Data Augmentation using Counterfactuals: Proximity vs Diversity

MGM Mehedi Hasan and Douglas A. Talbert
Department of Computer Science
Tennessee Tech University
Cookeville, USA
mmehediha42@tntech.edu, dtalbert@tntech.edu

Abstract

Counterfactual explanations are gaining in popularity as a way of explaining machine learning models. Counterfactual examples are generally created to help interpret the decision of a model. In that case, if a model makes a certain decision for an instance, the counterfactual examples of that instance reverse the decision of the model. Counterfactual examples can be created by craftily changing particular feature values of the instance. Though counterfactual examples are generated to explain the decision of machine learning models, we have already explored that counterfactual examples can be used for effective data augmentation. In this work, we want to explore what kind of counterfactual examples work best for data augmentation. In particular, we want to generate counterfactual examples from two perspectives: *proximity* and *diversity*. We want to observe which perspective works best in this regard. We demonstrate the efficacy of these approaches on the widely used "Adult-Income" dataset. We consider several scenarios where we do not have enough data and use each of these approaches to augment the dataset. We compare these two approaches and discuss the implications of the results.

Introduction

There has been a desire for explanations of how complex computer systems make decisions for quite some time. The need for explanations can be dated back to some of the earliest work on expert systems (Buchanan and Shortliffe 1984). Explanations are critical for machine learning (ML), especially as machine learning-based systems are being used to inform decisions in societally critical domains such as finance, healthcare, education, and criminal justice.

Wachter et al. (Wachter, Mittelstadt, and Russell 2017) argue that there are three important aims for explanations: (1) to inform and help the person understand why a particular decision was reached, (2) to provide grounds to contest the decision in the case of an undesirable outcome, and (3) to understand what would need to change in order to get a desirable result in the future, based on the current decision making model. A counterfactual explanation (CE) of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output, and it can be a good candidate to fulfill the three aims proposed by Wachter et. al. In interpretable machine learning, counterfactual explanations can be used to explain predictions of individual instances. In this paper, we use the terms counterfactuals and counterfactual examples interchangeably.

Counterfactual examples are increasingly seen as enhancing the autonomy of people subject to automated decisions by allowing people to navigate the rules that govern their lives (Barocas, Selbst, and Raghavan 2020). This helps people recognize whether to contest the decision making process and facilitates direct oversight and regulation of algorithms (Wachter, Mittelstadt, and Russell 2017; Selbst and Barocas 2018). Specifically, counterfactual examples provide this information by showing feature-perturbed versions of the same person who would have otherwise received the loan (Mothilal, Sharma, and Tan 2020).

Motivation and Contribution: With small datasets, overfitting becomes much harder to avoid. In this case, if our training dataset itself is small, overfitting is more likely to occur. We might overfit both the training data as well as the validation set. Outliers become much more dangerous with small dataset since even just a few outliers will form large proportion and significantly alter the model. The smaller our sample size, the more likely outliers are to skew our findings. To alleviate these problems, it's very important to have a large dataset. However, in reality, we cannot always have as large dataset as we would like because of some practical constraints. In this case, it will be helpful if we can find a way to augment our small dataset.

We have already seen that CEs are good candidate to augment a small datasets (Hasan and Talbert 2021). However, in our previous work, CEs were generated randomly i.e. without careful consideration to *diversity* or *proximity*. Now, we want the process of data augmentation by CEs to be as efficient as possible i.e. we want to generate CEs that are specially catered for data augmentation purpose. In other words, we are specially interested in generating "targeted" CEs. Here the targeted CEs will be generated based on certain criteria that contribute most in achieving certain targets, and here our target is the data augmentation process. It may be noted that we are not particularly interested how significantly these CEs contribute towards explanation as we are generating CEs targeting data augmentation only. In this work, we explore how proximity and diversity affect the success of data augmentation using CEs. We carefully choose different values for proximity and diversity and observe the resulting performance. We use a well-known dataset to conduct the experiments in different steps and analyze the results to measure the extent to which each of the properties (i.e. proximity and diversity) contributes to successful augmentation of the dataset. In this work, we do not consider data augmentation for correcting for class imbalance and leave that as future work. We are particularly interested in dealing with a situation where we have a small dataset.

