Deep Generative Model for Robust Imbalance Classification

Xinyue Wang, Yilin Lyu, Liping Jing
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
{xinyuewang, yilinlv, lpjing}@bjtu.edu.cn

Abstract

Discovering hidden pattern from imbalanced data is a critical issue in various real-world applications including computer vision. The existing classification methods usually suffer from the limitation of data especially the minority classes, and result in unstable prediction and low performance. In this paper, a deep generative classifier is proposed to mitigate this issue via both data perturbation and model perturbation. Specially, the proposed generative classifier is modeled by a deep latent variable model where the latent variable aims to capture the direct cause of target label. Meanwhile, the latent variable is represented by a probability distribution over possible values rather than a single fixed value, which is able to enforce uncertainty of model and lead to stable prediction. Furthermore, this latent variable, as a confounder, affects the process of data (feature/label) generation, so that we can arrive at well-justified sampling variability considerations in statistics, and implement data perturbation. Extensive experiments have been conducted on widely-used real imbalanced image datasets. By comparing with the state-of-the-art methods, experimental results demonstrate the superiority of our proposed model on imbalance classification task.

1. Introduction

The imbalanced data is inevitable in real-world applications especially in computer vision. For example, the background samples extremely outnumber the foreground samples in object detection [21]; the amount of healthy persons dominates that of the patients with lung cancer in medical image processing [13]. The size ratio between frequent event (majority classes) and rare event (minority classes) may be 10:1 or even 1000:1, which makes rare event detection much more difficult [9, 39]. However, misclassifying rare events can result in heavy costs, *e.g.*, in disease diagnosis, failing to identify a patient would cause the loss of life.

The scarce occurrences of rare events impair the detection task to imbalance classification problem. It aims to predict the unknown variables (*e.g.*, event) based on their observed features using a model estimated on a training dataset, which is a common statistical problem and has attracted much interest from various communities. Many methods have been proposed and perform successfully when training and testing data have similar joint distribution of features. Unfortunately, imbalanced data cannot always guarantee this because of two main reasons. Firstly, the observed features usually fall into two categories, one containing "direct causes" of the target variable, where the conditional distribution of the target given these features will not change when adding any other features, and the other having "noisy features" which do not affect the expected outcomes of the target [18]. Secondly, it is hard to directly determine the "direct causes" from insufficient training data especially for the minority classes. In this case, the learning model often obtains a good coverage of the majority samples whereas the minority samples are distorted.

From a probability theory perspective, it is difficult to build a stable model for invariant prediction from limited observed samples [42]. To date, many methods have been proposed to guarantee the stability as much as possible, which can be roughly divided into two categories: data perturbation and model perturbation. The former kind aims to characterize the underlying data distribution by approximating the data generation process. This mechanism, in imbalance classification, is adopted to augment the minority classes and then help the subsequent classifier determine the proper class boundaries [1, 25]. Even though they obtain promising performance, in such two-stage learning framework, data augmentation and classifier construction are implemented separately, which limits their applications. The later one tries to sufficiently assess the uncertainty of data by introducing uncertainty into learning model (*e.g.*, learning a probability distribution on the weights of neural network [4, 33] or constructing the loss function in an ambiguity set of the empirical distribution [31]). This strategy has ability to express uncertainty with few data and make reasonable predictions, however, it results in much more com-

plicated learning model and higher computational cost.

Thus, in this paper, we focus on robust imbalance classification to determine the essential factors of the target label so that the expected label value conditional on them is stable. Our idea is motivated by the literatures on causal inference and probabilistically generative model. To capture the cause-effect structure relationship, we propose a deep generative classifier (DGC) with the aid of deep latent variable model. It simultaneously learns the latent variable from the process of training data generation (including input features and labels) and approximates the Bayes' rule using importance sampling on the latent variable. Data generation process is formulated by minimizing the worst-case expectation of optimal transport cost between real and generative data distributions. DGC produces predictions by comparing the likelihood of labels on the learned latent variables for a given input. From this view, the proposed model can be taken as a joint generative model to approximate the joint distribution of input features, labels and latent variables, where the latent variables can characterize the essential structure hidden in the original data and can be used as the direct cause of labels. Therefore, it is expected to obtain stable prediction for new coming data. In summary, our contributions include:

- A deep generative classifier for imbalanced data is proposed by a deep latent variable model, where the latent variable is expressed via a probability distribution over possible values rather than a fixed value. Thus, it can be taken as a latent confounder affecting the whole learning process.

- The proposed method has ability to simultaneously implement data perturbation (via feature/label generation process) and model perturbation (via enforcing uncertainty on latent variables and minimizing the worst-case expectation), which can reduce the variability of label estimation and lead to stable prediction.

- We theoretically analyze the generalization error bounds of the proposed model, and efficiently optimize it by stochastic variational inference.

- A series of experiments are conducted to demonstrate its advantage over the state-of-the-art imbalance classification methods. Especially, it can produce good performance on the minority class, while maintaining a reasonable overall accuracy.

The remaining of this paper is organized as follows. The related work will be reviewed in Section 2. Then, the proposed method will be given in detail. In Section 4, we will describe and discuss the experiments. At last, conclusions and future work are briefly provided.

2. Related work

The class imbalance has been a frequent but challenging issue and attracts a significant amount of interest from the community of computer vision and machine learning. Most traditional classifiers have ability to optimize the overall prediction accuracy, however, they typically favor the majority classes and fail to classify the minority classes [15, 34]. Over the years, researchers have devised many methods for tackling class imbalance problem. Two basic strategies are resampling (oversampling or undersampling) and cost-sensitive learning [5, 13]. However, those methods are usually designed for low dimensional feature space and hard to cope with high dimensional data, like images, audio signals, etc. The emerging research surge of deep learning gave us the inspirations for an alternative strategy to deal with more complicated imbalanced data. Recent works mainly focus on two facets: data perturbation by extending resampling strategy to deep learning model and model perturbation by introducing uncertainty in model parameters or loss function, as shown in Figure 1.

