A Synthetic Minority Oversampling Method Based on Local Densities in Low-Dimensional Space for Imbalanced Learning

Zhipeng Xie[1,2(✉)], Liyang Jiang[1], Tengju Ye[1], and Xiao-Li Li[3]

[1] School of Computer Science, Fudan University, Shanghai, China
{xiezp,13210240017,13210240039}@fudan.edu.cn
[2] Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China
[3] Institute of InfoComm Research, Fusionopolis Way, Singapore, Singapore
xlli@i2r.a-star.edu.sg

Abstract. Imbalanced class distribution is a challenging problem in many real-life classification problems. Existing synthetic oversampling do suffer from the curse of dimensionality because they rely heavily on Euclidean distance. This paper proposed a new method, called Minority Oversampling Technique based on Local Densities in Low-Dimensional Space (or MOT2LD in short). MOT2LD first maps each training sample into a low-dimensional space, and makes clustering of their low-dimensional representations. It then assigns weight to each minority sample as the product of two quantities: local minority density and local majority count, indicating its importance of sampling. The synthetic minority class samples are generated inside some minority cluster. MOT2LD has been evaluated on 15 real-world data sets. The experimental results have shown that our method outperforms some other existing methods including SMOTE, Borderline-SMOTE, ADASYN, and MWMOTE, in terms of G-mean and F-measure.

Keywords: Imbalanced learning · Oversampling method · Local densities · Dimensionality reduction

1 Introduction

Imbalanced distribution of data samples among different classes is a common phenomenon in many real-world classification problems, such as fraud detection [1] and text classification [2]. In this paper, we focus on two-class classification problems for imbalanced data sets, where the class that contains few samples is called the minority class, and the other that dominates the instance space is called the majority class. The imbalanced data sets have degraded the learning performance of existing learning algorithms and posed a challenge to them for the hardness to learn the minority class samples.

Confronted with the problem of imbalanced learning, some simple yet effective methods have been proposed to generate extra synthetic minority samples in order to

M. Renz et al. (Eds.): DASFAA 2015, Part II, LNCS 9050, pp. 3–18, 2015.
DOI: 10.1007/978-3-319-18123-3_1

balance the distribution between the majority class and the minority class [3][4][5][6], which are called synthetic oversampling methods. SMOTE[3], Borderline-SMOTE[4], ADASYN[5], and MWMOTE[6] are typical examples of this kind of algorithms. All these algorithms generate synthetic minority samples in two main phases. The first phase is to identify those informative minority class samples, and the second phase is to interpolate a synthetic minority class sample between those informative minority class samples and their nearby ones. The difference among them exists in the way of how the synthetic samples are generated. SMOTE algorithm [3] is the first and simplest synthetic oversampling method, which treats all the minority class samples equally. To generate a synthetic minority sample, it first draws seed samples randomly from the whole set of minority class samples in the seed drawing phase, and then calculates the k nearest neighbors in the minority class for each seed sample and generates new synthetic samples along the line between the seed sample and its nearest minority neighbors. As an improvement over SMOTE, Borderline-SMOTE [4] only draw seed samples from those dangerous minority samples at borderline. A minority class sample is at borderline if there are more majority class samples than minority ones in its m nearest neighbors. Borderline-SMOTE first identifies the borderline minority class samples, and then uses them as seed samples for generating the synthetic samples because they are most likely to be misclassified by a classifier. However, all the borderline samples are treated equally. To adaptively draw seed samples, ADASYN algorithm [5] adaptively assigns weights to the minority class samples. A large weight enhances the chance for the minority class sample serving as a seed sample in the synthetic sample generation process. Both Borderline-SMOTE and ADASYN share a synthetic sample generation process that is similar to the one used by SMOTE: the synthetic minority class samples are generated by interpolation randomly between the seed samples and their K-nearest neighbors of the minority class. Recently, a new algorithm MWMOTE [6] is proposed to identify the hard-to-learn informative minority class samples, to assign them weights according to their Euclidean distance from the nearest majority class samples, and then to generate synthetic samples inside minority class clusters. It has been illustrated in [6] that MWMOTE can avoid some situations that the other methods will generate wrong and unnecessary synthetic samples.

