Global-and-Local Aware Data Generation for the Class Imbalance Problem

Wentao Wang*[†] Suhang Wang[‡] Wenqi Fan[§] Zitao Liu[¶] Jiliang Tang*

Abstract

In many real-world classification applications such as fake news detection, the training data can be extremely imbalanced, which brings challenges to existing classifiers as the majority classes dominate the loss functions of classifiers. Oversampling techniques such as SMOTE are effective approaches to tackle the class imbalance problem by producing more synthetic minority samples. Despite their success, the majority of existing oversampling methods only consider local data distributions when generating minority samples, which can result in noisy minority samples that do not fit global data distributions or interleave with majority classes. Hence, in this paper, we study the class imbalance problem by simultaneously exploring local and global data information since: (i) the local data distribution could give detailed information for generating minority samples; and (ii) the global data distribution could provide guidance to avoid generating outliers or samples that interleave with majority classes. Specifically, we propose a novel framework GL-GAN, which leverages the SMOTE method to explore local distribution in a learned latent space and employs GAN to capture the global information, so that synthetic minority samples can be generated under even extremely imbalanced scenarios. Experimental results on diverse real data sets demonstrate the effectiveness of our GL-GAN framework in producing realistic and discriminative minority samples for improving the classification performance of various classifiers on imbalanced training data. Our code is available at `https://github.com/wentao-repo/GL-GAN`.

Keywords: imbalanced data, adversarial learning

1 Introduction

The classification performance heavily relies on the quality and quantity of the training data [11]. However, in many real-world applications, due to some practical concerns such as privacy and time cost, only limited labeled data can be collected. Meanwhile, such data could be imbalanced. Specifically, some classes have significantly larger number of data samples while others have very limited amount of data, which is called class imbalance problem [10]. For instance, in fake news detection [24], the majority of news in the collected data are true news while only a small portion of news are fake news. The imbalanced data has negative impacts on the classifier training since the standard classifiers tend to be overwhelmed by the majority classes while ignoring the minority classes [3]. Furthermore, even though minority classes may only take extremely small ratio of one data set, for some applications like medical diagnosis, misclassifying a minority class sample is usually more severe than misclassifying a majority one [16].

Oversampling has been proven to be an effective way to alleviate the class imbalance problem by oversampling minority samples into the imbalanced data set [17]. As one of the most popular oversampling methods, Synthetic Minority Over-sampling Technique (SMOTE) [2] generates new synthetic minority samples by performing linear interpolation operations between existing minority samples and their nearest neighbors within the same class. As shown in Figure 1, by applying the SMOTE method, new synthetic minority samples are generated along with the linear interpolation between two existing minority samples.

Despite the success of SMOTE and its variants [8, 9], they still face some challenges. First, SMOTE-based methods only consider the local neighbor relationship of each minority sample, while the global distribution is totally ignored. Without considering the global distribution of the given data, the generated minority samples could not fit the real data distribution. For instance, the generated samples in Figure 1 are either located on the null space of the given data samples or interleaved with majority data samples. Second, the interpolation operations performed by these methods on raw feature space may not generate realistic data samples. For instance, for the given text data which lies in discrete space, SMOTE-based methods cannot guarantee their generated texts are readable.

Therefore, in this paper, we study the class imbalance problem by simultaneously exploring both global and local information. The local data distribution provides detailed local information for generating minority samples; and the global data distribution provides guid-

*Michigan State University. {wangw116, tangjili}@msu.edu.
[†] Work was done when interned at TAL AI Lab.
[‡] Pennsylvania State University. szw494@psu.edu.
[§] City University of Hong Kong. wenqifan03@gmail.com.
[¶] TAL AI Lab, TAL Education Group. liuzitao@100tal.com.

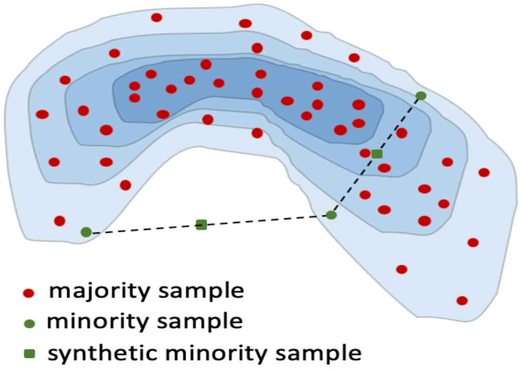


Figure 1: An example of imbalanced data and SMOTE method. The synthetic minority samples are generated along the dash line between two minority samples.

ance from a global view to avoid generating samples that interleave with majority samples or fall in the null space of the given data. We are faced with two challenges: (i) how to explore global data distribution for minority sample generation; and (ii) how to simultaneously leverage global and local distribution information to generate realistic and discriminative synthetic minority samples. Recently, generative adversarial learning [7] has shown promising results in generating realistic data samples [7, 20] through estimating the latent global data distribution, which paves us a way to solve these two challenges. Hence, we propose a novel framework which leverages oversampling techniques to capture local data structure and generative adversarial learning to explore global data distribution. The contributions of our work are summarized below:

- We identify the importance of both global and local distribution information in tackling the class imbalance problem.
- We propose a novel generative adversarial framework, GL-GAN, to generate realistic and discriminative minority samples by exploring both global and local distributions.
- We conduct extensive experiments on diverse real data sets to demonstrate the effectiveness of GL-GAN on alleviating the class imbalance problem.

The rest of this paper is organized as follows. Section 2 summarizes related works. Our proposed GL-GAN framework is introduced in Section 3. Empirical studies and case studies are reported in Section 4 and Section 5 separately. Finally, we conclude this work in Section 6.