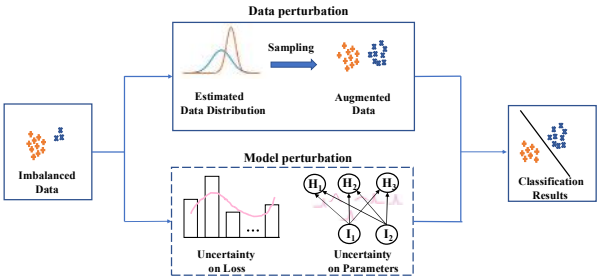


Figure 1. Two strategies for deep imbalanced learning.

To compensate the skewed distribution (caused by imbalanced data), deep generative models such as Variational autoencoder (VAE) [16] and Generative Adversarial Network (GAN) [11] are introduced to generate synthetic samples in the original feature space for imbalance classification. A simple way is to directly apply VAE on the given imbalanced data to capture the dimensional dependencies via latent variable, and then generate new samples from the learned latent variable [38, 43]. This strategy suffers from the drawback of VAE, which assumes that the data follow a single Gaussian distribution. When samples have mixture distributions, VAE cannot generate artificial data with sharp edges or fine details. To handle this case, Guo et al. [12] model latent representation via two Gaussian distributions with opposite means. Unfortunately, this idea is only useful for binary classification and cannot directly deal with multi-class imbalanced data.

An alternative generation strategy is GAN, which provides an effective way to learn mapping from the latent encoding space to original data space. To consider the difference between different classes, conditional GAN

(cGAN) [10] is introduced to generate class-specific minority samples [7]. GAN-based generation methods are usually fed with a random noise, which may result in a highly entangled process and disrupt the orientation-related features [6]. To solve this problem, researchers proposed BAGAN [25] by integrating AE and cGAN via a two-step framework. It learns the latent codes via AE and feeds them to cGAN instead of the random noise. However, attempting to oversample the minority class using GAN may lead to boundary distortion [29]. To make the latent features much more discriminative, a discriminative feature-based sampling (DFBS) method is proposed in [22]. DFBS adopts a supervised autoencoder with triplet loss to extract the latent features, then generates synthetic samples in the latent space by a random combination method. This generation strategy has the ability to bring samples within the same class closer together and push those from different classes further apart. The generated samples, in these methods, are likely close to the mode of the minority class, while new samples around the boundaries are required for reliable classifier [5, 35].

The previous methods can be roughly called as two-stage strategy, one for generating synthetic data, and the other for training classifier on the augmented data. Even though they sufficiently exploit data perturbation for imbalance classification performance, there may be a gap between data generation and classifier training. Recently, a generative adversarial minority oversampling (GAMO) method [27] is proposed to seamlessly integrate them by a three-player adversarial game between a convex generator, a multi-class classifier network, and a real/fake discriminator. GAMO generates new samples within the convex hull of the real minority-class samples. However, the convex hull of a minority class would be far from the true data distribution, which may generate less informative or even overlapped samples.

Besides data perturbation, model perturbation is also a good way to improve the stability of learning process, which is useful for limited data [42]. Among them, Bayesian network is widely used to offer uncertainty estimation via its parameters in form of probability distributions [4]. Recently, this strategy is extended to convolutional neural network [33]. By using a prior distribution to integrate out the parameters, they are estimated across many models during training, which has ability to prevent overfitting and obtain robust prediction. However, the learning process is time-consuming due to that much more parameters have to be estimated for characterizing the distributions of original network weights. To overcome the limitation of observed data, researchers proposed distributionally robust learning models with ambiguity set containing all (continuous or discrete) distributions that can be converted to the (discrete) empirical distribution at bounded cost [32]. Although this kind of model perturbation strategy prefers to robust prediction, it leads to more complicated model and higher computational complexity.

Motivated by the above problems, in this paper, we propose a new imbalance classification method via deep generative model to sufficiently exploit both data perturbation and model perturbation.

3. Proposed method

Inspired by the dual roles of generative model, summarizing past data (including prior knowledge) and generating synthetic observations, in this section, we design a deep generative classifier for imbalanced data.

3.1. Preliminaries and notations

Let calligraphic letter (*i.e.*, $\mathcal{X}$) indicate sets, capital letter (*i.e.*, X) for matrix, and lower case letter (*i.e.*, $\mathbf{x}$) for vector. Given a joint distribution over $\mathcal{X} \times \mathcal{Y}$, the training set is with N points from C classes, which are independent and identically distributed. The i-th training point $(\mathbf{x}_i, \mathbf{y}_i)$ contains feature information $\mathbf{x}_i \in \mathbb{R}^D$ and label information $\mathbf{y}_i \in \mathbb{R}^C$, where $\mathbf{y}_i$ is a C-dimensional one-hot vector. Denote $N = \sum_{c=1}^{C} n_c$, where n_c is the size of the c-th class. For an imbalance classification setting, there may be a big variance among $\{n_1, \ldots, n_c, \ldots, n_C\}$. Usually, the size of the largest class may be ten or even hundred times larger than that of the smallest class. In this paper, our goal is to learn a predictive model, which can make a robust prediction for new coming data point $\mathbf{x}^* \in \mathbb{R}^D$. In such an imbalanced situation, it is hard to guarantee the training data used to build the classification model follow the same distribution with the testing data, especially for minority classes [5].