Although these synthetic oversampling methods have achieved some satisfactory results for imbalanced learning, they still have their deficiencies. Firstly, all of them for these methods rely heavily on the Euclidean distance in the calculation of K-nearest neighbors, which may suffer from the curse of dimensionality, especially when the dimensionality of the sample space is high. Secondly, the synthetic generation process used by SMOTE, Borderline-SMOTE, and ADASYN does not take the cluster structure into consideration. Last but not least, we think the local minority density should have its position in determining the importance of a minority class sample for the generation of synthetic minority samples.

To solve these problems, we propose a new algorithm, which consists of three main steps. It first applies t-SNE algorithm to reduce the dimensionality of the training samples into a two-dimensional space, where each sample is represented as a two-dimensional vector. Then, a density-peak algorithm is used to learn the cluster structure of the training samples in the low-dimensional space. The importance of a minority class sample is measured by taking two factors into consideration: local

majority count and local minority density. Local majority count indicates how many majority class samples appear in the *K*-nearest neighbors of the minority class sample. The higher the local majority count is, the harder is to make the correct decision for the sample. The local minority density of a given minority sample indicates the density of minority class samples around it. The lower the local minority density is, the more likely is to generate a synthetic sample from the sample. Finally, based on the importance measurement of the minority samples, synthetic minority samples are generated, which are located inside some minority cluster.

The whole paper is organized as follows. Section 2 describes our proposed method MOT2LD (Minority Oversampling Technique based on Local Densities in Low-Dimensional Space) in detail. Section 3 presents the experimental results. Finally, we summarize the whole paper and point out possible directions for future work.

2 The Proposed Method

The objective in this paper is to exploit and integrate modern dimensionality reduction and clustering techniques in order to solve the problems that existing synthetic oversampling methods are facing. The proposed algorithm, called Minority Oversampling Technique based on Local Densities in Low-Dimensional Space (or **MOT2LD** in short), consists of five major steps as listed in Table 1.

- The first step is to reduce the dimensionality of the representation of training samples. By dimensionality reduction, each sample in high-dimensional space can be mapped into a point in a low-dimensional space. Dimensionality reduction can be thought of a kind of metric learning technique, leading to a better distance metric between samples.
- The second step is to discover the cluster structure of the minority class samples in the low-dimensional space. A new density-based clustering algorithm called DPCluster is exploited, which is capable of determining the cluster number automatically. It is desirable that the generated synthetic minority samples are within some cluster, instead of "between clusters".
- The third step is to detect and filter out outliers and noises in the set of minority samples, for the existence of outliers and noises may do harm to the quality of the generated synthetic minority samples.
- The fourth step is to assign weights to the minority samples, indicating their importance for oversampling. The weight is measured as the product of the local majority count and the inverse of local minority density.
- The final step is to generate the synthetic minority samples according to the importance weights of the training minority samples. We also restrict that the synthetic minority samples should be inside some minority cluster, to avoid generating synthetic samples between different clusters.

The details of MOT2LD are described below.

Table 1. The framework of MOT2LD algrorithm

Algorithm:	Minority Oversampling Technique based on Local Densities in Low-Dimensional Space
Input:	
NSamples:	A set of majority class samples (Negative class)
PSamples:	A set of minority class samples (Positive class)
K:	The number of nearest neighbors observed when filtering noise samples
NumToGen:	The number of synthetic minority samples to be generated
Output:	
Y:	The set of synthetic minority samples that are generated

Procedure Begin

Step 1: (Dimensionality Reduction)

Use t-SNE algorithm to reduce the dimensionality of the dataset, where each data sample x_i is represented as a low-dimensional image y_i in a low-dimensional space.

Step 2: (Clustering of Minority Class Samples)

Use Density Peak Clustering algorithm to partition the set of minority class samples into a number of clusters $Cl_1,...,Cl_s$, where s is the number of clusters. As byproduct, we can also get the local minority density ρ_i for each minority sample i.

Step 3: (Outlier Detection and Noise Filtering)

For each minority class sample, if its local minority density is zero, it will be treated as outlier and get removed. In addition, we also count the number of majority class samples in its K-nearest neighbors. If all the K neighbors are from majority class, then the minority sample is a noise to be filtered.

Step 4: (Weight Assignment)

Assign an importance weight $Importance(i)$ to each minority class sample i as a product of its local majority count $\gamma(i)$ and the inverse of its local minority density $\rho(i)$.

