiments carried out it was found that SMOTE possibily would perform better if SMOTE is carried out followed by under sampling of the original minority class from the resampled data. This is because there seems to be a large bias towards the oversampled class after using SMOTE. While there are ways to oversample the majority class also, this result hasn't been studied in this paper. The choice of classifier is critical in the performance because some classifiers have a strong bias or strong variance and the dataset used is very critical in narrowing down on the classifier type. For the binary classification problem used in this study, logisitic regression seems to generalize well in comparsion to other classifiers.

5 References

[1] Paula Branco, Luis Torgo, Rita Ribeiro (2015) "A Survey of Predictive Modelling under Imbalance Distributions", *arXiv.org*, Cornell University

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

[3] Haibo He , Yang Bai , Edwardo A. Garcia , Shutao Li (2008), "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", IEEE International Joint Conference on Neural Networks 2008

[4] Azra Ramezankhani, Omid Poumik, Jamal Shahrabi (2014) "The Impact of Oversampling with SMOTE on the Performance of 3 Classifiers in Prediction of Type 2 Diabetes", Society for Medical Decision Making