Convex optimization for actionable & plausible counterfactual explanations

André Artelt[*] and Barbara Hammer [†]

CITEC - Cognitive Interaction Technology
Bielefeld University - Faculty of Technology
Inspiration 1, 33619 Bielefeld - Germany

Abstract. Transparency is an essential requirement of machine learning based decision making systems that are deployed in real world. Often, transparency of a given system is achieved by providing explanations of the behaviour and predictions of the given system. Counterfactual explanations are a prominent instance of particular intuitive explanations of decision making systems. While a lot of different methods for computing counterfactual explanations exist, only very few work (apart from work from the causality domain) considers feature dependencies as well as plausibility which might limit the set of possible counterfactual explanations. In this work we enhance our previous work on convex modeling for computing counterfactual explanations by a mechanism for ensuring actionability and plausibility of the resulting counterfactual explanations.

1 Introduction

Nowadays we are faced with an increasing deployment of machine learning (ML) and artificial intelligence (AI) based decision making systems in the real world - e.g. predictive policing [1] and loan approval [2, 3]. Because of the high impact of many of these systems, policy makers demand transparency and interpretability of such decision making systems - first approaches have already been manifested in legal regulations like the EUs GDPR [4]. There exist a wide variety of methods for explaining ML and AI based decision making systems and thus meeting the demands for transparency and interpretability. A popular class of explanation methods, that are not tailored to a specific model but rather universal, are model agnostic methods: Feature interaction methods [5], feature importance methods [6] and example based methods [7]. Popular instances of example based methods are influential instances [8], prototypes & criticisms [9] and counterfactual explanations [10]. A counterfactual explanation is a change of the original input that leads to a different (specific) prediction or behavior of the decision making system - *what has to be different in order to change the prediction of the system?* Such an explanation is considered to be intuitive and useful because it proposes changes to achieve a desired outcome, i.e. it provides actionable feedback [11, 10]. Furthermore, there exists strong evidence that explanations by humans are often counterfactual in nature [12]. We will focus on theses types of explanations in this work.

[*] Corresponding author: aartelt@techfak.uni-bielefeld.de

[†] We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society.*

2 Counterfactual explanations

Counterfactual explanations (often just called counterfactuals) contrast samples by counterparts with minimum change of the appearance but different class label [10, 11] and can be formalized as follows:

Definition 1 (Counterfactual explanation [10])**.** *Let $h : \mathbb{R}^d \to \mathcal{Y}$ be a given prediction function. Computing a counterfactual $\vec{x}_{cf} \in \mathbb{R}^d$ for a given input $\vec{x}_{orig} \in \mathbb{R}^d$ is phrased as an optimization problem:*

$$\underset{\vec{x}_{cf} \,\in\, \mathbb{R}^d}{\arg\min}\; \ell\left(h(\vec{x}_{cf}), y_{cf}\right) + C \cdot \theta(\vec{x}_{cf}, \vec{x}_{orig}) \tag{1}$$

where $\ell(\cdot)$ denotes a suitable loss function, y_{cf} the requested prediction, and $\theta(\cdot)$ a penalty term for deviations of $\vec{x}_{cf}$ from the original input $\vec{x}_{orig}$. $C > 0$ denotes the regularization strength.

While the classical formalization Eq. (1) is model agnostic - i.e. it does not make any assumptions on the model $h(\cdot)$ -, it can be beneficial to rewrite the optimization problem Eq. (1) in constrained form [13]:

$$\underset{\vec{x}_{\text{cf}} \,\in\, \mathbb{R}^d}{\arg\min}\; \theta(\vec{x}_{\text{cf}}, \vec{x}_{\text{orig}}) \tag{2a}$$

$$\text{s.t. } h(\vec{x}_{\text{cf}}) = y_{\text{cf}} \tag{2b}$$

In our previous work [13] we have shown that the constrained optimization problem Eq. (2) can be turned (or efficiently approximated) into convex programs for many standard machine learning models including generalized linear models, quadratic discriminant analysis, nearest neighbor classifiers, etc. Since convex programs can be solved efficiently [14], the constrained form [13] becomes superior over the original black-box modeling [10] if we have access to the underlying model $h(\cdot)$.