The rest of this paper is organized as follows: Next, we present the necessary background material to understand the concept. After that we discuss related work done in this regard. Then, we run relevant experiments to test our proposed approach, which will be followed by the discussion on the experimental results. Finally, we draw conclusion to our work with some proposed future research directions.

Background

Wachter et. al (Wachter, Mittelstadt, and Russell 2017) proposed the most commonly accepted approach of generating CEs by minimizing the following loss function, which was later refined by Molnar (Molnar 2019):

$$L(x_i, x_i', y', \lambda) = \lambda \cdot (\hat{f}(x_i') - y_i')^2 + d(x_i, x_i') \qquad (1)$$

Here, the term $\lambda \cdot (\hat{f}(x_i') - y_i')^2$ represents the quadratic distance between the model prediction ($\hat{f}(x_i')$) for the counterfactual x_i' for an instance of interest x_i and the desired outcome y_i', which the user must define in advance. The second term $d(x_i, x_i')$ is the distance d between the instance of interest x_i to be explained and the desired counterfactual x_i'.

The parameter λ plays an important role here, which balances the distance in prediction i.e. $\lambda \cdot (\hat{f}(x_i') - y_i')^2$ against the distance in feature values i.e. $\hat{f}(x_i')$. The loss is solved by choosing an appropriate value of λ, and the solution returns a counterfactual x_i'. The value of λ dictates the kind of compromise we want to make in our preference for counterfactuals. For example, if we choose a higher value of λ that means we prefer counterfactuals that are closer to the desired outcome y_i'. On the other hand, if we go for a lower value λ, we prefer counterfactuals x_i' that are very similar to the instance of interest, x_i, in the feature values. A very large value of λ indicates that, the instance with the prediction that comes closest to y_i' will be selected, no matter how far it is away from x_i.

The choice of λ depends on the user, as he/she must decide how to balance the requirement that the prediction for the counterfactual matches the desired outcome with the requirement that the counterfactual is similar to x_i. Wachter et. al suggest instead of selecting a value for λ, we can select a tolerance ϵ. The tolerance indicates how far away the prediction of the counterfactual instance is allowed to be from y_i'. We can write this constraint in the following way:

$$|\hat{f}(x_i') - y_i'| \leq \epsilon \qquad (2)$$

We can use any suitable optimization algorithm to minimize this loss function in Eq. (2). For example, if we have access to the gradients of the machine learning model, we can use gradient-based methods like RMSprop optimizer (Tieleman and Hinton 2012) or Adam. To the best of our knowledge, all the recent works on counterfactual examples generation use Eq. (1) with little or no modification (Mahajan, Tan, and Sharma 2019; Mothilal, Sharma, and Tan 2020; Russell 2019; Sokol and Flach 2019).

Proximity and Diversity: Finding CEs is usually formulated as an optimization problem. In a sense, it is similar to finding adversarial examples (Goodfellow, Shlens, and Szegedy 2014), where we perturb the data to change the outcome. In this case, however, we need perturbations that not only change the output of a machine learning model, but also are diverse and feasible to change. CEs are usually generated from two perspectives: proximity or diversity. The goal of a typical CE generation model is to generate an actionable counterfactual set. In this case, we need individual CEs that are feasible with respect to the original input. However, we also need diversity among the generated counterfactuals to provide different ways of changing the outcome class. In this case, consideration may be given to the proximity of the explanations to the original input as well as the diversity of those explanations, i.e. the range of suggested changes to the explanations in question (Mothilal, Sharma, and Tan 2020).

