International Journal of Internet, Broadcasting and Communication Vol.11 No.4 37-42 (2019)
http://dx.doi.org/10.7236/IJIBC.2019.11.4.37

IJIBC 19-4-6

Experimental Analysis of Equilibrization in Binary Classification for Non-Image Imbalanced Data Using Wasserstein GAN

Zhi-Yong Wang[1,2], Dae-Ki Kang[2]

[1]*Weifang University of Science and Technology, China*
[2] *Department of Computer Engineering, Dongseo University, Busan, Korea*

wangzhiyong6688@163.com, *dkkang@gmail.com*

Abstract

In this paper, we explore the details of three classic data augmentation methods and two generative model based oversampling methods. The three classic data augmentation methods are random sampling (RANDOM), Synthetic Minority Over-sampling Technique (SMOTE), and Adaptive Synthetic Sampling (ADASYN). The two generative model based oversampling methods are Conditional Generative Adversarial Network (CGAN) and Wasserstein Generative Adversarial Network (WGAN). In imbalanced data, the whole instances are divided into majority class and minority class, where majority class occupies most of the instances in the training set and minority class only includes a few instances. Generative models have their own advantages when they are used to generate more plausible samples referring to the distribution of the minority class. We also adopt CGAN to compare the data augmentation performance with other methods. The experimental results show that WGAN-based oversampling technique is more stable than other approaches (RANDOM, SMOTE, ADASYN and CGAN) even with the very limited training datasets. However, when the imbalanced ratio is too small, generative model based approaches cannot achieve satisfying performance than the conventional data augmentation techniques. These results suggest us one of future research directions.

Keywords: *Wasserstein GAN, Over-sampling, Imbalanced data, Data augmentation methods*

1. Introduction

Imbalanced data may cause learning bias (also called suboptimal results) [1]. This learning bias can expand the decision boundary of the majority class and thereby deteriorate minority class recognition performance. Oversampling replicates the minority samples so that the distribution is balanced. The most commonly used oversampling methods are Random Sampling [2], Synthetic Minority Over-sampling Technique (SMOTE) [3] and Adaptive Synthetic Sampling (ADASYN) [4]. Our approach takes advantage of Generative Adversarial Network (GAN) [5] to capture the modes of distribution and to overcome

Manuscript Received: September. 2, 2019 / Revised: September. 8 , 2019 / Accepted: September. 15, 2019
Corresponding Author: dkkang@dongseo.ac.kr
Tel: +82-51-320-1724, Fax: +82-51-327-8955
Department of Computer Engineering, Dongseo University, Busan, Korea

imbalance problem in the original datasets by artificially generating data samples. We compare Conditional GAN (CGAN) [6] and Wasserstein GAN (WGAN) [7] with conventional data augmentation methods (SMOTE and ADASYN) through a large amount of experiments based on 16 common datasets.

2. Related Work

Random Oversampling is one of the earliest resampling methods which adds minority classes samples by easily duplicating the member of minority classes, which is proven to be robust, but the information learned by the model is too special to be generalized and easily over-fitted [2]. To cope with this problem, SMOTE analyzes minority class and synthesizes new samples according to the minority class. However, SMOTE has two main problems. First, there is some blindness in the selection of k-nearest neighbors. The choice of k value needs to be determined by the user. In addition, SMOTE is prone to the problem of marginalization. This type of marginalization will blur the boundaries of positive and negative class. This can be easily aggravated during the training phase and increases the difficulty of classification. Instead of sampling from k values, ADASYN adaptively generates minority samples according to minority distribution.

3. GAN Topologies for Experiments

In these experiments, we have adopted WGAN because it tries to approximate Earth Mover (EM) distance for smooth measurement of the distances between distributions. The architecture of a generator in our WGAN based on CNN is shown in Table 1.

Table 1. The architecture of the generator of our WGAN. "#" is the number of attributes.