3.2. Deep generative classifier (DGC)

To effectively handle imbalanced data, we design a deep generative model to simultaneously mine the direct cause of target label and build a stable generative imbalanced classifier.

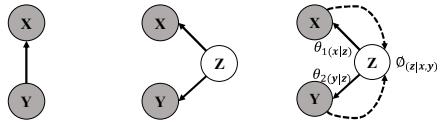


Figure 2. The architecture of graphical models for a) the traditional generative classifier and b) the proposed DGC. c) is the parameterized version of b). The nodes denote observed or latent variables. The solid lines and dash lines represent generative and recognition model respectively.

Giving the training dataset, generative classifier aims to build a model of "how data for a class looks like". Mathematically, it will learn the joint probability distribution

$p(\mathbf{x}, \mathbf{y})$ by estimating the parameters of $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y})$ (as shown in Figure 2(a)). In prediction phase, the label y^* of a new coming sample x^* can be assigned via the Bayes rule. The well-known generative classifier, Naive Bayes classifier, has obtained promising performance if the features are independent to each other, *i.e.*, the conditional distribution can be factorized along all features via $p(\mathbf{x}|\mathbf{y}) = \prod_{d=1}^{D} p(x_d|\mathbf{y})$. Unfortunately, this assumption is not appropriate for the complicated case, *e.g.*, image data, since all "natural" images always show a lot of spatial regularity, but this kind of factorized distribution cannot present discontinuities across hypersurfaces.

Inspired by deep learning, we propose a deep generative classifier with the aid of deep latent variable model. Our goal is to mine the direct cause and implement stable prediction on the target label, even for imbalanced data. Let $\mathbf{z}$ denote the direct cause of target label, in this case, any noisy feature $\mathbf{v}$ will not affect the prediction result, *i.e.*, $p(\mathbf{y}|\mathbf{z}, \mathbf{v}) = p(\mathbf{y}|\mathbf{z})$ always holds. Meanwhile, $\mathbf{z}$ is expected to capture the essential structure of the original data, *i.e.*, both $\mathbf{x}$ and $\mathbf{y}$ can be most likely generated from $\mathbf{z}$. To achieve these two goals, our generative model is designed as shown in Figure 2(b). In this model, the joint probability distribution $\mathrm{p}(\mathbf{x}, \mathbf{z}, \mathbf{y})$ can be factorized as $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$. This model has a good by-product that the conditional distribution $p(\mathbf{x}|\mathbf{y})$ can be written by

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} = \frac{\int p(\mathbf{x}, \mathbf{z}, \mathbf{y}) d\mathbf{z}}{\int p(\mathbf{x}, \mathbf{z}, \mathbf{y}) d\mathbf{z} d\mathbf{x}}. \tag{1}$$

which is not factorized along the original features, thus, it is expected to be more powerful on real-world complicated applications.

To build the graphical model from training data, we introduce its parameterized version in Figure 2(c). Mathematically, the input feature $\mathbf{x} \in \mathbb{R}^D$ and corresponding label $\mathbf{y} \in \mathbb{R}^C$ are generated by the latent vector $\mathbf{z} \in \mathbb{R}^k$ and parameter $\theta = \{\theta_1, \theta_2\}$ as follows.

$$p(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p(\mathbf{z}) p_{\theta_1}(\mathbf{x}|\mathbf{z}) p_{\theta_2}(\mathbf{y}|\mathbf{z}). \tag{2}$$

For convenient computing, $\mathbf{z}$ is assumed to be sampled from a standard Gaussian prior, *i.e.*, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I^k})$. During training process, the proper $\mathbf{z}$ will be recognized with the aid of parameter ϕ, *i.e.*, $\mathbf{z} = \phi(\mathbf{x}, \mathbf{y})$.

3.3. Inference learning for DGC model

One way to infer deep generative model is to postulate that they are trying to minimize certain divergence between the true data distribution P of $\mathcal{D} = (X, Y)$, and the generative distribution P_G of the generated data $\hat{\mathcal{D}} = (\hat{X}, \hat{Y})$ from the observed training data.

As mentioned above, imbalanced data cannot provide sufficient information for training classifier, which will be easy to result in overfitting problem and might further constitute poor predictors beyond the training dataset. This is the main reason that the good performance cannot be obtained on minority classes. Thus, we infer the proposed deep generative model by utilizing Wasserstein distance [2, 36]. Comparing with KL-divergence or JS-divergence, Wasserstein distance has the ability to measure the similarity between the distributions well and preserve the transitivity in latent space due to the much weaker topology.

From the optimal transport (OT) point of view, we aim at minimizing OT cost between real data distribution P and generative distribution P_G,

$$W_c(P, P_G) = \inf_{\Gamma \in \mathcal{P}_\Gamma} \mathbb{E}_{\left((X,Y),(\hat{X},\hat{Y})\right) \sim \Gamma} [c((X, Y), (\hat{X}, \hat{Y}))] \tag{3}$$

here $\mathcal{P}_\Gamma = \left((X, Y) \sim P, (\hat{X}, \hat{Y}) \sim P_G\right)$. Minimizing (3) is equal to precisely minimize the worst-case expectation, which has ability to provide an upper confidence bound on the out-of-sample error. Thus, it is expected to produce good solution on limited data.

