GENERATING COUNTERFACTUAL EXPLANATIONS USING CARDINALITY CONSTRAINTS

Rubén Ruiz-Torrubiano
Institute of Digitalization and Informatics
IMC Krems University of Applied Sciences
Krems, 3500, Austria
ruben.ruiz@fh-krems.ac.at

ABSTRACT

Providing explanations about how machine learning algorithms work and/or make particular predictions is one of the main tools that can be used to improve their trusworthiness, fairness and robustness. Among the most intuitive type of explanations are counterfactuals, which are examples that differ from a given point only in the prediction target and some set of features, presenting which features need to be changed in the original example to flip the prediction for that example. However, such counterfactuals can have many different features than the original example, making their interpretation difficult. In this paper, we propose to explicitly add a cardinality constraint to counterfactual generation limiting how many features can be different from the original example, thus providing more interpretable and easily understantable counterfactuals.

1 INTRODUCTION

Explainable Artificial Intelligence (XAI) can be defined as the study and implementation of methods than provide visibility into how an AI system makes decisions, predictions and executes its actions (Rai, 2020). In general, two principal dimensions can be defined to classify XAI methods: whether the method requires knowledge of the model being explained, and whether the explanations refer to the model itself or its predictions (Du et al., 2019).
In the first case, knowing the internal workings of a particular algorithm results in *model-specific* approaches, whereas those that handle machine learning models basically as black-boxes and can therefore be applied to a more general class of algorithms are called *model-agnostic*. Examples of model-specific approaches are DNN-specific methods like those proposed in (Simonyan et al., 2014; Fong & Vedaldi, 2017; Du et al., 2018). By contrast, model-agnostic explanations use approaches like perturbations to determine feature contributions based on how sensitive the prediction target reacts when changing those features (Robnik-Šikonja & Bohanec, 2018; Ribeiro et al., 2016; Liu et al., 2019). Conterfactual explanations (Wachter et al., 2017) can be considered to belong to this type of explanations. The main idea is the following: let's consider a machine learning model f_θ and an input data point $\mathbf{x}$. We want to find data points $\hat{\mathbf{x}}$ such that they are the closest points to $\mathbf{x}$ such that the prediction target is different $f_\theta(\mathbf{x}) \neq f_\theta(\hat{\mathbf{x}})$. These data points should help the user understand what features one would need to change in $\mathbf{x}$ to flip the prediction target. For instance, let $\mathbf{x}$ represent a loan application, and f_θ a model trained to predict if the loan will be paid off or will result in default. Having a prediction of default would likely result in the loan application being rejected. A counterfactual example $\hat{\mathbf{x}}$ would provide an explanation on which features would need to have been different in order to reach the opposite decision (e.g. by reducing the loan amount or having a higher income).
One of the main obstacles for using counterfactual explanations in practice is the amount of features that are different in the original example and the counterfactual: the higher, the more complicated and unintuitive counterfactual explanations can be. Even if a counterfactual is close to the original example in feature space (say, in terms of the Euclidean distance between $\mathbf{x}$ and $\hat{\mathbf{x}}$), slight changes in a high number of features can have a negative effect on its interpretability. Therefore, we propose in this work to add *cardinality constraints* to counterfactual generation methods in order to ensure that the explanations provided do not diverge from the original example by more than k features.

Table 1: Comparison of counterfactuals obtained without cardinality constraints and with a cardinality constraint of $k = 2$ and $k = 3$. Differences are shown in bold.

	Age	Sex	BP	Cholesterol	NaToK
Original	16	M	LOW	HIGH	12.006
Unconstrained	**17**	M	**NORMAL**	**NORMAL**	**11.29**
$k = 2$	**15**	M	LOW	HIGH	**22.82**
$k = 3$	**15**	M	**HIGH**	HIGH	**11.04**

Specifically, we provide an extension for the CERTIFAI framework (Sharma et al., 2020) with cardinality constraints to answer the question of whether such *sparse* counterfactuals can be generated effectively and efficiently. In general, there is a trade-off between sparse counterfactuals and those that minimize other metrics like diversity or proximity (Mothilal et al., 2020). In this study, we focus on sparsity as our main criterium and leave using sparsity in combination with other measures as future work.

2 METHODOLOGY

The CERTIFAI framework uses a custom genetic algorithm to find those counterfactuals $\hat{\mathbf{x}}$ that minimize the distance $d(\mathbf{x}, \hat{\mathbf{x}})$ to a given data point $\mathbf{x}$. In this paper, we restrict ourselves to tabular data and use as distance function the sum of the the L_1 distance (for continuous features) and a matching distance for categorial values as proposed in Sharma et al. (2020). We forked the publicly available repository from the authors and implemented an additional cardinality constraint by penalizing those individuals with a cardinality (number of modified features with respect to the input example) higher than the target value k. We compare the best counterfactuals generated by CERTIFAI without cardinality constraints (which are distance-based) with our counterfactuals and analyze their interpretability using random examples. To have a better overview, we also calculate the mean cardinality between the constrained and unconstrained counterfactuals for each training example in the dataset used (see next section). We provide more details on the implementation as well as links to our code in Appendix A.