Layer	Feature maps	Activation	Kernel size	padding
Input	100	-	-	-
Dense_1	1*#	ReLU	-	-
Conv2D_2	1*#*25	ReLU	2	1
Conv2D_3	1*#*50	ReLU	2	1
Conv2D_4	1*#*1	tanh	2	1

The input layer of the generator is a random vector which is produced from a normal distribution with 100 dimensions. The following layer in the generator is a fully connected layer with dimensions of 1*#, in other words, one multiplied by the number of attributes (denoted as #). Note that different dataset usually consists of different number of attributes. After the fully connected layer, there is a series of convolutional layers with ReLU activation and batch normalization with momentum=0.8. The output of the generator is 1*#*1 which should match the dimensionality of the input layer in the discriminator. Please refer to Table 1 for more detail. And the architecture of a discriminator in our WGAN based on CNN is shown in Table 2.

Table 2. The architecture of the discriminator of our WGAN. "#" is the Number of attribute.

Layer	Feature maps	Activation	Kernel size	strides	padding
input	1*#*1	-	-	-	-
Conv2D_1	1*#*4	LeakyReLU	2	1	1
Conv2D_2	1*#*9	LeakyReLU	2	1	1
Conv2D_3	1*#*18	LeakyReLU	2	1	1
Conv2D_4	1*#*9	LeakyReLU	2	1	1

Flatten_5	171	-	-	-	-
Dense_6	1	-	-	-	-

The input of the discriminator is also a type of vector. It can be a real sample or a generated one, with dimensions of 1*#*1 followed by a series of convolutional layers which paired with batch normalization with momentum=0.8 and Dropout layer. We adopt ReLU as activation in the generator network whereas in the discriminator, we use LeakyReLU with alpha=0.2. The Flatten layer is able to change a tensor data to a vector. The optimizer of the WGAN is RMSProp with learning rate=5e-5. This architecture is summarized in Table 2.

4. Experimental Results

We have performed over-sampling on sixteen datasets in order to evaluate our idea for alleviating data imbalance problems. All these datasets are available from the UCI Machine Learning Repository [8].

First, we divide datasets into 'train' and 'test'. The test set should be balanced and has about 50% instances of minority class. For example, if class '+' has 100 instances and class '–' has 500 instances (100,500) in the original dataset, then we divide the set so that the training set has (50,450), and the test set has (50,50), as shown in Table 3. Second, we train a neural network and SVM with the training set and classify the test set as the baseline without over-sampling. For the neural network, we set the number of layers N=4 in which the number of nodes is 100, 50, 100, 2, respectively. As for the setup of SVM, we use cost parameter as 0.1, linear kernel, and one-versus-rest strategy.

Table 3. divided datasets

DATASET Name	#training set	#test set
BREAST-CANCER	(40,166)	(40,40)
BREAST-W	(121,338)	(120,120)
COLIC	(73,159)	(73,73)
COLIC-ORIG	(62,182)	(62,62)
CREDIT-A	(154,230)	(153,153)
CREDIT-G	(150,550)	(150,150)
DIABETES	(134,366)	(134,134)
HEART-STATLOG	(60,90)	(60,60)
HEPATITIS	(34,125)	(33,33)
IONOSPHERE	(63,162)	(63,63)
KR-VS-KP	(764,906)	(763,763)
LABOR	(30,47)	(30,30)
MUSHROOM	(1458,2750)	(1458,1458)
SICK	(116,3426)	(115,115)
SONAR	(49,63)	(48,48)
VOTE	(84,183)	(84,84)

We use several different machine learning algorithms for our experiments. In the results shown in Table 4 (taking BREAST-CANCER dataset as an example), SVM_ BASELINE and NN_ BASELINE are trained by imbalanced training dataset (BREAST-CANCER) which has 286 instances including 80 '+' instances and 206 '-' instances. We divide it into a training dataset and a test dataset where the test data has (40,40) and the training data has (40,166). SVM_ BASELINE and NN_ BASELINE are trained by imbalanced training dataset (40,166), obtaining accuracy values 50.0 and 50.0 respectively. SVM_WHOLE (SVM trained with

