Internship Report: Causal Theory Applied to Machine Learning Fairness

Maxence Nesme

July 26, 2022

1 Introduction

The aim of this document is to provide an overview and our findings about the current state of the causal theory applied to the development of fair AI-based models. From as far as we know, literature is mainly concerned about the improvement or creation of new methods for assessing fairness or on the ethical issues inherent to the use of counterfactual values, in particular on immutable parameters such as Race and Gender. However few papers discussed the applicability of those methods when a datascientist effectively develop an AI-based model, that is the choice we have made. Our work contribution is tripartite:

- We provide a discussion on the use of causal methods applied to fairness on a real-case dataset: The German Credit Dataset.
- Following from this discussion we describe a workflow to be used to better evaluate the relevance of one causal method compared to another through the example of the Counterfactual Fairness.
- Finally, we provide implementation of our method. The code can be found here[1]

2 Background

In the following section we will give a quick overview of the current work on fairness and notably applied to Machine Learning Algorithms. We review the necessary knowledge about Causal Theory. While we do our best to summarize it we encourage interested readers to explore the various paper on this subject.

[1] https://gitlab.tech.orange/frederic.guyard/fairness

2.1 Fairness of ML Algorithms

The increasing use of machine learning in decision taking in our society and the various scandals around artificial systems about fairness questions lead a growing part of actors to manifest interest in developing ethical and fair AI models in their domain. As shown in [6], Those debates are mainly pushed by western stakeholders and particularly through the writing of whitepapers and guidelines. [6] listed the most depectited principles of among a great number of documents, ordered by frequency of appearance and showing that no consensus has been found on what composes an ethical AI:

1. Transparency
2. Justice & Fairness
3. Non-maleficence
4. Responsibility
5. Privacy
6. Beneficence
7. Freedom & Autonomy
8. Trust
9. Sustainability
10. Dignity
11. Solidarity

Among the definitions of an ethical AI, fairness plays an important place. It is often refered as the prevention, monitoring, or mitigation of bias and discrimination but also as the respect of diversity, inclusion and equality. Some documents also supports the importance of fair access to AI and its benefits. Bias are inherent to Machine Learning. They are the consequences of technological choices that were made designing the AI-based process but they also come from the human society. Being spread along the whole pipeline of artificial models, they can be of various type: historial/strucutal bias, data generation bias, learning bias, evaluation bias, etc. In order to leverage those concerns, contradictory methods have been proposed such as increasing the dataset size so that it would naturally lead to lower bias in the whole process which seems not to agree with EU's policy on data management and GDPR for instance. Fairness vary with situations, culture, applications, it is not the same at the individual or group level. Following this idea the literature proposed various fairness metrics to estimate and considerate this notion: group and individual metrics that all have in common to be observable. We write some of them below. We denote by $\hat{Y}$ the output of the ML algorithm, and by A the sensitive parameter on which we want to measure fairness, for instance Sex, Race, Age, etc.

$$\text{Statistical Parity: } P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0) \tag{1}$$

$$\text{Predictive Parity: } P(Y = 1|\hat{Y} = 1, A = 0) = P(Y = 1|\hat{Y} = 1, A = 1) \tag{2}$$

$$\text{Overall Accuracy Equality: } P(Y = \hat{Y}|A = 0) = P(Y = \hat{Y}|A = 1) \tag{3}$$

$$\text{Equalized Odds: } P(\hat{Y} = 1|Y = y, A = 0) = P(\hat{Y} = 1|Y = y, A = 1) \quad \forall y \in [0, 1] \tag{4}$$

Facilitating implementations, they are yet to be sufficient, some of them are contradictory or not compatible. Depending of the domain of applications one could show fairness where there is in reality discrimination, the opposite being also true. Hence the necessity to adopt another paradigm: Causality.

2.2 Causality

[12] described causation as so:

> *A variable X is a cause of a variable Y if Y in any way relies on X for its value.*

Literature on Causality is divised in two frameworks: Structural Causal Models (SCM) and Potential Outcome (PO). In this document we chose to focus on the first one. Before going deeper in the causal theory, we have to introduce a bit of graph knowledge:

2.2.1 Graph Background

We introduce a bit of Graph theory that will be useful when dealing with the causal theory.