2 Related Work

Existing works for tackling the class imbalance problem can be roughly classified into three categories: data-level methods, algorithm-level methods and hybrid methods [13]. Our GL-GAN is a data-level method.

Undersampling [25, 16] and oversampling [2, 8, 9] are two fundamental data-level solutions. Briefly, undersampling approaches downsize the majority class by removing majority samples, while oversampling approaches upsize the minority class by generating minority samples [15]. Oversampling with replacement, also called random oversampling [6], is the simplest oversampling approach that randomly duplicates existing minority samples to augment the minority class. However, the random oversampling method often makes the decision boundary of the classifier smaller and causes the classifier to over-fit [8]. As an improved approach, SMOTE [2] inflates the minority class by producing synthetic minority samples instead of duplicating existing minority samples. Different from SMOTE-based methods [2, 8, 9] that utilize Euclidean distance to perform interpolation operations, some recent work [1, 23] introduced Mahalanobis distance into synthetic minority samples generation process and achieved good performance on classifier training.

Recently, more and more researchers have been attracted by the generative adversarial learning due to its great power on generating different kinds of realistic synthetic data samples. The pioneer work introduced by [7] presented Generative Adversarial Networks (GAN) to learn the real data distribution through a minimax game between a generator G and a discriminator D. The generator G produces synthetic samples to fool the discriminator D, while the discriminator D judges whether the input samples come from the generator or from the real data set. These two components fight against each other and improve themselves gradually [5]. Some recent research applied generative adversarial learning to solve the class imbalance problem. For instance, conditional GAN [20] is adopted in [4] for producing minority samples effectively. BAGAN [18], is a data augmentation model that can alleviate the class imbalance problem by modifying the discriminator D in the traditional GAN. However, the local structure of the given minority samples is not explored by these aforementioned models, so some generated synthetic samples may be close to the decision boundary and hard to be utilized to train a classifier.

Our GL-GAN is inherently different from existing works and, hence, able to generate more realistic and discriminative synthetic minority samples.

3 The Proposed Framework

In this paper, we focus on the binary class imbalance problem. Given an imbalanced sample set $\mathcal{X}_{org}$ containing a majority sample set $\mathcal{X}_{maj}$ and a minority sample set $\mathcal{X}_{min}$, our goal is to generate a set of realistic and

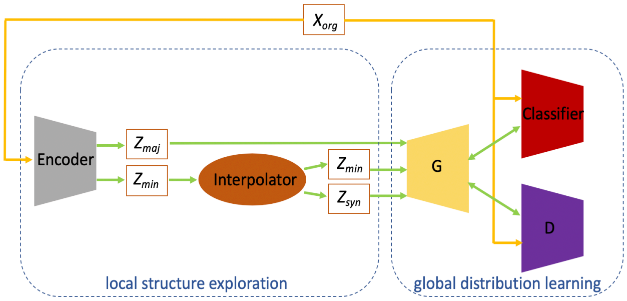


Figure 2: An overview of GL-GAN.

discriminative synthetic minority samples $\mathcal{X}_{syn}$ so that, comparing with training only on the original imbalanced sample set $\mathcal{X}_{org}$, the classification performance of classifiers can be greatly improved by training on the balanced augmentation sample set $\mathcal{X}_{org} \cup \mathcal{X}_{syn}$.

As shown in Figure 2, our GL-GAN is composed of two modules, local structure exploration and global distribution learning. The former is designed for generating latent representations of minority samples through exploring the local distribution information, and the latter aims to produce realistic and discriminative minority samples that can fit the global distribution. Next, we will introduce details of each module.

3.1 Local Structure Exploration

The local structure exploration module consists of two components, i.e., an encoder E and a local data representation interpolator I, specifying for two different tasks separately.

3.1.1 Discriminative Representation Learning

In many cases, directly generating synthetic data samples in raw feature space by local-based oversampling techniques such as SMOTE may cause several problems.

Firstly, as we demonstrated before, these methods cannot generate realistic synthetic samples for some specific data types like text data. Secondly, the generated minority data samples may interleave with majority samples. This motivates us to first learn discriminative latent representations of the raw data, then exploit the local data structure in the learned latent space. The advantages of doing this are as follows: (i) By learning a low-dimensional latent representation, we can preserve the most important information of the data while drop some noisy information; and (ii) During the latent representation learning process, we can enforce the latent representations of data samples belonging to the same class to be closed to each other.

Deep autoencoders have been proved to be an effective way to extract important information from high-dimensional data using low-dimensional representations [22]. Typically, an autoencoder consists of two components: an encoder E and a decoder Q. The encoder E takes the high-dimensional data as input and maps them to the corresponding latent representations. The decoder Q recovers these learned latent representations back to the raw feature space. The goal of training an autoencoder is to minimize the reconstruction error between the input data and the reconstructed data produced by the decoder Q, which can be defined as

$$\mathcal{L}_{rec}(Q\left(E\left(\mathcal{X}_{org}\right)\right), \mathcal{X}_{org}) = \frac{1}{|\mathcal{X}_{org}|} \sum_{x_i \in \mathcal{X}_{org}} \|Q\left(E\left(x_i\right)\right) - x_i\|_2^2. \tag{3.1}$$

In our GL-GAN framework, we propose to embed the given real data samples into a latent space with majority samples in one cluster and minority samples in another cluster, and these two clusters should be faraway from each other. To do that, we aim to reduce the interleaving between the synthetic generated minority samples and the given majority samples. Formally, this process can be described by

$$\begin{aligned} \mathcal{L}_{clu} = \frac{1}{|\mathcal{X}_{maj}|} \sum_{x_i \in \mathcal{X}_{maj}} \|E(x_i) - \overline{z}_{maj}\|_2^2 + \frac{\lambda_1}{|\mathcal{X}_{min}|} \\ \sum_{x_i \in \mathcal{X}_{min}} \|E(x_i) - \overline{z}_{min}\|_2^2 - \lambda_2 \|\overline{z}_{maj} - \overline{z}_{min}\|_2^2, \end{aligned} \tag{3.2}$$

where $\overline{z}_{maj}$ and $\overline{z}_{min}$ are mean of the latent representations of majority sample set X_{maj} and minority sample set X_{min}, respectively. λ_1 and λ_2 are two hyper-parameters controlling the weights. Starting from here, we use Λ or λ to represent hyper-parameters.