the whole data) and NN_WHOLE (NN trained with the whole data) are trained by (166,166) training data in which 126 '+' instances of the training data are generated by GAN (CGAN or WGAN). The GAN's are trained by all instances in the dataset, i.e., 286 instances including 80 '+' instances and 206 '-' instances. Models named SVM_(50,50) (i.e. SVM with 50% instances of minority class and 50% instances of majority class) and NN_ (50,50) (i.e. NN with 50% instances of minority class and 50% instances of majority class) are using (40,103) subset of the training dataset for GAN (CGAN or WGAN) to obtain a balanced dataset. In models named SVM_(+,) and NN_(+,), we feed all '+' instances to GAN to make a balanced dataset. For the models, SVM_ KMSMOTE and NN_ KMSMOTE, we perform over-sampling using SMOTE. SVM_ ADASYN and NN_ ADASYN are from ADASYN oversampling. Finally, we use random oversampling for SVM_ RANDOM and NN_ RANDOM.

Table 4 reports the overall performances in terms of accuracy for different datasets over-sampling for the minority classes. The number of the new samples generated by all methods for the minority class is equal to number of majority instances minus number of minority instances. As can be observed, CGAN and WGAN do not outperform for all datasets. LABOR dataset (shown in italic in Table 4) has only 137 samples, which is the smallest one in all datasets, however, its CGAN and WGAN performance is significantly better than the conventional data augmentation techniques. Meanwhile, WGAN can also slightly outperform CGAN.

Table 4. Performance Accuracy

DATASET	SVM_ BASELINE	NN_ BASELINE	SVM_ RANDOM	NN_ RANDOM	SVM_ KMSMOTE	NN_ KMSMOTE	SVM_ADASYN	NN_ADASYN	GAN	SVM_ WHOLE	NN_ WHOLE	SVM_ (50,50)	NN_ (50,50)	SVM_ (+,)	NN_ (+,)
BREAST-CANCER	**50.0**	**50.0**	**53.0**	**62.0**	**59.0**	**62.0**	**60.0**	**61.0**	**CGAN**	**61.0**	**53.0**	**61.0**	**61.0**	**61.0**	**59.0**
									WGAN	**50.0**	**52.0**	**50.0**	**49.0**	**50.0**	**51.0**
BREAST-W	86.0	74.0	90.0	79.0	91.0	80.0	95.0	88.0	CGAN	95.0	95.0	86.0	92.0	85.0	95.0
									WGAN	95.0	82.0	93.0	80.0	92.0	78.0
COLIC	80.0	50.0	79.0	55.0	79.0	66.0	77.0	50.0	CGAN	79.0	65.0	81.0	59.0	80.0	59.0
									WGAN	84.0	65.0	83.0	50.0	82.0	50.0
COLIC-ORIG	50.0	50.0	71.0	49.0	64.0	50.0	71.0	50.0	CGAN	60.0	57.0	50.0	50.0	50.0	52.0
									WGAN	62.0	50.0	62.0	41.0	62.0	34.0
CREDIT-A	50.0	67.0	50.0	61.0	50.0	57.0	50.0	55.0	CGAN	50.0	59.0	50.0	55.0	50.0	59.0
									WGAN	50.0	67.0	50.0	53.0	50.0	50.0
CREDIT-G	**54.0**	**50.0**	**72.0**	**50.0**	**70.0**	**50.0**	**70.0**	**50.0**	**CGAN**	**56.0**	**50.0**	**58.0**	**51.0**	**57.0**	**56.0**
									WGAN	**62.0**	**50.0**	**60.0**	**50.0**	**64.0**	**50.0**
DIABETES	52.0	50.0	66.0	50.0	67.0	50.0	68.0	50.0	CGAN	70.0	64.0	48.0	47.0	48.0	48.0
									WGAN	66.0	50.0	55.0	54.0	55.0	50.0
HEART-STATLOG	82.0	50.0	80.0	50.0	79.0	50.0	79.0	50.0	CGAN	81.0	72.0	81.0	71.0	81.0	60.0
									WGAN	84.0	50.0	84.0	50.0	85.0	50.0
HEPATITIS	**50.0**	**50.0**	**73.3**	**66.7**	**73.3**	**50.0**	**76.7**	**50.0**	**CGAN**	**53.3**	**43.3**	**53.3**	**53.3**	**73.3**	**73.3**
									WGAN	**52.5**	**50.0**	**52.5**	**50.0**	**52.5**	**50.0**
IONOSPHERE	64.0	51.0	66.0	65.0	67.0	56.0	68.0	56.0	CGAN	75.0	62.0	56.0	74.0	74.0	55.0
									WGAN	63.0	55.0	63.0	55.0	64.0	61.0
KR-VS-KP	73.0	76.0	73.0	75.0	73.0	74.0	73.0	80.0	CGAN	73.0	69.0	78.0	79.0	74.0	88.0
									WGAN	77.0	78.0	77.0	87.0	77.0	90.0
LABOR	*65.0*	*50.0*	*95.0*	*60.0*	*95.0*	*50.0*	*95.0*	*50.0*	*CGAN*	*95.0*	*40.0*	*95.0*	*70.0*	*95.0*	*55.0*
									WGAN	*100.0*	*50.0*	*100.0*	*87.5*	*100.0*	*56.25*
MUSHROOM	95.0	50.0	95.0	88.0	95.0	92.0	95.0	88.0	CGAN	95.0	95.0	95.0	95.0	95.0	95.0
									WGAN	95.0	55.0	95.0	79.0	93.0	90.0
SICK	**50.0**	**50.0**	**79.0**	**70.0**	**77.0**	**69.0**	**76.0**	**62.0**	**CGAN**	**79.0**	**72.0**	**51.0**	**52.0**	**50.0**	**51.0**
									WGAN	**77.0**	**50.0**	**77.0**	**50.0**	**77.0**	**50.0**

