A Genetic Algorithm to Optimize SMOTE and GAN Ratios in Class Imbalanced Datasets

Hwi-Yeon Cho
Department of Computer Science, Kwangwoon University
Seoul, Republic of Korea
hwyn.cho@gmail.com

Yong-Hyuk Kim
Department of Computer Science, Kwangwoon University
Seoul, Republic of Korea
yhdfly@kw.ac.kr

ABSTRACT

Class imbalance is one of the problem easily encountered in the fields of data analysis and machine learning. When there is an imbalance in learning dataset, machine learning models become biased and learn inaccurate classifiers. To resolve such data imbalance problems, a strategy that increases the volume of data of minority classes is often used by applying the synthetic minority oversampling technique (SMOTE). Furthermore, the use of generative adversarial networks (GANs) for data oversampling has recently become more common. This research used a genetic algorithm to search and optimize the combinations of oversampling ratios based on the SMOTE and GAN techniques. The case in which the proposed method was used was compared with the cases in which a single technique was used to train either the imbalanced data or oversampled data. From the results, it was established that the classifier that learned the oversampled data with the optimized ratio using the proposed method was superior in classification performance.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**; *Machine learning approaches*;

KEYWORDS

genetic algorithm, machine learning

ACM Reference Format:
Hwi-Yeon Cho and Yong-Hyuk Kim. 2020. A Genetic Algorithm to Optimize SMOTE and GAN Ratios in Class Imbalanced Datasets. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3398153

1 INTRODUCTION

A class imbalanced dataset refers to a dataset in which the ratios of data belonging to different classes are not equal but are skewed to one side. Imbalanced data is a problem that is commonly encountered in data classification using machine learning (ML). When training is conducted in a state of imbalanced data distribution, the ML-based classifier has a limitation characterized by over-fitting the data of a class with a high proportion in the training data. When the data are skewed to one side, the classifier is highly likely to misclassify the data belonging to the minority class by comparing it with the data belonging to the majority class.

A typical strategy for processing imbalanced data is the synthetic minority oversampling technique (SMOTE) [1]. SMOTE is a technique that intelligently expands the data of the minority class and shows better classification performance than the oversampling technique using simple replication. Recently, there have been various studies on generative adversarial networks (GANs) [2], and data sampling has been attempted using GANs, in addition to SMOTE, thereby improving classifier performance [3]. However, many experiments and tests should be conducted to improve the classification performance by adjusting the ratio of classes. Such a process may be difficult and inefficient. Study was conducted to find the optimal SMOTE ratio by using the support vector regression (SVR) technique, and the results were better than those of previous methods [4]. In this study, a genetic algorithm (GA) is used to search for various combinations of oversampling ratios based on the SMOTE and GAN techniques and to improve the performances of classifiers.

This paper is organized as follows: in Section 2, the proposed GA-based oversampling ratio optimization method is explained, and in Section 3, the experiments and results are discussed. Finally, the paper is concluded in Section 4.

2 GENETIC ALGORITHM

Encoding: a chromosome refers to a combination of oversampling ratios obtained using various techniques. Each gene of a chromosome indicates how much oversampling will be done using either the SMOTE or GAN technique, i.e., the sampling ratio for each technique.

Fitness: the fitness of each chromosome is calculated using the performance of a trained classifier. The data of the minority class in the training dataset are oversampled according to the combination represented by the chromosome, and the classifier is trained using the sampled dataset. The performance of the trained classifier is evaluated using the training dataset, and the F_1 score, which is an evaluation metric, is used for fitness.

Selection: we use the Roulette wheel method, which is the most representative selection method.

Crossover: since a chromosome is a one-dimensional array with a certain length, a one-point crossover, which exchanges the chromosomes of parents at an arbitrary point, is used as the crossover operator.

Mutation: two arbitrary points are chosen in the chromosome, and the mutation operator is used to exchange the genes at the two points. In other words, two techniques are selected, and the

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07…$15.00
https://doi.org/10.1145/3377929.3398153

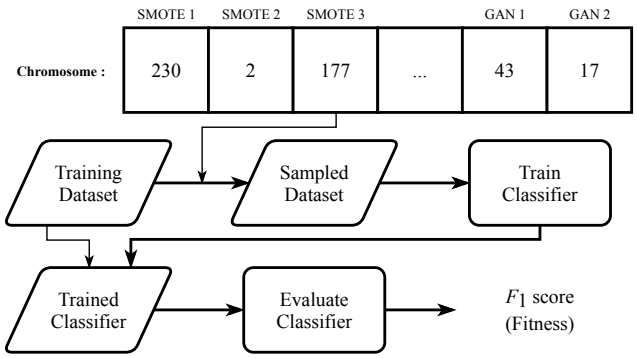

Figure 1: Example of fitness calculation

oversampling ratios of these techniques are exchanged.

