Wei *et al. J Wireless Com Network* (2022) 2022:22
https://doi.org/10.1186/s13638-022-02105-7

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Mixed-type data generation method based on generative adversarial networks

Ning Wei[1], Longzhi Wang[1], Guanhua Chen[1], Yirong Wu[1,2], Shunfa Sun[1] and Peng Chen[1*]

*Correspondence:
chenpeng@ctgu.edu.cn
[1] The College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China
Full list of author information is available at the end of the article

Abstract

Data-driven based deep learing has become a key research direction in the field of artificial intelligence. Abundant training data is a guarantee for building efficient and accurate models. However, due to the privacy protection policy, research institutions are often limited to obtain a large number of training data, which would lead to a lack of training sets circumstance. In this paper, a mixed-type data generation model based on generative adversarial networks is proposed to synthesize fake data that have the same distribution with the real data, so as to supplement the real data and increase the number of available samples. The model first pre-trains the autoencoder which maps given dataset into a low-dimensional continuous space. Then, the generator constructed in the low-dimension space is obtained by training it adversarially with discriminator constructed in the original space. Since the constructed discriminator not only consider the loss of the continuous attributes but also the labeled attributes, the generator nets formed by the generator and the decoder can effectively learn the intrinsic distribution of the mixed data. We evaluate the proposed method both in the independent distribution of the attribute and in the relationship of the attributes, and the experiment results show that the proposed generate method has a better performance in preserve the intrinsic distribution compared with other generation algorithms based on deep learning.

Keywords: Generative adversarial network, Autoencoder, Mixed type data

1 Introduction

In the Internet era, with the convergence and integration of information technology and human production and life, big data has exerted a significant impact on economic development, social governance and people's lives. Through big data analysis, user groups can be more reasonably divided to provide more accurate services. However, when the big data platform provides a large amount of data to some technology companies for data analysis, it will inevitably increase the risk of users' privacy information disclosure, which is the focus of concern in the financial and medical fields. In order to reduce the negative impact of privacy information disclosure, the United States, the European Union, China and other countries or organizations continue to improve privacy protection regulations to regulate enterprises and individuals, so as to reduce or limit the sharing and opening of data [1].

In this context, big data analysis and research often encounter problems such as lack of data and too few training samples. In order to solve this problem, the current research ideas are mainly carried out from two aspects: information hiding and data generation. From the perspective of data hiding, for example, health care organization (HCO) can reduce the risk of information leakage by interfering with potential identifiable attributes through generalization, suppression and randomization and then sharing data [2, 3]. However, criminals can still restore the personal tags corresponding to the data through the remaining attribute information, so as to restore the original data.

With the development of deep learning and various learning model proposed, data generation based methods have attracted more and more attention in the field of data privacy protection. Its main idea is to capture the potential distribution structure of data sets by learning from very limited real data, and then generate synthetic data having similar distribution with the real data, so as to solve the problem of data deficiency [4]. In this work, we focus on generating high-dimensional mixed-type (continuous and discrete) data, compared with single-type data no matter continuous or discrete, which is a more important and challenging problem on its own. We propose a new data generation architecture which combines the versatility of an autoencoder with the recent success of Adversarial Networks (GANs) on complex data type. To assess the quality of the synthetic data, we define several new metrics that evaluate the performance of synthetic mixed-type data compared to the original data.

2 Related works

Nowadays, depth-generation model has been proved to be a highly flexible and expressible unsupervised learning method that can capture the potential structure of complex high-dimensional data. The well-trained depth generation model can effectively simulate the complex distribution of high-dimensional data and generate synthetic data similar to the original data [5, 6]. Early work on data generation are more widely based on Variational Autoencoder(VAE) [7], such as Variational Lossy Autoencoder [8], DVAE++ [9] and ShapeVAE [10]. These method have been shown to be efficient and accurate to capture the latent structure of vast amounts of complex high-dimensional data. However, they can not handle data with discrete featrues let alone continuous and discrete mixed data generation. Recently, Nazábal [11] proposed a general framework named HI-VAE, which is suitable for heterogenous data generation and presents competitive predictive performance in supervised task.