Therefore, the autoencoder in our GL-GAN can be trained by minimizing the following loss function:

$$\mathcal{L}_A = \mathcal{L}_{rec} + \Lambda_1 \mathcal{L}_{clu} + \Lambda_2 R(\theta). \tag{3.3}$$

Here $R(\theta)$ is the regularizer of the model parameters θ. Once the autoencoder is trained well, the latent representation of sample x_i can be given as $z_i = E(x_i)$.

3.1.2 Local-based Data Generation

With the learned latent representations, we can generate synthetic minority samples in the latent space by exploring the local structure of the sample set $\mathcal{Z}_{min}$, which is the latent embedding of the minority sample set $\mathcal{X}_{min}$. In our GL-GAN, we adopt SMOTE as the implementation of the local data interpolator I because of its simplicity.

For any minority sample $z_i \in \mathcal{Z}_{min}$, SMOTE 1) discovers k nearest neighbors $\{z_i^1, z_i^2, \ldots, z_i^k\}$ of z_i within the same minority class set $\mathcal{Z}_{min}$, 2) randomly picks up any one nearest neighbor z_i^n $(n \in [1, k])$ from the set $\{z_i^1, z_i^2, \ldots, z_i^k\}$ and chooses a random number $\eta \in [0, 1]$. Hence, a new synthetic minority sample z_i' could be created by $z_i' = z_i + \eta\left(z_i^n - z_i\right)$. The second step can be repeated N times, and, finally, $N \times |\mathcal{Z}_{min}|$ synthetic minority samples will be generated when executing the same process on every minority sample in $\mathcal{Z}_{min}$. After the synthetic minority sample set Z_{syn} is obtained, we can get a balanced augmentation sample set $\mathcal{Z} = \mathcal{Z}_{maj} \cup \mathcal{Z}_{min} \cup \mathcal{Z}_{syn}$ in the latent space.

3.2 Global Distribution Learning For making the generated minority samples in $\mathcal{Z}_{syn}$ more realistic and discriminative, we introduce a generative adversarial learning model to learn the global information of given samples and modify samples in $\mathcal{Z}_{syn}$ accordingly.

3.2.1 Discriminator *D* The role of the discriminator D is to differentiate if a data sample is real or fake. For a data sample who comes from the given real data set, the discriminator D labels it as a real sample. If a data sample is synthetically generated by the generator G, it will be classified as a fake sample. The discriminator D and the generator G fight against each other and improve themselves gradually. The loss function for training the discriminator D can be written as

(3.4)
$$\mathcal{L}_D = \frac{1}{|\mathcal{X}_{org}|} \sum_{x_i \in \mathcal{X}_{org}} \|D(x_i) - 1\|_2^2 + \frac{\lambda}{|\mathcal{Z}|} \sum_{z_i \in \mathcal{Z}} \|D(G(z_i)) - 0\|_2^2.$$

In equilibrium, the discriminator D cannot find the difference between real and synthetic samples, which means the quality of synthetic data generated by the generator G are approximate to the real data.

3.2.2 Classifier *C* For making sure that generated data samples can have expected labels, we introduce a classifier C in our GL-GAN. Specifically, the classifier C also takes both real samples and synthetic samples generated by the generator G as input. Since the input of G in GL-GAN is the balanced augmentation sample set $\mathcal{Z}$, every output of the generator G, i.e. $G(z_i)$, has its corresponding label. The classifier C works on labeled data samples and makes classification for them. The loss function for training the classifier C in our GL-GAN is

(3.5)
$$\mathcal{L}_C = \frac{1}{|\mathcal{X}_{org}|} \sum_{x_i \in \mathcal{X}_{org}} \|C(x_i) - \Gamma_{x_i}\|_2^2 + \frac{\lambda}{|\mathcal{Z}|} \sum_{z_i \in \mathcal{Z}} \|C(G(z_i)) - \Gamma_{z_i}\|_2^2.$$

Here Γ_{x_i} and Γ_{z_i} are true labels of real sample x_i and latent representation z_i, respectively. By introducing the classifier C into the traditional GAN, the generator G is forced to produce synthetic samples which can be classified by C correctly.

