Counterfactual data augmentation for small datasets

Daniel Bazar (314708181) and Peleg Shefi (316523638)

May 2024

Abstract

In today's world data is an expensive commodity and attaining it is a difficult task. Data augmentation offers a potential solution to reduce the dependency on large datasets. This work explores the use of counterfactuals, originally employed in eXplainability, for data augmentation. By utilizing counterfactuals to expand small datasets, our method improves model performance on classification and regression tasks. The method was tested on four distinct datasets and showed a meaningful improvement on all of them. This work makes the use of counterfactual augmentation more robust and not only limited to binary classification tasks.

1 Introduction

Acquiring large datasets is essential for model performance, but obtaining them can be more challenging than training the model. This can be due to budget limitations, rare data, or privacy issues. Even when data is readily available, it may be unbalanced or with spurious correlations. These challenges arise even before the data science pipeline begins.

There are two main approaches to solve such data problems, at the algorithmic level or in the data level [2, 3, 7, 11], this work is regarding the latter. Common methods to tackle the unbalanced classes problem, are Random over-sampling (ROS), Random under-sampling (RUS), and Synthetic Minority Over-Sampling Technique (SMOTE). A newer method which tries to solve the small dataset problem is the use of Generative adversarial network (GAN).

Counterfactual (CF) examples, originated in model eXplainability, by explaining the importance each feature ("cause") had on the outcome of the model ("effect"). In simple terms which features to change and by how much, will alternate the outcome of the model [8, 10]. The use of CF for data creation [5, 11] has been researched before but for tabular data, it is an under searched field [11].

In this work, by using CF to create more data, the models' performance has improved on binary, multi-class, and regression tasks. Moreover, a new combined approach of CF and classic over-sampling is introduced to overcome the difficulties of CF with imbalanced data. For example, on the regression task the model improved by 2.9% for the RMSE and by 6.66% for the R^2. The method was tested on 4 distinct datasets - Adult [1], German Credit [9], Cirrhosis [4] and Diabetes [6].

The rest of the paper is structured as follows: Sect. 2 will be an overview of related work. Then,

in Sect. 3 the method is introduced. Sect. 4 is the conducted experiments and their results. Lastly, there will be a summary and future work.

2 Related work

Sampling methods are very common to solve data problems, more specifically, imbalanced data. ROS balances the class distribution by randomly adding data points of the minority classes to the training data. With RUS, a selected number of data points of the majority class are randomly removed (this method was not used in this study because it is not suitable for small datasets). These methods have a few drawbacks. Since ROS copies minority-class instances, no new information is added to the dataset, which can lead to over-fitting. On the other hand, by randomly removing examples, RUS can discard important data [11]. In SMOTE, new data records in the minority class are created by interpolating between several instances of the same class. By doing so instead of copying instances, it generates new ones, which can prevent the over-fitting problem. As SMOTE is the best performing out of the three we will compare our work to it as a benchmark.

Temarz and Keane [11] are one of the first to use CF for data augmentation; by pairing data points from the majority class to their closest opposite data point. Then, points near the decision line without a pair will be paired to a synthetically created CF point. Another work is by Wang et al. [12], whom used CF to over-sample. Both works addressed only the imbalance data issues on binary classification tasks. In this work, we will try and expand the use of CF augmentation in two ways. Firstly, to solve not only binary classification but, also multi-class and regression tasks. Secondly, to solve a slightly more general problem, of small datasets.

3 Counterfactual synthetic data

In this section, we will describe more formally what counterfactuals are, then how they are used and finally why mathematically it works. We used the DiCE library, introduced by Mothilal et. al [10] for our CF generation.

3.1 Definitions

Let there be a model m and a data point x the output is a set of k new synthetic data points $(c_1, ..., c_k)$ all with a different result than x. For the use of eXplainability the CF needs to be diverse and feasible.