For our case, given the latent variable Z, X and Y are independent to each other as shown in Figure 2(c), *i.e.*, $X \perp Y|Z$, thus we have

$$P_G(\mathbf{x}, \mathbf{y}) = \int_{\mathcal{Z}} P_G(\mathbf{x}|\mathbf{z}) P_G(\mathbf{y}|\mathbf{z}) P_z(\mathbf{z}) d\mathbf{z} \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{D}. \tag{4}$$

Among it, $P_G(\mathbf{x}|\mathbf{z})$ deterministically maps $\mathbf{z}$ to $\mathbf{x}$, similarly, $P_G(Y|Z)$ for mapping $\mathbf{z}$ to $\mathbf{y}$ with two functions $G_X : \mathcal{Z} \to \mathcal{X}$ and $G_Y : \mathcal{Z} \to \mathcal{Y}$, respectively.

According to [36], instead of finding a coupling Γ between $(X, \hat{X})$ and $(Y, \hat{Y})$, it is sufficient to find a conditional distribution $Q(Z|X, Y)$ such that its Z marginal distribution (Q_Z) is identical to the prior distribution P_Z, here Q_Z can be computed by

$$Q_Z := \mathbb{E}_{(X,Y)\sim P}[Q(Z|X, Y)] = \mathbb{E}_{X\sim P_X, Y\sim P_Y}[Q(Z|X, Y)] \tag{5}$$

In this situation, the OT cost can be re-written as

$$\inf_{\mathcal{Q}: \mathcal{Q}_Z = P_Z} \mathbb{E}_{P_X} \mathbb{E}_{P_Y} \mathbb{E}_{\mathcal{Q}(Z|X,Y)} [c_1(X, G_X(Z)) + c_2(Y, G_Y(Z))] \tag{6}$$

where $c_1(\mathbf{x}, \hat{\mathbf{x}})$ and $c_2(\mathbf{y}, \hat{\mathbf{y}})$ are measurable cost functions with non-negative value.

In order to implement a numerical solution, the above constrained problem can be relaxed by adding a penalty, *i.e.*,

$$\begin{aligned} W_c(P, P_G) = \inf_{Q(Z|X,Y) \in \mathcal{Q}} & \mathbb{E}_{P_X} \mathbb{E}_{P_Y} \mathbb{E}_{Q(Z|X,Y)} \big[c_1\big(X, G_X(Z)\big) \\ & + c_2\big(Y, G_Y(Z)\big)\big] + \lambda \cdot D_Z(Q_Z, P_Z). \end{aligned} \tag{7}$$

where $\mathcal{Q}$ is any nonparametric set of recognition model. $D_{\mathbf{z}}$ is an arbitrary divergence between Q_Z and P_Z. In this work, the maximum mean discrepancy (MMD) is adopted since it shares the properties of divergence functions and has the

ability to form an unbiased U-estimator [8]. The other reason is that MMD is conjuncted with the subsequent stochastic gradient descent (SGD) methods. $\lambda > 0$ is the hyperparameter to trade-off the corresponding terms.

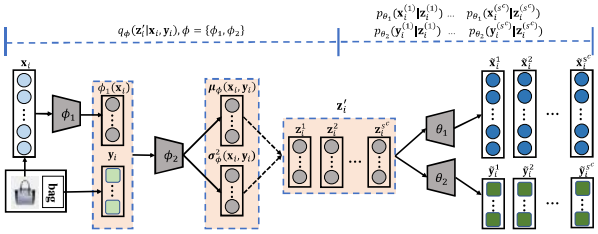


Figure 3. Neural network version of the proposed DGC.

To efficiently handle the objective function, we adopt stochastic optimization technique. Then, the objective function can be calculated along each training sample as follows.

$$\begin{aligned}&\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i; \theta, \phi)\\ &= \inf_{q_\phi(\mathbf{z_i}|\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{Q}} \mathbb{E}_{p_{\mathbf{x}_i}} \mathbb{E}_{p_{\mathbf{y}_i}} \mathbb{E}_{q_\phi(\mathbf{z_i}|\mathbf{x}_i,\mathbf{y}_i)}[c_1(\mathbf{x}_i, g_{\theta_1}(\mathbf{x}_i|\mathbf{z}_i)) \\ &\quad + c_2(\mathbf{y}_i, g_{\theta_2}(\mathbf{y}_i|\mathbf{z}_i))] + \lambda \cdot D_{\mathbf{z}}(q_\phi(\mathbf{z}_i|\mathbf{x}_i, \mathbf{y}_i), p(\mathbf{z}_i)).\end{aligned} \tag{8}$$

Among it, c_1 measures the distance between the given data $\mathbf{x}_i$ and the generated data $\hat{\mathbf{x}}_i = g_{\theta_1}(\mathbf{x}_i|\mathbf{z}_i)$ on feature space via $c_1(\mathbf{x}_i, \hat{\mathbf{x}}_i) = ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2^2$. c_2 evaluates the difference between the ground truth label information $\mathbf{y}_i$ and the generated label information $\hat{\mathbf{y}}_i = g_{\theta_2}(\mathbf{y}_i|\mathbf{z}_i)$. Specially, we adopt the cross-entropy loss function, *i.e.*, $c_2(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \mathbf{y}_i \log \hat{\mathbf{y}}_i + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i)$ to measure the difference on label space. Note the parameters need to be optimized include $\theta = \{\theta_1, \theta_2\}$ and $\phi = \{\phi_1, \phi_2\}$, as shown in Figure 3.