The GANs model have achieved great success in the field of synthesize image generation, such as MMD-GAN [12], AdaGAN [13] and WGANs [14], which adopts the idea of antagonistic game and consists of two parts, generator $G(\cdot)$ and discriminator $D(\cdot)$: the generator learns the distribution of the real samples and generates fake data to simulate the real data; the discriminator aims to distinguish between the real data and the fake data [15, 16].

With the practical application and theoretical development of GANs, more and more data scientists scholars have turned their attention to the this model [17]. At present, most researches related to GANs are focused on continuous datasets, but the application of big data science usually involves discrete variables with multi-label features. Training networks with discrete outputs is a main challenge that curbs the application of the

GANs in the field of big data analysis. The main difficulty behind this is that the output of the network is always transformed by softmax function into a multinominal distribution. However, sampling from this distribution is not a differentiable operation, which curbs the gradient flow to back propagate during the training of GANs for data with discrete features. To tackle this problem, the Gumbel-softmax technique is proposed to be equipped in the VAE and GANs based method for sequences discrete data generation [18–20]. Aiming at the same problem, seqGAN [21] proposes a stochastic strategy based on reinforcement learning to avoid the back propagation of discrete sequences.

Another method to avoid the back propagation of discrete data is Adversarially regularized autoencoders(ARAE) [22]. The author transforms the discrete words learned from text into continuous potential feature space, and uses GANs to generate potential feature distribution, which effectively improve the training stability and obtain a loss more correlated with sample quality. medGAN proposed by Choi et al. [23] is inspired from this concept, which can learn the realistic healthcare patient records and generate the synthesize data. The model hybrid the autoencoder with GANs, which first pre-train an autoencoder and then the generator maps latent code space back to original space, and the discriminator receives the fake data from generator or sample from real data to form an adversarial learning.

To improves the medGAN for generating of multi-label variables, Camino et al. [24] proposed Multi-categorical GANs based on the concept of medGAN. The idea behind it is to encode the multi-label variables into a binary representation using one-hot encodings [25], and apply Gumbel-Softmax [18] to solved the problem of multi-label data back propagation which improves the computation stability and convergence speed.

To the extent of our knowledge, most of the GANs based data generation work are focus on single type feature data generation, numerical type or discrete type. Apart from these research, we propose a mixed-type date generation model based on GANs, which improves the performance of mixed-type data generation by leveraging the fact that autoencoder has the ability to learn the intrinsic characteristic of mixed-type features and build the generator in the code space. The proposed framework equip the Gumel-softmax technique to deal with the problem of undifferential of discrete random varialbes, and optimized the loss function to balance the gradient flow coming from different mixed type features. We also provide elaborate empirical evaluation for generation model based on the Lending Club datasets. The results demonstrate that the proposed method has better performance than state-of-the-art VAE based method [11] not only in terms of approximation of distribution for single feature by also for approximation of the correlation between features.

3 Methods

3.1 Description of mixed-type data

In this paper, we assumes that the features of the data is composed by two types: numerical type and muti-label type. The data space is defined as $\mathcal{S} = (\mathcal{W} \times \mathcal{V})$, where the numerical space $\mathcal{W} = \mathcal{W}_1 \times \cdots \times \mathcal{W}_M (\mathcal{W} \in \mathbb{R}^M)$. In numerical space, we define random vector as $\boldsymbol{x} = (x^1, \ldots, x^M) \in \mathcal{W}$. The multi-label space is formed as $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_N$, Where $\mathcal{V}_i$ represent each multi-label feature(such as men and women, some possible occupation, etc.), the number for each categories per label is defined as $d_i = |\mathcal{V}_i|$. We

also define the random variable in space $\mathcal{V}$ as $\boldsymbol{v} = (v^1, v^2, \ldots, v^N) \in \mathcal{V}$, and each label variable v^i is encoded by one-hot and denoted as a vector $y^i \in \{0,1\}^{d_i}$. So the random variable in space S can be fully expressed as $\mathcal{S} = (\boldsymbol{x}, \boldsymbol{y}) = (x^1, \ldots, x^M, y^1, \ldots, y^N)$, and $y^i = (y^{i,1}, \ldots, y^{i,d_i})$.