The counterfactuals from Definition 1 are also called closest counterfactuals because they try to find a point that is as close as possible to the original sample. Is was observed that this often leads to adversarials which might not be useful for explaining the models behaviour [15]. When this is an issue, additional plausibility constraints are added [13, 16, 17, 15] often modeled based on densities like stated in Definition 2.

Definition 2 (δ-plausible counterfactual [15])**.** *Let $h : \mathbb{R}^d \to \mathcal{Y}$ be a prediction function and $p(\cdot)$ a class dependent density. We call a counterfactual explanation $(\vec{x}_{cf}, y_{cf})$ of a particular sample $\vec{x} \in \mathbb{R}^d$ δ-plausible iff the following holds:*

$$\vec{x}_{cf} = \underset{\vec{x}_{cf} \,\in\, \mathbb{R}^d}{\arg\min}\; \theta(\vec{x}_{cf}, \vec{x}) \quad \textit{s.t. } h(\vec{x}_{cf}) = y_{cf} \;\; \textit{and} \;\; p(\vec{x}_{cf}; y_{cf}) \geq \delta \tag{3}$$

where $\delta > 0$ denotes a minimum density at which we consider a sample plausible.

When it comes to communicate the explanation to the use, we have two possibilities: We can either present the solution $\vec{x}_{\text{cf}}$ to the user or the difference $\vec{\delta} = \vec{x}_{\text{orig}} - \vec{x}_{\text{cf}}$ as an explanation to the user. Since the popularity of counterfactual explanations comes from the recommendation of actions that lead to a desired goal (actionable feedback to the user), it is natural to communicate the difference $\vec{\delta}$ to the user. However, when computing and communicating the difference $\vec{\delta}$, we implicitly assume that all features are independent of each other - i.e. we assume that we can arbitrarily and independently change all features at the same time. This is a strong assumption which might not always be true in practice - e.g. it could be the case that some features are anti-correlated so that they can not be both increased or decreased at the same time. Another problem with the plain difference $\vec{\delta}$ as an explanation is that it assumes that the features are somewhat meaningful and interpretable, which is not always the case like in case of images[1].

Actionable and plausible counterfactuals Instead of directly optimizing the final counterfactual $\vec{x}_{\text{cf}}$, we propose to optimize over an *action vector* which encodes the actions/changes applied to the original sample $\vec{x}_{\text{orig}}$ which then yield the final counterfactual $\vec{x}_{\text{cf}}$. We define a function that maps a given action vector to a (potentially counterfactual) explanation $\vec{x}_{\text{cf}}$ - note that $\vec{x}_{\text{cf}}$ might not necessarily be a valid counterfactual because the property of being a counterfactual depends on a prediction function $h(\cdot)$ which the function $f(\cdot)$ is not aware of, $f(\cdot)$ only applies an action vector to the original sample $\vec{x}_{\text{orig}}$.

$$f : \vec{\delta} \mapsto \vec{x}_{\text{cf}} \tag{4}$$

Since Eq. (4) should depend on the original sample $\vec{x}_{\text{orig}}$, we explicitly note the dependency on $\vec{x}_{\text{orig}}$ by writing:

$$f(\vec{\delta}; \vec{x}_{\text{orig}}) = \vec{x}_{\text{cf}} \tag{5}$$

The action vector $\vec{\delta}$ can live in the same space like the original sample $\vec{x}_{\text{orig}}$ - i.e. $\vec{\delta}$ and $\vec{x}_{\text{orig}}$ share the same features. However, it is also possible that $\vec{\delta}$ lives in some kind of latent space - i.e. changing some abstract attributes. In other words, the function $f(\cdot)$ applies the changes $\vec{\delta}$ to the original sample $\vec{x}_{\text{orig}}$ and embeds the result in the data space that is used for making predictions - i.e. it might have to embed text into some vector space, consider feature dependencies, apply one-hot-encodings, etc.

