Counterfactual Explanation for Tabular Dataset

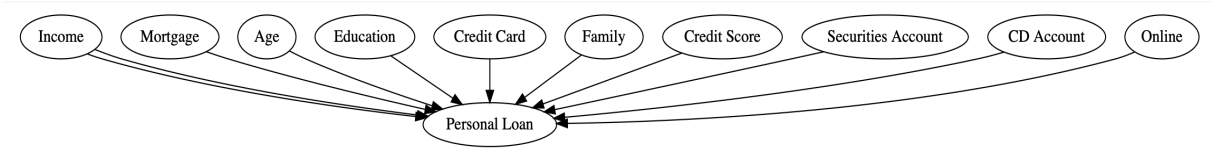
Sudip Khadka
Machine Learning for Scientific Computing (AMSC808X)
Department of Quantitative Finance
Department of Mathematics
Robert H Smith School of Business
University of Maryland, College Park

Abstract:
Explainable Artificial Intelligence (XAI) has provided an explainability framework to address the challenges associated with the black-box and white-box problems in Machine Learning by improving transparency in model predictions. This paper provides a review of empirical research focused on investigating the white-box problem of machine learning models through the application of the Counterfactual Explanation (CFE) technique proposed by DICE and Wachter. This paper also aims to illuminate the process of generating counterfactuals and explore the distinctions between naive counterfactual generation and the counterfactuals proposed by DICE and Wachter's models. Finally, it specifically emphasizes adjusting for confounding variables increases the out-of-sample prediction accuracy while reducing bias in counterfactual generations.

Introduction:
Counterfactual Explanation is the explainability framework that describe the smallest adjustment to the local features value that is needed to change the prediction to a predefined output. It is the local, example-based post-hoc explanation methods that examines whether a particular value of the feature is necessary for the model to arrive at its prediction *(Christoph).* Counterfactuals are often framed in terms of attributing a given outcome to a particular cause. For example, a commercial bank ran a marketing campaign to attract customers to open a loan account. Assuming our model predicted the person did not open loan account, the counterfactual instance explains, what small change would need to the feature so that the person would open an account. The rest of the paper is organized in the following manner. Section I explains the naive model and the comparison of the counterfactual model proposed by the *Wachter et al. 2017* and DICE. Section II talks about proposed Confounder identifying filtration technique.

Counter Factual Generation and Comparison

Section I explains the naive counterfactual generation and the comparison of the counterfactual model proposed by the *Wachter et al. 2017* and DICE. Back to our example here is a simple Structural Causal Model.



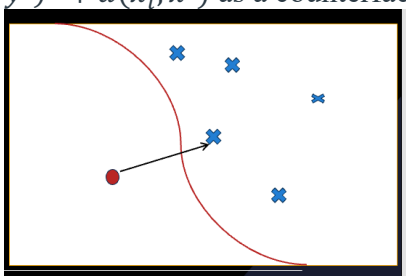
Counterfactual Generation: A simple way to generate counterfactual is through trial and error $\tau = pr(income_i\ is\ caused\ of\ y * | income_i = a,\ y = y*)$. We keep everything constant and only change Income, see how often does the classifiers outcome change. Each of the perturbed inputs that can change the value of y i.e. personal Loan considered as a counterfactual example. Another approach is through optimization. First, we define a loss function. This loss takes as input the instance of interest, a counterfactual and the desired (counterfactual) outcome. Then, we can find the counterfactual explanation that minimizes this loss using an optimization algorithm *(Guidotti)*. To find a counterfactual example that minimize the distance between original instance(x) and the counterfactual instance(x') subject to the prediction that is identical to the desired output.

$$\arg\min_{x'} d(x, x')\ subject\ to\ f(x') = y^*$$

The model proposed by *Wachter et al. 2017* is very similar to the naive counterfactual generation but it utilized the Euclidean distance to measure the instances *(Christoph)*. Wachter model is designed to offer just enough information to influence a decision without needing the individual to comprehend its inner workings to utilize it.

$$arg\min_{x'} \max_{\lambda} \lambda(f_W(x') - y')^2 + d(x_i, x')$$

$d(x_i, x')$ is the $\sum_{i=1}^{n} \frac{|x_i - x'_i|)}{j \in median_{\{1,\ldots,n\}}(|x_{j,i} - median_{L \in \{1,\ldots,n\}}(x_{L,i})|)}$ where total distance is the sum of all feature wise distance which are scaled by the inverse of the median absolute deviation of feature. Instead of using mean as the center and summing over the squared distance, it uses the Euclidean distance (1). It involves repeatedly solving for x' and maximizing λ until a solution that is close enough original instance is achieved. $f(.)$ is the feed forward neural network with Adam loss in our case. Like naive, counter factual generation, we start each iteration with various random values of x' and choose the most optimal minimizer that satisfies $arg\min_{x'} \max_{\lambda} \lambda(f_W(x') - y')^2 + d(x_i, x')$ as a counterfactual.