Step 5: (Synthetic Sample Generation)

For each minority sample i, set $prob(i) = \frac{Importance(i)}{Z}$ where $Z = \sum_{i \in minority\ class} Importance(i)$.

for i := 1 to NumToGen

1) Randomly draw a minority sample x_s as the seed sample, according to the probability distribution $\{ prob(i): i \in minority\ class\}$
2) Choose another minority sample x_t from the minority cluster that x_s belongs to.
3) Generate one synthetic minority sample $x_{new} = \alpha \times x_s + (1 - \alpha) \times x_t$, where α is a random number between 0 and 1.
4) Add x_{new} into Y.

end for

Procedure End

2.1 Dimensionality Reduction via t-SNE

Due to the curse of dimensionality, the commonly-used distance metrics that work well in low-dimensional space may have significantly-degraded performance in high-dimensional space. To alleviate this problem, dimensionality reduction is an important preprocessing step for many machine learning tasks such as clustering and classification. A lot of methods have been proposed to embed objects, described by either high-dimensional vectors or pairwise dissimilarities, into a lower-dimensional space [7]. Principal component analysis (PCA) [8] seeks to capture as much variance as possible. Multidimensional scaling (MDS) [9] tries to preserve dissimilarities between items. Traditional dimensionality reduction methods such as Principal Component Analysis and Multidimensional Scaling usually focus on keeping the low-dimensional representations of dissimilar data points far apart. However, for high-dimensional data that lies on or near a low-dimensional non-linear manifold, it is usually more important to keep the low-dimensional representations of very similar data points close together. Locally linear embedding (LLE) [10] attempts to preserve local geometry. Stochastic Neighbor Embedding (SNE) [11] is an iterative technique that aims at retaining the pairwise distances between the data points in the low-dimension space, which is similar to MDS. However, SNE differs from MDS in that it makes use of a Gaussian kernel such that the similarities of nearby points contribute more to the cost function. As such, it preserves mainly local properties of the manifold.

In this paper, we adopt a recently developed dimensionality reduction algorithm, called t-Distributed Stochastic Neighbor Embedding (t-SNE) [12], which is an extension to the well-known original Stochastic Neighbor Embedding (SNE) algorithm [11]. The t-SNE algorithm was proposed originally for the visualization of high-dimensional data points, which can transform the high-dimensional data set into two or three-dimensional data. The reason why we choose to use t-SNE is that it is capable of capturing much of the local structure of high-dimensional data very well, while also revealing global structure such as the presence of clusters. The first capability provides the quality of *K*-nearest neighbors, while the second capability makes it easy to discover the cluster structure of the minority class samples. Both these two capabilities are fundamental to the proposed MOT2LD algorithm. A brief description of t-SNE goes as follows.

In SNE [11] or t-SNE [12] algorithm, the high-dimensional Euclidean distances between data points are transformed into conditional probabilities that one data point would pick another data point as its neighbor.

$$p_{j|i} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2/(2\sigma_i^2)\right)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/(2\sigma_i^2))} \tag{1}$$

where σ_i is the variance of the Gaussian that is centered on data point x_i, and $\frac{\|x_i - x_k\|^2}{2\sigma_i^2}$ represents the dissimilarities between two data points that are measured as the scaled squared Euclidean distance. The value of σ_i is chosen by a binary search such that the Shannon entropy $\mathrm{H}(P_i) = -\sum_j p_{j|i}\log p_{j|i}$ of the distribution over neighbors equals to $\log u$, where u is a user-specified perplexity with 15 as default value.

In the high-dimensional space, the joint probabilities p_{ij} is defined to be the symmetrized conditional probabilities, that is:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2}$$

where n denotes the number of data points.

In t-SNE, a Student t-distribution with one degree of freedom is employed as the heavy-tailed distribution in the low-dimensional space. The joint distribution is defined as:

$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{3}$$

Based on the definitions (2) and (3), the goal of t-SNE is to minimize the difference between the two joint probability distributions P and Q. The Kullback-Leibler divergence between the two joint probability distributions P and Q, which measures their difference, is given by:

$$\mathrm{C} = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} = \sum_i \sum_j (p_{ij} \log p_{ij} + p_{ij} \log q_{ij}) \tag{4}$$

Therefore, we take the Kullback-Leibler divergence as the objective function to be minimized. Its gradient can be written as:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij}) \left(1 + \left\|y_i - y_j\right\|^2\right)^{-1} (y_i - y_j) \tag{5}$$

For the detailed derivation procedure of the expression (5), please refer to the Appendix A in [12].