Replacement: the replacement operator is used to maintain population diversity through an operation that replaces the parents with the offspring.

3 EXPERIMENTS AND RESULTS

We used a credit card fraud detection dataset in the experiments. This dataset is one of the representative class imbalanced datasets, and includes credit card transactions that were carried out in Europe for two days in September, 2013. The total number of datasets is 284,807; out of these 284,315 are normal transactions while 492 are fraud transactions. In the experiments, 75% of the total data was used as the training dataset while the remaining 25% was used to evaluate the trained model.

We used a deep-neural network (DNN)-based classifier in the experiment. The number of hidden layers in the neural network is 2, and the number of nodes in the hidden layer is 20 and 11. The neural network was implemented using *PyTorch*[1].

A total of 13 oversampling strategies with different parameters were used to increase the volume of data in the minority class in the experiments. The strategies can be divided into three main types: oversampling strategies using *k*-NN-based SMOTE, SVM algorithm-based SMOTE, and GAN. Specifically, there are three oversampling strategies that use the *k*-NN-based SMOTE, eight that use the SVM algorithm-based SMOTE, and two that use the GAN. The SMOTE strategies were provided by *Imbalanced-Learn*[2], and the GAN was implemented using *PyTorch*.

As described above, a total of 13 oversampling strategies were developed, and using the GA, we conducted experiments to investigate how to set and combine the oversampling ratios of each technique to find the optimal combination. The following parameters of GA were used: the length of each chromosome was 13, and the sum of all genes of each chromosome was initialized to be between 381 and 1,524. Each generation had a population size of 128, and a total of 100 generations were searched. Furthermore, 16 parents were selected in each generation and 16 offspring were transferred to the next generation. The probability of mutation was 1%. The performance of the trained classifier was compared with that of a classifier trained by oversampling with the optimal combination searched using the GA, and that of a classifier trained by oversampling using a strategy characterized by a certain ratio (the ratio was set to 200% with an increase of 381, 300% with an increase of 762, 400% with an increase of 1143, and 500% with an increase of 1,524). Our code for this paper and experiment is on *GitHub*[3].

As described in Section 2, we evaluated the trained classifier with the training dataset and calculated the F_1 score as the fitness. Table 1 shows the results of comparing the F_1 scores of the scenario where the imbalanced dataset was trained, that in which 13 techniques with four fixed ratios were used, and that in which the data sampled was trained with a ratio optimized using the GA. When the GA was used, the performance was better than those in all other cases.

Table 1: Results for each oversampling strategy

Case		F_1 score	Std.
Imbalanced	Train	0.8271	*NA*
	Evaluate	0.7797	*NA*
Single strategy (42 cases)	Train	0.8929	0.045
	Evaluate	0.8553	0.0373
Proposed strategy	Train	**0.9822**	*NA*
	Evaluate	**0.8895**	*NA*

4 CONCLUSION

In this study, we used the GA to explore and optimize the combination of oversampling ratios for each strategy using SMOTE and GAN. A comparison was made between the proposed method and the case in which the class imbalanced dataset was trained, and the one in which training was conducted using a single technique with oversampled data at a certain ratio. As a result, the classifier that learned the oversampled data with a ratio optimized using the proposed method was superior in classification performance in every case. Through this method, we confirmed the possibility of optimizing the combination of oversampling ratios.

ACKNOWLEDGMENTS

This research was a part of the project titled 'Marine Oil Spill Risk Assessment and Development of Response Support System through Big Data Analysis', funded by the Korea Coast Guard.

REFERENCES

[1] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[2] Hwi-Yeon Cho and Yong-Hyuk Kim. 2019. Stabilized training of generative adversarial networks by a genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 51–52.

[3] Akhilesh Gangwar and Vadlamani Ravi. 2019. WiP: Generative Adversarial Network for Oversampling Data in Credit Card Fraud Detection. In *Information Systems Security - Proceedings of the 15th International Conference (Lecture Notes in Computer Science)*, Vol. 11952. Springer, 123–134.

[4] Jae-Hyun Seo and Yong-Hyuk Kim. 2018. Machine-Learning Approach to Optimize SMOTE Ratio in Class Imbalance Dataset for Intrusion Detection. *Computational Intelligence and Neuroscience* 2018 (2018), 9704672:1–9704672:11.

[1] https://pytorch.org/

[2] https://imbalanced-learn.readthedocs.io/

[3] https://github.com/hwyncho/GECCO-2020-PyTorch.git/