Definition 2.2.1 *Given X a set of vertices (or nodes) and V a set of edges, a graph $G = (X, V)$ is a collection of nodes connected by edges. If the edges of G are directed we call G a directed graph, otherwise we call it an undirected graph.*

Definition 2.2.2 *We call two nodes adjacent if they are connected by an edge. The node at the tail of the edge is called a parent of the other node. Respectively the other node is called a child. Let $X_i \in X$ we denote the set of its parents by pa_i or $pa(X_i)$.*

Definition 2.2.3 *A path is a sequence of adjacent nodes. A directed path is a path where all edges are directed in the same direction. If a directed path connect two nodes then the one at the beginning of the path is called ancestor of the other one, respectively the one at the end is called descendant.*

Definition 2.2.4 *A Directed Acyclic Graph is a directed graph with no cycles between variables, i.e. that no variables is at the same time a start and an end of a directed path.*

Definition 2.2.5 *(Causal Graph) A Causal Graph is a DAG with the following assumptions:*

- *(Local Markov Assumption) A node is independent of all its non-descendant given its parents.*
- *Adjacent nodes are dependent*
- *A parent is a direct caus of all its children*

Given the previous assumptions we have the Bayesian Factorization:
$$\text{Let } X = \{x_1, ..., x_n\} \ , \ P(x_1, ..., x_n) = \prod_i p(x_i|pa_i)$$

2.2.2 Structural Causal Models Framework

A SCM is a set of equations defining endogenous variables as a function of other variables: i.e.

Definition 2.2.6 *SCM: Given a Directed Acyclic Graph G, a set of endogenous variables X, as set of exogenous variables U, a SCM defines the endogenous variables as a function of other variables: i.e.*

$$\forall X_i \in X \quad X_i := f(pa_i, U_i) \tag{5}$$

Where pa_i is the set of parent variables of X_i and U_i an exogenous variable that is a cause of X_i. A SCM describe a causal graph that could be of use for certain causal fairness notions.

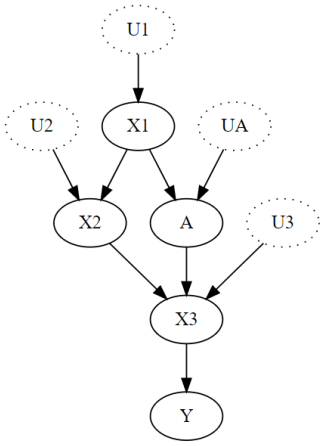
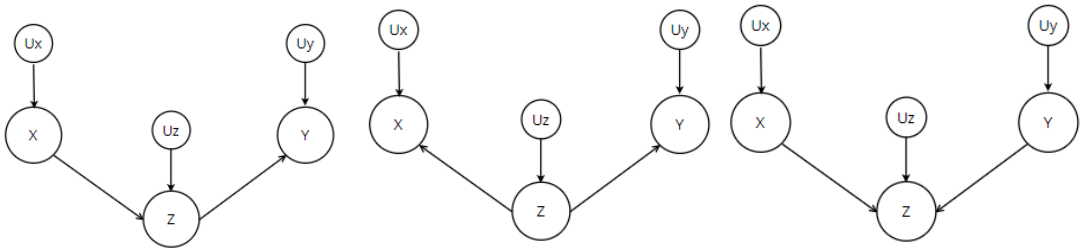


Figure 1: A DAG, nodes in plain line are endogenous variables while those in dotted line are exogenous

In causal graphs we denote three types of different structures with different properties, association and causal flows. Figure 2 shows those three structures. We note by Ux, Uy and Uz the unobserved influences on variables respective variables. The causation and association flows differently depending on them:

- For a Chain and a Fork, as in Figure 2a and Figure 2b, we say that X and Y are associated or likely dependent. Things change when conditioning on Z. When conditioning on Z the variable X and Y are conditionaly independent.

- For a Collider, as in Figure 2c, X and Y are independent. Again things change when conditioning on Z. When doing so X and Y become associated or likely dependent.



(a) A Chain (b) A Fork (c) A Collider

Figure 2: The three main structures that could be found in a Causal Graph.

Understanding the causation and association flow in those structures is important because it permits to introduce one of the main criterion allowing one to identify the causal effect in the Strucutal Causal Models framework: The Backdoor Criterion. Before introducing it we have to define what is a blocked path and d-separation:

Definition 2.2.7 *(Blocked Path) We say that the path between X and Y is blocked conditioning on a set of variable Z if:*

- *There is a Fork or a Chain in the path and the center of the Fork/Chain belongs to Z.*
- *There is a collider in the path and nor the center nor the center's descendants belongs to Z.*

Basically as conditioning of the center of a Fork and a Chain blocks the association flows, if a path contains a Chain or a Fork by conditioning on its center we block the association flow. For the collider not conditioning on the center nor the descendants of the center gives us the same effect.