The final optimization problem for computing an action vector that leads to a valid counterfactual explanation is phrased as the following optimization problem:

$$\underset{\vec{\delta} \in \mathbb{R}^m}{\arg\min} \; \theta(\vec{\delta}) \tag{6a}$$

$$\text{s.t. } h\left(f(\vec{\delta}; \vec{x}_{\text{orig}})\right) = y_{\text{cf}} \tag{6b}$$

[1] In case of images, we have pixel as features which are not informative at all.

The original modeling Eq. (2) can be obtained as a special case of Eq. (6) if $f(\vec{\delta};\vec{x}_{\text{orig}}) = \vec{x}_{\text{orig}} + \vec{\delta}$ and $\theta(\cdot)$ denotes a suitable metric.

In this work, we focus on two aspects: Feature dependencies and plausibility. In section 3.1 we propose a realization of $f(\cdot)$ that takes care of potential feature dependencies - i.e. feature can not be changed independently. And in section 3.2 we consider a realization of $f(\cdot)$ where the action vectors corresponds to selecting primitives/prototypes that are combined into the final sample - i.e. some kind of sparse coding via a given codebook (similar to a latent space approach). Finally, in section 4 we empirically evaluate our proposed modelings on several data sets and models.

Related work As already mentioned, there exist a wide variety of work that deals with the computation of counterfactual explanations Definition 1 as well as considering additional aspects like plausibility [13, 16, 17, 15]. However, usually these methods assume that all features can be changed independently from each other - i.e. no further constraints are posed on the actionability of the difference $\vec{\delta} = \vec{x}_{\text{orig}} - \vec{x}_{\text{cf}}$ - only methods that use structural causal models or some predefined set of valid actions take potential feature dependencies into account [18, 19]. In addition, those existing methods often have high computational complexity by relying on non-convex optimizations like integer programming.

A method for computing actionable recourse based on probabilistic counterfactual explanations is proposed in [18]. This method heavily relies on a given or estimated probabilistic causal model that encodes all variable dependencies - the counterfactuals itself are then computed by solving integer programs.

Another (early) work [20] is concerned with actionable recourse of linear classifiers. For a given set of actionable actions, their method solves an inter program for computing the final actionable counterfactual explanation.

In [17] a method called FACE for computing feasible and actionable counterfactual explanations is proposed. Instead of computing a single change or action vector that leads to the counterfactual, a path of intermediate samples is constructed that finally lead to the counterfactual explanation - assuming that moving from one sample to a nearby sample is always feasible and actionable.

The authors of [19] propose to use a variational autoencoder for learning feasible/actionability constraints from labeled data or user feedback. These learned constraints are then used when computing the final counterfactual explanations.

Other work like [21, 22, 23] use some kind of autoencoder to learn a latent space and then compute the counterfactuals in this latent space to get plausible and meaningful counterfactual explanations.

All these methods need some kind causal graph as an input and are computational infeasible for many models or are even limited to linear models - on the other hand, methods that work for arbitrary models come without any formal guarantees.

3 Actionable and plausible counterfactual explanations

3.1 Feature dependencies

Instead of assuming independence of all features, we aim for a mechanism that allows us to consider potential feature dependencies when computing counterfactual explanations.

Assuming a fixed but unknown data generating process, the simplest measurement of feature dependencies is to consider the covariance matrix $\mathbf{\Sigma} \in \mathcal{S}_{+}^{d}$:

$$\mathbf{\Sigma} = \mathbb{E}\Big[\big(\vec{X} - \mathbb{E}[\vec{X}]\big)\big(\vec{X} - \mathbb{E}[\vec{X}]\big)^\top\Big] \tag{7}$$

where $\vec{X}$ denotes the random vector of the underlying generating process.

