Submitted 4/23; Published 9/23

Quantile-Based Balanced Sampling: A Novel Approach for Imbalanced Classification

Spencer Churchill CHURCHILL@IONQ.COM
IonQ
College Park, MD 20740, USA

Editor:

Abstract

Imbalanced datasets pose a significant challenge to machine learning algorithms. Traditional methods often perform poorly on minority classes due to the skewed distribution of class samples. This paper introduces the Quantile-Based Balanced Sampling (QBS) algorithm for handling imbalanced datasets. The QBS algorithm is an under-sampling method that aims to balance the dataset by selecting a representative subset of the majority class samples based on their distance to quantiles in the feature space. We demonstrate the effectiveness of QBS by comparing its performance to existing under-sampling techniques on various benchmark datasets. Experimental results show that QBS consistently achieves competitive performance in terms of F1-score, precision, recall, and accuracy.

Keywords:
Imbalanced Learning, Undersampling, Quantile-Based Sampling, Classification

1 Introduction

Imbalanced datasets are common in many real-world applications, such as fraud detection, medical diagnosis, and text classification. The skewed distribution of class samples in these datasets can lead to biased predictions and poor performance on minority classes. Several under-sampling methods have been proposed to address this issue, including random under-sampling, Tomek links, and neighborhood cleaning rules. However, these methods often struggle to capture the underlying structure of the majority class and may discard valuable information.

In this paper, we propose the Quantile-Based Balanced Sampling (QBS) algorithm, an under-sampling technique that aims to preserve the structure of the majority class while balancing the dataset. The key idea of QBS is to select a subset of majority class samples based on their proximity to quantiles in the feature space. We also provide a comprehensive comparison of QBS to existing under-sampling methods on benchmark datasets, demonstrating its effectiveness in various settings.

2 Quantile-Based Balanced Sampling Algorithm

The Quantile-Based Balanced Sampling (QBS) algorithm is an innovative undersampling technique that leverages quantile information to select representative samples from the majority class. The main steps of the algorithm are:

1. Count the unique non-minority class labels c, minority class samples m, and features f.
2. Create an empty set d and add all minority class samples to it.
3. Calculate the number of quantiles q such that $f^q = c * m$.
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.

We provide a Python implementation A.1 of the QBS algorithm and a comparison code to evaluate its performance against existing under-sampling methods.

3 Experimental Results

We evaluate the QBS algorithm on two benchmark datasets: the Iris dataset and the Wine dataset. We compare the performance of QBS to several under-sampling techniques, including random under-sampling, Tomek links, neighborhood cleaning rules, and others. The performance metrics include accuracy, precision, recall, F1-score, and computation time.

Our experimental results show that the QBS algorithm consistently achieves competitive performance across various metrics. Specifically, QBS outperforms other methods in terms of F1-score on both datasets, indicating its effectiveness in handling imbalanced datasets.

4 Conclusion

In this paper, we introduce the Quantile-Based Balanced Sampling (QBS) algorithm, a novel under-sampling technique for imbalanced datasets. QBS selects a representative subset of majority class samples based on their distance to quantiles in the feature space, effectively balancing the dataset without discarding valuable information. Experimental results on benchmark datasets demonstrate the effectiveness of QBS in comparison to existing under-sampling methods. We believe that the QBS algorithm is a valuable addition to the toolkit of machine learning practitioners dealing with imbalanced datasets.

Acknowledgments and Disclosure of Funding

The authors would like to express their gratitude to their family for proofreading the paper and providing valuable feedback about the algorithm. There were no funding sources for this research. The authors declare that they have no competing interests or related financial activities outside the submitted work.

Appendix A.

A.1 Code Implementation

The code implementation of the Quantile-Based Balanced Sampling (QBS) algorithm is provided below in Python:

```
import numpy as np
from itertools import product
from scipy.spatial import KDTree

def qbs(X, y, version=1):
    # Identify unique classes and their counts
    unique_classes, class_counts = np.unique(y, return_counts=True)

    # Determine the minority class and its count
    minority_y = unique_classes[np.argmin(class_counts)]
    minority_y_count = np.min(class_counts)

    # Split the dataset into minority and majority class samples
    minority_X = X[y == minority_y]
    majority_X = X[y != minority_y]
    majority_y = y[y != minority_y]

    # Calculate the number of quantiles and feature count
    feature_count = X.shape[1]
    quantile_count = int(np.ceil(np.log(minority_y_count * (len(unique_classes) - 1)) / np.l

    # Compute the quantiles for each feature
    quantiles = np.array([np.percentile(majority_X[:, i], np.linspace(0, 100, quantile_count

    # Generate all possible permutations of the quantiles
    quantile_permutations = np.array(list(product(*quantiles)))

    # Build a KDTree for efficient nearest neighbor search
    tree = KDTree(majority_X)

    # Find the closest samples in the majority class for each quantile permutation
    k = 1 if version == 1 else minority_y_count
    closest_samples_idx = np.unique(tree.query(quantile_permutations, k=k, workers=-1)[1])

    # Combine the minority class samples and the selected majority class samples
    X_resampled = np.vstack((minority_X, majority_X[closest_samples_idx]))
    y_resampled = np.hstack((np.full(minority_y_count, minority_y), majority_y[closest_sampl

    return X_resampled, y_resampled
```