Definition 2.2.8 *(d-separation) We say two sets of nodes X and Y d-separated by a third Z if all paths between any nodes of each set X and Y are blocked conditioning on Z.*

Note that Z can be empty and that X and Y should be pairwise disjoints. Thanks to those definitions we can now introducte the Backdoor Criterion:

Definition 2.2.9 *(Backdoor Criterion) A backdoor path between X and Y is a path which contains an arrow into X.*

Given a pair of nodes (A, Y) we say that a set of variables X satisfies the Backdoor Criterion iff X blocks all the backdoor paths between A and Y and no node in X is a descendant of A.

The set that satisfies the backdoor criterion is a sufficient set. It is a sufficient condition to identify the causal effects. Pearl introduced the principle of intervention, that is often used in the following, being very close to another concept. Therefor it is important to understand what is the interventional distribution:

Definition 2.2.10 *(Interventional Distribution)*
The probability $P(Y = y|do(A = a))$ is obtained after re-running the modified generation process (generally the SCM) where we set $A = a$. Intuitively it is the distribution obtained after setting the sensitive parameter to the value a to the whole population.

The interventional density is different from the one obtained by conditioning on the sensitive parameter: when conditioning on the sensitive parameter we look at the part of the population with the corresponding sensitive parameter value whereas the interventional distribution is the density we get after intervening on the sensitive parameter for every individual.

Literature provided a great number of Causal Fairness Notions, we encourage the reader to have a look to [11] as it well summarize a great number of it. In the following of this document we will talk in general about Causal Fairness Notions that needs a graph to be used through the example of Counterfactual Fairness [9]. While we do know about the ethical issues that come along with it as described in [5][4][15][7], it does not remove anything to the following discussion.

Definition 2.2.11 (Counterfactual Fairness [9]) *A predictor $\hat{Y}$ of Y is said to be counterfactually fair given a sensitive attribute $A = a$ and any observed variables (endogenous variables) X if*

$$P(\hat{Y}_{A\leftarrow a} = y|X = x, A = a) = P(\hat{Y}_{A\leftarrow a'} = y|X = x, A = a) \tag{6}$$

for all y and $a' \neq a$

Where $\hat{Y}_{A\leftarrow a'}$ stands for the value of the prediction had the sensitive parameter A of the instance had been a' and all things not depending on A being kept fixed. This concept has the advantage to fit with the intuition: a prediction model is fair with respect to Counterfactual Fairness if the prediction of an factual (observed) individual and its counterfactual individual is equal. A Counterfactual Individual is the individual that we would have observed in the same world, if a variable was intervened on and set to another value, all other things that do not depend on it being kept as we observed it. That is to say, for instance, "would have I been hired had I been of the opposite gender, in the same world all other things being fixed ?". It is very close to the concept of intervetion, the only differences is that we keep every thing that do not depend on the sensitive parameter fixed to compute the counterfactual instance.

To get the counterfactual instance and counterfactual prediction, one has to use the 3 steps process described by Pearl [12].
Given a SCM as in equation 5.

- Abduction: get the probability $P(U = u)$ thanks to the probability $P(U = u|X = x, A = a)$
- Action: intervene on the sensitive attribute A to another value a' and update the SCM with the corresponding value of A
- Prediction: predict the outcome $\hat{Y}$ using the SCM and the updated probability $P(U|X = x, A = a)$

[10]'s Appendix provides an example of the computation of a counterfactual prediction.

To ensure Counterfactual Fairness, one has to know the causal graph, i.e. the underlying causal data generation process, that is in general impossible to obtain. It is important to note the dependency of this concept to a causal graph. Given a graph it is in general possible to ensure Counterfactual Fairness with respect to the graph. That is to say, that this concept is relative to it. In practice we will aim at discovering the true graph by an expert comitee or by a causal discovery step. But what if one comes up with two different causal graphs ensuring Counterfactual Fairness for a same task? Which one is really fairer? We further describe this issue in the following.

2.2.3 Potential Outcome Framework

While not being the one on which we decided to focus we provide a bit of background on what is the potential outcome framework and how it differs from the SCM framework. The main advantage of the Potential Outcome Framework is that one do not need the complete knowledge of the causal graph to determine the causal effect of a variable on another. This come with some withdraws too: the causal effect determined is global and we cannot have a fine grained view of the different causal path that lead to the global effect observed. Indeed, there is no way to know if the causal effect obtained came from direct or indirect path. This could be useful when dealing about fairness because one could argue that direct path between a sensitive parameter and a outcome could be discrimatory while arguing that indirect path could be acceptable. Potential Outcome Framework takes its source from medical studies where we want to indentify the causal effect of a treatment, thus the sensitive parameter A is often refered as treatment.