By standardizing the covariance Eq. (7), we obtain the correlation matrix $\tilde{\mathbf{\Sigma}} \in \mathcal{S}_{+}^{d}$ which can be stated as follows:

$$\tilde{\mathbf{\Sigma}} = \mathrm{diag}(\mathbf{\Sigma})^{-\frac{1}{2}}\mathbf{\Sigma}\,\mathrm{diag}(\mathbf{\Sigma})^{-\frac{1}{2}} \tag{8}$$

In the remainder of this work, we use the correlation matrix Eq. (8) for encoding feature dependencies - however, note that correlation does not necessarily denote a causal relationship - furthermore, non-linear dependencies are also not covered. We still think that considering only linear feature dependencies can already be beneficial and is better than assuming independence of all features which might be unrealistic for many real world scenarios.

Usually, the true covariance matrix is not known in practice and therefore we have to work with some kind of approximation. It might be the case that some entries in the covariance matrix are known or at least some educated guesses by domain experts are available. Otherwise we have to estimate it from a given data set. Since the straight forward estimation by using the maximum likelihood estimator[2] of Eq. (7) is unstable in high dimensions, we use a method for estimating a sparse covariance matrix [24][3]:

$$\hat{\mathbf{\Sigma}}^{-1} = \underset{\hat{\mathbf{\Sigma}}^{-1} \in \mathcal{S}_{+}^{d}}{\arg\min}\; \mathrm{trace}\left(\mathbf{\Sigma}_{\text{emp}}\hat{\mathbf{\Sigma}}^{-1}\right) - \log\left(\det(\hat{\mathbf{\Sigma}}^{-1})\right) + \alpha\|\hat{\mathbf{\Sigma}}^{-1}\|_1 \tag{9}$$

where $\mathbf{\Sigma}_{\text{emp}} \in \mathcal{S}_{+}^{d}$ denotes the empirical covariance matrix that has been estimated from a given data set using the maximum likelihood estimator,$\|\cdot\|_1$ denotes the sum of the absolute values of the off-diagonal coefficients of the given matrix and $\alpha > 0$ is a hyperparameter that controls the sparsity of the covariance matrix (higher values lead to more sparsity).

[2] Also called empirical covariance where we replace the expectations by averages.

[3] The choice of this particular method is kind of arbitrary - there exist many other methods for estimating covariance matricies in (possibly) high dimensional spaces.

3.1.1 Realization of the action mapping function

In order to ensure that the final counterfactual is actionable according to linear feature dependencies, we define the action mapping function $f(\cdot)$ as follows:

$$f(\vec{\delta}; \vec{x}_{\text{orig}}) = \vec{x}_{\text{orig}} + \tilde{\mathbf{\Sigma}}(\vec{x}_{\text{orig}})\vec{\delta} \tag{10}$$

where $\tilde{\mathbf{\Sigma}}(\vec{x}_{\text{orig}})$ denotes the correlation matrix which encodes the specific linear feature dependencies for the particular sample $\vec{x}_{\text{orig}}$. While one could use a global correlation matrix $\tilde{\mathbf{\Sigma}}$ which does not depend on $\vec{x}_{\text{orig}}$, it might be beneficial to be able to use different correlation matrices for different users or groups. By this we can take differences in feasible actions for different users into account.

In the context of Eq. (10), changing some feature in $\vec{\delta}$ might result in a change (increase or decrease) of a different feature - i.e. it might be impossible to increase two features at the same because of some anti-correlation. While this formalization Eq. (10) allows us to encode some feature dependencies, note that it is still limited in the sense that time is ignored - i.e. all changes might not take place at the time but be applied in some order which itself could have some consequences on the previous changes.

3.2 Plausibility

We ensure plausibility by assuming that all data samples are must be composed from a set of given primitives/prototypes arranged column wise in a matrix $\mathbf{P} \in \mathbb{R}^{d \times m}$ as well as a base $\vec{b} \in \mathbb{R}^d$. We therefore define $f(\cdot)$ as follows:

$$f(\vec{\delta}; \vec{x}_{\text{orig}}) = \mathbf{P}\left(z(\vec{x}_{\text{orig}}) + \vec{\delta}\right) + \vec{b} \tag{11}$$

where $z : \mathbb{R}^d \to \mathbb{R}^m_+$ denotes a function that encodes the given sample in a latent space - i.e. the selection and weighting of the given primitives in $\mathbf{P}$. The primitives could be given (i.e. hand engineered) or learned by some kind of dictionary/representation learning (including PCA and ICA).