3.2.3 Generator *G* Different from the traditional generator G that takes a set of random noise following some prior distribution as input, during the model training phase, the generator G in our GL-GAN is fed with the balanced augmentation sample set $\mathcal{Z}$. Since there are two types of latent representations in $\mathcal{Z}$, i.e., the latent representations of real samples in $\mathcal{Z}_{maj}$ and $\mathcal{Z}_{min}$, denoted as $\mathcal{Z}_{org} = \mathcal{Z}_{maj} \cup \mathcal{Z}_{min}$, and the latent representations of synthetic samples in $\mathcal{Z}_{syn}$, the generator G should be able to project latent representations $\mathcal{Z}_{org}$ back to the raw feature space as well as produce synthetic data samples that can fool the discriminator D. Therefore, the loss for training generator G includes three different types: the reconstruction loss $\mathcal{L}_{rec}$ for mapping latent representations $\mathcal{Z}_{org}$ back to the raw feature space, the discriminator loss $\mathcal{L}_{(G,D)}$ produced by the discriminator D for evaluating the difference between the real data samples and data samples generated by G, and the classifier loss $\mathcal{L}_{(G,C)}$ brought by the classifier C for making classification on the generated data samples of G. Formally, the loss function for training the generator G in our GL-GAN can be defined as

(3.6)
$$\begin{aligned}\mathcal{L}_G &= \mathcal{L}_{rec}(G(\mathcal{Z}_{org}), \mathcal{X}_{org}) + \lambda_1 \mathcal{L}_{(G,D)} + \lambda_2 \mathcal{L}_{(G,C)} \\ &= \frac{1}{|\mathcal{X}_{org}|} \sum_{x_i \in \mathcal{X}_{org}, z_i \in \mathcal{Z}_{org}} \|G(z_i) - x_i\|_2^2 + \frac{\lambda_1}{|\mathcal{Z}|} \\ &\quad \sum_{z_i \in \mathcal{Z}} \|D(G(z_i)) - 1\|_2^2 + \frac{\lambda_2}{|\mathcal{Z}|} \sum_{z_i \in \mathcal{Z}} \|C(G(z_i)) - \Gamma_{z_i}\|_2^2.\end{aligned}$$

After the whole framework is trained well, the generator G is able to produce a set of realistic and discriminative synthetic minority samples.

3.3 Objective Function of GL-GAN With local structure exploration module and global distribution learning module introduced above, the final objective function of GL-GAN is given as:

$$\text{(3.7)} \quad \min_{\theta_G, \theta_C} \max_{\theta_D} \mathcal{L}_{rec}(G(\mathcal{Z}_{org}), \mathcal{X}_{org}) + \Lambda_1 \mathcal{L}_{(G,D)} + \Lambda_2 \mathcal{L}_{(G,C)}$$

where θ_G, θ_C and θ_D are the parameters of generator G, classifier C and discriminator D, respectively.

3.4 Algorithm In this subsection, we present our GL-GAN framework in Algorithm 1.

As shown in Algorithm 1, we train the autoencoder part at first to make sure the autoencoder could map the input data samples into two far-way clusters in the latent space. After pre-training the autoencoder, we utilize the encoder E to obtain the latent representations of the input data samples. Then, the local data interpolator I can be applied in the learned latent space to generate a set of synthetic minority samples within the same cluster. In order to train the generative adversarial learning part more effectively, we use the knowledge learned by the pre-trained autoencoder to initialize the generative model. Specifically, the discriminator D and

Algorithm 1 The algorithm of GL-GAN.

```
Require: an imbalanced sample set $X_{org}$
Initialize the parameters of autoencoder.
Pre-train the autoencoder to obtain the latent representations $\mathcal{Z}_{org} = \mathcal{Z}_{maj} \cup \mathcal{Z}_{min}$ of $\mathcal{X}_{org}$.
Apply SMOTE method for $\mathcal{Z}_{min}$ to get the synthetic minority sample set $\mathcal{Z}_{syn}$.
Form a balanced augmentation sample set $\mathcal{Z} = \mathcal{Z}_{maj} \cup \mathcal{Z}_{min} \cup \mathcal{Z}_{syn}$ in the learned latent space.
repeat
    for discriminator-epochs do
        Train the discriminator $D$ with augmented latent sample set $\mathcal{Z}$ and real sample set $\mathcal{X}_{org}$. (Sec. 3.2.1)
    end for
    for classifier-epochs do
        Train the classifier $C$ with augmented latent sample set $\mathcal{Z}$ and real sample set $\mathcal{X}_{org}$. (Sec. 3.2.2)
    end for
    for generator-epochs do
        Train the generator $G$. (Sec. 3.2.3)
    end for
until model convergence
```

the classifier C have the same architecture with the encoder E except both D and C have one more layer. The last layer of the discriminator D is a dense layer with a softmax activation function for producing binary outputs and the last layer of the classifier C is a dense layer for producing classification results. The parameters learned by the encoder E will be used to initialize the discriminator D and the classifier C during the generative model training phase. Similarly, the generator G is initialized by the weight parameters learned by the decoder Q since they have the same architecture.

4 Experiments

In this section, we conduct experiments to verify the effectiveness of our proposed GL-GAN framework. We aim at answering the following two questions:

- Can the proposed GL-GAN framework generate discriminative minority samples for improving the classification performance of imbalanced data?
- What is the impact of each module of GL-GAN?

We begin by introducing the data sets and experimental settings, then we compare GL-GAN with several state-of-the-art related methods on the classification task to answer the first question. We then analyze the impact of each module of GL-GAN to answer the second question.

4.1 Experimental Settings In order to test how the generated synthetic samples alleviate the binary class imbalance problem, we utilize the classification performance of different classifiers training on various augmented sample sets as the evaluation indicator.