To obtain an unbiased estimate of $\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i; \theta, \phi)$, we can sample $\mathbf{z}_i$ via $\mathbf{z}_i \sim q_\phi(\mathbf{z}_i|\mathbf{x}_i, \mathbf{y}_i)$. More specifically, by taking advantage of *reparameterization trick* for *Gaussian* distribution, we sample $\epsilon \sim \mathcal{N}(0, \mathbf{I^k})$ and reparametrize $\mathbf{z}_i = \mu_\phi(\mathbf{x}_i, \mathbf{y}_i) + \epsilon \odot \sigma_\phi(\mathbf{x}_i, \mathbf{y}_i)$. For imbalanced data, there are only few samples in minority classes. To make up for the limitation of input data, a set of latent codes $\{\mathbf{z}_i^{(j)}\}_{j=1}^{s^c}$ is sampled if the corresponding input point $\mathbf{x}_i$ belongs to the minority classes and $\mathbf{y}_i = c$, where s^c is the number of oversampling latent codes for each instance of the c-th class. The oversampling codes will be used to generate synthetic data in the original feature space via decoder ($\mathbf{z}_i^{(j)}$ for $\mathbf{x}_i^{(j)}$) and infer more robust classifier $p(\mathbf{y}|\mathbf{z})$.

The model parameters $\{\theta, \phi\}$ can be iteratively updated by minimizing (8) with the aid of stochastic variational inference [17]. Once having the optimal model, a generative classifier will be built, where the conditional distribution $p(\mathbf{y_c}|\mathbf{x})$ for the c-th class has its support as a low-dimensional manifold. Then, the proper label vector $y^* \in \mathbb{R}^C$ of any new coming point $x^* \in \mathbb{R}^D$ can be obtained by an approximation to Bayes' rule with importance sampling $\mathbf{z}_c^s \sim q(\mathbf{z}|\mathbf{x}^*, \mathbf{y}_c)$:

$$p(y^*|x^*) \approx softmax_{c=1}^{C}\left\{ \log \frac{1}{S} \sum_{s=1}^{S} \mathcal{L}(\mathbf{x}^*, \mathbf{y}_c, \mathbf{z}_c^s; \theta, \phi) \right\} \tag{9}$$

S is the sampling size. In experiment, it is set to be 10 for prediction.

3.4. Error bound analysis

In this subsection, we will theoretically analyze the generalization error bounds of the proposed model. Motivated by [3], we can define the generalization of input feature and label generation process by measuring the difference between the population real data distribution ($\mathcal{P}_{real}$) and generated data distribution ($\mathcal{P}_G$). The generalization error will be acceptable if the population distance between $\mathcal{P}_{real}$ and $\mathcal{P}_G$ is close to the empirical distance between the observed real data distribution ($\tilde{\mathcal{P}}_{real}$) and its generated distribution ($\tilde{\mathcal{P}}_G$). In the proposed deep latent variable model, given latent code Z, the input feature information X is independent to input label Y, *i.e.*, $(X \perp Y|Z)$. In this case, the data distribution $(X, Y) \sim \mathcal{P}_{real}$ can be factorized into two parts $X \sim \mathcal{X}_{real}$ and $Y \sim \mathcal{Y}_{real}$, similar to $\mathcal{P}_G$, $\tilde{\mathcal{P}}_{real}$ and $\tilde{\mathcal{P}}_G$.

Definition 1 *For the empirical real distribution ($\tilde{\mathcal{X}}_{real}$, $\tilde{\mathcal{Y}}_{real}$) with N training examples, a generated distribution ($\tilde{\mathcal{X}}_G$, $\tilde{\mathcal{Y}}_G$) generalizes under the distribution distance $d(\cdot,\cdot)$ with generalization error $\delta_1, \delta_2 > 0$ if the following holds with high probability,*

$$|E(X) - E(\tilde{X})| \leq \delta_1 \tag{10}$$

$$|E(Y) - E(\tilde{Y})| \leq \delta_2 \tag{11}$$

where $E(X)$ and $E(Y)$ indicate the population distance between the real and generated distributions on feature and label information respectively. $E(\tilde{X})$ and $E(\tilde{Y})$ are the corresponding empirical distances.

(10) and (11) can be proved by the following two theorems and the detailed proof is given in Supplementary.

Theorem 1 *For any $X \in \mathbb{R}^{D\times N}$ $(N, D > 0)$,*

$$E(X) \leq E(\tilde{X}) + \sqrt{\frac{\log \delta_1}{-2N}(\max_i d_i)^2}$$

holds with probability at least $1 - \delta_1 (\delta_1 > 0)$ over uniformly choosing an empirical version ($\tilde{X}$) of X. Here, $d_i = d(\tilde{\mathbf{x}}_i^{(real)}, \tilde{\mathbf{x}}_i^{(G)}) = ||\tilde{\mathbf{x}}_i^{(real)} - \tilde{\mathbf{x}}_i^{(G)}||_2^2$.

Theorem 2 *Given prior label probabilities $\{p_1, \ldots, p_c, \ldots, p_C\}$ (where $p_c = P(y = c)$ and $\sum_{c=1}^{C} p_c = 1$) and the conditional latent variable densities $\{f_1, f_2, \ldots, f_C\}$ (where $f_c = f(\mathbf{z}|y = c)$) , following [30], the error rate of generative classifier can be formulated and bounded:*

$$|E(Y) - E(\tilde{Y})| = 1 - \int \max\{p_1 f_1(\mathbf{z}), \ldots, p_C f_C(\mathbf{z})\} d\mathbf{z} \leq \delta_2.$$

Table 1. Summarization of the experimental datasets.

Dataset	Shape	Classes	IR	Training Set	Testing Set
MNIST	$28 \times 28 \times 1$	10	100	4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40	980, 1135, 1032, 1010, 982, 892, 985, 1028, 974, 1009
Fashion-MNIST	$28 \times 28 \times 1$	10	100	4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40	1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000
SVHN	$32\times 32 \times 3$	10	56.25	4500, 2000, 1000, 800, 600, 500, 400, 250, 150, 80	1744, 5099, 4149, 2882, 2523, 2384, 1977, 2019, 1660, 1595
CelebA	$64\times 64 \times 3$	5	100	15000, 1500, 750, 300, 150	2660, 5422, 423, 3587, 636

DGC aims to build a robust imbalance classifier by modeling generation processes of input feature and label. The theoretical analysis is to derive generalization error bounds of these two generation processes. Theorem 1 shows that the bound on feature generation will be tight when generated instances are realistic (small $max_i d_i$), which is empirically proven in Fig.6. Theorem 2 indicates the bound on label generation will be tight when the ground-truth label has a higher probability while other labels have much lower probabilities, which is empirically proven by the classification accuracy.