SONAR	50.0	46.0	48.0	50.0	49.0	43.0	48.0	51.0	CGAN	60.0	49.0	56.0	34.0	57.0	50.0
									WGAN	50.0	50.0	50.0	50.0	50.0	51.0
VOTE	96.0	85.0	96.0	89.0	96.0	93.0	95.0	88.0	CGAN	96.0	94.0	96.0	92.0	96.0	92.0
									WGAN	96.0	88.0	96.0	95.0	96.0	90.0

Fig. 1 illustrates the performance of all methods on the LABOR dataset. It can be observed that the total running result on neural network (a) is not as good as on SVM (b). This should be attributed to the fact that there are too few samples in the dataset which results in that the neural network cannot be trained as well. It can be found that the worst performance in (a) is the CGAN_NN_WHOLE. Its accuracy is 40.0 which is even less than NN_BASELINE. Nevertheless, the corresponding result of WGAN (WGAN_NN_WHOLE) is 50.0, equal to NN_BASELINE. This suggests that WGAN performs more robustly even if the training samples are very small. In contrast, the stability of CGAN is slightly insufficient. In addition, the conventional data augmentation techniques can achieve higher accuracy for some situations in (a) and (b). This indicates the efficacy of the conventional data augmentation approach in the type of problems.

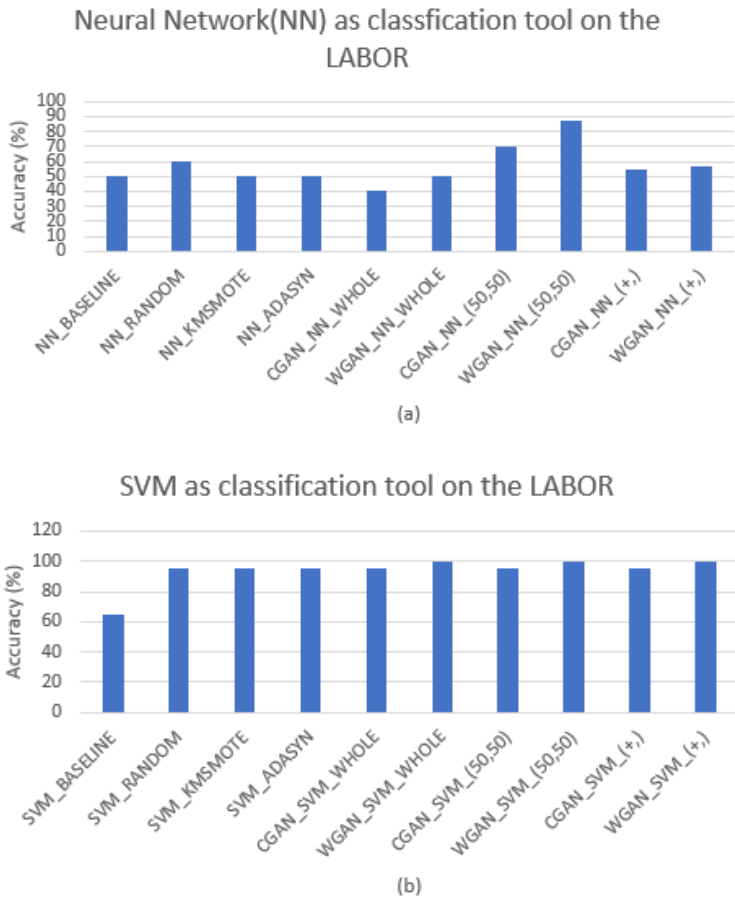

Figure 1. Comparative analysis of the conventional data augmentation techniques (RANDOM, KMSMOTE, ADASYN) and CGAN, WGAN on the LABOR dataset. (a) Performance on Neural Network (NN). (b) Performance on SVM.

5. CONCLUSION

In the future work, we will make attempt to use Bayesian-GAN since it is superior to CGAN and WGAN [9]. And we will study the more profound effects of imbalanced ratio on WGAN to provide reference on using it for data augmentation and various related tasks [10,11].

ACKNOWLEDGEMENT

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2018R1D1A1A02050166).

REFERENCE