Mainly, when using the Potential Outcome Framework, you want to estimate the Individual Treatment Effect:

$$\tau := Y_{i,A\leftarrow 1} - Y_{i,A\leftarrow 0} \tag{7}$$

Where i refer to an individual. We call $Y_{i,A\leftarrow 0}$ the potential outcome under a specific treatment value, i.e. under a specific value of a sensitive parameter. It is the counterfactual value of Y had the sensitive parameter been 0.

Following the ITE, ATE is defined as the expectation of all the ITEs over the whole population:

$$ATE := E[Y_{A\leftarrow 1} - Y_{A\leftarrow 0}] \tag{8}$$

While being of interest, ITE and ATE are difficul to compute with given observable data as we are not able to observe the counterfactual values. This is known as the fundamental problem of causal inference.

However under some assumptions the POF makes the estimation of ITE and ATE easier:

- Unconfoundedness: The potential outcome is independent of the observed treatment (sensitive parameter) given the set of confounding variables. Therefor it posits that all confounding variables are observable.
- Positivity: For any value of the confounding variables, for any value of the treatment, the probability to get this form of treatment is strictly positive.
- Stable Unit-Tretment Value Assumption (SUTVA):
  - No-Interference: The outcome of an individual is not influenced by the treatment of the others.
  - Consistency: If the observed treatment is $A = a$ then Y is the potential outcome under $A = a$, i.e. $Y_{A\leftarrow a} = Y$. Also formulated as $Y = Y_{A\leftarrow 1}A + (1 - A)Y_{A\leftarrow 0}$ when A is binary.

To explain those four assumptions we will take an example: assume happiness (Y) is caused whether someone offer you a drink or not (A) and by the color of your shoes (X). The color of your shoes could be black or orange. A is the treatment, X the covariate (a confounding of A and Y) and Y the outcome. While potential outcome framework does not require the use of the complete graph to be used, we will provide one for illustration purposes. The graph of the situation is the following:

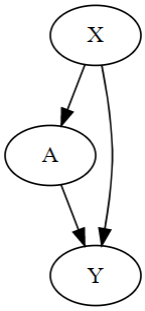


Figure 3: The graph of the situation

- Unconfoundedness: There should no unobserved variables that is a common cause of "be offered a drink or not" and "be happy or not". "Be offered a drink or not" is randomly assigned when conditioning on X.

- Positivity: instead of explaining what is a situation where positivity is achieved, we will explain what is a situation without positivity for explaination purposes. No-positivity states that every people with orange shoes is offered a drink while no-one with black shoes is offered one. Thus, is it because the shoes are black or because you are not offered a drink that you are no happy ?

- No-Interference: The fact that people are offered a drink or not doesn't influence one happiness.

- Consistency: Assume X well chosen. If an individual is offered a drink, then he will be happy no matter what the drink is.

It is important to note that those assumptions are very difficult to ensure at a point that it will mostly not be the case. Moreover in order to increase the chance to reach the Unconfoundedness assumption you have to conditionate on as many as confounding variables as possible. Doing that the set of confounding variables will get bigget and it will be nearly impossible to ensure the positivity assumption. Reverse being also true. It is known as the Positity-Unconfoundedness Tradeoff.

Under those four assumptions, the following simplify ATE:

$$\text{Adjustement Formula: } ATE = E_X[E_Y[Y|A=1, X=x] - E_Y[Y|A=0, X=x]] \tag{9}$$

The assumptions described under the Potential Outcome Framework can also be found in the SCM framework as both relies on the same theory:

- No-Interference: It is often implicit in causal graph.

- Consistency: follows from the axioms of SCM

- Unconfoundedness: it is ensured if the Backdoor Criterion is satisfied

- Positivity: it is an assumption that has to be ensured

As said before , Potential Outcome Framewok was not the one on which we decided to focus. A significant number of notions come from it, it could be of interest to evaluate their relevance.

2.3 Related Work

[8] evaluated the sensitivity of counterfactual fairness to unmeasured confounder, that is an unobserved variable that is a common cause of the sensititve parameter and the output variable. This approach is close to ours as it explores the relevance of counterfactual hypothesis. We differ in the point that our approach permits not only to explore the sensitivity of the notion with respect to unmeasured confounder but enables a great number of experimentations that we believe necessary to the use of Counterfactual Fairness on real cases. [14] described an algorithm that

learns a classifier minimizing Counterfactual Fairness across various graph, our approach is complementary: we could test the impact of the discovery step on the counterfactual fairness for instance, and assess the relevance of using discovered graph as a batch of input for their algorithm. The possibilities given by such a tool are wide.