If it happens to be the case that all samples in the convex hull of the primitives $\text{covx}(\mathbf{P}, \vec{b})$ are δ-plausible according to [15], we can easily ensure δ-plausibility (under Definition 2) of the final counterfactual explanations as stated in Lemma 1.

Lemma 1. *Assuming that for given primitives $\mathbf{P}, \vec{b}$, all sample $\vec{x} \in$ covx$(\mathbf{P}, \vec{b})$ are δ-plausible (Definition 2) for a fixed and given $\delta > 0$.*

Then, when using Eq. (11) *as a realization of the action function Eq.* (4) *and the additional linear constraints $\sum_i (z(\vec{x}_{orig}) + \vec{\delta})_i = 1$, $\vec{\delta} \geq \vec{0}$, the resulting counterfactual explanations $\vec{x}_{cf}$ Eq.* (6) *are guranteed to be δ-plausible.*

3.3 Convex optimization for actionable & plausible counterfactuals

In our previous work [13], we show how to rewrite the constraint Eq. (2b) as a set of convex constraints or a set of convex programs for many different ML models[4]. We assume that we can rewrite the constraint Eq. (2b) as a set of convex functions $\phi_i(\cdot)$:

$$\phi_i(\vec{x}) \leq 0 \quad \forall\, i \tag{12}$$

Substituting our proposed parametrization Eq. (10) into Eq. (12) yields:

$$\phi_i\left(f(\vec{\delta})\right) \leq 0 \quad \forall\, i \tag{13}$$

If our realization of the action mapping function Eq. (4) is convex - our realization for feature dependencies and plausibility are both affine and thus convex -, the original constraints Eq. (12) are convex and the concatenation of two convex functions is convex [14], the new constraints Eq. (13) are also convex. Convexity of the objective Eq. (6a) depends on the chosen regularization $\theta(\cdot)$ - it is clear that convexity is given in case of the weighted Manhattan or L2 norm.

Our proposed modeling for actionable & plausible counterfactuals therefore nicely fits into our previously proposed convex modeling framework for computing counterfactual explanations - i.e. we can use convex optimization for efficiently computing counterfactual explanations for many different standard ML models.

4 Experiments

We empirically verify our proposed modeling for actionable & plausible counterfactuals by comparing them to unconstrained counterfactual and check if and how they differ. The Python implementation of the experiments is available on GitHub[5] whereby we use MOSEK[6] as a solver for all mathematical programs.

4.1 Feature dependencies

Data sets We use the following three standard benchmark data sets: The "Iris Plants Data Set" [25], the "Breast Cancer Wisconsin (Diagnostic) Data Set" [26] and the "Wine data set" [27].

Models We use a softmax regression classifier and a general learning vector quantization (GLVQ) classifier with 3 prototypes per class.

Setup We perform the following experiment over a 3-fold cross validation: In order to improve numerical stability, the training and test samples are standardized before the classifier is fitted and the correlation matrix is estimated from the training data - we use Eq. (9) and $\alpha = 0.8$ for estimating a sparse covariance matrix. For all correctly classified samples in the test set, we compute a normal closest counterfactual Eq. (2) and a closest actionable counterfactual (under Eq. (10)) - in both cases we use L1 norm as a regularization $\theta(\cdot)$ and use

[4] For some models we propose model specific convex approximations.

[5] `https://github.com/andreArtelt/ActionableCounterfactualsConvexProgramming`

[6] We gratefully acknowledge an academic license provided by MOSEK ApS.

Data set	Iris	Wine	Breast cancer
Softmax regression	0.74	0.1	11.12
GLVQ	0.63	0.62	11.52

Table 1: Median Euclidean distance between the closest counterfactual and the closest actionable counterfactual

a random target label. For each pair of counterfactuals we compute their Euclidean distance and the number of overlapping features - we consider a feature to be overlapping if it is in both cases equal to zero or not equal to zero.