3.2 The proposed mixGAN

The mixGAN proposed in this paper first pre-trains an autoencoder, which maps the mixed data space to a low-dimensional continuous space. Due to the fact that the intrinsic feature of the data can be more efficiently represent in the mapped low-dimensional continuous code space, the generator $G(\cdot)$ of the mixGAN is established in code space. The discriminator $D(\cdot)$ is established in the original mixed-type data space to identify the real data or fake data. The mixGAN is obtained by joint antagonistic learning between the generative network $G(\cdot)$ and discriminator D, and trained across over the original space and code space. Our mixGAN model is represented from the Pre-autoencoder to GANs respectively.

3.2.1 Pre-autoencoder

The autoencoder is composed by a encoder and a decoder. The encoder compresses the original high-dimentsional data to the low-dimension code space. Then, the decoder maps the code space back to the original data space. The auto-encoder network is trained to obtain encoder and decoder network, so that after the original data x go through the whole autoencoder system, the output of the network is a good approximation $\hat{x}$ to the input. Our proposed Pre-autoencoder modifies the traditional autoencoder by replacing the last output layer with a mixed-type layer output, which is formed by $N+1$ parallel features extraction Dense layers as shown in Fig. 1. At the end this parallel structure are the activation output function to transfer the components back to their original features.

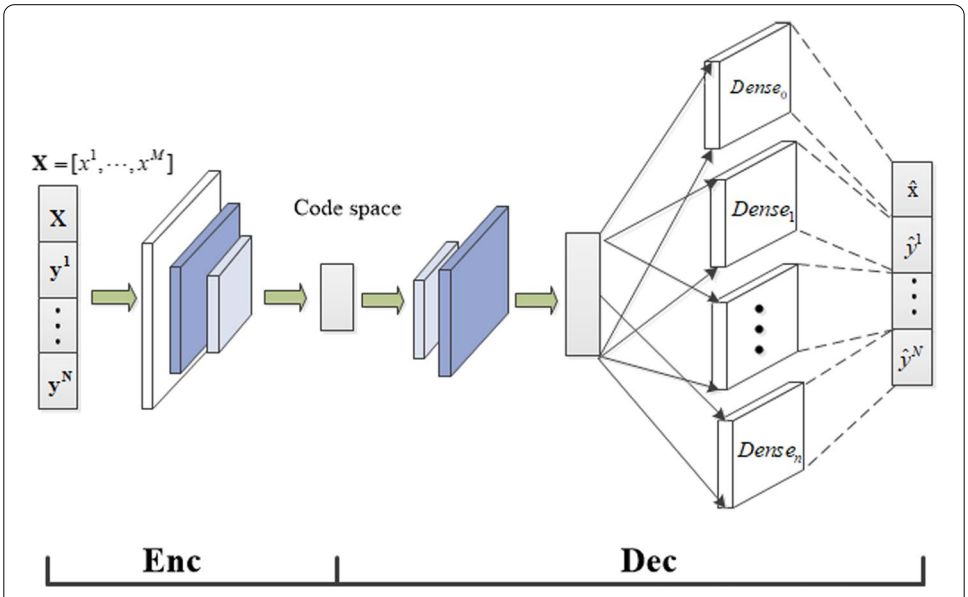


Fig. 1 Autoencoder

The parallel structure of the output layer model not only guarantees the independence of the each single feature but also maintains the interdependence between features.

The encoder network is simply composed by two layers FCN. The decoder network is firstly composed by two FCN mapping the code space to a continuous lower vector, after that, there is an $N+1$ parallel data type separation networks $\text{Dense}^0, \ldots, \text{Dense}^N]$, where Dense^0 represents the generation of multiple numerical vector $\boldsymbol{x} = [x^1, \ldots, x^M]$, which are activated by sigmoid layer. $[\text{Dense}^1, \ldots, \text{Dense}^N]$ represents the generation of N one-hot encoded vectors $\boldsymbol{y} = [y^1, \ldots, y^N]$, which is activated by Gumbel-Softmax layer for output. Finally, all the output results are concatenate together to obtain the generated mixed data $\hat{s} = [\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}] = [\hat{x}^1; \ldots; \hat{x}^M; \hat{y}^1; \ldots; \hat{y}]$. The model is shown in Fig. 1.