A gradient descent method can be implemented to find out the map points in the low-dimensional space that minimizes the Kullback-Leibler divergence, following the gradient (5). To initialize the gradient descent process, we sample map points randomly from an isotropic Gaussian with small variance (10^{-8} by default) that is centered at the origin.

Through applying the t-SNE algorithm described above to the training samples inclusive of minority class and majority class, each sample is mapped to a point in a low-dimensional space. The map points in the low-dimensional space can better reveal the implicit structure of the high-dimensional data, especially when the high-dimensional data are lying on several different low-dimensional manifolds.

2.2 Density Peak Clustering in Low-Dimensional Space

Dimensionality reduction can be thought of as unsupervised distance metric learning, in that every dimensionality reduction approach can essentially learn a distance metric in the low-dimensional space [13]. Equivalently speaking, after high-dimensional data get mapped to map points in a low-dimensional space, we can derive a new distance

metric between data points in the low-dimensional space, which is normally of higher quality than the original distance metric in the high-dimensional space. A better distance metric usually leads to higher quality of calculated *K*-nearest neighbors or density, and in turn yields a better clustering result. The global cluster structure is helpful to synthetic oversampling methods, because each generated synthetic sample should be inside some minority cluster. MWMOTE [6] uses an average-linkage agglomerative clustering algorithm [14] to derive the cluster structure of minority class. In our method, we use a simple clustering algorithm called Density Peak Clustering (DPCluster in short) [15]. DPCluster assumes that the cluster centers are defined as local maxima in the density of data points, or in other words, the cluster centers are surrounded by neighbors with lower density. It also assumes that the cluster centers are at a relatively large distance from any points with a higher local density. According to these two assumptions, DPCluster calculates two quantities for each data point: one is its local density, and the other is its distance from points of higher density, which play important roles in the clustering solutions and are defined as follows [15].

Definition 1. (Local Density) The local density of a data point i is defined as:

$$\rho_i = \rho(i) = \sum_j \chi(d_{ij} - d_c) \tag{6}$$

where d_{ij} denotes the distance between two data points i and j, $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a cutoff distance.

In this paper, the distance between two data points is calculated as the Euclidean distance in the low-dimensional space, the value of d_c is set such that the average local density over all points equal to 2% of the total number of points. Because the clustering is applied only on minority class samples, the local density is also called the **local minority density** in this paper.

Definition 2. (Distance from points of higher density) For any data point i, its distance δ_i from points of higher density is measured as the minimum distance between the point and any other point with higher density:

$$\delta_i = \delta(i) = \min_{j:\rho_j > \rho_i} d_{ij} \tag{7}$$

For the data point with the highest local density y_i, δ_i is defined to be its maximal distance from any other point, that is, $\delta_i = \max_j d_{ij}$.

Based on those quantities, the clustering process consists of two steps. The first step is to identify the cluster centers which are the points with anomalously large value of ρ and relatively large value of δ, because cluster centers normally has high densities. In our implementation, this paper, a point is thought of as a cluster center if its local density is larger than 80% of all the data points, and its distance from points of higher density is among the top 5% of all the data points. As such, the number of clusters is automatically determined as the number of cluster centers identified, where each cluster center represents a unique cluster. In this step, the points with a high δ value

and a low ρ can be treated as outliers. The second step is to assign the remaining data points to the same cluster as its nearest neighbor of higher density. This assignment step is performed in a single pass, which is much faster than other clustering algorithms such as k-means [16].

2.3 Outlier Detection and Noise Filtering

In MOT2LD, we detect outliers and filter noises, in the low-dimensional space with two strategies (Step 3).

- Strategy 1 (Outlier Detection): During the clustering process described in section 2.2, we calculate the local minority density ρ_i for each minority class sample i. If the local minority density ρ_i equals to zero, then the sample i is likely to be an outlier, because there is no minority samples surrounding it. Therefore, it is deleted from the set of minority class samples, and gets removed from subsequent processing.
- Strategy 2 (Noise Filtering): We calculate the set $NN(i)$ of *K*-nearest neighbors for a minority class sample i in the low-dimensional space. If the *K*-nearest neighbors of a minority class sample in the low-dimensional space are all from the majority class, then the minority sample i is likely to be a noise sample because it is surrounded by only the majority class samples. It is then filtered out of the minority sample set.