4.1.1 Data Sets The experiments are conducted on five real data sets, i.e., USPS, Sensorless Drive Diagnosis, Gas Sensor Array Drift, Madelon and Gisette. Sensorless Drive Diagnosis and Gas Sensor Array Drift are publicly from the UCI data repository[1] and the rest three can be obtained from Feature Selection data repository[2]. Since all these five data sets are class balanced, we construct the imbalanced data set for each of them according to the following three steps. Firstly, we randomly choose one class as majority class and another one as minority class to obtain a balanced binary data set. Then we divide 80% data samples of the balanced binary class data set as the candidates set and the rest as the test set. Lastly, we artificially imbalance the candidates set to form the imbalanced data set by utilizing a predefined imbalanced ratio r. For instance, if $r = 0.01$, then 99% minority samples will be removed from the candidates set so that the ratio of the minority samples to the majority samples in the imbalanced data set is 0.01. Table 1 provides the statistical information of five imbalanced data sets obtained by the aforementioned three steps when the imbalanced ratio $r = 0.01$.

Table 1: Statistical information of imbalanced data sets.

Data Set	# Features	# Majority	# Minority
USPS	256	744	7
Sensorless Drive Diagnosis	48	4256	42
Gas Sensor Array Drift	128	1549	15
Madelon	500	1040	10
Gisette	5000	2800	28

4.1.2 Classifiers Since our goal is to generate synthetic minority samples for improving the classification performance, we introduce several classifiers to help to evaluate the quality of the generated samples. Three representative classifiers, i.e., Multi-layer Perceptron (MLPClassifier), Linear Support Vector Classification (LinearSVC) and AdaBoost are adopted in our experiments. We train these classifiers on the training sets augmented by the synthetic minority samples generated by our model or baselines and test them on the corresponding test data sets. All these three classifiers are implemented by the *scikit-learn* package[3] in Python, and we use their default settings in all experiments.

4.1.3 Evaluation Metrics For measuring the classification performance of classifiers, we introduce three different metrics, macro F1-score, micro F1-score and Matthews correlation coefficient (MCC) [19] into our experiments. The value of MCC is in the range $[-1, 1]$, in

[1] https://archive.ics.uci.edu/ml/index.php

[2] http://featureselection.asu.edu/datasets.php

[3] https://scikit-learn.org/stable/index.html

Table 2: Classification performance of classifiers on the USPS data set.

Evaluation		Method							
Classifier	Metrics	Imbalanced	Random	SMOTE	MDO	NRAS	SWIM	BAGAN	GL-GAN
MLPClassifier	macro F1	0.7712	0.8840	0.8825	0.8914	0.8415	0.6443	0.8208	**0.8937**
	micro F1	0.7969	0.8899	0.8887	0.8947	0.8503	0.6595	0.8356	**0.8985**
	MCC	0.6232	0.7839	0.7823	0.7858	0.7022	0.4731	0.6872	**0.7990**
LinearSVC	macro F1	0.8473	0.8580	0.8580	0.8834	0.8408	0.6184	0.8836	**0.8912**
	micro F1	0.8589	0.8681	0.8680	0.8865	0.8497	0.6380	0.8896	**0.8957**
	MCC	0.7346	0.7510	0.7510	0.7683	0.7010	0.4440	0.7838	**0.7913**
AdaBoost	macro F1	0.8024	0.7878	0.8036	0.8436	0.7883	0.8900	0.7848	**0.8920**
	micro F1	0.8218	0.8095	0.8221	0.8558	0.8098	0.8906	0.8077	**0.8966**
	MCC	0.6689	0.6441	0.6662	0.7291	0.6444	0.7865	0.6433	**0.7942**

Table 3: Classification performance of classifiers on the Sensorless Drive Diagnosis data set.

Evaluation		Method							
Classifier	Metrics	Imbalanced	Random	SMOTE	MDO	NRAS	SWIM	BAGAN	GL-GAN
MLPClassifier	macro F1	0.7697	0.8642	0.8700	0.8580	0.8510	0.8439	0.9078	**0.9334**
	micro F1	0.7809	0.8666	0.8722	0.8608	0.8542	0.8476	0.9086	**0.9338**
	MCC	0.6251	0.7608	0.7699	0.7513	0.7406	0.7299	0.8310	**0.8755**
LinearSVC	macro F1	0.8195	0.8425	0.8425	0.8455	0.8420	0.8435	**0.9281**	0.8714
	micro F1	0.8250	0.8462	0.8462	0.8490	0.8457	0.8471	**0.9285**	0.8735
	MCC	0.6939	0.7277	0.7277	0.7322	0.7269	0.7292	**0.8659**	0.7721
AdaBoost	macro F1	0.8962	0.9683	0.9686	0.9955	0.9835	0.9924	0.9817	**0.9959**
	micro F1	0.8973	0.9683	0.9687	0.9955	0.9835	0.9924	0.9817	**0.9959**
	MCC	0.8119	0.9385	0.9392	0.9910	0.9676	0.9849	0.9638	**0.9918**

which MCC = 1 indicates a perfect prediction, MCC = 0 means the prediction made by a classifier is no better than the random prediction and MCC = -1 represents total wrong between the prediction and the observation.

4.2 Effectiveness Evaluation For evaluating the effectiveness of our GL-GAN framework on alleviating the binary class imbalance problem, we compare the quality of the synthetic samples generated by GL-GAN with several representative and state-of-the-art oversampling methods, including: 1) Imbalanced, which directly uses original imbalanced data sets without adding minority samples; 2) Random [6], i.e., random oversampling, which inflates minority class by duplicating existing minority samples; 3) SMOTE [2], which generates minority samples by performing linear interpolation operations between minority samples and their nearest neighbors; 4) MDO [1], which produces minority samples that have the same Mahalanobis distance from the considered class mean with existing minority samples; 5) NRAS [21], which performs a noise removal process on the minority class first and then constructs synthetic samples from the remaining samples; 6) SWIM [23], which utilizes the distribution information of majority class to generate minority samples located at the same Mahalanobis distance from the majority class; and 7) BAGAN [18] which takes random noise as input and produces synthetic samples to balance the imbalanced data set. We adopt the implementations of Random and SMOTE methods provided by literature [14] and of MDO and NRAS methods provided by literature [12] in all experiments with default settings. BAGAN is developed upon its public source code[4].