4. Experiments

In this section, a series of experiments are conducted to evaluate the proposed method. The code is available at `https://github.com/lvyilin/DGC`.

4.1. Datasets

The experimental data includes two single-channel image sets (MNIST [19] and Fashion-MNIST [41]) and two three-channel image sets (SVHN [28] and CelebA [23]). These datasets are not significantly imbalanced in nature. Following GAMO [27], we created imbalanced datasets by randomly selecting instances with different sizes from different classes in order of their indices. For CelebA, only five non-overlapping classes (*blonde, black, bald, brown*, and *gray*) are kept. Details of these datasets are shown in Table 1. Obviously, there is a large imbalance ratio (IR: ratio of the number of instances from the largest class to that of the smallest class) from 56.25 to 100. Note that datasets with imbalance ratio over 10 can be regarded as highly imbalanced [9, 39]. Larger ratio will make the learning problem more difficult. For each dataset, the classification performance is evaluated on the public testing set. All the methods run on five imbalanced subsets with same size (for each dataset) to mitigate any bias generated due to randomization. The average results and corresponding statistics are recorded.

4.2. Methodology

In order to validate the imbalance classification performance, five well-known and widely used metrics [14, 26, 37] are adopted: recall of minority class (R_{MI}), precision of majority class (P_{MA}), average class specific accuracy (ACSA), macro-averaged geometric mean (G_{macro}) and macro-averaged F-measure (F_{macro}). Larger value indicates better performance.

We choose two kinds of methods as baselines, one for robust learning with data perturbation and the other with model perturbation. The first kind contains BAGAN [25], DFBS [22], and GAMO [27]. Among them, BAGAN and DFBS firstly augment imbalanced data and then adopt the existing methods to train classifier. GAMO is one-stage and outperforms the recently proposed deep oversampling method [1]. The second type includes mmDGMs [20] and BayesCNN [33]. mmDGMs adopts a latent variable to affect the generation process on feature information, and the discriminative model for label prediction. BayesCNN implements robust prediction by introducing uncertainty on the parameters of convolutional neural which is a good architecture for images.

The parameters of all algorithms we compared with are adopted from their original papers or determined by experiments. For our **DGC** model, the size of latent space (k) is set to 64 for the first three datasets and 128 for the fourth dataset to capture more information. $\lambda \in \{1, 10\}$ is tuned by 5-fold cross-validation technique.

4.3. Results and discussion

The experimental results are listed and analyzed from two facets: comparing the proposed DGC with baselines and investigating its stability.

- **Comparing classification performance**

To make a fair comparison, the trained models are evaluated by the same public testing set for each dataset. The overall classification performance (average value and the standard variance in terms of ACSA, G_{macro} and F_{macro}) on four benchmark datasets are listed in Table 2. The best

Table 2. Comparing overall classification performance on experimental datasets.

	MNIST			Fashion-MNIST		
Method	ACSA	F_{macro}	G_{macro}	ACSA	F_{macro}	G_{macro}
BAGAN	0.8848 ± 0.02	0.8785 ± 0.02	0.9295 ± 0.01	0.7814 ± 0.01	0.7610 ± 0.01	0.8546 ± 0.01
DFBS	0.7812 ± 0.04	0.7838 ± 0.04	0.8683 ± 0.02	0.5135 ± 0.17	0.4620 ± 0.27	0.6382 ± 0.21
GAMO	0.8826 ± 0.01	0.8794 ± 0.01	0.9308 ± 0.00	0.7929 ± 0.01	0.7880 ± 0.01	0.8740 ± 0.00
BayesCNN	0.9158 ± 0.01	0.9141 ± 0.01	0.9512 ± 0.01	0.7934 ± 0.01	0.7835 ± 0.01	0.8701 ± 0.01
mmDGMs	0.9066 ± 0.02	0.9039 ± 0.02	0.9449 ± 0.01	0.8091 ± 0.00	0.7984 ± 0.01	0.8796 ± 0.00
DGC	**0.9480 ± 0.00**	**0.9474 ± 0.00**	**0.9704 ± 0.00**	**0.8364 ±0.00**	**0.8314 ±0.00**	**0.9010 ±0.00**
	SVHN			CelebA		
Method	ACSA	F_{macro}	G_{macro}	ACSA	F_{macro}	G_{macro}
BAGAN	0.6785 ± 0.01	0.6719 ± 0.01	0.7876 ± 0.01	0.5972 ± 0.00	0.5152 ± 0.02	0.6554 ± 0.01
DFBS	0.4788 ± 0.03	0.4745 ± 0.03	0.6539 ± 0.02	0.2109 ± 0.00	0.1335 ± 0.01	0.2664 ± 0.01
GAMO	0.6474 ± 0.01	0.6457 ± 0.01	0.7784 ± 0.01	0.6409 ± 0.02	0.5903 ± 0.03	0.7472 ± 0.02
BayesCNN	0.5511 ± 0.03	0.5392 ± 0.03	0.6998 ± 0.02	0.5517 ± 0.04	0.4936 ± 0.04	0.6534 ± 0.04
mmDGMs	0.7220 ± 0.01	0.7291 ± 0.01	0.8291 ± 0.01	0.3760 ± 0.03	0.0618 ± 0.14	0.3754 ± 0.04
DGC	**0.7493 ±0.01**	**0.7535 ± 0.01**	**0.8501 ± 0.00**	**0.6755 ± 0.03**	**0.6454 ± 0.02**	**0.7779 ± 0.03**

Table 3. Comparing prediction performance on the smallest class (R_{MI}) and largest class (P_{MA}).