In this model, the Gumbel-softmax sampling technique is used to sample the discrete distribution, which widely used for discrete data generation, since it has the ability to solve the problem of discrete random data back-propagation [18]. Gumbel-softmax sampling technique models the hidden variable as a discrete multinomial distribution, and the transformation process satisfies the following formula:

$$y^{j,k} = \frac{\exp((\log(a^{j,k}) + g_k)/\tau)}{\sum_{i=1}^{d_j} \exp((\log(a^{j,i}) + g_i)/\tau)} \tag{1}$$

where $j = 1, \ldots, N$, $k = 1, \ldots, d_j$, and a^j is the output of full connection layer Dense_j, and $a^{j,k}$ is the output of Dense_j's k-th component. $\tau \in (0, \infty)$ is a hyperparameter greater than zero, which controls the softening degree: the higher the τ value is, the smoother the distribution; The lower the τ value is, the closer the generated distribution is to the discrete One-Hot distribution. In the process of training, the real discrete distribution can be approached gradually by gradually decreasing τ. Let g_i be i.i.d samples drawn from $\text{Gumbel}(0,1) = -\log(-\log(u_i))$ with $u_i \sim U(0,1)$.

Our pre-autoencoder loss function is shown in (2), which is compose of two parts: the the mean square error is utilized for the loss of numerical type and cross entropy error is utilized for the loss of multi-label type. Before input the training data to our model, we will first normalize the numerical features to (0,1), which can balance the two type of the loss in (2) and address the problem that the numerical type loss would dominate all loss and lead to poor performance for multi-lable type data approximation.

$$L_{\text{rec}} = \frac{1}{B} \sum_{i=1}^{B} \left(\sum_{m=1}^{M} \left(x_i^m - \hat{x}_i^m\right)^2 + \sum_{j=1}^{N} \sum_{k=1}^{d_j} \left(-y_i^{j,k} \log \hat{y}_i^{j,k}\right) \right) \tag{2}$$

where x^m represents the m-th component of $\boldsymbol{x}$, $y^{j,k}$ represents the k-th component of multi label feature y^j, and B is the size of training batch.

3.2.2 Generative adversarial network

The generative confrontation network consists of two network modules: the generator network and the discriminator network [15]. The generator $G(z; \theta_g)$ learns the distribution of the training data, and converts the input random prior distribution into a generated sample $G(z)$ with a similar distribution to the training data. The discriminator $D(x; \theta_d)$ is a two type classifier used to determine whether the input data set is a real sample or a generated fake

sample, that is, the discriminator will output a larger probability for real data, and a smaller probability for false data. In the training process, $G(\cdot)$ and $D(\cdot)$ are made to play against each other until the data generated by $G(\cdot)$ can "cheat" $D(\cdot)$. the optimization goal of the above game process can be expressed as:

$$\min_G \max_D V(D,G) = E_{x\sim P_{\text{data}}}[\log D(x)] + E_{x\sim P_z}[\log(1 - D(G(z)))] \tag{3}$$

where P_{data} represents the distribution of real samples, and P_z represents a random prior distribution subject to $\mathcal{N}(0,1)$. In the process of alternating training $G(\cdot)$ and D, the parameter optimization follows the following iterative formula:

$$\theta_d \leftarrow \theta_d + \alpha\nabla_{\theta_d}\frac{1}{B}\sum_{i=1}^{B}(\log(x_i) + \log(1 - D(G(z_i))) \tag{4}$$

$$\theta_g \leftarrow \theta_g + \alpha\nabla_{\theta_g}\frac{1}{B}\sum_{i=1}^{B}\log D(G(z_i)) \tag{5}$$

where B is the size of each training batch, and α is the iterative step size of the optimizer.

3.2.3 The architecture of mixGAN

The proposed mixGAN is constructed across the code space and original space. The method is inspired by the recent successes in discrete data generation using GANs [24], which addressed the difficulty of discrete random variable back propagation by using Gumbel-softmax sampling technique. We use the encoder which comes from the pre-trained autoencoder to map the original data to a low-dimensional continuous code space, where we build the GANs based generator.

basedUtilize this concept, the generator network $G(z)$ transfer the standard gaussian variable $z \sim N(0,1)$ to code space, then, the Decoder network $\text{Dec}(\cdot)$ maps the generated continuous variable back to original space $\hat{s}$. This process is shown in Fig. 2, and can be expressed as $\text{Dec}(G(z))$ appeared in generation loss (7). The discriminator $D(\cdot)$ is build in the original space, which judges weather the input item is real or fake by using the discrimination loss (6).