2.4 Weight Assignment

As to measuring the importance of a minority class sample for synthetic minority sample generation, there are three facts that deserve our attention:

- The first fact is that the borderline points of a cluster normally have low local minority densities ρ, but the interior points usually have high local minority densities. For classification, the borderline points are more informative than the interior points. Therefore, the points with lower local densities should be given higher probabilities when chosen as the seed samples for generating synthetic minority samples. In other words, for a given minority sample i, if the number of minority class samples, whose distance to i is less than the cutoff distance d_c, is low, then its weight should be increased.
- The second fact is that for two minority clusters of different densities, the samples in the cluster of lower density should get more chances to serve as seed samples for generating synthetic minority samples than those in the cluster of higher density. This fact leads to the same conclusion as the first fact: minority samples of lower minority density should be given more weight.
- The third fact is that a minority sample is hard to make the correct decision if there are many majority class samples in its *K*-nearest neighbors. As such, we use the local majority count $\gamma(i)$ to indicate how many majority samples occur in the *K*-nearest neighbors of a given minority sample i. Minority samples with higher local majority count should be given higher probability of serving as seed samples for generating synthetic minority samples.

For a given minority class sample i, its importance $importance(i)$ is defined as the product of its local majority count $\gamma(i)$ and the inverse of its local minority density $\rho(i)$:

$$importance(i) = \frac{\gamma(i)}{\rho(i)} \tag{8}$$

The importance weight of a given minority class sample is an indicator of the importance for generating synthetic minority sample from it. A large weight implies that the sample needs to generate many synthetic minority samples nearby it.

To illustrate the rationale behind the weighting scheme, we construct a simple example explained as follows. In Fig. 1, there are totally 550 points. Among these points, 500 points represented as blue dots are drawn randomly from a bivariate Gaussian $\mathcal{N}(\mu_0, \Sigma_0)$, and 50 points as red plus-signs are drawn from another bivariate Gaussian $\mathcal{N}(\mu_1, \Sigma_1)$, where $\mu_0 = (1,2)$, $\mu_1 = (1,-1)$, and $\Sigma_0 = \Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$. Next, we shall examine three minority class samples labeled by 1, 2, and 3.

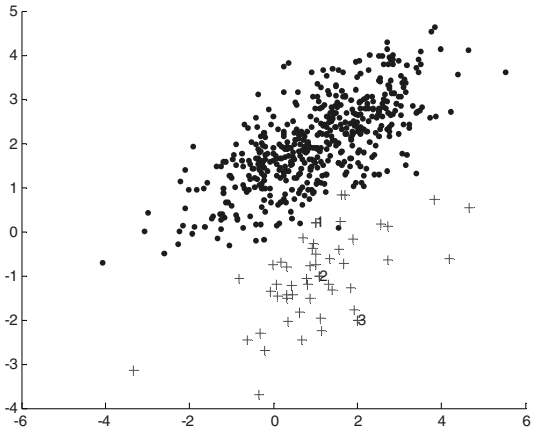

Fig. 1. An example of Gaussian-distributed Minority and Majority Samples

First, let us focus only on the minority class samples. The samples 1 and 3 are at the borderline of minority class, while the sample 2 is an interior point of minority class. As illustrated in Fig. 2, where each circle is centered at sample 1, sample 2, or sample 3, and the radius of each circle equals to the chosen cutoff value d_c. Clearly, the local minority density of sample 1, $\rho(1)$, equals to 5, because there are five minority class samples in its circle. Similarly, the local density of sample 2, $\rho(2)$, is 16, while the local density of sample 3, $\rho(3)$, is only 2.

Next, we examine the local majority count for the three minority samples. For the sample 1, there are two majority samples in its 5-nearest neighbors, so its majority count equals to 2, that is $\gamma(1) = 2$. For the sample 2 and the sample 3, there are no majority samples appearing in their 5-nearest neighbors, so we have $\gamma(2) = \gamma(3) = 0$.