Diversity is calculated with *determinantal point process*, and its in Eq. 1

$$Diversity = det(\frac{1}{1 + dist(c_i, c_j)}) \tag{1}$$

Proximity of a CF set is calculated by summation of distances between each CF point and the original x.

$$Proximity = -\frac{1}{k}\sum_{i=1}^{k} dist(c_i, x) \tag{2}$$

Based on the two terms a loss function is defined:

$$C(x) = \underset{c_1,...,c_k}{\arg\min} \frac{1}{k}\sum_{i=1}^{k} y \times loss(m(c_i), y) + \frac{\lambda_1}{k}\sum_{i=1}^{k} dist(c_i, x) - \lambda_2 \times Diversity(c_1, ...c_k) \tag{3}$$

Equation 3, is a combined loss of 3 terms - a term to minimize the distance between the CF and models prediction, term for proximity and finally a term for diversity. In the equation, c_i is a CF data point, $m(\cdot)$ is some kind of a predictive model, and λ_1, λ_2 are weight hyper-parameters representing the trade-off between proximity and diversity.

Then, the function is optimized by a method of our choosing such as SGD or genetic algorithm. Another important note, model m is a black-box to the method and therefore can be changed as well.

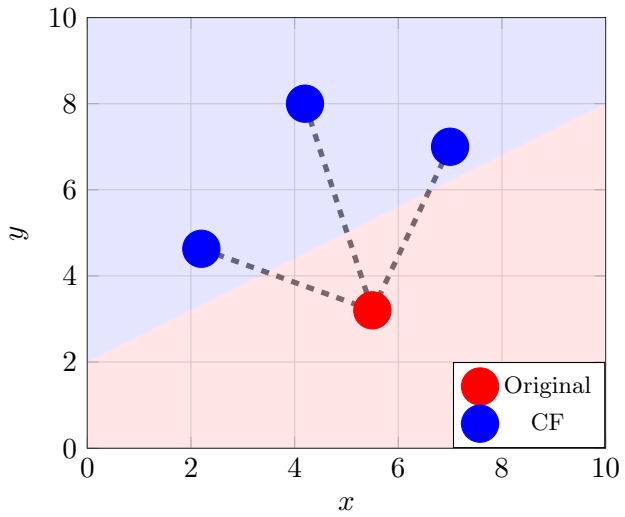


Figure 1: A visual demonstartion of a *red* data point with 3 potential *blue* CF points

3.2 Data point generation

We used the DiCE library [10] to generate CF points. For each dataset, we experimented with two cases. The first is balancing data, meaning ensuring classes are distributed uniformly. The second, is to randomly sample a fraction (hyper-parameter) of the original data. Finally, the new synthetic points are added to the original dataset.

We used a different model for the CFs generation then the experimented models, to eliminate any bias towards the CF examples, which are in fact generated by the other model. At the same time, we also wanted to make sure that CFs are transferable between different models.

DiCE implementation supports not only binary classification but, also multi-class classification and regression. This is done by specifying the wanted class and the desired range, respectively. In our case, to determine the wanted class for each sample in a multi-class case, we sampled from all classes except the query one, relative to their weights. In the regression case, we determined the desired range to be at least one *STD* (standard deviation) away towards the direction of the mean.

4 Experiments and results

The experiments were conducted on four datasets. Two binary classification, a multi-class classification, and a regression tasks. For each one, different models and augmentation methods were extensively tested.

4.1 Datasets

Adult dataset [1] is a very common binary classification dataset, aimed at predicting whether an individual's income is above or below 50K. It has 14 attributes, 6 continuous, 8 nominal, and an overall 45,222 instances. Due to its size, the Adult dataset allows us to sample a smaller portion, simulating a small dataset. This approach enables us to compare the results with those obtained using the original large dataset. For our analysis, we considered 5% of the original data for augmentation and compared the results

German credit dataset [9] classifies people described by a set of attributes as good or bad credit risks. It has 8 attributes, 4 numeric, 4 categorical, and overall 1,000 instances.

Diabetes dataset [6] has 10 both numeric and categorical attributes, and overall 442 instances.The target variable is a quantitative measure of disease progression one year after baseline, making this a regression task.

Cirrhosis dataset [4] is a medical dataset. It has 17 features and 418 instances, and it is a multi-class classification task.

Synthetic dataset This dataset was created by us. It is a multi-class classification task with 3 classes with a $50\%, 20\%, 30\%$ ratio. It has 12 numeric attributes, with 600 instances. It was used as a proof of concept to check that our idea is indeed feasible. The reason this dataset was needed was due to running time issues, explained in Sect. 4.4.