Results The median Euclidean distances between the closest counterfactual and the closest actionable counterfactual is shown in Table 1. On average, we observe in all cases significant differences between the closest counterfactual and the closest actionable counterfactional. This suggests that there might be substantial differences in both types of counterfactuals. However, the Euclidean distance itself does not tell anything about the chosen features - i.e. are the same or different features changed. Therefore, we plot the the number of overlapping features in Fig. 1 - recall that we consider a feature to be overlapping if it is in both cases equal to zero or not equal to zero (i.e. we count the number of same changed features). We observe that in most cases the same features are changed - although there are always a few outliers. Only in case of the breast cancer data set we observe a significant difference in the choose features - on average there is a difference of two features. This shows that, depending on the data set, the covariance matrix and the model, it can actually happen that a closest actionable counterfactual uses different features than a closest counterfactual explanations.

Example For an illustrative (but real) example, consider the following correlation matrix that has been estimated from the Iris data set:

$$\begin{pmatrix} 1. & 0. & 0.07913005 & 0.04025559 \\ 0 & 1. & 0. & 0. \\ 0.07913005 & 0. & 1. & 0.16517646 \\ 0.04025559 & 0. & 0.16517646 & 1. \end{pmatrix} \tag{14}$$

When computing the closest counterfactual explanation (under a softmax regression) of a specific sample $\vec{x}_{\text{orig}}$, we find that:

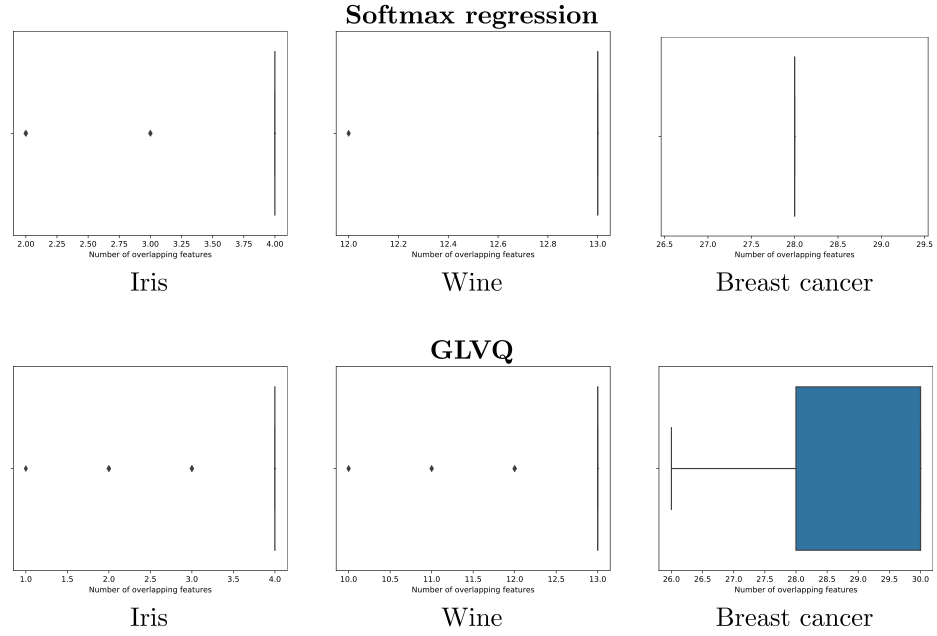
$$\vec{x}_{\text{cf}} = (-1.53442, 0., 0., 0.)^\top \tag{15}$$

However, when considering the correlation matrix Eq. (14), the closest actionable counterfactual at the same sample $\vec{x}_{\text{orig}}$ turns out to be:

$$\vec{x}_{\text{cf}} = (0., 0., 0., -1.36813)^\top \tag{16}$$

Because of the correlations in Eq. (14), the selected feature to be changed is different for closest counterfactual vs. closest actionable counterfactual.

