Postprint of IDP-SMOTE Resampling Algorithm for Imbalanced Classification

Authors: Sheng Kai, Liu Zhong, Zhou Dechao, Feng Chengxu

Date: 2018-05-20T00:00:00+00:00

Abstract

Traditional classification algorithms are prone to misclassifying minority classes when dealing with imbalanced data. To enhance the classification accuracy of minority class samples, this paper proposes a sampling algorithm termed IDP-SMOTE based on improved density peak clustering. First, the density peak clustering algorithm is enhanced through Box-Cox transformation and the #1; criterion, enabling automatic discrimination of cluster centers and outliers. Subsequently, the improved density peak clustering algorithm is integrated with the SMOTE upsampling technique to eliminate noisy data, and synthetic samples are generated within subclass boundaries based on the local density and neighbor distances of minority class samples. The proposed algorithm effectively mitigates boundary ambiguity arising from upsampling, ameliorates issues of intra-class imbalance and the learning difficulty of boundary samples, while simultaneously achieving automatic clustering and resampling, thereby preventing human factor interference. Experimental comparisons validate the effectiveness and adaptability of the proposed algorithm.

Full Text

IDP-SMOTE Resampling Algorithm for Imbalanced Classification

Sheng Kai, Liu Zhong, Zhou Dechao, Feng Chengxu
College of Electronic Engineering, Naval University of Engineering, Wuhan, Hubei 430033, China

Abstract

Traditional classification algorithms tend to misclassify minority samples when dealing with imbalanced data. To improve classification accuracy for minority classes, this paper proposes a novel resampling algorithm based on an improved

density peaks clustering method, named IDP-SMOTE. First, the density peaks clustering algorithm is enhanced using Box-Cox transformation and sigma criteria to enable automatic identification of cluster centers and outliers. Then, this improved clustering algorithm is combined with the SMOTE oversampling technique to remove noisy data and synthesize new samples within sub-class regions based on the local density and nearest neighbor distances of minority samples. This approach effectively avoids boundary ambiguity caused by oversampling, addresses within-class imbalance, and mitigates learning difficulties for boundary samples, while achieving automatic clustering and resampling without manual intervention. Experimental comparisons demonstrate the effectiveness and adaptability of the proposed algorithm.

Keywords: imbalanced data; classification; resampling; density peaks clustering

0 Introduction

Classification is a fundamental technique for knowledge acquisition in machine learning and data mining, aiming to build predictive models from labeled data. Most conventional classification algorithms assume balanced training datasets where each class contains roughly equal numbers of samples. However, real-world data often exhibits significant class imbalance, with certain classes having far fewer instances than others. When trained with accuracy-maximizing objectives, these algorithms become biased toward the majority class, increasing the risk of misclassifying minority instances. This is particularly problematic in applications such as anomaly detection and disease diagnosis, where minority classes are often of greater interest and misclassification carries higher costs. Consequently, imbalanced data classification has become a major research focus in machine learning.

Oversampling techniques address this issue by increasing minority class samples to achieve class balance. Simple random oversampling replicates existing minority samples, but may lead to severe overfitting. In 2002, Chawla et al. proposed SMOTE, which generates synthetic minority samples by randomly interpolating along lines connecting each minority instance to its k nearest neighbors. However, SMOTE ignores the underlying data distribution and selects neighbors somewhat arbitrarily, often resulting in blurred class boundaries and inserted noise. To address within-class imbalance, He et al. introduced ADASYN in 2008, which synthesizes more samples in low-density minority regions. Nevertheless, this method can still place synthetic samples within the majority class distribution.

Various clustering-based sampling techniques have since emerged to solve this problem. Barua et al. proposed CBSO based on hierarchical clustering in 2011; Bunkhumpornpat et al. developed DB-SMOTE using DBSCAN in 2012; Cao et al. presented a Gaussian mixture model-based approach in 2014; and Chen et

al. introduced KM-SMOTE based on K-means clustering in 2015. These methods perform oversampling within minority sub-clusters but suffer from high computational complexity and require manual parameter tuning, which is challenging for unknown data distributions.