For each imbalanced data set, we apply baselines and our model to generate synthetic minority data samples and then form different augmented data sets for training classifiers. Table 2 to Table 6 list the classification performance of three different classifiers on five test data sets. We conduct each experiment ten times and report average results. From these tables, we make the following observations: (i) Compared with the imbalanced set, the classification performance generally increases with the oversampling techniques, which shows the importance of oversampling. (ii) In most cases, with GL-GAN, the classification performance of classifiers outperforms with baselines, which implies the high quality of synthetic minority samples generated by GL-GAN. This is because local-based oversampling methods like SMOTE may produce some synthetic minority samples which are interleaved with existing majority samples or located in the null space of the given data set, while only global distribution information is explored in BAGAN and the generated synthetic samples may overlook the local structure of the given minority samples.

4.3 Components Analysis In order to investigate the impact of each module in our GL-GAN framework, we implement two models employing part of components contained in GL-GAN to generate synthetic minority samples and compare the quality of generated samples with GL-GAN. First, we combine the autoen-

[4] https://github.com/IBM/BAGAN

Table 4: Classification performance of classifiers on the Gas Sensor Array Drift data set.

Evaluation		Method							
Classifier	Metrics	Imbalanced	Random	SMOTE	MDO	NRAS	SWIM	BAGAN	GL-GAN
MLPClassifier	macro F1	0.3879	0.7880	0.8116	0.6540	0.7182	0.6974	**0.8891**	0.8881
	micro F1	0.5376	0.8003	0.8201	0.6833	0.7424	0.6985	**0.8908**	0.8884
	MCC	0.1077	0.6518	0.6832	0.4819	0.5592	0.3967	**0.7933**	0.7782
LinearSVC	macro F1	0.8580	0.9270	0.9296	0.6588	0.9216	0.3822	0.9290	**0.9694**
	micro F1	0.8619	0.9270	0.9296	0.6866	0.9216	0.5126	0.9296	**0.9695**
	MCC	0.7511	0.8560	0.8610	0.4871	0.8445	0.1606	0.8666	**0.9391**
AdaBoost	macro F1	0.3825	0.4620	0.4739	0.5299	0.5295	0.4795	0.4995	**0.5976**
	micro F1	0.5255	0.5418	0.5352	0.5963	0.6082	0.5445	0.5352	**0.6082**
	MCC	0.0689	0.0940	0.0690	0.3423	0.3174	0.0954	0.0653	**0.2181**

Table 5: Classification performance of classifiers on the Madelon data set.

Evaluation		Method							
classifier	Metrics	Imbalanced	Random	SMOTE	MDO	NRAS	SWIM	BAGAN	GL-GAN
MLPClassifier	macro F1	0.3333	0.3346	0.3364	0.4513	0.4440	0.4088	0.3376	**0.4821**
	micro F1	0.5000	0.5006	0.5008	0.4931	0.5213	0.5128	0.5019	**0.5260**
	MCC	0.0	0.0132	0.0120	-0.0165	0.0628	0.0465	0.0439	**0.0640**
LinearSVC	macro F1	0.3324	0.3333	0.3367	0.4643	0.4306	0.3894	0.3376	**0.4390**
	micro F1	0.4981	0.5000	0.5000	0.4942	0.5092	0.5058	0.5019	**0.5212**
	MCC	-0.0439	0.0	0.0	-0.0131	0.0276	0.0237	0.0439	**0.0657**
AdaBoost	macro F1	0.3354	0.3325	0.3400	0.3529	**0.4860**	0.4504	0.3325	0.3612
	micro F1	0.5000	0.4981	0.5000	0.4942	0.4885	0.4962	0.4981	**0.5019**
	MCC	0.0034	-0.0439	0.0	-0.0324	-0.0233	-0.0094	-0.0439	**0.0092**

coder component and the local data interpolator component together to obtain a new model called *Auto-only*. In Auto-only, the encoder E maps all given data samples into a latent space and the new synthetic minority samples are generated by the local data interpolator I. This procedure is the same with the first module in GL-GAN. However, instead of importing all latent representations into the generator G, Auto-only model employs the decoder Q to project all latent representations back to the raw feature space. Second, for studying the functionality of GAN-based module, we adopt conditional GAN [20] to generate synthetic minority samples. The conditional GAN takes random noise as input and produces synthetic samples with minority class label.

We conduct experiments on two real data sets and display the experimental results in Figure 3 and Figure 4, respectively. Here we can see, despite autoencoder (Auto-only) or conditional GAN (cGAN) can also produce synthetic minority samples for training a classifier, the quality of generated synthetic samples are not good enough, especially in the extremely imbalanced scenario ($r = 0.01$). However, since our GL-GAN could simultaneously explore the global and local information through combining the advantages of local-based oversampling techniques and generative adversarial learning together, the synthetic samples produced by GL-GAN could be more helpful for training a better classifier.

