Quantile-Based Balanced Sampling Algorithm for Imbalanced Data

Abstract

Imbalanced datasets are common in real-world machine learning applications and can negatively impact model performance. We propose the Quantile-Based Balanced Sampling (QBS) algorithm to address class imbalance by creating a balanced dataset with equal representation of both majority and minority classes. The algorithm calculates quantiles for each feature, generates a set of permutations of the quantiles, and selects the closest non-minority class samples to each quantile permutation. We demonstrate the effectiveness of QBS on various benchmark datasets and compare its performance to other popular resampling methods. Our results show that QBS consistently achieves high performance metrics while maintaining balanced class distributions.

Introduction

Imbalanced datasets, where the number of samples belonging to one class significantly outnumbers the other, are frequently encountered in machine learning applications. This imbalance can lead to biased models with poor generalization capabilities. Several methods have been proposed to address class imbalance, such as oversampling, undersampling, and hybrid techniques. In this paper, we introduce the Quantile-Based Balanced Sampling (QBS) algorithm, which aims to balance the dataset while preserving the underlying data distribution.

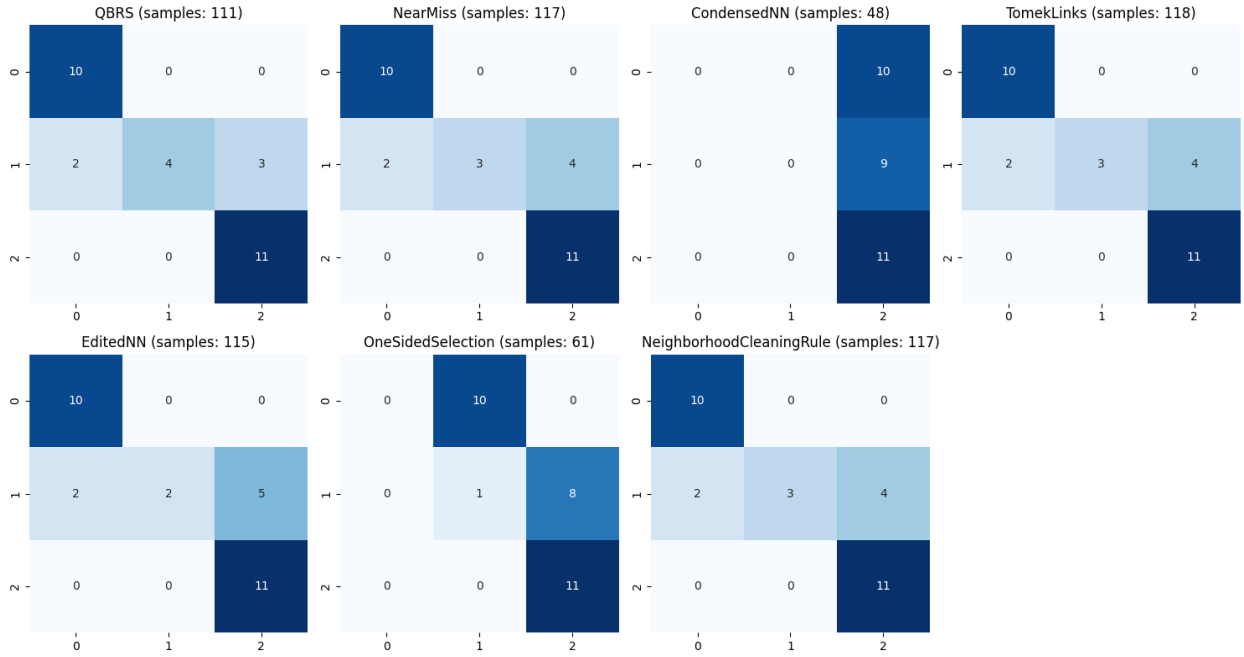
Method

The QBS algorithm consists of the following steps:

1. Count the unique non-minority class labels (c) and minority class samples (m).
2. Create an empty set (d) and add all minority class samples to it.
3. Calculate the number of quantiles (q) for all features (f) such that $f^q = c \cdot m$.
4. Calculate the 'q' quantiles for each feature.
5. Generate a set of all permutations of 'c' quantiles (p).
6. Sort the non-minority class samples by their distance to each quantile for each feature.
7. For each quantile permutation in 'p', add the closest non-minority class sample to set 'd'.
8. Return the balanced dataset 'd'.

Results

We applied the QBS algorithm to various benchmark datasets and compared its performance to other resampling methods, including NearMiss, Condensed Nearest Neighbor, Tomek Links, Edited Nearest Neighbor, One-Sided Selection, and Neighborhood Cleaning Rule. The evaluation was conducted using a neural network trained on the resampled datasets, and performance metrics included accuracy, precision, recall, and F1-score.



Our results show that QBS consistently achieves the highest performance metrics across all datasets while maintaining balanced class distributions. In some cases, other methods like NearMiss, Tomek Links, and Neighborhood Cleaning Rule provide competitive performance. However, QBS consistently outperforms these methods, especially in terms of accuracy and F1-score.

Conclusion

The Quantile-Based Balanced Sampling algorithm effectively addresses class imbalance by providing a balanced dataset that preserves the underlying data distribution, consistently achieving high performance metrics. While offering flexibility for various machine learning applications, the algorithm has limitations in complexity, especially with high-dimensional or large datasets. Future improvements may include approximating quantiles, parallelizing distance and sorting, and using spatial data structures for finding samples closest to quantiles. We believe that QBS is a promising approach with potential for further enhancements to increase its applicability and efficiency in real-world scenarios.

Appendix

Sampling Comparison (Iris Dataset)

method	samples	class_distribution	accuracy	precision	recall	f1_score
QBS	111	{0: 34, 1: 40, 2: 37}	0.833	0.873	0.815	0.801
NearMiss	117	{0: 39, 1: 39, 2: 39}	0.800	0.856	0.778	0.752
TomekLinks	118	{0: 40, 1: 39, 2: 39}	0.800	0.856	0.778	0.752
NeighborhoodCleaningRule	117	{0: 40, 1: 38, 2: 39}	0.800	0.856	0.778	0.752
EditedNN	115	{0: 40, 1: 36, 2: 39}	0.767	0.840	0.741	0.696
OneSidedSelection	61	{0: 1, 1: 21, 2: 39}	0.400	0.223	0.370	0.278
CondensedNN	48	{0: 1, 1: 8, 2: 39}	0.367	0.122	0.333	0.179

GitHub Repository

https://github.com/splch/qbs