3 The use of Causal Methods on Real Cases: The Measuring Problem

3.1 The German Credit Risk Dataset

The German Credit is often used as a standard example in the ethical machine learning domain. It is composed of 20 explanatory variables and 1000 records, and is exploited in fair machine learning to evaluate and mitigate the bias caused by two sensible parameters: Age and Sex. Despite knowing about the incorrectness of The German Credit Dataset as discussed in [3], we will use it as an example as it doesn't remove anything to our arguement. Discovering the data generation graph is a tricky process that can vary depending on various parameters: the type of graph discovery algorithm, the parameters used with respect to it, the pre-processing steps, the dataset, etc. Each combination of those parameters can lead to a different discovered graph, with very different edges. In our experiments on German, using two diferent algorithms PC and GES, it lead to two very different graphs. Even modifying the Conditional Independence Test was sufficient to discover another unique one. Figure 4 is an example of our statement.

(a) Graph discovered with GES

(b) Graph discovered with PC

Figure 4: Two graph discovered on German Credit Dataset using the python package causal-learn [2]

Once discovered, the graph provides information about the causal relations between each variable that we can use to ensure fairness, or at least to mitigate the bias. As said before, the discovery step is unperfect. Not only the discovevered graph comports wrongly discovered and directed edges but it could also retrieve

[2]https://github.com/cmu-phil/causal-learn

partially the true graph corresponding to the data generation process. Facing the wrongly directed edges, one could try to direct them himself by using common sense or with the help of an expert comitee. However, even such methods could lead to differents graphs: perception of a situation vary among individuals and it is not likely to get a consensus on the graph of a situation. Because a significant number of causal methods inherently use the discovered causal graph, determining which graph better represent the reality is critical. Actually, it is even worse when considering counterfactuals values. Counterfactuals are computing with respect to the graph discovered and, as the real graph never can be obtained, it is impossible to measure to what extent the computed counterfactuals instances are distant to the real ones. That is the case for Counterfactual Fairness, leading to what we called the Measuring Problem.

3.2 The Measuring Problem

Lets focus on the Counterfactual Fairness. Counterfactual Fairness is graph dependent: to evalutate to what extent a model respects the criterion we have to compare the output distribution of the model on the different counterfactual worlds that are computed with a discovered graph. Moreover, when designing a Counterfactually Fair model by construction we have to use a set of equations that describe the data generation process, or at least a graph, which is often done by using a discovered graph. One way to obtain a model that ensures Counterfactual Fairness by construction is to select only non descendant variables of the sensitive parameter in the discovered graph. Obtaining different graphs means differents sets of non-descendants of the sensitive parameter and thus "two" counterfactual fairness for a same situations with no possibility to tell which one is closer to the one we would get using the real one. Indeed, comparing the output distribution over the counterfactual worlds makes no sense because the counterfactual instances are computed using the same discovered graph that was used to ensure the fairness notion by construction.

One possible first way could be as [13], [2] to use statistical measures to compare the output distribution of the two models between the sensitive group and the non sensitive group and choose the one that better reduce the distance between those two densities. Actually, some statistical notions compare the label distribution Y coming from the dataset and the prediction distribution $\hat{Y}$. In out view, it seems problematic because the distribution of Y could be biased: the observation process leading to the observed dataset is unknown meaning that we do not know if it has been biased with respect to what we refer to the Real World Data Distribution, that is the global distribution from which the dataset was sample and that is generally not observed. The value of the notion is no more than a measure of the ability of the model to preserve the bias oberved in the dataset. [16] refered to them as bias preserving metrics.

Some other statistical notions such as Statistical Parity or Conditional Statistical Parity do not make such comparison. However as said in [9] supplementary material, there is no reason to think that a model counterfactually fair respects

the Statistical Parity. To the best of our knowledge and our experiments, there is no proof that a counterfactually fair model ensures any statistical fairness notions.

Hence the Measuring Problem, when considering the Counterfactual Fairness notions, two different graphs could lead to two differently counterfactually fair model with no one being able to tell which one is closer to the one we would get using the real graph.

4 Causal Generator Tool

Following from what we discussed before we designed an experimental workflow to explore the impact of the causal discovery step on the afterward fairness obtained in the output distribution. This workflow leverage the Measuring Problem by permitting to explore the behavior of the different causal methods with respect to the modelling process chosen on a large variety of situations.