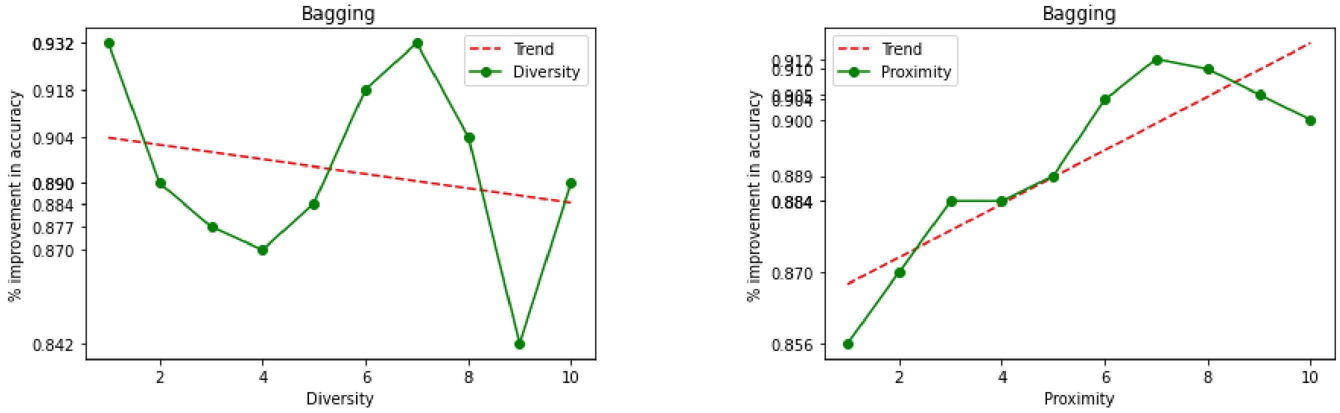
We use the DiCE model that adapts diversity metrics to generate diverse CEs that can offer users multiple options (Mothilal, Sharma, and Tan 2020). The authors incorporate feasibility using the proximity constraint from Wachter et al. (Wachter, Mittelstadt, and Russell 2017).

In DiCE, diversity is captured by building on determinantal point processes (DPP), which has been adopted for solving subset selection problems with diversity constraints (Kulesza and Taskar 2012).

Related Work

Mothilal et. al extended the work of Wachter et. al (Wachter, Mittelstadt, and Russell 2017) and provided a method to construct a set of counterfactuals with diversity (Mothilal, Sharma, and Tan 2020). Ribeiro et al. (Ribeiro, Singh, and Guestrin 2016) proposed a feature-based approach, LIME, that fits a sparse linear model to approximate non-linear models locally.

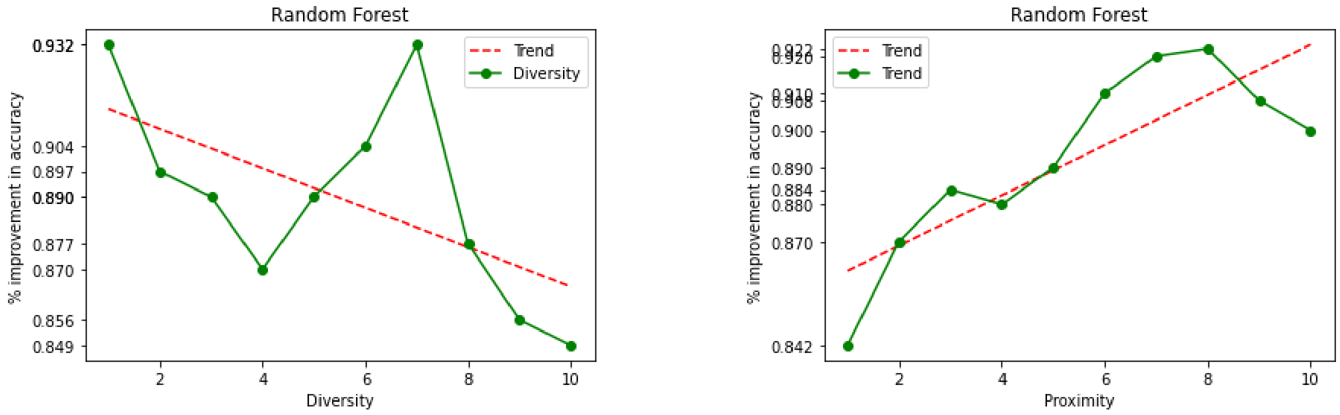
There has been several works on data augmentation especially on image data. One such work uses Random Erasing method where it randomly selects a rectangle region in an image and erases its pixels with random values (Zhong et al. 2020). However, this work focuses on image data. Liu et al. lay the groundwork for formal causal language in Data Augmentation and proposed a data augmentation method for neural machine translation (Liu, Kusner, and Blunsom 2021). This method works by interpreting language models and phrasal alignment causally. Mikołajczyk et al. presented a data augmentation method based on image style transfer, which allows to generate the new images of high perceptual quality that combine the content of a base image with the appearance of another ones (Mikołajczyk and Grochowski 2018). There has been some peripheral works of counterfactual data augmentation in natural language processing for Mitigating Gender Stereotypes in Languages (Zmigrod et al.