The proposed mixGAN is an architechture coupling the pre-autoencoder model and GANs structure, which combines the ability that the pre-autoencoder can capture the mixed-type data information and the ability of GANs which has high performance for continuous data generation. At the same time, the limitation of the discrete data learning ability of GANs is solved by this architechture.

As shown in Fig. 2, the data generated by the generator $G(\cdot)$ is decoded before being imported into the discriminator. It can be seen that the discriminator D's judgment of the authenticity of the data is performed in the original space. In the training process, the loss functions for discriminator $D(\cdot)$ and generator $G(\cdot)$ are represented in (6) and (7):

$$L_d = \frac{1}{B}\sum_{i=1}^{B}(\log D(x_i) + \log(1 - D(\text{Dec}(G(z_i))))) \tag{6}$$

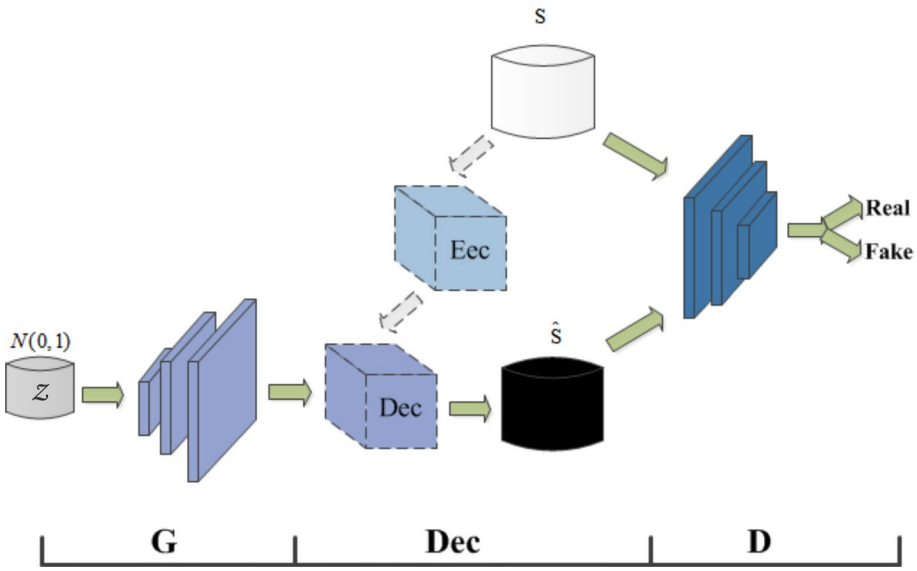


Fig. 2 mixGAN

$$L_g = \frac{1}{B}\sum_{i=1}^{B} \log D(\mathrm{Dec}(G(z_i))) \tag{7}$$

During the main training phase, the gradients flow from the discriminator to the decoder and afterwards to the generator, and the decoder will be fine-tuned while optimizing the generator.

4 Experiment

To assess the performance of our model, we use HI-VAE method [11] as a benchmark, we uses it as a benchmark for comparative evaluation. HI-VAE distinguishes between different feature types in the data when encoding and decoding, and designs a corresponding probability model for each type. According to the probability model corresponding to each features, the HI-VAE encoder processes the feature individually, and aggregates all attribute processing results to generate the code. The HI-VAE decoder performs the inverse process of the above processing, that is, the code is converted into various feature values and concatnate together.

4.1 Data acquisition

Our training dataset is a subset of high dimensional bank customs, which is hosted by Lending Club [26]. We randomly sampling 10,000 recorders from the original dataset, which are partitioned by 9:1 for training set and test set. The original dataset has 31 features, and we removed the 7 of them which have constant value. We rearrange the features of the dataset, so that the features of the dataset matches our data model; first 15 features are numerical type and the rest 9 features are mult-label type with One-Hot coded. Hence, we have $s_i = [x^1; \ldots; x^{15}; y^1; \ldots; y^9]$, and category number for each label type is listed as (2, 2, 2, 12, 2, 7, 29, 4, 3).

There is a common problem in the big data processing, that is, most time the numerical type values always have quite different magnitude than One-hot coded label type. Therefore, if we training the model using the raw data, the gradient flow come from the numerical type will dominant the back propagation, which will weaken the learning ability and reliability. In our experiment, we utilize Min–Max normalization method to stretch the range of the numerical features into 0–1, in order to make their ranges have similar magnitude with the one-hot coded multi-label features. Empirically, the normalization process not only improves the accuracy of the model but also accelerate the convergence of the training.