To overcome these limitations, this paper integrates the Density Peaks (DP) clustering algorithm with SMOTE to propose a novel resampling method called IDP-SMOTE. First, we improve DP by establishing automatic criteria for identifying cluster centers and outliers. Then, we cluster each class using this improved algorithm while removing noise. Finally, to prevent synthetic minority samples from falling into majority class regions, we generate new samples within minority sub-clusters, adjusting oversampling weights based on local density to prioritize minority sub-clusters and boundary instances.

---

1 Improved Density Peaks Clustering Algorithm

Rodriguez and Laio proposed the DP clustering algorithm in 2014 based on two assumptions: (1) cluster centers have the highest local density among their neighbors, and (2) different cluster centers are relatively far apart. The core steps are:

a) Calculate local density ρ_i for each sample point x_i:

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c)$$

where d_{ij} represents the distance between samples x_i and x_j, and d_c is a cutoff distance typically set to the 1% or 2% percentile of all pairwise distances. For small datasets, an exponential kernel can be used:

$$\rho_i = \sum_{j \neq i} \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right)$$

b) Compute nearest neighbor distance δ_i (distance to the nearest higher-density point): For the point with maximum global density, δ_i is set to the global maximum distance. The calculation is:

$$\delta_i = \begin{cases} \min_{j:\rho_j > \rho_i}(d_{ij}), & \text{if } \exists j \text{ s.t. } \rho_j > \rho_i \\ \max_j(d_{ij}), & \text{otherwise} \end{cases}$$

c) Generate a decision graph with ρ_i on the x-axis and δ_i on the y-axis. Cluster centers exhibit both high local density and large nearest neighbor distance, while outliers have large δ_i but very small ρ_i. These can be manually selected from the decision graph.

d) Assign remaining points to the same cluster as their nearest higher-density neighbor, excluding cluster centers and outliers.

While DP eliminates the need for preset cluster parameters, its identification of centers and outliers (step c) remains manual. To automate this, we introduce Box-Cox transformation from statistical economics to normalize the distributions of ρ_i and δ_i, then apply sigma criteria for automatic identification.

Given a positive sequence $X = \{x_0, x_1, ..., x_{N-1}\}$, the Box-Cox transformation is:

$$x_i^{(\lambda)} = \begin{cases} \frac{x_i^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ \ln(x_i), & \lambda = 0 \end{cases}$$

where λ is a transformation parameter determined by maximizing the log-likelihood function:

$$f(x,\lambda) = -\frac{N-1}{2} \ln \left(\sum_{i=0}^{N-1} \frac{(x_i^{(\lambda)} - \overline{x^{(\lambda)}})^2}{N} \right) + (\lambda - 1) \sum_{i=0}^{N-1} \ln(x_i)$$

We define the discrimination rules for cluster centers and noise points as:

$$E_C = \{i \mid [\delta_i' \geq \mu_{\delta'} + 3\sigma_{\delta'}] \cap [\rho_i' \geq \mu_{\rho'} + \sigma_{\rho'}]\}$$

$$E_N = \{i \mid [\delta_i' \geq \mu_{\delta'} + 2\sigma_{\delta'}] \cap [\rho_i' < \mu_{\rho'} - 2\sigma_{\rho'}]\}$$

where δ_i' and ρ_i' are transformed values, μ denotes mean, and σ denotes standard deviation. E_C identifies cluster centers while E_N identifies noise.

The improved DP algorithm is summarized below:

Algorithm 1: Improved-DP
Input: Dataset D to be clustered
Output: Class labels $idxC$, local density ρ, nearest neighbor distance δ

1. Calculate all pairwise distances between samples;
2. Determine cutoff distance d_c;
3. Compute local density ρ_i for each sample using Eq. (1) or (2);
4. Compute nearest neighbor distance δ_i using Eq. (4);
5. Apply Box-Cox transformation to ρ and δ;
6. Determine cluster centers E_C and noise points E_N using Eqs. (7) and (8);
7. Assign labels to remaining points to complete clustering.