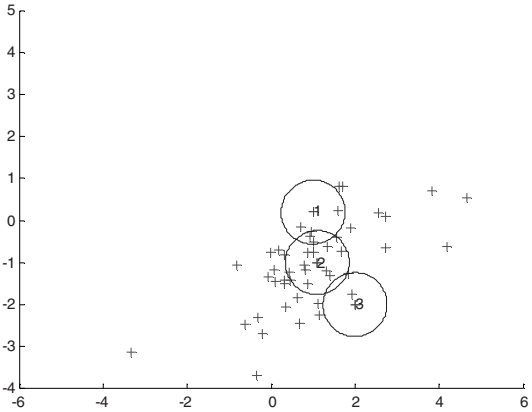

Fig. 2. Local minority densities for three minority class samples

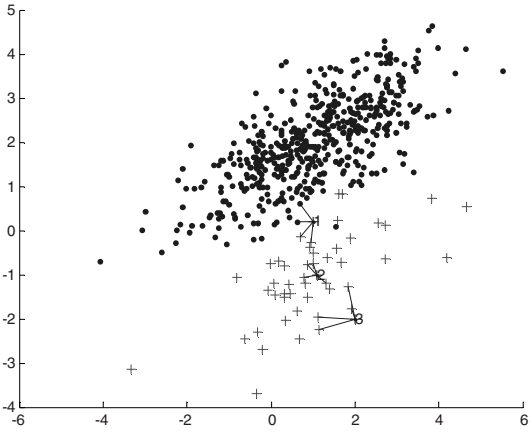

Fig. 3. Local majority counts for three minority class samples

Following the definition of importance weight in Equation (8), we can calculate the importance weights of the three minority samples:

$$Importance(1) = \frac{\gamma(1)}{\rho(1)} = \frac{2}{5};$$

$$Importance(2) = \frac{\gamma(2)}{\rho(2)} = 0;$$

$$Importance(3) = \frac{\gamma(3)}{\rho(3)} = 0.$$

Clearly, a minority class sample is given a high importance weight, if it has a high local majority count and a low local minority density.

2.5 Generation of Synthetic Minority Samples

Before we describe the generation of synthetic samples, we first transform the importance weights of minority samples into a probability distribution that indicates the probability that a minority sample is selected as the seed sample:

$$prob(i) = \frac{Importance(i)}{\sum_{j \in minority\ class} Importance(j)} \tag{9}$$

To generate a synthetic minority sample, a minority sample x_s is selected randomly as the seed sample according to the probability distribution. Let S denote the cluster that contains x_s. We then select a second minority sample x_t that belongs to the minority cluster S. A new synthetic minority sample is thus generated by random interpolation between the two minority samples x_s and x_t.

3 Experimental Results

To evaluate the effectiveness of the proposed MOT2LD method, we compare it with four other synthetic oversampling methods: SMOTE[3], Borderline-SMOTE[4], ADASYN[5], and MWMOTE[6], on 15 data sets from the UCI machine learning repository [17]. The data sets with more than two classes are transformed to two-class problems. Table 2 lists detailed information about the data sets and how the majority and minority classes.

On each data set, we randomly split it into two parts of (almost) the same size, one for training set and the other for testing set. Synthetic oversampling method is applied on the training set.

Accuracy is the most commonly-used evaluation metric for classification problems. However, the accuracy measure suffers greatly from the imbalanced class distribution, and thus is not suitable for imbalance classification [18]. To assess the classifier performance on imbalanced two-class classification problem, a confusion matrix is constructed as shown in Table 3, where *TP* denotes the number of true positive, *FP* denotes the number of false positive, *FN* denotes the number of false negative, and *TN* denotes the number of true negative. Two evaluation metrics derived from confusion matrix are used in this paper to assess learning from imbalanced data sets. They are G-mean and F-measure [18].