(a) Change in diversity.

(b) Change in proximity.

Fig. 1: (a) Performance of Bagging (Bootstrap Aggregation) applied to decision trees when CEs are generated varying diversity. (b) Performance of Bagging applied to decision trees when CEs are generated varying proximity.



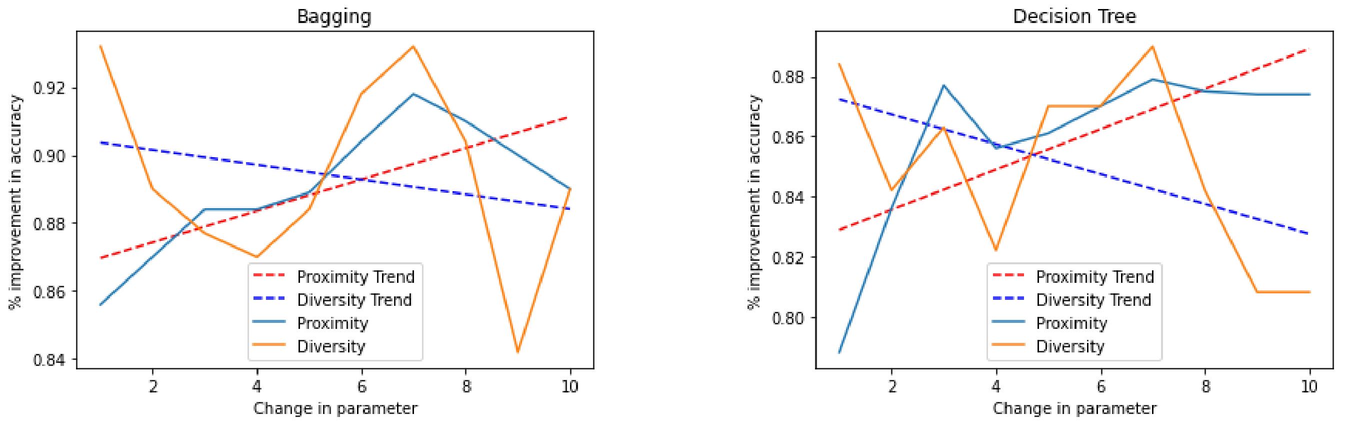
(a) Change in diversity.

(b) Change in proximity.

Fig. 2: (a) Performance of Random Forest when CEs are generated varying diversity. (b) Performance of Random forest when CEs are generated varying proximity.

2019) and reducing bias (Kaushik, Hovy, and Lipton 2019). The authors used counterfactual examples as a tool to mitigate those issues but not as a tool to augment the dataset. However, there has not been that much work done on tabular data augmentation except for Synthetic Minority Oversampling (SMOTE) (Chawla et al. 2002), which works by creating synthetic observations based upon the existing minority observations (Chawla et al. 2002). In this case, SMOTE selects examples that are close in the feature space, draws a line between the examples in the feature space, and selects a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space (Chawla et al. 2002; He and Ma 2013). The synthetic example looks too much like the original data and does not bring any new information to the dataset. Hasan et. al explored on the effectiveness of CEs to augment a small dataset (Hasan and Talbert 2021). There has been some efforts on using Generative Adversarial Networks(GANs) to augment a small dataset (Frid-Adar et al. 2018). However, there has been a work that showed empirically that CEs are more viable compared to GANs for data augmentation purpose (Hasan and Talbert 2021). CEs apparently do contribute useful information and put points in space (Hasan and Talbert 2021).