4.2 Implementation details

The proposed pre-autoencoder of the model contains two hidden FCM layers for both encoder and decoder, all the layers are activated by tanh function. We empirically set the latent continuous code space to 72 dimension, and the hyperparameter τ appeared in Gumbel-Softma activation function is set as 0.6.

For GANs training, the generator $G(\cdot)$ and discriminator $D(\cdot)$ of GANs are all implemented based on FCM with 3 layers for each, which are [256, 128, 72] and [128, 64, 1]. The batch normalization skill is also used between the layers. Referring to the work in [24], the hidden layers in $G(\cdot)$ are activated by Tanh function, while the hidden layer of $D(\cdot)$ are activated by LeakyRelu function. We use Adam algorithm to optimize the model, and set the learning rate $\mathrm{lr} = 0.002$ and set weight decay as 0.001. The batch size is set as $B = 100$. Finally, the training time of the pre-autoencoder is 52.30s, and the training time of the mixGAN model is 880.64s.

5 Results

To evaluate the performance of the GANs is widely known as a difficult task [27]. Borji [27] provides a range of commonly used metrics used for assessing the performance of the GANs, but they are not suitable for big data generation evaluation. In this paper, we suppose that if the generated data have a good approximation to the original data, it should satisfy the following two conditions: firstly, in terms of each single feature, the distribution of the generated value should be as close as possible to the real data distribution; secondly, The dependency among features should be similar to that of real data. Based on the above assumptions, we evaluates the performance of the mixGAN from perspective of the distribution approximation for single feature and the correlation maintenance between features.

5.1 Distribution approximation for single feature

To evaluate the approximation for independent distribution in each feature, we deals with the features of the numeric type and the label type respectively. For the numerical type x^i, we quantified the interval (0–1) into 10 bins, by which we can calculate the histograms of the generated and the original feature. After that, we pair each histogram bin using the original real distribution and the generated fake data $(P_{\mathrm{real}}, P_{\mathrm{fake}})$. Similar to the concept of the joint histogram, if the two random variables have similar distribution, the paired points $(P_{\mathrm{real}}, P_{\mathrm{fake}})$ should located diagonally alone joint distribution coordinate plane.

The similar concept is applied to the mult-label type features. We can see that each component of the $y^{i,j}$ is either 1 or 0, since the label type y^i has been one-hot coded. Hence, we accumulate all data across the each feature component $y^{i,j}$ and denoted it with P_{real} and P_{fake} for original label feature and generated label feature. It can be proved that if the synthesized label type features y^i have a good approximation to the original data, the paired points $(P_{\text{real}}, P_{\text{fake}})$ should also distribute alone the diagonal of the coordinate plane.

Following these concept, we plot the paired points $(P_{\text{real}}, P_{\text{fake}})$ in the Fig. 3, where the (a) is drawn by using our proposed mixGAN, and (b) is drawn using the HI-VAE proposed in [11]. The circular point represents the label type, and the star point represents the numerical type. We can find Fig. 3 that mixGAN has a apparently better performance than HI-VAE in independent feature approximation, since the paired points come from mixGAN, not only the numerical type or label type, are all distributed more closer to the diagonal than HI-VAE.

5.2 Correlation maintenance between features

The basic idea for assessing the correlation between features of generated data is: in generated dataset, the impact for a feature f_i come from the rest of the features should be as similar as possible to the original data. According to the concept, we establish a learning model to estimate the feature f_i by using the rest features. The model is formulated as a multi classification task, when f_i is of multi-label type, and formulate it as a regression task when f_i is of numerical type. We denote the estimation loss for f_i by using the real data as E^i_{real}, and E^i_{fake} for the generated data.

In testing, all the estimation model is formed by FCN, but the loss function is formulated depend on the feature type of f_i. We formulate the loss function for numerical feature f_i as $\frac{1}{N}\sum_{j=1}^{N}(x^i_j - \hat{x}^i_j)^2$, and formulate the loss function as $\frac{1}{N}\sum_j^N I(y^i_j = \hat{y}^i_j)$ where $I(\cdot)$ is indicator function, and N is the total number of samples in testing set.