Table 2. Characteristics of the experimental data sets

Data Sets	Minority Class	Majority Class	Features	Minority	Majority	Imbalance Ratio
Statlandsat	4	other	37	415	4435	0.09:0.91
Yeast	ME3, ME2, EXC, VAC, POX, ERL	other	8	304	1180	0.21:0.79
Ecoli	im	other	7	77	259	0.23:0.77
PageBlocks	Graphic, Vert.line, Picture	other	10	231	5245	0.04:0.96
BreastCancer	Malignant	Benign	9	239	444	0.34:0.66
Glass	5, 6, 7	other	9	51	163	0.24:0.76
Vehicle	van	other	18	199	647	0.24:0.76
Libra	1, 2, 3	other	90	72	288	0.20:0.80
Abalone	18	other	7	42	689	0.06:0.94
Vowel	0	other	10	90	900	0.09:0.91
Pima	1	0	8	268	500	0.35:0.65
Ionosphere	bad radar	good radar	34	126	225	0.36:0.64
Segment	Grass	other	19	330	1980	0.14:0.86
BreastTissue	CAR, FAD	other	9	36	70	0.34:0.66
Wine	3	other	13	48	130	0.26:0.74

Table 3. Confusion Matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Based on the confusion matrix in Table 3, the evaluation metrics, G-mean and F-measure, are defined as follows:

- G-mean is a good indicator for performance assessment of imbalanced learning by combining the accuracies on the positive class and negative class samples.

$$G_{Mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$

where $\frac{TP}{TP+FN}$ is the accuracy on the positive class and $\frac{TN}{TN+FP}$ is the accuracy on the negative class.

- F-measure make a combination of precision and recall of the positive samples:

$$\mathrm{F}_{\mathrm{measure}} = \frac{2 \times recall \times precision}{recall + precision}$$

where $recall = \frac{TP}{TP+FN}$ and $precision = \frac{TP}{TP+FP}$.

We use the CART [16] decision tree as the classification model in our experiments. Throughout the experiments, we do not fine-tune the parameters in our algorithm. All the parameters take default values as indicated in Section 2. Table 4 and Table 5 summarize the results of SMOTE, Borderline-SMOTE, ADASYN, and MOT2LD on the 15 experimental data sets. The reported performance results are all averaged over 20 independent runs. At each run, the data set is randomly divided into two parts of approximately equal size: one for training set and the other for testing set. The number of synthetic minority samples that generated by the compared oversampling methods is two times the number of minority samples in the training set. On each data set, the best result is highlighted with underlined bold-face type.

Table 4. Comparison of G-mean on experimental data sets

Data Sets	ADASYN	Borderline-SMOTE	SMOTE	MWMOTE	MOT2LD
Statlandsat	0.717	0.714	**0.723**	0.719	**0.723**
Yeast	**0.783**	0.781	**0.783**	0.779	0.778
Ecoli	0.830	0.824	0.844	0.836	**0.851**
PageBlocks	**0.869**	0.858	0.861	0.858	0.868
BreastCancer	0.898	0.903	0.909	**0.910**	0.907
Glass	0.884	**0.893**	0.887	0.880	0.892
Vehicle	0.891	0.900	0.900	0.894	**0.903**
Libra	0.744	0.742	0.764	0.761	**0.769**
Abalone	0.557	0.561	0.5998	0.580	**0.619**
Vowel	0.947	0.923	0.941	0.927	**0.947**
Pima	0.664	0.648	0.662	0.658	**0.669**
Ionosphere	0.835	**0.857**	0.829	0.849	0.824
Segment	**0.997**	0.996	0.996	**0.997**	0.996
BreastTissue	0.717	0.712	0.692	0.681	**0.725**
Wine	0.949	**0.949**	0.943	0.947	0.947

Table 5. Comparison of F-measures on experimental data sets

Data Sets	ADASYN	Borderline-SMOTE	SMOTE	MWMOTE	MOT2LD
Statlandsat	0.513	0.514	**0.519**	0.507	0.507
Yeast	0.646	**0.650**	0.647	0.643	0.642
Ecoli	0.741	0.735	0.758	0.749	**0.765**
PageBlocks	**0.728**	0.726	0.718	0.682	0.700
BreastCancer	0.890	0.896	0.902	**0.904**	0.900
Glass	0.835	**0.848**	0.832	0.828	**0.848**
Vehicle	0.836	0.845	0.846	0.835	**0.847**
Libra	0.630	0.641	**0.650**	0.643	0.636
Abalone	0.304	0.322	0.321	0.334	**0.359**
Vowel	**0.876**	0.863	0.867	0.864	0.858
Pima	0.575	0.554	0.573	0.567	**0.580**
Ionosphere	0.790	**0.818**	0.781	0.804	0.776
Segment	0.995	**0.996**	0.995	**0.996**	**0.996**
BreastTissue	0.635	0.630	0.605	0.594	**0.646**
Wine	**0.923**	**0.923**	0.917	0.916	**0.923**

From Table 4 and Table 5, it can be seen that MOT2LD has achieved 8 best results out of the 15 data sets among all the compared methods, in both G-mean and F-measure, which is much better than the others including SMOTE, Borderline-SMOTE, ADASYN, and MWMOTE, which have mostly achieved 2 or 3 best results out of the 15 data sets.