Fig. 1: Box plots of the number of overlapping features of closest counterfactuals and closest actionable counterfactuals.



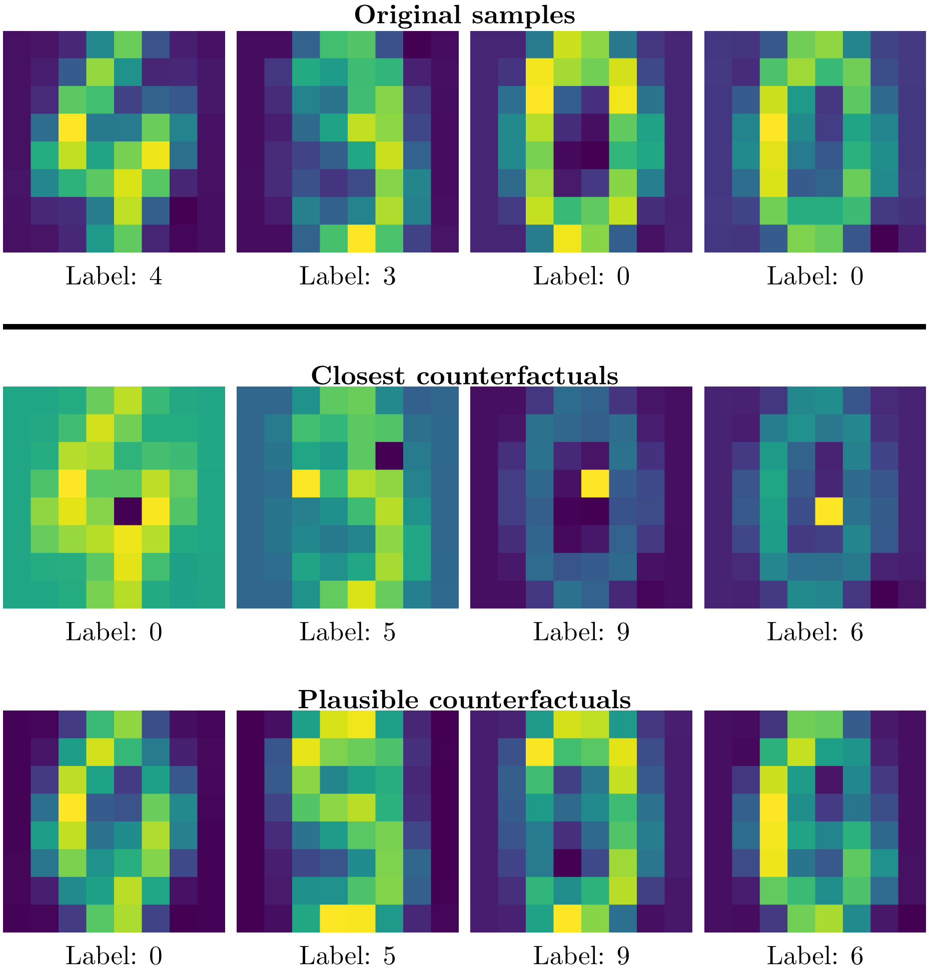
Iris Wine Breast cancer

Iris Wine Breast cancer

4.2 Plausibility

In order to evaluate plausibility of the counterfactuals from section 3.2, we compute closest counterfactuals and plausible counterfactuals (according to the action function Eq. (11)) of the Digits data set [28] under a softmax regression model. We use sparse dictionary learning to learn 10 prototypes/primitives that are used to encode all samples. We show a few examples of original, closest and plausible counterfactuals in Fig. 2. We observe that most cases the closest counterfactual looks like an adversarial whereas in case of the plausible counterfactual the target class can be often clearly recognized. However, note that we are looking at a few samples only and a “true” evaluation would require an extensive user study since perception of plausibility is difficult to formalize.

Fig. 2: Closest vs. plausible counterfactual explanations. First row shows the original samples, the closest and plausible counterfactuals are shown in the second and third row. The original label or target label is given below each image.