In Fig. 4, we plot all the paired points $(E_{\text{real}}, E_{\text{fake}})$ in the plane, where (a) and (b) are the estimated errors by using the mixGAN and HI-VAE [11]. It shows that the proposed mixGAN method is superior to HI-VAE in the maintenance of features correlation especially better for numerical type features.

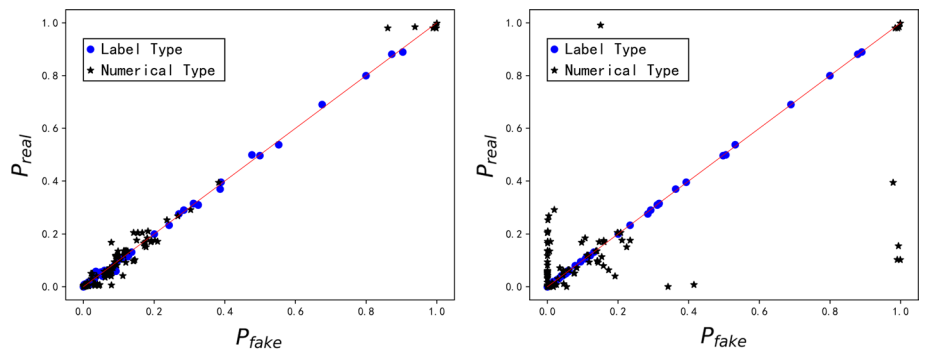
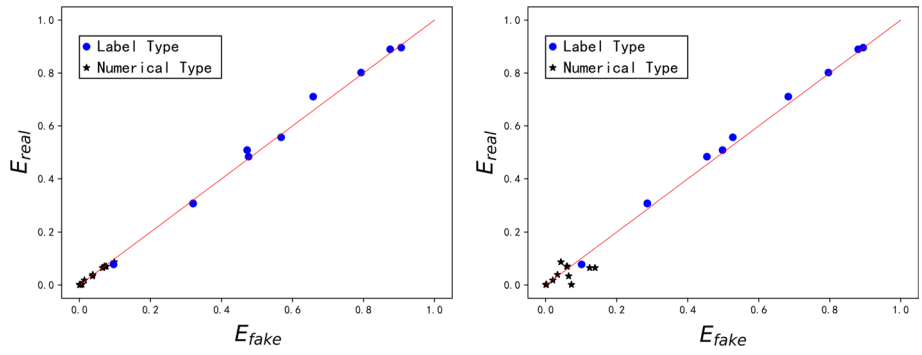


Fig. 3 Generated the distribution map for matching the independent attributes of data and real data



(a) mixGAN (b) HI-VAE

Fig. 4 Generated multi-attribute matching distribution map of data and real data

6 Discussion

In this work, we proposed mixGAN, which uses generative adversarial framework to generate the synthetic mixed-type data. Apart from the traditional method, our framework improves the performance of generated data by leveraging the fact that autoencoder has the ability to learn the intrinsic characteristic of mixed-type features and build the generator in the code space, which also solved the problem of gradient back propagation for discrete variables. We also provide elaborate empirical evaluation for generation model based on the Lending Club datasets, which demonstrate that our method has better performance not only in terms of approximation of distribution for single feature but also for approximation of the correlation between features.

In the future, we are planning to improve the robustness of the model, so that the model can generate synthesis mixed-type data even when some of the features are missing in the original data samples. This will widen the extent of application of our model.

Abbreviations

GANs: Generative adversarial networks; HCOs: Health care organization; VAE: Variational autoencoder; FCN: Fully convolutional networks.

Acknowledgements

The authors thank the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Author Contributions

NW and PC conceived and designed the study. LW and GC developed the simulations and performed the computation. NW, LW and PC wrote the paper. SS, YW reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Key Research and Development Program of China through the Ministry of Science and Technology of China (No. 2016YFC0802500), in part by the National Natural Science Foundation of China (No. 61871258), and in part by the National Social Science Fund of China (No. 20BTQ066).

Availability of data and materials

The datasets used in this study is hosted by Lending Club [26].

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]The College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China. [2]Center for Governance Studies, Beijing Normal University, Zhuhai 519087, China.