	MNIST		Fashion-MNIST		SVHN		CelebA	
Method	R_{MI}	P_{MA}	R_{MI}	P_{MA}	R_{MI}	P_{MA}	R_{MI}	P_{MA}
BAGAN	0.5354	0.8541	0.7306	0.5709	0.1630	0.4222	0.0192	0.5064
DFBS	0.5946	0.5118	0.4412	0.3395	0.1946	0.2173	0.0522	0.2174
GAMO	0.6394	0.8812	0.7928	0.6165	0.2813	0.4047	0.2302	0.6687
BayesCNN	0.7578	0.8896	0.8474	0.6022	0.1263	0.3276	0.1063	0.5225
mmDGMs	0.6525	0.8459	0.8160	0.5942	0.3740	0.4301	0.0006	0.4110
DGC	**0.8276**	**0.9270**	**0.8864**	**0.6900**	**0.4882**	**0.5348**	**0.2987**	**0.7603**

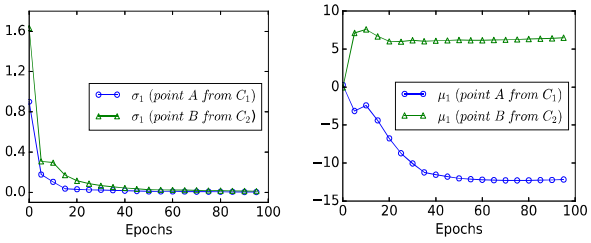
results are highlighted in bold. As expected, the proposed DGC obtained the best performance in all cases.

From this result, we can get following observations. DFBS, as a two-stage method, performs worse than other methods, because it cannot guarantee generated instances are useful to create margins among classes. GAMO, as a one-stage method, performs better than DFBS, however it is hard to trade off the discriminator and classifier, so that the generated images look real but cannot demonstrate the intrinsic structure of classes. BAGAN is superior to DFBS because it adopts AE to determine class distribution on latent space. Unfortunately, the initialization strategy of subsequent GAN may push BAGAN to fall into the mode collapse problem. BayesCNN cannot obtain satisfying results even though it adopts model perturbation strategy, because imbalanced data contains too few instances to sufficiently train the complicated model. mmDGMs takes advantage of generative model and perturbation on latent code. However, it adopts the discriminative classifier to determine class boundaries, which limits its performance on highly imbalanced data. This result confirms that DGC can construct effective imbalanced classifier by integrating data and model perturbation in a unified deep generative model.

In real-world applications, the minority class is of more interest. Thus, it is expected that the classifier can return a higher recall on minority class and maintain higher precision on majority class. Table 3 lists the recall of the smallest class (*i.e.*, the percentage of minority instances correctly predicted) and the precision of the largest class (*i.e.*, the percentage of correct predictions in majority class). It can be seen that, for Fashion-MNIST, five baselines increase the performance on minority class but their performances on majority class are not significantly improved. For CelebA, baselines definitely output worse results on minority class. This result indicates that existing methods cannot determine the clear boundaries among classes. Fortunately, for both the minority class and majority class, DGC consistently obtains the best and high-quality classification results. Pairwise t-test is conducted along each evaluation metric at 95% confidence level. Fortunately, the p-value between DGC and any baseline is below 0.01, which demonstrates DGC has ability to significantly improve classification performance.

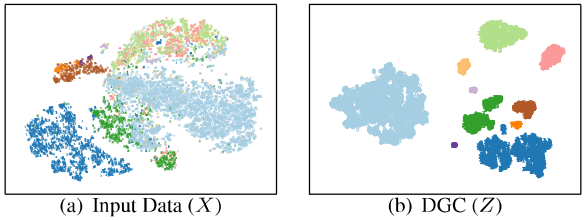
- **Investigating stability of DGC**

To investigate the learning process of DGC, we display the convergence of latent code on its mean (μ) and standard deviation (σ) over epochs. It is hard to show all latent codes since there are k latent features and each input data has a latent code. Therefore, we randomly select two



(a) (b)

Figure 4. Convergence of (a) σ values and (b) μ values (only the first latent feature is demonstrated for two instances A and B which are randomly selected from two classes.



(a) Input Data (X) (b) DGC (Z)

Figure 5. tSNE analysis of (a) input data (X) and (b) latent codes (Z) obtained by DGC on Fashion-MNIST dataset.

(a) The largest class (b) The smallest class

Figure 6. Generated Fashion-MNIST images by DGC for (a) the largest class and (b) the smallest class.

training instances (A and B belonging to class C_1 and C_2) from Fashion-MNIST, and demonstrate the convergence of the first latent feature (μ_1, σ_1). As aforementioned, the variational posterior distribution $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$are approximated as Gaussian distributions which become more confident over epochs, decreasing standard deviation as shown in Figure 4(a). An interesting thing is that two means (μ for A and B) are separated and converge over epochs in Figure 4(b).