A.2 Analysis and Results

In this section, we provide a detailed analysis of the results obtained from the experimental evaluation of the Quantile-Based Balanced Sampling (QBS) algorithm. The performance metrics used for comparison are accuracy, F1 score, and balanced accuracy.

Table 1: Performance comparison of undersampling techniques on the Iris dataset

Method	Samples	Class Distribution	Accuracy	Precision	Recall	F_1 Score
QBS	39	{2: 15, 1: 13, 0: 11}	1	1	1	1
NeighborhoodCleaningRule	58	{1: 24, 2: 23, 0: 11}	0.96	0.961538	0.956522	0.956336
TomekLinks	63	{1: 26, 2: 26, 0: 11}	0.946667	0.944127	0.942029	0.941919
RandomUnderSampler	33	{0: 11, 1: 11, 2: 11}	0.946667	0.950617	0.942029	0.941587
AllKNN	57	{1: 24, 2: 22, 0: 11}	0.933333	0.940476	0.927536	0.92667
CondensedNearestNeighbour	14	{0: 11, 1: 2, 2: 1}	0.933333	0.940476	0.927536	0.92667
EditedNearestNeighbours	54	{1: 22, 2: 21, 0: 11}	0.933333	0.940476	0.927536	0.92667
InstanceHardnessThreshold	36	{2: 14, 0: 11, 1: 11}	0.933333	0.940476	0.927536	0.92667
RepeatedEditedNN	54	{1: 22, 2: 21, 0: 11}	0.933333	0.940476	0.927536	0.92667
ClusterCentroids	33	{0: 11, 1: 11, 2: 11}	0.88	0.90625	0.869565	0.864373
NearMiss	33	{0: 11, 1: 11, 2: 11}	0.84	0.885714	0.826087	0.813387
OneSidedSelection	14	{0: 11, 1: 3}	0.693333	0.5	0.666667	0.555556

Table 2: Performance comparison of undersampling techniques on the Wine dataset

Method	Samples	Class Distribution	Accuracy	Precision	Recall	F_1 Score
QBS	41	{1: 17, 0: 13, 2: 11}	0.932584	0.941126	0.934938	0.93779
TomekLinks	65	{1: 31, 2: 21, 0: 13}	0.910112	0.922454	0.915033	0.918255
InstanceHardnessThreshold	39	{0: 13, 1: 13, 2: 13}	0.910112	0.908586	0.900178	0.903062
NearMiss	39	{0: 13, 1: 13, 2: 13}	0.853933	0.861871	0.860665	0.860147
NeighborhoodCleaningRule	55	{1: 26, 2: 16, 0: 13}	0.853933	0.875238	0.867201	0.85311
RandomUnderSampler	39	{0: 13, 1: 13, 2: 13}	0.842697	0.862043	0.825312	0.833182
ClusterCentroids	39	{0: 13, 1: 13, 2: 13}	0.808989	0.815108	0.820559	0.817629
CondensedNearestNeighbour	21	{0: 13, 1: 4, 2: 4}	0.775281	0.769542	0.781937	0.766123
OneSidedSelection	25	{0: 13, 2: 9, 1: 3}	0.730337	0.825	0.743316	0.736667
EditedNearestNeighbours	38	{1: 18, 0: 13, 2: 7}	0.719101	0.690453	0.695781	0.690821
RepeatedEditedNN	38	{1: 18, 0: 13, 2: 7}	0.719101	0.690453	0.695781	0.690821
AllKNN	45	{1: 22, 0: 13, 2: 10}	0.662921	0.583276	0.609329	0.588449

From Tables 1 and 2, we can observe that the QBS algorithm consistently achieves better performance compared to other undersampling techniques across both datasets. This indicates that the quantile-based approach is effective in selecting representative samples from the majority class, leading to improved classification performance.

References

Christopher M Bishop. *Pattern recognition and machine learning.* Springer, 2006.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357, 2002.

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

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

Ivan Tomek. Two modifications of cnn. In *IEEE Transactions on Systems, Man, and Cybernetics*, pages 769–772, 1976.

David L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972.