Received: 10 September 2021 Accepted: 10 March 2022
Published online: 25 March 2022

References

1. W. Zhong, Y. Jinali, Design of personal data privacy disclosure traceability mechanism in big data environment. China Bus. Mark. **8**, 117–121 (2014)
2. U.D. of Health and Human Services, et al., *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule* (US Department of Health and Human Services, Washington, DC, 2012), Available at: https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html, Accessed 26 Sept 2018
3. K. El Emam, S. Rodgers, B. Malin, Anonymising and sharing individual patient data. BMJ **350**, 1139 (2015)
4. A.L. Buczak, S. Babin, L. Moniz, Data-driven approach for creating synthetic electronic medical records. BMC Med. Inform. Decis. Mak. **10**(1), 59 (2010)
5. D.J. Rezende, S. Mohamed, Variational inference with normalizing flows (2015). arXiv preprint arXiv:1505.05770
6. N.I.U. Bin, M.L.W.U. Peng, A behavior data set extension method based on generative adversarial network. Comput. Technol. Dev. **29**(07), 43–48 (2019)
7. D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: *ICLR 2014: International Conference on Learning Representations (ICLR) 2014* (2014)
8. X. Chen, D.P., Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, P. Abbeel, Variational lossy autoencoder, in: *ICLR 2017: International Conference on Learning Representations 2017* (2017)
9. A. Vahdat, W. Macready, Z. Bian, A. Khoshaman, Dvae++: Discrete variational autoencoders with overlapping transformations, in: *ICML 2018: Thirty-Fifth International Conference on Machine Learning* (2018), pp. 5035–5044
10. C. Nash, C.K.I. Williams, The shape variational autoencoder: a deep generative model of part-segmented 3d objects. Comput. Graph. Forum **36**(5), 1–12 (2017)
11. A. Nazábal, P.M. Olmos. Z. Ghahramani, I. Valera, Handling incomplete heterogeneous data using vaes (2018). arXiv preprint arXiv:1807.03653
12. Y. Li, K. Swersky, R. Zemel, Generative moment matching networks, in: *Proceedings of The 32nd International Conference on Machine Learning* (2015), pp. 1718–1727
13. I.O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, B. Schölkopf, Adagan: Boosting generative models, in: *31st Annual Conference on Neural Information Processing Systems (NIPS 2017)* (2017), pp. 5424–5433
14. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan (2017). arXiv preprint arXiv:1701.07875
15. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in *Advances in Neural Information Processing Systems 27 (2014)*, pp. 2672–2680
16. S.Y. Zhao, J.W. Li, Generative adversarial network for generating low-rank images. Acta Autom. Sin. **44**(05), 64–74 (2018)
17. C. Mengting, Z. Yuanping, Research and application progress of generative adversarial networks. Comput. Eng. **45**(09), 222–234 (2019)
18. E. Jang, S. Gu, B. Poole, Categorical reparameterization with Gumbel-Softmax, in: *ICLR 2017: International Conference on Learning Representations 2017* (2017)
19. C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: a continuous relaxation of discrete random variables, in: *ICLR 2017: International Conference on Learning Representations 2017* (2017)
20. M.J. Kusner, J.M. Hernández-Lobato, Gans for sequences of discrete elements with the Gumbel-Softmax distribution (2016). arXiv preprint arXiv:1611.04051
21. L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)* (pp. 2852–2858). *Association for the Advancement of Artificial Intelligence (AAAI)* (2017) (In Press). (2016), pp. 2852–2858
22. J.J. Zhao, Y. Kim, K. Zhang, A.M. Rush, Y. LeCun, Adversarially regularized autoencoders, in: *ICLR 2018: International Conference on Learning Representations 2018* (2018)
23. E. Choi, S. Biswal, B.A. Malin, J. Duke, W.F. Stewart, J. Sun, Generating multi-label discrete patient records using generative adversarial networks, in: *Machine Learning for Healthcare Conference* (2017), pp. 286–305
24. R.D. Camino, C. Hammerschmidt, R. State, Generating multi-categorical samples with generative adversarial networks (2018). arXiv preprint arXiv:1807.01202
25. S. Suh, S. Choi, Gaussian copula variational autoencoders for mixed data (2016). arXiv preprint arXiv:1604.04960
26. https://www.lendingclub.com
27. A. Borji, Pros and cons of gan evaluation measures. Comput. Vis. Image Underst. **179**, 41–65 (2019)

Publisher's Note