[Figure 1: see original paper] demonstrates the clustering performance of Improved-DP on several datasets. Subfigures (a) and (e) show the original D31 and Spiral datasets; (b) and (f) display the δ_i distributions; (c) and (g)

show the transformed distributions that better approximate normality; (d) and (h) present the final clustering results, where cluster centers and outliers are marked with △ and ⋆ symbols, respectively. The results confirm that Improved-DP achieves good clustering performance with strong adaptability.

---

2 IDP-SMOTE Algorithm

To automatically identify sub-clusters according to data distribution, perform oversampling within minority sub-clusters, and avoid within-class imbalance and noise effects, we propose the IDP-SMOTE algorithm based on Improved-DP.

Algorithm 2: IDP-SMOTE
Input: Training set D, oversampling coefficient β
Output: Resampled training set $D_{resample}$

1. Define D_S and D_L as minority (positive) and majority (negative) subsets of D;
2. Apply Improved-DP to cluster D_S and D_L, obtaining cluster labels, ρ_i and δ_i for each sample;
3. Determine total minority sampling quantity G based on the ratio after noise removal: $G = m_s \times \beta - m_s$, where m_s and m_l are minority and majority sample counts, and β controls the balance level ($\beta = 1$ means perfect balance);
4. Determine sampling quantity G_i for each minority sub-cluster: $G_i = G \times \frac{1/s_i}{\sum_{j=1}^{n}(1/s_j)}$, where n is the number of sub-clusters and s_i is the size of sub-cluster i;
5. Calculate sampling weight for each minority sample: $r_i = \frac{1/\delta_i}{\sum_{j=1}^{s_i}(1/\delta_j)}$, giving higher weights to boundary samples with smaller δ_i;
6. Compute individual sampling quantity $g_i = r_i \times G_i$ for each minority sample;
7. For each minority sample x_i:
   - 7.1 Find all neighbors within distance threshold δ_i' in its sub-cluster;
   - 7.2 Randomly select a neighbor x_s and generate synthetic sample: $x_{new} = x_i + \text{rand}(0,1) \times (x_s - x_i)$, repeated g_i times;
8. Generate the final resampled training set $D_{resample}$.

Improved-DP determines cluster numbers, centers, and noise points automatically. We apply it to both classes, exclude noise, and retain the computed ρ_i and δ_i parameters. Step 4 uses the inverse of sub-cluster sizes (following CBSO' s approach) to determine sampling quantities per sub-cluster. Since smaller δ_i indicates boundary positions, we assign higher sampling weights to boundary samples in Step 5. Step 7 adapts CBSO's oversampling process but uses distance threshold δ_i' instead of fixed k-NN to avoid manual k-value selection.

---

3 Experiments

3.1 Datasets

We evaluate our approach on seven datasets from the UCI Machine Learning Repository [11]: Abalone, German, Glass, Leaf, Letter, Vehicle, and Wine. For binary classification testing, multi-class datasets were converted. Table 1 summarizes the dataset characteristics.

3.2 Performance Metrics

We assess imbalanced classification performance using F-measure, G-means, and AUC [11]. These metrics are defined using the confusion matrix shown in Table 2 .

F-measure evaluates minority class recognition, balancing precision and recall:

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

where Recall $= \frac{TP}{TP+FN}$ and Precision $= \frac{TP}{TP+FP}$.

G-means measures overall classification effectiveness on both classes:

$$G\text{-means} = \sqrt{TPR \times TNR}$$

where $TPR = \frac{TP}{TP+FN}$ and $TNR = \frac{TN}{TN+FP}$. A classifier biased toward one class will yield low G-means.