4 Conclusion and Future Work

In this paper, we propose a new synthetic oversampling method MOT2LD for imbalanced learning. MOT2LD first maps samples into a low-dimensional space using t-SNE algorithm, and discovers the cluster structure of the minority class in the low-dimensional space by DPCluster. It then assigns importance weights to minority samples as the products of the local majority count and the inverse of local minority density.

To finalize this paper, we would like to list several directions for our future work:

Firstly, it may be interesting to study the effect of supervised dimensionality reduction technique as a proprecessing step. If we could make use of supervised information in the dimensionality reduction algorithm to maximize the separation between minority class and majority class, it is expected that a better results would be achieved.

Secondly, although synthetic oversampling methods have achieved satisfactory results for imbalanced learning, a lot of other methods do exist. Recently, there are some model-based oversampling methods such as SPO [20][21] and MoGT [22]. SPO [20][21] assumes that the minority samples follow a multivariate Gaussian distribution. It estimates its mean vector and covariance matrix and then draws extra minority sample from the probability distribution. MoGT [22] assumes another probabilistic model called mixture of Gaussian Trees. It is similar to the Gaussian Mixture model, but differs in that Gaussian Tree can be thought of as a restricted kind of Gaussian distribution, which has much less parameters to be estimated. How to combine synthetic oversampling methods and model-based oversampling ones is a challenging problem.

Acknowledgements. This work is supported by National High-tech R&D Program of China (863 Program) (No. SS2015AA011809), Science and Technology Commission of Shanghai Municipality (No. 14511106802), and National Natural Science Fundation of China (No. 61170007). We are grateful to the anonymous reviewers for their valuable comments.

References

1. Fawcett, T.E., Provost, F.: Adaptive Fraud Detection. Data Min. Knowl. Disc. **3**(1), 291–316 (1997)
2. Mladenić, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive bayes. In: Proceedings of the 16th International Conference on Machine Learning, pp. 258–267 (1999)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
4. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: a new oversampling method in imbalanced data sets learning. In: Proceedings of International Conference on Intelligent Computing, pp. 878–887 (2005)
5. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1322–1328 (2008)
6. Barua, S., Islam, M.M., Yao, X., Murase, K.: MWMOTE - majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans. Knowl. Data Eng. **26**(2), 405–425 (2014)
7. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality reduction: a comparative review. Tilburg University Techical Report, TiCC-TR 2009–005 (2009)
8. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**, 417–441 (1933)
9. Torgerson, W.S.: Multidimensional scaling I: theory and method. Psychometrika **17**, 401–419 (1952)
10. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by Locally Linear Embedding. Science **290**(5500), 2323–2326 (2000)
11. Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: Advances in Neural Information Processing Systems, vol. 15, pp. 833–840 (2002)
12. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)

13. Liu, Y.: Distance metric learning: a comprehensive survey. Research Report, Michigan State University (2006)
14. Voorhees, E.M.: Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Inf. Process. Manage. **22**(6), 465–476 (1986)
15. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**, 1492–1496 (2014)
16. MacQueen, J.: Some methods for classifications and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability, University of California Press, pp. 281–297 (1967)
17. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, 2013 [http://archive.ics.uci.edu/ml]
18. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
19. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC press (1984)
20. Cao, H., Li, X.L., Woon, Y.-K., Ng, S.K.: SPO: structure preserving oversampling for imbalanced time series classification. In: Proceedings of IEEE International Conference on Data Mining (2011)
21. Cao, H., Li, X.L., Woon, Y.K., Ng, S.K.: Integrated oversampling for imbalanced time series classification. IEEE Trans. Knowl. Data Eng. **25**(12), 2809–2822 (2013)
22. Pang, Z.F., Cao, H., Tan, Y.F.: MOGT: oversampling with a parsimonious mixture of Gaussian trees model for imbalanced time-series classification. In: Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6 (2013)