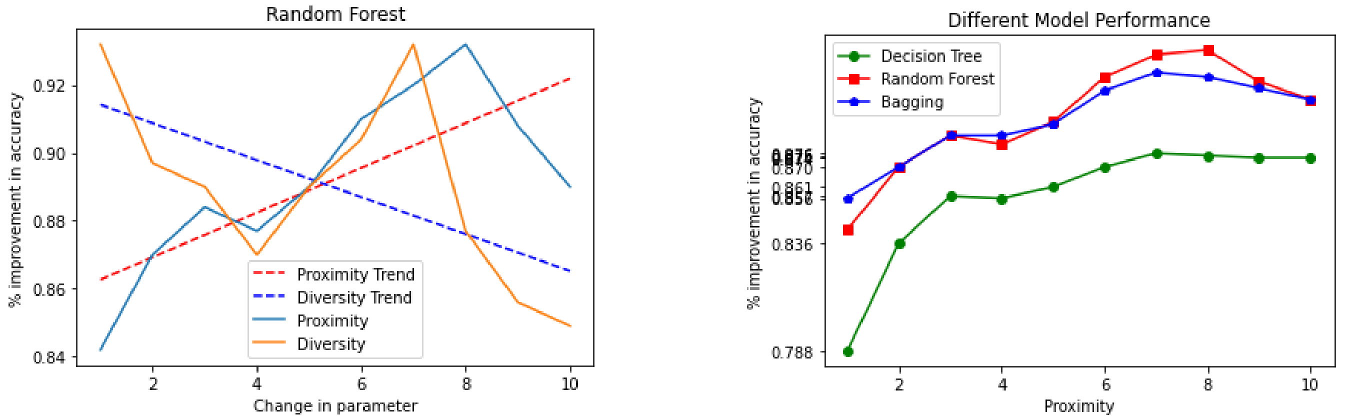
Even though it has been showed that CEs can be great alternative for data augmentation purpose (Frid-Adar et al. 2018). However, to the best of our knowledge, there is no work available that considers proximity and diversity with special attention in this purpose i.e., which of them impacts more in the data augmentation purpose. In this work, we explore which one of these two plays the dominant role in supporting the data augmentation process.



(a) Change in diversity and proximity.

(b) Change in diversity and proximity.

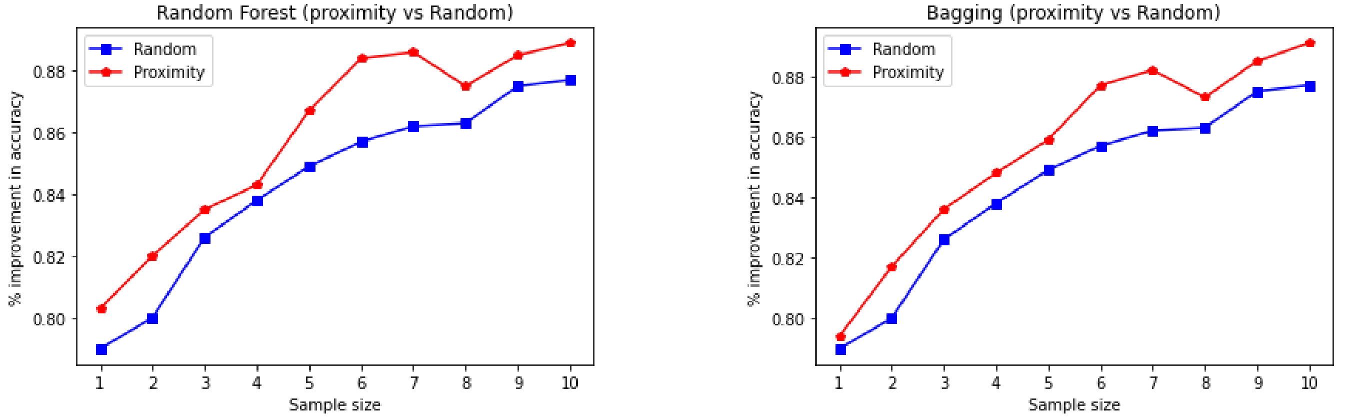
Fig. 3: (a) Performance of Bagging when CEs are generated varying both diversity and proximity. (b) Performance of Decision Tree when CEs are generated varying both diversity and proximity.