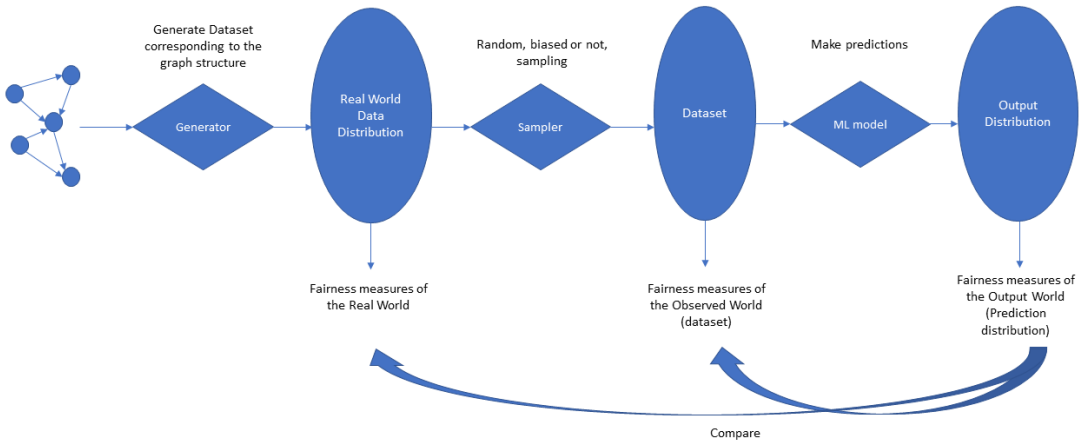


Figure 5: The experimental workflow: Takes a graph structure as input. Data is then generated according to it. An observation process samples a possibly biased dataset from the generated real world data distribution. A causal fairness notions is applied with respect to the dataset and after a discovery step. At the end predictions are made by the machine learning algorithm. Fairness is measured at each step of the process

Figure 5 shows the experimental process. We generate what we call a Real World Data Distribution corresponding to an input graph. Weights of each arrows, i.e. the influence of a variable on another, can be modified manualy. We then define an observation process that can produce a biased (or not) sample from the Real World Distribution that we call the Observed World. By applying causal discovery steps we can then apply a causal notion and feed a Machine Learning

algorithm with the afterward modified data. As we control the whole process we can therefore measure fairness at each step of the workflow and evaluate the real impact of such modeling on fairness. Pictures of the tool are given in Appendix. Such an experimental environment is beneficial for several reasons:

- It enables to explore various experimental settings and situations such as: presence of unmeasured confounders, small datasets, undistinguishable causal relations due to very low causal strength, when variables are a cause and a consequence of another, etc.
- Fairness can be evaluated more precisely and more insights are obtained to compare which method/modeling is more suitable to one case than another.
- It permits to test causal discovery algorithms on various graph structures, data samples, and causal strength between variables

4.1 Data Generation

To generate data according to a graph structure we used two approaches: the following process described in the Chapter 8.1 of [1] and one using conditional probabilities. Each variable is considered as binary. Both process give the same results but each is more suitable to certain usecases:

- The first one is more suitable when considering fast and straightforward experimentations. It also permits to compute counterfactual values as the model structure is known
- The second approach enables finer grained experiments by controlling each conditional probabilities manually

4.1.1 First approach

[1] used this formula to calculate a conditional probability:

$$p(y = 1|x_1, ..., x_m) = \sigma(w_0 + \sum_{i=1}^{M} w_i x_i) \tag{10}$$

Where $\sigma(a) = (1 + exp(-a))^{-1}$ is the logisitc sigmoid and $x = (x_0, ..., x_M)^T$ is a vector of parent states. This formula gives a straightforward method to generate data: considering that each variables is binary, variables that are not consequences of any other one could be generated with a Bernoulli Law of a manually specified parameter p while every other variables could be generated with a Bernoulli Law of parameter p' where $p' = \sigma(w_0 + \sum_{i=1}^{M} w_i x_i)$
The following example describe this process:

Here the data generation process could be described as follows:

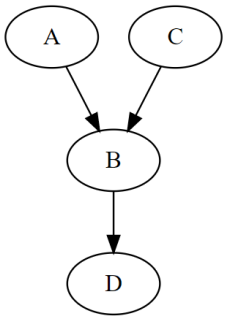


Figure 6: An example of a graph

$$
\begin{aligned}
&A \sim Bern(0.5)\\
&C \sim Bern(0.3)\\
&B \sim Bern(\sigma(w_{B,0} + w_{B,1}A + w_{B,2}C))\\
&D \sim Bern(\sigma(w_{D,0} + w_{D,1}B))
\end{aligned}
$$

That is to say that a value for A and C is generated following the Bernoulli law. Then the value of $p_1 = \sigma(w_{B,0} + w_{B,1}A + w_{B,2}C)$ is computed given the precedent values obtained for A and C. The obtained p_1 is used as the parameter for the Bernoulli law to generate the value of B. We proceed the same way for D.