3 RESULTS AND DISCUSSION

We performed experiments using the drug200 dataset (obtained from Kaggle) to generate one counterfactual for each training example with and without cardinality constraints. In this dataset, there are in total 5 features, both continuous and categorical. Using this dataset, CERTIFAI computed counterfactuals with an average cardinality of $\hat{k} = 3.1$. We set the cardinality constraint to $k = 2$ and $k = 3$ features to get counterfactuals that are as easy as possible to interpret and compare the generated counterfactuals with the unconstrained ones. Table 1 shows a comparison of counterfactuals obtained for a random example. As can be seen in the second row, the unconstrained counterfactuals generated by CERTIFAI can be different than the original sample in many features. Interpreting such a counterfactual might be difficult, as the only feature that stays with the same value in this case is the gender. By contrast, the low-cardinality counterfactuals generated with our approach are more easily interpretable: in the case $k = 2$ the counterfactual can be intepreted as 'the target would change if the age is 15 and the NaToK ratio increases to 22.92'. The last row shows a counterfactual constrained to have a maximum of $k = 3$ different features, which is also more easily interpretable than the unconstrained counterfactual. We provide additional experiments with another dataset in Appendix B to further support our results.

4 CONCLUSIONS

In this paper, we have presented a modification of the CERTIFAI framework to obtain low-cardinality counterfactuals as model-agnostic explanations. The presented results show that the cardinality-constrained counterfactuals are more easily interpretable. As future work, we plan to design more effective genetic operators and validate our approach with larger datasets.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Mengnan Du, Ninghao Liu, Qingquan Song, and Xia Hu. Towards explanation of dnn-based prediction with guided feature inversion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1358–1367, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220099. URL https://doi.org/10.1145/3219819.3220099.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019. ISSN 0001-0782. doi: 10.1145/3359786. URL https://doi.org/10.1145/3359786.

Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017. doi: 10.1109/ICCV.2017.371.

Shusen Liu, Zhimin Li, Tao Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. NLIZE: A Perturbation-Driven Visual Interrogation Tool for Analyzing and Interpreting Natural Language Inference Models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1): 651–660, January 2019. ISSN 1941-0506. doi: 10.1109/TVCG.2018.2865230. URL https://ieeexplore.ieee.org/document/8454904. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pp. 607–617, New York, NY, USA, January 2020. Association for Computing Machinery. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372850. URL https://doi.org/10.1145/3351095.3372850.

Arun Rai. Explainable AI: from black box to glass box. *Journal of the Academy of Marketing Science*, 48(1):137–141, January 2020. ISSN 1552-7824. doi: 10.1007/s11747-019-00710-5. URL https://doi.org/10.1007/s11747-019-00710-5.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "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*, KDD '16, pp. 1135–1144, New York, NY, USA, August 2016. Association for Computing Machinery. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL https://dl.acm.org/doi/10.1145/2939672.2939778.

Marko Robnik-Šikonja and Marko Bohanec. Perturbation-Based Explanations of Prediction Models. In Jianlong Zhou and Fang Chen (eds.), *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Human–Computer Interaction Series, pp. 159–175. Springer International Publishing, Cham, 2018. ISBN 978-3-319-90403-0. doi: 10.1007/978-3-319-90403-0_9. URL https://doi.org/10.1007/978-3-319-90403-0_9.

Shubham Sharma, Jette Henderson, and Joydeep Ghosh. CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pp. 166–172, New York, NY, USA, February 2020. Association for Computing Machinery. ISBN 978-1-4503-7110-0. doi: 10.1145/3375627.3375812. URL https://doi.org/10.1145/3375627.3375812.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017. URL http://arxiv.org/abs/1711.00399.

A Implementation Details

As outlined in Section 2, our implementation[1] is a fork of the publicly available Python code[2] provided by the authors of Sharma et al. (2020). We defined a new distance function `card_distance` that calculates the amount of different features between a candidate counterfactual and the original data point. This distance function provided a deviation Δ between the target maximum cardinality and the cardinality of the current counterfactual. A linear penalty term was then calculated using a coefficient c_{card} that was chosen in such a way that it effectively resulted in the cardinality constraint being considered as a hard constraint (i.e. $c_{card}\Delta >> \max fitness$). The original distance-based objective function used in CERTIFAI remained unchanged (i.e. the best counterfactuals found by CERTIFAI are scored according to this function). As a base model, we used the original model provided in the code which is a multi-layer perceptron with a hidden layer of $h = 25$ neurons. We note that the results obtained are independent of the base model used, as our method is model-agnostic. The genetic algorithm is run for 10 generations and the probability of mutation and crossover are set to $p_m = 0.2$ and $p_c = 0.5$.

B Additional Experiments

In order to further validate our results, we performed additional experiments using the Car Evaluation dataset (obtained from the UCI Machine Learning repository[3]). This dataset contains six categorial features, where the target shows the acceptance level of the car according to the features (buying price, maintenance price, number of doors, capacity in terms of persons to carry, the size of the luggage boot and an estimated level of safety). The dataset comprises 1728 rows. We run our algorithm to find counterfactuals with a maximum of $k = 3$ different features. In this case, the average cardinality was $\hat{k} = 1.6$, showing that our algorithm was able to find counterfactuals which are often of cardinality $k = 2$ or even $k = 1$. Table 2 shows an additional comparison between an unconstrained counterfactual calculated by CERTIFAI and sparse counterfactuals calculated with our method. As can be seen, using the cardinality constraint can lead to sparser counterfactuals were changing already only very few features can already lead to flip the prediction target.

Table 2: Comparison of counterfactuals obtained without cardinality constraints and with a cardinality constraint of $k = 2$ and $k = 3$. Different values compared to the original example are shown in bold.

	Buying	Maint	Doors	Persons	Lug_boot	Safety
Original	vhigh	med	2	2	small	med
Unconstrained	**low**	**low**	2	**more**	small	**high**
$k = 2$	vhigh	med	2	**more**	**big**	med
$k = 3$	vhigh	med	**4**	**more**	**big**	med

[1] https://github.com/IMC-UAS-Krems/CERTIFAI_card

[2] https://github.com/Ighina/CERTIFAI

[3] https://archive.ics.uci.edu/dataset/19/car+evaluation