(a) Change in diversity and proximity.

(b) Performance with varying proximity.

Fig. 4: (a) Performance of Random Forest when CEs are generated varying both diversity and proximity. (b) Performance of different models when CEs are generated varying proximity.



(a) Proximity-based CEs.

(b) Proximity-based CEs.

Fig. 5: (a) Performance of Random Forest when CEs are generated varying proximity. (b) Performance of Bagging CEs are generated varying proximity.

Implementation

We apply a technique introduced by Mothilal et. al (Mothilal, Sharma, and Tan 2020) to generate the counterfactual examples (CEs). We generated counterfactual examples using a shallow artificial neural network (ANN) and then used those counterfactual examples in other models. We used different models to experiment with the generated counterfactual examples to avoid potential bias that might arise when the same model that generated the CEs is again used to test those CEs. At the same time, we also wanted to make sure that CEs generated by one model are transferable to another model.

Dataset

In this experiment, we consider the *Adult-Income* data set, which contains demographic, educational, and other information based on the 1994 Census database and is available on the UCI machine learning repository (Kohavi and Becker 1996). The data set is credited to Ronny Kohavi and Barry Becker (Kohavi 1996). It involves using personal details such as education level, hours of work per week, etc. to predict whether an individual will earn more or less than $50,000 per year. The Adult-Income data set is a widely used standard machine learning data set and has become a de facto data set for counterfactual example experiments (Karimi et al. 2020; Mothilal, Sharma, and Tan 2020; Mahajan, Tan, and Sharma 2019). We obtained 8 features, namely, hours per week, education level, occupation, work class, race, age, marital status, and sex by applying the preprocessing based on a prior analysis (Zhu 2016). In this case, the ML model's task is to classify whether an individual's income is over $50,000.

Experiments

We trained an artificial neural network (ANN) model using the Adult-Income dataset. In this case, we generate CEs for several scenarios. Here are couple of example scenarios: (i) we generate CEs varying the proximity but keeping diversity fixed, (ii) we generate CES varying the diversity but keeping the proximity fixed, and (iii) we generate CEs varying both the proximity and diversity.

There is a work that showed CEs can be an effective alternative for augmenting a small dataset (Hasan and Talbert 2021). In this work, we are mainly interested to find out the impact of diversity and proximity in the data augmentation process. We run our experiments in different scenarios as mentioned above.

To test how proximity and diversity impact augmenting a dataset using CEs, we reduced the original dataset by 80%. This truncated dataset became our new baseline dataset, which we wanted to augment. The rationale behind this idea is if, for some experiments, we do not have enough data to adequately train a model then CEs can be used to effectively supplement the given data.

In such a case, we would use that small dataset to train a model and use this trained model to generate CEs. These CEs along with the original dataset can then be used to train another model. In the latter case, choosing a different model than the one used to generate the CEs is preferable. In this way, we can avoid the bias that might potentially be created using the same model for both the purposes. This also paves the way for the generated CEs to become model-agnostic i.e., we can train any classifier with the generated CEs.

Fig. 1(a) and Fig. 1(b) show the performance of a bagging applied to decision trees when CEs are generated based on diversity and proximity, respectively. In this case the trend shows that proximity-based CEs perform better and more consistently than diversity-based CEs. We observe similar situation in the case of Random Forest (Fig. 2) as well. Fig. 3 (a), Fig. 3 (b), and Fig. 4 (a) show direct contrast in performance of proximity-based and diversity-base CEs for different models. Fig. 4 (b) shows how different values of proximity affect data augmentation process. Fig. 5 (a) and Fig. 5 (b) show how proximity-based CEs perform compared to randomly generated CEs (Hasan and Talbert 2021).