4.1.2 Conditonal Probability Approach

Data created by a causal graph should verify two properties. The first one is the Bayesian Factorization Formula:

Definition 4.1.1 *Bayesian Factorization Formula: Given a Directed Acyclic Graph G, and a probability density P:*

$$P(x_1, ..., x_n) = \prod_i P(x_i|pa_i) \tag{11}$$

Where the vector $(x_1, ..., x_n)$ design a state of the variables in the Graph

The other one is the Law of total probability applied to each node of the Graph:

Definition 4.1.2 *Given a DAG G and X the set of its nodes,*

$$\forall X_i \in X \quad \sum_j P(X_i = x_j|pa_i) = 1 \tag{12}$$

Basically, applied to binary variables and the variable D of Figure 6 that means: $P(D = 0|B = 0) + P(D = 1|B = 0) = 1$ and $P(D = 0|B = 1) + P(D = 1|B = 1) = 1$

With those two properties one can set manually every Conditonal Probability of a given graph and then generate data according to them.

5 Experiments

5.1 Automatic Counterfactual Processing

In the following we describe an experimental process using specific graph and a specific adjency matrix for understanding purposes. Note that this setting was tried for several other random graphs, the results are similar and given in the following.

5.1.1 Experiment settings

We used the following adjency matrix that describes the graph of Figure 7:

$$A_{9\times 9} = \begin{bmatrix} 0 & 0 & -6 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & -5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & -3 & -4 \\ 0 & 0 & 0 & -7 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Where $A_{i,j} \neq 0$ signifies a causal influence of the variable i on the $j-th$ one. The sensitive parameter is assumed to be x_0 and the variable to be predicted is assumed to be x_8. We generated the Real World Data Distribution by using the process described in the sections before with each root variable following a Bernoulli Law of parameter 0.5. 100,000 instances were created as well as their counterfactual value with respect to the sensible parameter.

Having the Real World Data Distribution we extracted a 10,000 long dataset. The latter was biased by ensuring that $P(x_8 = 1|x_0 = 1)$ is equal to 0.7. Then we used a Discovery Step to get an experimental discovered graph from this biased dataset as in Figure 8.

Note that the edges $x_3 - x_6$ and $x_1 - x_9$ do not provide any information on the direction of the causal relation. We handled double causation by constructing 4 graphs corresponding to every combinations of the problematic causal relations: i.e. one with $x_3 \rightarrow x_6$ and $x_1 \rightarrow x_9$, another one with $x_3 \leftarrow x_6$ and $x_1 \rightarrow x_9$, etc. We do know that this method is highly computationaly expensive: we create about 2^n different graphs where n is the number of double causation relations, however we consider it to be enough as a first step. For each combination of causal relation direction we check the un-cyclicity of the obtained graph and remove them otherwise. We then splitted the dataset in train/test parts with ratio 80/20. We used the approach described in [14] to train a "Counterfactually Fair" classifier. The purpose of this method is to train a model so that it reduces the counterfactual inequalities accross different causal models. First we infered a SCM for each graph, we postulated it to be Linear as primary approximation. For each SCM

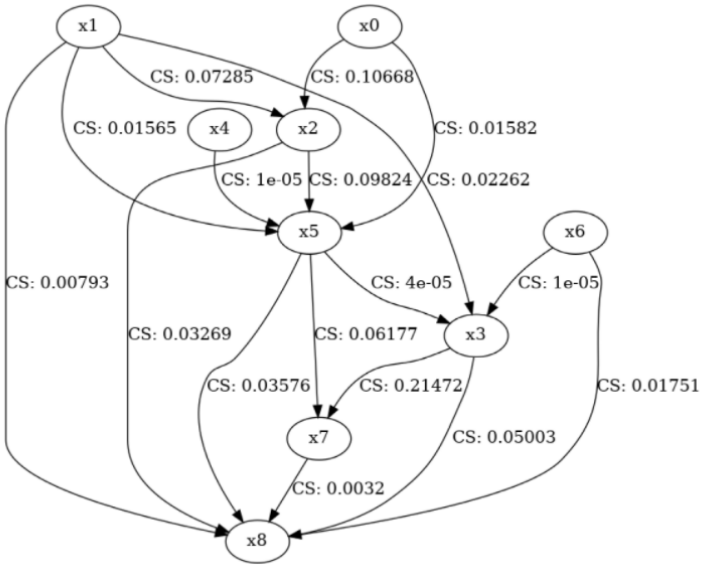


Figure 7: The graph used for the data generation process. CS stands for Causal Strength