The region with blue is the selection region and the red is rejection. Above equation highlights what is the minimum change that is required to get into the other side of the decision boundary. Let's see our example from the bank marketing campaign.

Original Instance

True Label: 0

Predcited Label: [0]

Features of the 3 th training example:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard
2803	43.0	18.0	41.0	1.0	0.3	3.0	0.0	0.0	0.0	1.0	0.0

Counterfactual Instance:

Desired Level 1:

Features of the countefactual:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard
0	30.0	6.0	32.0	2.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0

To get into the other side of the decision boundary or the change required for the person to open the loan account. If the individual is 13 years younger, possesses 6 year of work experience, has a family size of 2, an average credit score of 1, holds a bachelor's degree, and lacks an online bank account will open the loan account. It seems plausible that they might require loan, given their lower income, lesser qualifications, and large family size.
Nevertheless, it's crucial to acknowledge that this approach only ensures validity, meaning it selects all counterfactuals minimizing distance, while overlooking other critical properties such as Actionability, Sparsity and Data manifold Closeness *(Guidotti)*.

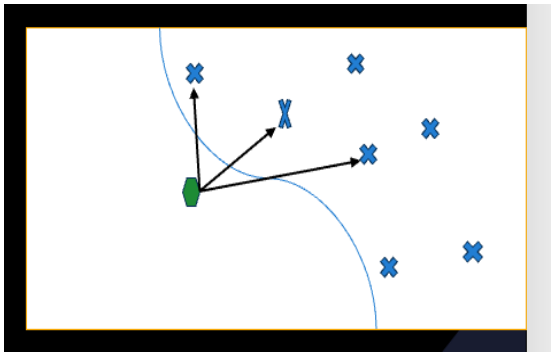
Similarly, the model processed by [*Mothilal, Ramaravind Kommiya, et al]* a Diverse Counterfactual Explanations (DICE) model extends *Wachter et al. 2017* by constructing an optimization problem that considers the diversity of the generated counterfactual examples. It generates the set of diverse counterfactual instance for any differentiable classifier by using the Determinantal Point process. We can find the DICE counterfactual instance by solving the following optimization problem.

$$C(x) = \arg\min_{C_1,\ldots,C_n} \frac{1}{n}\sum_{i=1}^{n} yloss(f(C_i),\ y)\frac{\lambda_1}{n}\sum_{i=1}^{n} dist(C_i,\ x) - \lambda_2 dpp(C_1, \ldots\ldots,\ C_n)$$

$$dpp = det(K)$$

$$where,\ K_{i,j} = \frac{1}{1 + dist(C_i, C_j)}$$

dpp measures the sparsity among the counterfactuals and λ's are the hyperparameter that balance the loss functions. yloss(.) is a metric that minimizes the distance between $(f(C_i),\ y)$ using uses L1 distance, n is total CF to be generated and $f(.)$ is the black box model (or Feed Forward Neural Network in our case)



Let's again examine a straightforward example illustrating how the counterfactual is selected using the decision boundary. Unlike *Wachter et al*, now we have three sets of counterfactual examples. Let's see our example from the bank marketing campaign.

Query instance (original outcome : 0)

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard	Personal Loan
0	57	31	32	3	1.4	1	0	1	1	1	1	0

Counterfactual Instance:

Diverse Counterfactual set (new outcome: 1)

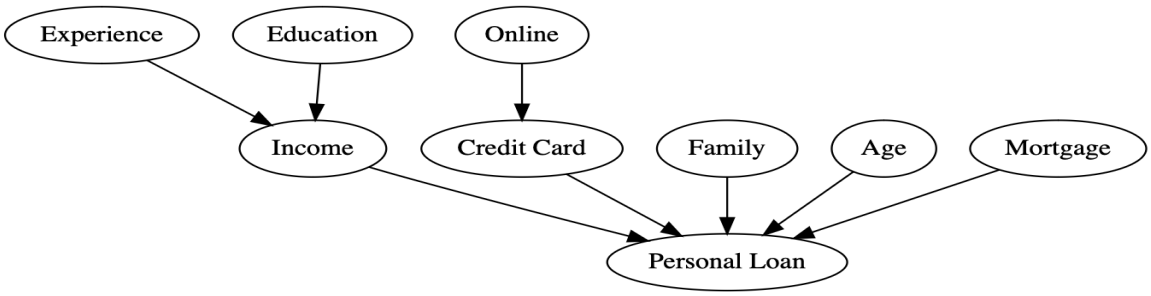
	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard	Personal Loan
0	57	31	119	3	1.4	1	0	1	1	1	1	1
1	26	31	32	3	1.4	1	0	1	1	1	1	1
2	27	31	32	3	1.4	1	0	1	1	0	1	1

To get into the other side of decision bounder we have three sets of possible solutions. First proposed solution is targeting individuals with income higher by 87,000 while keeping all other factor constant. Second, target individual who is younger by 31 years. Third, individual who is younger by 30 years and has an online account.