[1] B. Krawczyk, "Learning from Imbalanced Data: Open Challenges and Future Directions," Progress in Artificial Intelligence, Vol. 5, No. 4, pp. 221-232, 2016.
DOI: http://dx.doi.org/10.1007/s13748-016-0094-0

[2] C. X. Ling, and C. Li. "Data Mining for Direct Marketing: Problems and Solutions," KDD, Vol. 98, pp. 73-79. 1998.

[3] N.V. Chawla, K.W. Bowyer, L.O. Hall and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, Vol. Jun 1, No. 16, pp. 321-357, 2002.
DOI: https://doi.org/10.1613/jair.953

[4] H. He, Y. Bai, E.A. Garcia and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in 2008 IEEE International Joint Conference on Neural Networks, pp. 1322-1328, Jun. 1, 2008.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets." in Proc. Neural Information Processing Systems 2014, pp. 2672–2680, Dec. 8-13, 2014.

[6] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," arXiv preprint, pp. 1411.1784, Nov 6, 2014.

[7] M. Arjovsky, S. Chintala, L. Bottou, "Wasserstein Generative Adversarial Networks," in Proc. International Conference on Machine Learning, pp. 214-223, Jul 17, 2017.

[8] UCI Machine Learning Repository. [Online]. http://archive.ics.uci.edu/ml/

[9] Y. Saatci and A.G. Wilson, "Bayesian GAN," in Proc. Neural Information Processing Systems 2017, pp. 3622-3631, Dec. 4-9, 2017.

[10] G. Agrawal, and D.-K. Kang, "Wine Quality Classification with Multilayer Perceptron," International Journal of Internet, Broadcasting and Communication (IJIBC), 10(2):25-30, May 2018.
DOI: http://dx.doi.org/10.7236/IJIBC.2016.8.4.19

[11] Ho, J., and Kang, D.-K., "Ensemble-By-Session Method on Keystroke Dynamics based User Authentication," International Journal of Internet, Broadcasting and Communication (IJIBC), 8(4), November 2016.
DOI: https://doi.org/10.7236/IJIBC.2018.10.2.5