5 Case Studies

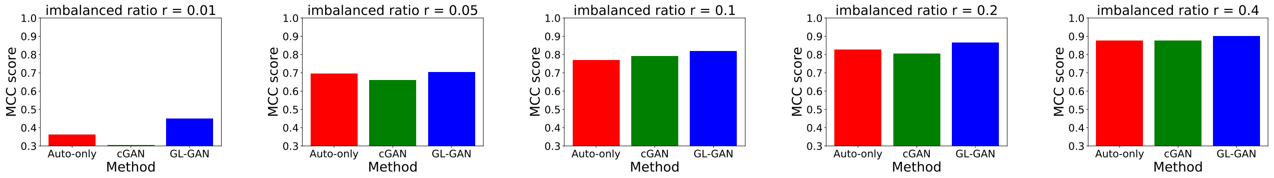
For verifying whether GL-GAN can produce more realistic synthetic minority samples, we visualize the synthetic samples generated on a handwritten digits data set MNIST[5]. Here we randomly choose images "4" as majority class and images "7" as minority class, and form the imbalanced data set as described in Sec 4.1.1.

5.1 Functionality of Autoencoder As we mentioned before, in order to avoid minority samples are generated in the null space of the given sample set or interleaved with majority samples, we require the encoder contained in our GL-GAN framework is able to map the given sample set $\mathcal{X}_{org}$ into two far-away clusters in the latent space, which can be achieved by Eq. (3.2). Here we utilize the MNIST data set to verify the usefulness of this design. In Figure 5, the right figure shows a snippet of images generated by the Auto-only method, in which the setting of the encoder is exactly same with the encoder E contained in our GL-GAN. As a comparison, we remove the loss function defined in Eq. (3.2) from the final loss function of the autoencoder, i.e., Eq. (3.3), and also apply SMOTE to generate synthetic minority samples in the latent space. In other words, the majority samples and minority samples are not required to be mapped far-away from each other in the latent space learned by the encoder, which is a common setting in the traditional autoencoder (AE). As the left figure shown, under this setting, the quality of generated synthetic minority samples "7" is worse than the Auto-only method generated ones. The reason is that, in the latent space, the synthetic minority samples generated by SMOTE may still have probability to interleave with majority samples if the majority cluster and minority cluster are

[5] http://yann.lecun.com/exdb/mnist/

Table 6: Classification performance of classifiers on the Gisette data set.

Evaluation		Method							
Classifier	Metrics	Imbalanced	Random	SMOTE	MDO	NRAS	SWIM	BAGAN	GL-GAN
MLPClassifier	macro F1	0.4618	0.6051	0.6161	0.6317	0.5888		0.4462	**0.8552**
	micro F1	0.5685	0.6528	0.6604	0.6710	0.6426	-	0.5558	**0.8568**
	MCC	0.2423	0.4246	0.4371	0.4486	0.4066		0.2418	**0.7277**
LinearSVC	macro F1	0.6053	0.6115	0.6115	0.8616	0.6718		0.3521	**0.8636**
	micro F1	0.6529	0.6571	0.6571	0.8636	0.7011	-	0.5086	**0.8650**
	MCC	0.4248	0.4318	0.4318	**0.7482**	0.5018		0.0930	0.7457
AdaBoost	macro F1	0.5226	0.5874	0.5669	0.3361	0.5718		0.5949	**0.6271**
	micro F1	0.5993	0.6400	0.6271	0.4850	0.6300	-	0.6199	**0.6679**
	MCC	0.3320	0.4001	0.3817	-0.0935	0.3848		0.2770	**0.4476**



(a) $r = 0.01$ (b) $r = 0.05$ (c) $r = 0.1$ (d) $r = 0.2$ (e) $r = 0.4$

Figure 3: MCC score of AdaBoost on the Gisette data set.

not far-away from each other. Hence, the generated synthetic images may not good enough. In short, these two figures demonstrate the loss function described by Eq. (3.2) is useful and indispensable.

5.2 Quality of Generated Image Data We also visualize the synthetic minority samples generated by SMOTE and our proposed GL-GAN framework on the MNIST data set. In the left figure of Figure 6, several synthetic samples produced by SMOTE looks like some intermediates between the majority class "4" and minority class "7". As we discussed before, due to only local neighbor relationships are utilized in SMOTE and the global information is totally ignored, SMOTE cannot avoid producing outliers or samples that interleaved with majority samples. On the contrary, our GL-GAN framework is able to generate more realistic synthetic minority samples. Since both global and local distributions are simultaneously explored in GL-GAN, the drawbacks of SMOTE can be overcame and high quality synthetic minority samples can be generated.

6 Conclusion and Future Work

In this paper, we propose a novel framework to solve the class imbalance problem through generating synthetic data samples for minority class. Different from local-based oversampling methods which only explore the local structure of minority samples and generative adversarial learning models which only utilize the global distribution information of all given samples, our GL-GAN framework considers both global and local information of the given data in the synthetic minority sample generation process. Extensive experimental results demonstrate that, comparing with existing baselines, our model can produce more realistic and discriminative synthetic minority samples, which are helpful for training better classifiers. In the future, we would extend our GL-GAN framework to the class imbalance problem of multi-class as well as some specific imbalanced application scenarios such as credit fraud detection.

Acknowledgements

Wentao Wang and Jiliang Tang are supported by the National Science Foundation (NSF) under grant numbers IIS1907704, IIS1714741, IIS1715940 and IIS1845081. Suhang Wang is supported by the National Science Foundation (NSF) under grant number IIS1909702.