Confounder identification and filtration technique

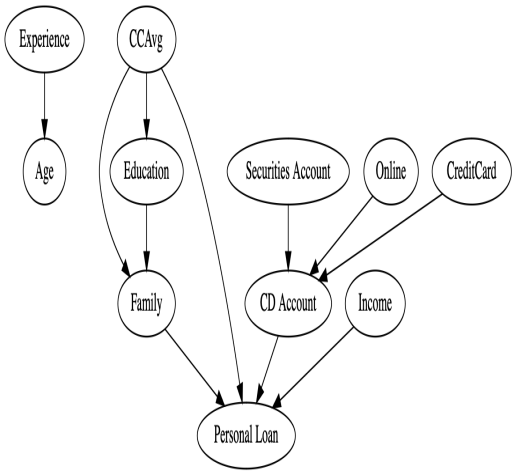
In Section II, the discussion begins with an inquiry into the reliable of the train process. Specifically, the focused is on whether the counterfactual is sufficiently dependable to inform critical decisions. It can be contended that DICE exhibits reliability in case where the cause-and-effect relationships are not overly intricate. Although DICE does not inherently address multicollinearity, it can be mitigated during the training process through the incorporation of regularizations such as Lasso and Ridge regularization. An illustrative example of simple hypothetical Directed Acyclic Graphs (DAGs).

Fig (1)

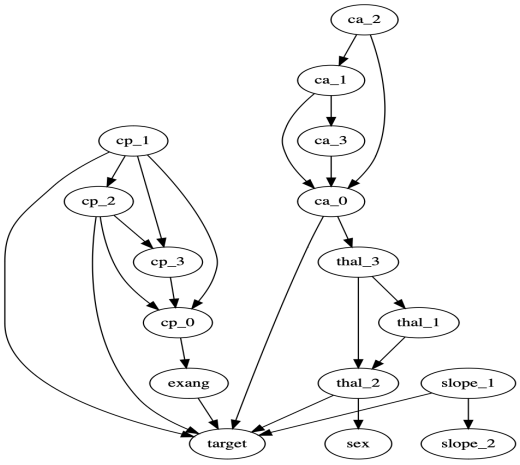


What if we have the confounders variable[1] and the DAGs is not as simple as before. In such case, Confounders are not adjusted and simply considered as a cause. Confounders make the model prediction and the generated counterfactual bias. An illustrative two examples of real Directed Acyclic Graphs (DAGs).

Fig(2) (Bank marketing Campaign.



Fig(3) Heart attack patients[2]



The DAGs are generated by Bayesian network which I will come back later. "CCAvg" is the confounding variable in our Casual Model. Work done by [*Wang, H., Wu, Z., & Xing, E. P. (2019)]* claims removing the confounding factors from the deep neural network improves the model prediction. *[Wang, H., Wu, Z., & Xing, E. P. (2019)]* proposed the Confounder Filtration Method (CFM) that filters out the potential confounders from the trading by removing the weight associate with the confounders while improving the generalizability of Deep Neural Network.
Propose Idea: I am taking very similar approach but instead of identifying the confounder during the training process like [*Wang]*, I am using Bayesian network to get the DAGs to identify the confounder before training and then masking them to get confounder free training model. DAGs are generated by Hill climbing search [3]estimator that learns network structure from data using the Bayesian network which can efficiently search possible DAGs structure *(khadija, Bouyakhsaine).* The DAGs obtained is the probabilistic relationship among variables based on the observed data. While DAGs obtained by Bayesian network show the relations between variables the model is unaware of outcome variable, I imposed additional constrain that any edge connected to the outcome must connect as a cause. Figure 2 and 3 exabits the generated DAGs.

Algorithms:
Bayesian network for generating DAGs:
Initialization: $\theta_0 = random\ initial\ structure$
Evaluate: $\hat{\theta}(\theta_0)$ = Score of θ_0, in our case score evaluation metric is Bayesian

[1] Confounding variable are those causes that influence both the independent (treatments) and the dependent variable (effect).

[2] Target is the possibility of heart attack that takes 0 as not heart attack and 1 as heart attack. The DAGs were generated by filtering out the anomalies.
[3] Hill climbing is the optimization algorithm used to find the best possible solution. It is a local search algorithm that captures the global maximum my making small improvement to the current solution.