Original samples

Closest counterfactuals

Plausible counterfactuals

5 Conclusion

In this work we extended our convex programming framework for computing counterfactual explanations by a mechanism where we optimize over the actions rather than directly the final counterfactual explanations. By this we were able to consider linear feature dependencies - i.e. getting rid of the assumption that all features are independent -, as well as ensuring plausibility of the final counterfactual by constructing it as a linear combination of given primitives/prototypes. Besides working out the modeling details, we also empirically evaluated our modeling on different data sets, where we observed significant differences in the counterfactuals considering estimated correlations versus counterfactuals assuming feature independence.

Since our formalization is limited to linear dependencies we plan to extend our modeling for non-linear dependencies as well as to consider more ML models like deep neural networks. We also would like to study the difference of closest and closest actionable counterfactuals from a psychological point of view - i.e. how are they perceived and in particular whether closest actionable counterfactuals are considered to be "better" or not.

References

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias - there's software used across the country to predict future criminals. and it's biased against blacks. 2016.

[2] Amir E. Khandani, Adlar J. Kim, and Andrew Lo. Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2010.

[3] Kaveh Waddell. How algorithms can bring down minorities' credit scores. The Atlantic, 2016.

[4] European parliament and council. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016.

[5] Brandon M. Greenwell, Bradley C. Boehmke, and Andrew J. McCarthy. A simple and effective model-based variable importance measure. CoRR, abs/1805.04755, 2018.

[6] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance. arXiv e-prints, page arXiv:1801.01489, Jan 2018.

[7] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and systemapproaches. AI communications, 1994.

[8] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pages 1885–1894, 2017.

[9] Been Kim, Oluwasanmi Koyejo, and Rajiv Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 2280–2288, 2016.

[10] 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.

[11] Christoph Molnar. Interpretable Machine Learning. 2019. https://christophm.github.io/interpretable-ml-book/.

[12] Ruth M. J. Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 6276–6282. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[13] André Artelt and Barbara Hammer. On the computation of counterfactual explanations - A survey. CoRR, abs/1911.07749, 2019.

[14] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, New York, NY, USA, 2004.

[15] André Artelt and Barbara Hammer. Convex density constraints for computing plausible counterfactual explanations. 29th International Conference on Artificial Neural Networks (ICANN), 2020.

[16] Yu-Liang Chou, Catarina Moreira, Peter Bruza, Chun Ouyang, and Joaquim Jorge. Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications, 2021.

[17] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodriguez, Tijl De Bie, and Peter A. Flach. FACE: feasible and actionable counterfactual explanations. CoRR, abs/1909.09369, 2019.

[18] Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals, 2021.

[19] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. CoRR, abs/1912.03277, 2019.

[20] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In danah boyd and Jamie H. Morgenstern, editors, Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019, pages 10–19. ACM, 2019.

[21] Amir Feghahati, Christian R. Shelton, Michael J. Pazzani, and Kevin Tang. Cdeepex: Contrastive deep explanations. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of Frontiers in Artificial Intelligence and Applications, pages 1143–1151. IOS Press, 2020.

[22] Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam H. Laradji, Laurent Charlin, and David Vázquez. Beyond trivial counterfactual explanations with diverse valuable explanations. CoRR, abs/2103.10226, 2021.

[23] Rachana Balasubramanian, Samuel Sharpe, Brian Barr, Jason D. Wittenbach, and C. Bayan Bruss. Latent-cf: A simple baseline for reverse counterfactual explanations. CoRR, abs/2012.09301, 2020.

[24] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432–441, 12 2007.

[25] Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. Annual Eugenics, 7 Part II:179–188, 1936.

[26] Olvi L. Mangasarian William H. Wolberg, W. Nick Street. Breast cancer wisconsin (diagnostic) data set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic), 1995.

[27] D. Coomans S. Aeberhard and O. de Vel. Comparison of classifiers in high dimensional settings. Tech. Rep. no. 92-02, 1992.

[28] E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits data set. https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits, 1998.