References

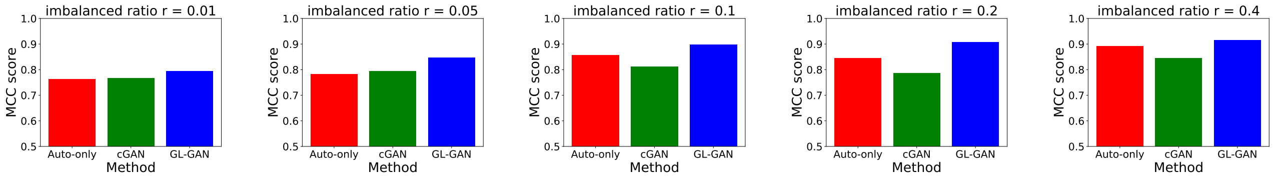
[1] L. Abdi and S. Hashemi, *To combat multi-class imbalanced problems by means of over-sampling techniques*, IEEE Transactions on Knowledge and Data Engineering, (2016), pp. 238–251.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *Smote: synthetic minority oversampling technique*, Journal of artificial intelligence research, 16 (2002), pp. 321–357.

[3] N. V. Chawla, N. Japkowicz, and A. Kotcz, *Special issue on learning from imbalanced data sets*, ACM Sigkdd Explorations Newsletter, 6 (2004), pp. 1–6.

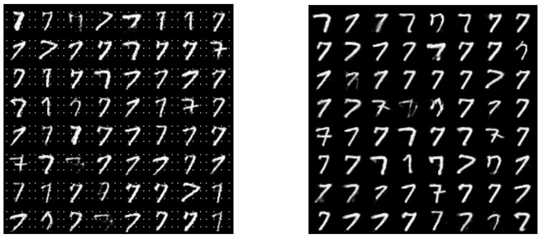
[4] G. Douzas and F. Bacao, *Effective data generation for imbalanced learning using conditional generative adversarial networks*, Expert Systems with applications, 91 (2018), pp. 464–471.

[5] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin, *Triangle generative ad-*



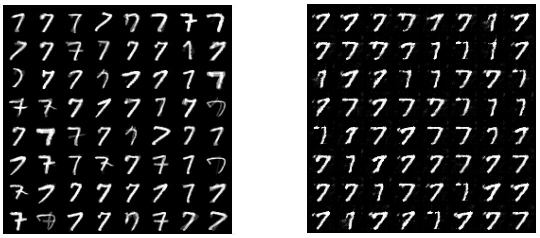
(a) $r = 0.01$ (b) $r = 0.05$ (c) $r = 0.1$ (d) $r = 0.2$ (e) $r = 0.4$

Figure 4: MCC score of AdaBoost on the USPS data set.



(a) Traditional AE (b) AE used in GL-GAN

Figure 5: Images generated by different autoencoders.



(a) SMOTE (b) GL-GAN

Figure 6: Images generated by SMOTE and GL-GAN.

versarial networks, in Advances in Neural Information Processing Systems, 2017, pp. 5247–5256.

[6] V. Ganganwar, *An overview of classification algorithms for imbalanced datasets*, International Journal of Emerging Technology and Advanced Engineering, 2 (2012), pp. 42–47.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in neural information processing systems, 2014, pp. 2672–2680.

[8] H. Han, W.-Y. Wang, and B.-H. Mao, *Borderline-smote: a new over-sampling method in imbalanced data sets learning*, in International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.

[9] H. He, Y. Bai, E. A. Garcia, and S. Li, *Adasyn: Adaptive synthetic sampling approach for imbalanced learning*, in Neural Networks. IJCNN. IEEE International Joint Conference on, IEEE, 2008, pp. 1322–1328.

[10] Y.-M. Huang, C.-M. Hung, and H. C. Jiau, *Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem*, Nonlinear Analysis: Real World Applications, 7 (2006), pp. 720–747.

[11] T. Kavzoglu, *Increasing the accuracy of neural network classification using refined training data*, Environmental Modelling & Software, 24 (2009), pp. 850–858.

[12] G. Kovács, *smote-variants: a python implementation of 85 minority oversampling techniques*, Neurocomputing, (2019).

[13] B. Krawczyk, *Learning from imbalanced data: open challenges and future directions*, Progress in Artificial Intelligence, 5 (2016), pp. 221–232.

[14] G. Lemaître, F. Nogueira, and C. K. Aridas, *Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning*, Journal of Machine Learning Research, 18 (2017), pp. 1–5.

[15] A. Liu, J. Ghosh, and C. E. Martin, *Generative oversampling for mining imbalanced datasets.*, in DMIN, 2007, pp. 66–72.

[16] X.-Y. Liu, J. Wu, and Z.-H. Zhou, *Exploratory undersampling for class-imbalance learning*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39 (2009), pp. 539–550.

[17] R. Longadge and S. Dongre, *Class imbalance problem in data mining review*, arXiv preprint arXiv:1305.1707, (2013).

[18] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, *Bagan: Data augmentation with balancing gan*, arXiv preprint arXiv:1803.09655, (2018).

[19] B. W. Matthews, *Comparison of the predicted and observed secondary structure of t4 phage lysozyme*, Biochimica et Biophysica Acta (BBA)-Protein Structure, 405 (1975), pp. 442–451.

[20] M. Mirza and S. Osindero, *Conditional generative adversarial nets*, arXiv preprint arXiv:1411.1784, (2014).

[21] W. Rivera, *Noise reduction a priori synthetic oversampling for class imbalanced data sets*, Information Sciences, 408 (2017), pp. 146–161.

[22] J. Schmidhuber, *Deep learning in neural networks: An overview*, Neural networks, 61 (2015), pp. 85–117.

[23] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, *Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance*, in 2018 IEEE International Conference on Data Mining, IEEE, 2018, pp. 447–456.

[24] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, *Fake news detection on social media: A data mining perspective*, ACM SIGKDD Explorations Newsletter, 19 (2017), pp. 22–36.

[25] S.-J. Yen and Y.-S. Lee, *Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset*, in Intelligent Control and Automation, Springer, 2006, pp. 731–740.