Information Criterion (BIC) where BIC = -2 * loglikelihood + total parameters * Log (training sample size)

Neighbor Generation: $N(\theta_0) = \{\theta_1, \theta_2, \ldots\ldots, \theta_n\}$, where θ_i represents a neighboring structure generated by making small modification to θ_0

Selection: $\theta' = \underset{\theta_i \in N(\theta_0)}{\arg\max}(\hat{\theta}(\theta_i) - \hat{\theta}(\theta_0)$

Iterate until: $\theta_{i+1} = \theta'$, if stopping criteria is meet terminate the loop.

Final_output: Constrain $\{(x_i, y_i)\ if\ x_i \neq target\ else\ (y_i, x_i) for\ i = 1,2,..,n\}$

Training model after incorporating the Confounder filtration:

Step 1: Masking Confounder identified by the DAGs:

Let, X be the input data with n X m dimensional.

mask_var be the index of the feature that need to be masked.

M represents binary matrix of size n X m, where each element $M_{ij} is$;

$$M_{ij} = \begin{cases} 1, if\ j < mask_{var} | j \geq mask_{var} + 1 \\ 0, \qquad otherwise \end{cases}$$

$$X_{masked} = X \odot M$$

Step 2: Training process of Neural network:

$$\hat{\theta} = \underset{\theta}{arg\,min}\, f(y, f(X_{masked}; \theta))$$

Step 3: Integrating confounder free training model into DICE optimization:

$$C(x) = \underset{C_1, \ldots\ldots, C_n}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} yloss(f(C_i), y) \frac{\lambda_1}{n} \sum_{i=1}^{n} dist(C_i, x) - \lambda_2 dpp(C_1, \ldots\ldots, C_n)$$

where,

$f(.) \equiv f(y, f(X_{masked}; \theta)$

Prediction Performance:

Models	Test Accuracy	Training Accuracy	Loss (MSE)
Before Confounder Adjustment	85.51%	91.8%	0.29
After Confounder Adjustment	90.43%	92.87%	0.25

Counterfactual Generation by DICE after Adjustment:

Query instance (original outcome : 0)

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard	Personal Loan
0	65	39	22	3	0.7	2	0	0	0	0	0	0

Diverse Counterfactual set (new outcome: 1)

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard	Personal Loan
0	31.0	-	-	2.0	-	-	-	-	-	-	-	1.0
1	31.0	-	-	-	-	-	-	-	-	-	-	1.0
2	-	-	105.0	-	4.3	-	-	-	-	-	-	1.0

The accuracy and the reliability of the counterfactual model are qualitative; there are no mathematical models available to assess the reliability of the generated counterfactual examples. The evaluation of the model relies on factors such as Actionability, Sparsity, Causality, Data Manifold, time complexity and model access *(Guidotti)*.

Properties	Actionability	Sparsity	Causality	Data Manifold	Low Time complexity	Model access	Confounding
Models							
Wachter et al	√	×	√	×	×	Model-agnostic	×
DICE	√	√	√	√	√	Model-agnostic	×
New DICE	√	√	√	√	√	Model-agnostic	√

Conclusion:

This paper delved into the examination of various counterfactual methods in the literature. Subsequently, I proposed a model aimed at reducing bias in counterfactual explanations while simultaneously enhancing model predictions. This was achieved through the strategic incorporation of controls for confounding factors in the tabular dataset.

References

1. Molnar, Christoph. “Interpretable Machine Learning.” *9.3 Counterfactual Explanations*, 21 Aug. 2023, christophm.github.io/interpretable-ml-book/counterfactual.html.
2. Mothilal, Ramaravind Kommiya, et al. “Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End.” *arXiv.Org*, 29 May 2021, arxiv.org/abs/2011.04917.
3. Wang, H., Wu, Z., & Xing, E. P. (2019). *Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for Healthcare Applications*. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6417810/#FN1
4. Guidotti, Riccardo. “Counterfactual Explanations and How to Find Them: Literature Review and Benchmarking - Data Mining and Knowledge Discovery.” *SpringerLink*, Springer US, 21 Sept. 2022, link.springer.com/article/10.1007/s10618-022-00831-6.
5. “Hill Climb Search¶.” *Hill Climb Search - Pgmpy 0.1.23 Documentation*, pgmpy.org/structure_estimator/hill.html
6. khadija, Bouyakhsaine. “Hill Climb Search: A Heuristic Optimization Algorithm _used to Estimate Dag (Directed Acyclic...” *Medium*, 8 Dec. 2023, medium.com/@bouyakhsainekhadija1999/hill-climb-search-a-heuristic-optimization-algorithm-used-to-estimate-dag-directed-acyclic-db0a1bfc0066#:~:text=By%20using%20Hill%20Climbing%20with,based%20on%20the%20observed%20data.
7. *Probabilistic Graphical Models: Principles and Techniques*, mcb111.org/w06/KollerFriedman.pdf.