4.2 Experiment workflow

Each dataset was tested with a couple of predictive models: XGBoost, Random Forest, and Logistic Regression (Linear in regression task). For the augmentation technique, SMOTE [3] is used as the main baseline, but basic random over-sampling is tested as well. Every technique is examined in two approaches, balancing and random sampling generation. In the balancing variation, we ensure uniform distribution of the classes, and in random sampling we don't enforce it. CFs are generated with three different model-agnostic methods readily available in the DiCE library - *random*, *genetic algorithm*, and *KD-Trees* (for counterfactuals within the training data). The results were compared by multiple metrics including F1, Accuracy, Precision, Recall, and ROC AUC (weighted in multi-class task). For the regression task, RMSE and R^2 are the used metrics. Lastly, the performance of all models and augmentation methods are tested with respect to each metric.

4.3 Results

Results for each dataset can be seen in the tables below. The best-performing augmentation method, for each metric-model combination, is in bold. The CF method has improved at least one metric in each dataset. The displayed metric for each DS was chosen based on the performance of the CF to best portray the benfits of the method.

	Whole	Sample	Random	SMOTE	CF_{random}	$CF_{genetic}$
Logistic regression	0.5574	0.5102	**0.7143**	0.6327	0.6122	0.6735
Random forest	0.4508	0.2653	0.8367	0.7347	**0.8571**	0.7959
XGBoost	0.5893	0.4898	0.6122	0.5918	0.6122	**0.6939**

Table 1: Adult DS - Recall

For the adult dataset, the usage of CF improved Recall and F1 compared to all other methods for all three models, this is seen in Fig. 2a and in table 1. Notably, using the Random Forest model with CF, substantially improves the *Recall* compared even to the original dataset. In the case of the German credit dataset with Random Forest, the *Precision* was improved by 1.8% compared to second

	Whole	Random	SMOTE	CF_{random}	$CF_{genetic}$
Logistic regression	0.7840	0.8407	**0.8462**	0.8000	0.8440
Random forest	0.7316	0.8942	0.8727	0.8560	**0.9109**
XGBoost	0.7862	0.8438	0.8271	0.8271	**0.8571**

Table 2: German credit DS - Precision

	Whole	Random	SMOTE	CF_{random}	$CF_{genetic}$
Logistic regression	0.8720	0.8682	0.8566	0.8653	**0.8915**
Random forest	0.8602	**0.8773**	0.8583	0.8708	0.8648
XGBoost	0.8679	0.8618	0.8468	0.8656	**0.8796**

Table 3: Cirrhosis DS - ROC AUC OVR

	Whole	Random	SMOTE	CF_{random}	$CF_{genetic}$
Logistic regression	-53.8534	-54.3299	-54.6513	**-53.4490**	-54.1484
Ridge	-53.8429	-54.3056	-54.6446	**-53.4434**	-54.1334
Lasso	-53.7742	-54.2660	-54.5531	**-52.7714**	-53.7406
Random forest	-53.7627	**-52.9805**	-54.5304	-53.2019	-54.3978
XGBoost	-53.9246	-63.8832	-54.3835	-52.9805	**-52.4092**