Discussion

Counterfactual examples generation method is model-agnostic, since it only works with the model inputs and output (Molnar 2019). As we have demonstrated in the experiments, we used one model (e.g. ANN) to generate counterfactual examples and different models (e.g. decision tree, Random Forests, and Bagging) to test the utility of the generated examples.

When the target is counterfactual explanations, there are certain criteria that need to be fulfilled. One of the foremost requirements is proximity i.e. a counterfactual instance should produce the predefined prediction as closely as possible by defining a relevant change in the prediction of an instance (i.e. the alternative reality) (Molnar 2019). At the same time, diversity is an important factor in counterfactual explanation, and it's often suggested to have diverse CEs. Generating a set of diverse explanations increases the likelihood of finding a useful explanation (Mothilal, Sharma, and Tan 2020)(Russell 2019). In a set of diverse CEs, each one proposes to change a different set of attributes.

However, we observe a different scenario when our target is data augmentation. Our experimental results show that proximity-focused CEs perform better than diversity-focused CEs. The reason behind this lies in the way proximity-focused CEs are generated. In the case of proximity, CEs should be as similar as possible to the instance regarding feature values. This criterion necessitates an appropriate distance measure between two instances. The counterfactual example should not only be close to the original instance, but should also change as few features as possible (Molnar 2019). To fulfill this criterion, an appropriate distance measure like the Manhattan distance is required. In other words, we get CEs that are very much similar to the original instances and do not change the model parameters significantly. However, in the case of diversity, the focus is given on how diverse the CEs are from each other and from the original instance. To fulfill the diversity criteria, often times features are changed randomly. We believe this causes the model to fail to generalize well.

Conclusion and Future Work

In recent literary work, Counterfactual explanation has garnered a lot of attention as an effective way to provide explanation for machine learning model's decision. We have observed that CEs can be a great alternative to augment a small dataset. In this work, however, we were interested to see how proximity-based CEs and diversity-based CEs fair in this purpose. We ran extensive experiments and found out that overall, proximity-based CEs perform better for data augmentation purpose. We conclude that if our target is to augment a dataset using CEs, we should emphasize more on proximity while generating them. In the future, we want to experiment with other datasets from different application areas. Additionally, we want to consider other models and other counterfactual example generation techniques. We want to explore how CEs can also be used to address class imbalance.

References

Barocas, S.; Selbst, A. D.; and Raghavan, M. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 80–89.

Buchanan, B. G., and Shortliffe, E. H. 1984. Rule-based expert systems: the mycin experiments of the stanford heuristic programming project.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16:321–357.

Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; and Greenspan, H. 2018. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, 289–293. IEEE.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hasan, M. G. M. M., and Talbert, D. A. 2021. Counterfactual examples for data augmentation: A case study. In *The International FLAIRS Conference Proceedings*, volume 34.

He, H., and Ma, Y. 2013. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

Karimi, A.-H.; Barthe, G.; Balle, B.; and Valera, I. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, 895–905.

Kaushik, D.; Hovy, E.; and Lipton, Z. C. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Kohavi, R., and Becker, B. 1996. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/adult.

Kohavi, R. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, 202–207.

Kulesza, A., and Taskar, B. 2012. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*.

Liu, Q.; Kusner, M.; and Blunsom, P. 2021. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 187–197.

Mahajan, D.; Tan, C.; and Sharma, A. 2019. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*.

Mikołajczyk, A., and Grochowski, M. 2018. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, 117–122. IEEE.

Molnar, C. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 607–617.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Russell, C. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 20–28.

Selbst, A. D., and Barocas, S. 2018. The intuitive appeal of explainable machines. *Fordham L. Rev.* 87:1085.

Sokol, K., and Flach, P. A. 2019. Counterfactual explanations of machine learning predictions: opportunities and challenges for ai safety. In *SafeAI@ AAAI*.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2):26–31.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.* 31:841.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13001–13008.

Zhu, H. 2016. Predicting earning potential using the adult dataset.

Zmigrod, R.; Mielke, S. J.; Wallach, H.; and Cotterell, R. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*.