AUC represents the area under the ROC curve, providing a comprehensive performance measure with values closer to 1 indicating better performance.

3.3 Experimental Setup and Results

We test IDP-SMOTE using Random Forest classifiers from Python' s scikit-learn library [12] with default parameters, comparing against SMOTE, KM-SMOTE, and DB-SMOTE. For fair comparison: k = 5 for nearest neighbors in SMOTE, KM-SMOTE, and DB-SMOTE; cluster count c = 2 for KM-SMOTE; neighborhood radius $\epsilon = 0.5$ and MinPts = 5 for DB-SMOTE; $\beta = 1$ for all methods. Each experiment randomly selects 75% of samples for training and 25% for testing, repeated 100 times to obtain average metrics. Results are shown in Table 3 .

The results show that SMOTE and its variants generally outperform direct classification on imbalanced data. However, SMOTE ignores sample distribution, while KM-SMOTE and DB-SMOTE rely heavily on proper parameter settings that require domain knowledge. Inaccurate parameters can cause synthetic samples to fall into majority class regions, degrading performance. IDP-SMOTE avoids subjective parameter input, removes noise, and adjusts sampling

weights to address within-class imbalance and boundary sample learning difficulties. This makes it more adaptive, achieving the highest overall win rate across all tested datasets.

4 Conclusion

This paper proposes IDP-SMOTE, a novel resampling algorithm for imbalanced classification that intelligently synthesizes minority samples based on spatial data distribution. Compared to previous methods, IDP-SMOTE offers four advantages: (1) Improved-DP clustering without shape constraints and without manual parameter tuning; (2) Noise removal across all classes; (3) Oversampling within minority sub-clusters to prevent synthetic samples from encroaching on majority class regions; (4) Adaptive sampling coefficients that mitigate within-class imbalance and boundary sample learning difficulties. Future work will extend the algorithm to multi-class problems for broader practical applicability.

References

[1] Li Yong, Liu Zhandong, Zhang Haijun. Survey of ensemble classification algorithms for imbalanced data [J]. *Computer Application Research*, 2014, 31(5): 1287-1291.

[2] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. *Journal of Artificial Intelligence Research*, 2002, 16(1): 321-357.

[3] He Haibo, Bai Yang, Garcia E A, et al. ADASYN: adaptive synthetic sampling approach for imbalanced learning [C] // Proc of IEEE International Joint Conference on Neural Networks. 2008: 1322-1328.

[4] Barua S, Islam M M, Murase K. A novel synthetic minority oversampling technique for imbalanced data set learning [C] // Proc of International Conference of Neural Information Processing. 2011: 735-744.

[5] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. DBSMOTE: density-based synthetic minority over-sampling technique [J]. *Applied Intelligence*, 2012, 36(3): 664-684.

[6] Chen Bin, Su Yidan, Huang Shan. Imbalanced data classification based on KM-SMOTE and random forest [J]. *Computer Technology and Development*, 2015, 25(9): 17-21.

[7] Cao Peng, Li Bo, Li Wei, et al. Hybrid sampling algorithm based on probability distribution estimation [J]. *Control and Decision*, 2014, 9(5): 815-820.

[8] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. *Science*, 2014, 344(6191): 1492-1496.

[9] Mehmood R, Bie Rongfang, Hussain D, et al. Fuzzy Clustering by Fast Search and Find of Density Peaks [C] // Proc of International Conference on Identification, Information, and Knowledge in the Internet of Things. 2016: 258-261.

[10] Bicego M, Baldo S. Properties of Box-Cox Transformation for Pattern Classification [J]. *Neurocomputing*, 2016, 218: 390-400.

[11] UCI machine learning repository [DB/OL]. [2017-06-10]. http://archive.ics.uci.edu/ml/datasets.

[12] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python [J]. *Journal of Machine Learning Research*, 2012, 12(10): 2825-2830.

Note: Figure translations are in progress. See original paper for figures.