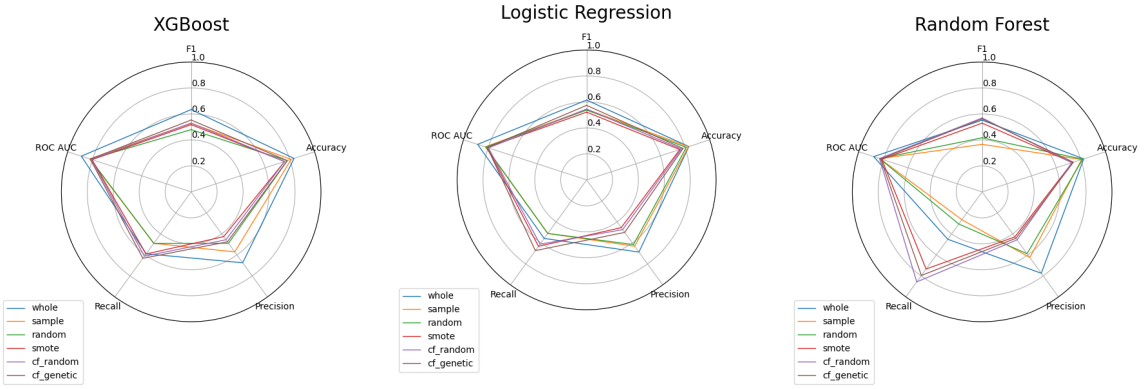
Table 4: Diabetes DS - RMSE

best performing method and by 24.5% with respect to the original dataset, this is seen in table 2. For the Cirrhosis DS, the *ROC AUC* improved by 1.6% compared to other augmentations and by 2.2% compared to the original. Lastly, in the regression experiment both *RMSE* and R^2 were enhanced. The overall performances and the best method for each dataset are presented in Tables 5 and 6.

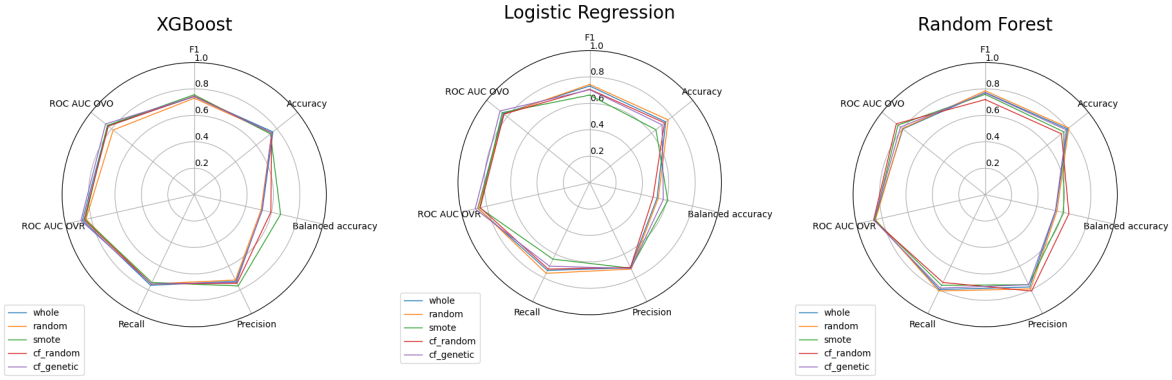
4.4 Downsides

Two major downsides occurred during our testing. First, the original use of CF required only a small number of points in-order to explain a model, this is not true in our case. This difference can lead to long augmentation times because the code was not suited for our use. Improving the efficiency of CF creation, specifically for this use case, can streamline the process.

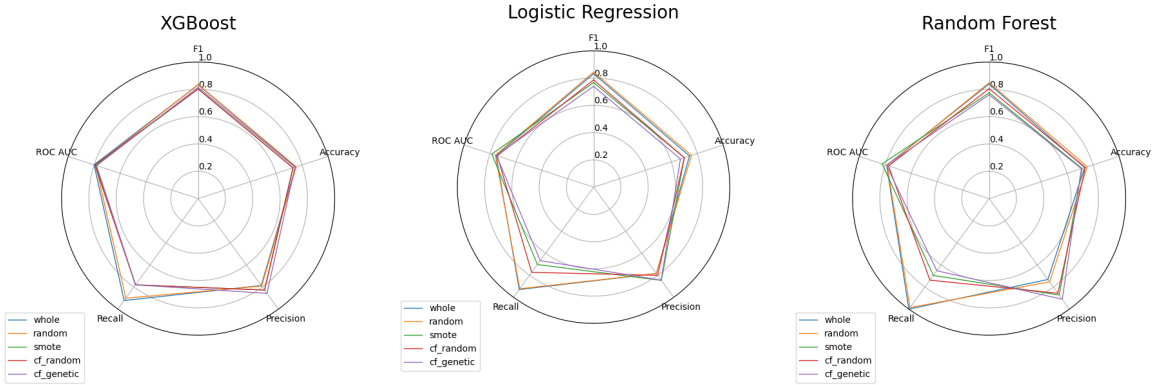
Secondly, if a dataset is extremely unbalanced, then the CF has a hard time augmenting rare categories. This also leads to long augmentation times. At first, we did not understand the long-running duration, so a synthetic dataset was created to test performance. Once the problem was isolated, we tried to use basic over-sampling to add instances to the very rare classes and then run CF augmentation. This method improved the running time without reducing performance. It is important to note that the performance of the CF augmentation on the synthetic dataset was better than almost every other configuration. It may be thanks to the fact that the synthetic data is probably much