Meanwhile, we adopt tSNE analysis [24] to visually present the discriminative ability of DGC. Taking Fashion-MNIST dataset as an example, tSNE is firstly applied on the training data, *i.e.*, projecting the original feature space to a 2-dimension space. When visualizing the instances, different classes are marked in different colors. The results on the original input feature space and the learned latent feature space are demonstrated in Figure 5(a) and 5(b). Obviously, it is not easy to directly separate classes in the original feature space. To be exciting, the class boundaries are clear in the latent space, which further confirms that the latent code $\mathbf{z}$ has the discriminative ability. This result further confirms that DGC is good at identifying the direct cause of target

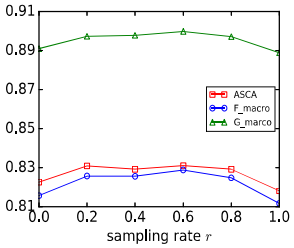


Figure 7. Effect of sampling rate (r) on DGC.

label. Figure 6 gives the generated images with $\mathbf{z}$. As expected, DGC indeed generates realistic and diverse images.

In experiments, the latent code is sampled from the learned Gaussian distribution. The number of latent codes (s^c) for each instance in the c-th class is set to be $\max\{1, r \times \frac{n_{max}}{n_c}\}$, where n_c is the size of c-th class, $n_{max} = \max\{n_c|_{c=1}^{C}\}$ is the size of the largest class, and r is the sampling rate. We test DGC under varying sampling rate r, as shown in Figure 7. DGC slightly benefits from large rate r, while its performance decreases when r is too large. It is reasonable because more oversampled similar codes may result in overfitting. This result demonstrates that sampling few latent codes is good enough to construct imbalanced classifier, which makes the training process more efficient.

5. Conclusions and future work

In this paper, we proposed a deep generative model for imbalance classification. It takes advantage of data perturbation and model perturbation to improve the prediction accuracy and learning stability on imbalanced data. DGC is inferred by utilizing Wasserstein distance. As a generative model, it can be improved by adopting improved architectures for future investigation, such as Self-Attention [44] or sliced-Wasserstein generative model [40], which can achieve good performance by considering more detailed features or efficient projection along random directions.

6. Acknowledgment

Liping Jing is the corresponding author. This work was supported in part by the National Natural Science Foundation of China under Grant 61822601, 61773050, and 61632004; the Beijing Natural Science Foundation under Grant Z180006; National Key Research and Development Program (2017YFC1703506); The Fundamental Research Funds for the Central Universities (2019JBZ110); Science and technology innovation planning foundation of colleges and Universities under the guidance of the Ministry of Education.

References

[1] Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[3] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning*, pages 224–232. JMLR. org, 2017.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *In Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[5] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49:1–56, 2016.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[7] Georgios Douzas and Fernando Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications*, 91:464–471, 2018.

[8] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Uncertainty in Artificial Intelligence*, pages 258–267. AUAI, 2015.

[9] Alberto Fernández, Salvador García, María José del Jesus, and Francisco Herrera. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data sets. *Fuzzy Sets and Systems*, 159(18):2378–2398, 2008.

[10] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Technical Report*, 2015.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[12] Ting Guo, Xingquan Zhu, Yang Wang, and Fang Chen. Discriminative sample generation for deep imbalanced learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2406–2412. AAAI Press, 2019.

[13] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

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

[15] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: a systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Rahul G Krishnan, Dawen Liang, and Matthew Hoffman. On the challenges of learning with inference networks on sparse, high-dimensional data. *arXiv preprint arXiv:1710.06085*, 2017.

[18] Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1617–1626. ACM, 2018.

[19] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998. *URL http://yann. lecun. com/exdb/mnist*, 10:34, 1998.

[20] Chongxuan Li, Jun Zhu, and Bo Zhang. Max-margin deep generative models for (semi-) supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2762–2775, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[22] Yi-Hsun Liu, Chien-Liang Liu, and Shin-Mu Tseng. Deep discriminative features learning and sampling for imbalanced data problem. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1146–1151. IEEE, 2018.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[25] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: data augmentation with balancing gan. *Computer Vision and Pattern Recognition*, 2018.

[26] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

[27] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1695–1704, 2019.

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[29] Shibani Santurkar, Ludwig Schmidt, and Aleksander Mkadry. A classification-based study of covariate shift in gan distributions. In *International Conference on Machine Learning*, pages 4487–4496, 2018.

[30] Salimeh Yasaei Sekeh, Brandon Oselio, and Alfred O Hero. Learning to bound the multi-class bayes error. *arXiv preprint arXiv:1811.06419*, 2018.

[31] Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

[32] Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

[33] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. A comprehensive guide to bayesian convolutional neural network with variational inference. *https://arxiv.org/pdf/1901.02731.pdf*, 2019.

[34] Paolo Soda. A multi-objective optimisation approach for class imbalance learning. *Pattern Recognition*, 44(8):1801–1810, 2011.

[35] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: reducing mode collapse in gans using implicit variational learning. In *In Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[36] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *Proceedings of International Conference on Learning Representations*, 2018.

[37] Vincent Van Asch. Macro- and micro-averaged evaluation measures [[basic draft]]. 2013.

[38] Zhiqiang Wan, Yazhou Zhang, and Haibo He. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[39] Fei Wu, Xiao-Yuan Jing, Shiguang Shan, Wangmeng Zuo, and Jing-Yu Yang. Multiset feature learning for highly imbalanced data classification. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[40] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3713–3722, 2019.

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[42] Bin Yu and Karl Kumbier. Three principles of data science: predictability, computability, and stability (pcs). *arXiv preprint arXiv:1901.08152*, 2019.

[43] Chunkai Zhang, Ying Zhou, Yingyang Chen, Yepeng Deng, Xuan Wang, Lifeng Dong, and Haoyu Wei. Over-sampling algorithm based on vae in imbalanced classification. In *International Conference on Cloud Computing*, pages 334–344. Springer, 2018.

[44] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.