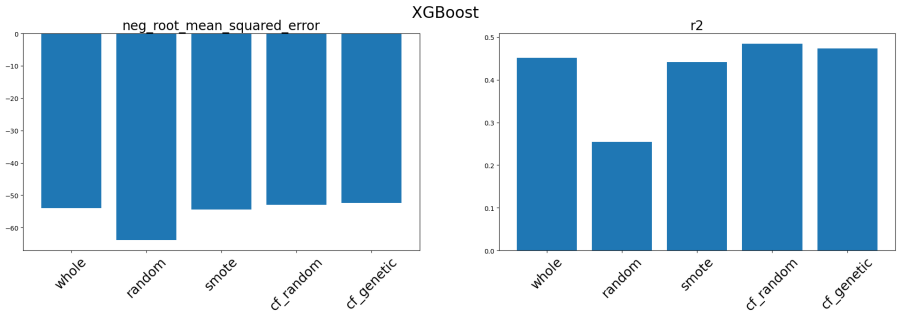
(a) Adult dataset



(b) Cirrhosis dataset



(c) German credit dataset



(d) Diabetes dataset

Figure 2: Each sub-figure plots the performance of all augmentation methods for a specific model. Each row is for a different dataset stated in the label and each column is a different model as stated in the title of each plot.

cleaner than realistic data, even though we added noise to it. Nevertheless, it is an indication that such improvements are achievable.

5 Summary

This work takes another step to make the use of tabular data more practical and affordable by enabling the use of smaller datasets. We showed, that by creation of synthetic data instances using CF, the overall performance of the model improved on all tasks in at least one metric as seen in Tables 5, 6. Despite the improved performance, two issues emerged when using CF this way. Both are regarding the running duration, mitigating them is left for future work. This was an interesting and different experience than we had in other courses, and it was a nice eye-opener into the world of research in general and in data science specifically.

	F1	**Accuracy**	**Precision**	**Recall**	**ROC AUC**
Adult	Whole	Whole	Whole	CF_{random}	Whole
German	Random	CF_{random}	$CF_{genetic}$	Whole	SMOTE
Cirrhosis	Random	Random	CF_{random}	Random	$CF_{genetic}$

Table 5: Overall Classification performance. Showing the best augmentation method for each combination of dataset and metric we found in the experiments.

	negative RMSE	R^2
Diabetes	$CF_{genetic}$	CF_{random}

Table 6: Overall Regression performance. Showing the best augmentation method for each metric we found in the experiments.

References

[1] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] A. Biswas, M. A. A. Nasim, A. Imran, A. T. Sejuty, F. Fairooz, S. Puppala, and S. Talukder. Generative Adversarial Networks for Data Augmentation, June 2023. arXiv:2306.02019 [cs].

[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*, 16:321–357, 2002.

[4] E. Dickson and A. Langworthy. Cirrhosis Patient Survival Prediction. UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5R02G.

[5] T. Dixit, B. Paranjape, H. Hajishirzi, and L. Zettlemoyer. Core: A retrieve-then-edit framework for counterfactual data generation. *arXiv preprint arXiv:2210.04873*, 2022.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. 2004.

[7] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[8] M. Höfler. Causal inference based on counterfactuals. *BMC medical research methodology*, 5:1–12, 2005.

[9] H. Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[10] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[11] M. Temraz and M. T. Keane. Solving the class imbalance problem using a counterfactual method for data augmentation. *Machine Learning with Applications*, 9:100375, Sept. 2022.

[12] S. Wang, H. Luo, S. Huang, Q. Li, L. Liu, G. Su, and M. Liu. Counterfactual-based minority oversampling for imbalanced classification. *Engineering Applications of Artificial Intelligence*, 122:106024, 2023.