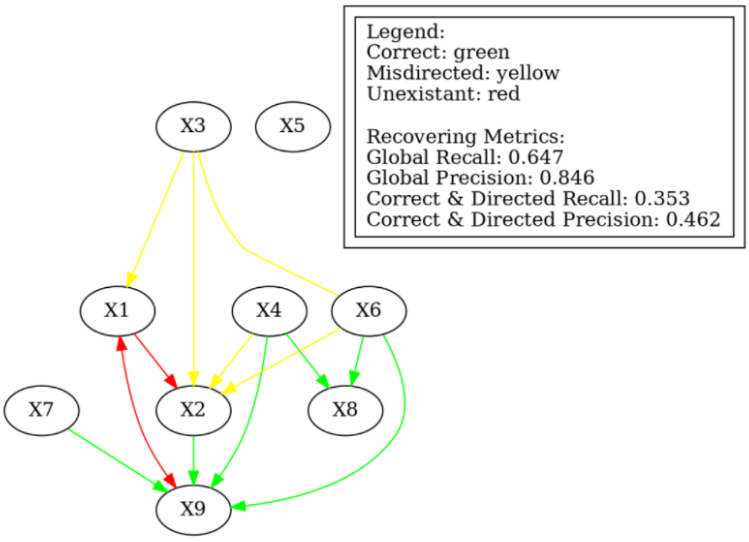


Figure 8: The graph discovered using PC Algorithm, note that variables are starting with x_1 rather than x_0 here. Knowing the real graph we are able to provide straight forward graph comparison utilities. The green edges are the edges that are correctly discovered and directed. The yellow ones those that are correctly discovered and wrongly directed. Finally, the red ones are edges that do not exist in the real graph.

we generated the counterfactual value of each instance of the dataset. We used a two layer Neural Network as binary classifier that we trained according to the

following loss:

$$\mathcal{L} := \frac{1}{n}\sum_{i=1}^{n} l(f(x_i, a_i), y_i) + \lambda \sum_{j=1}^{m} \frac{1}{n} \sum_{i=1}^{n} \sum_{a' \neq a_i} \mu_j(f, x_i, a_i, a') \quad (13)$$

$$\text{where:} \quad \mu_j(f, x_i, a_i, a') = max(0, |f(x_{i,A^j \leftarrow a_i}, a_i) - f(x_{i,A^j \leftarrow a'}, a')| - \epsilon) \quad (14)$$

x_i stands for the i-th instance variables,a_i for its sensible parameter, y_i for its label, f for the classifier, m is the number of causal graphs and n the number of instances in the dataset Mainly the first term of the loss is a classical loss that aims at ensuring that the model learns to make accurate predictions. The second one aims at penalizing the model so that it respects the "Counterfactual Fairness" notions for every graphs. Thus we have two hyperparameters to choose and fit : λ and ϵ. ϵ corresponds to the relaxation of the counterfactual fairness notions, i.e. to which extent do we consider the predictor to be counterfactually fair. λ is a coefficient managing the weight of the counterfactual penalty on the Neural Network. We decided to set ϵ to 0.1 and to use the Binary Cross Entropy for the first term of the loss. We used 200 epochs to train our model with a learning rate of 0.1. We tracked the performance of our algorithm in terms of prediction score using Precision and Recall. In terms of fairness we defined the following metric:

$$CounterfactualFairnessAccuracy := \frac{\#\{X | \hat{Y}_{A \leftarrow 0}(X) = \hat{Y}_{A \leftarrow 1}(X), X \in RWDD\}}{\#\{RWDD\}} \quad (15)$$

Where RWDD stands for Real World Data Distribution and # for the cardinality. Basically the meaning of such metric is to track the percentage of instances whose prediction is the same accross both real counterfactual worlds of the Real World Data Distribution, i.e. those created with the real graph generated at the beginning. Knowing the real counterfactuals we can now assess the real counterfactual fairness performance of a model that aims to be counterfactually fair with respect to the graphs discovered.

A plan of the complete experimental process using our tool is given in Appendix. The different steps used to create an automatic counterfactual processing are summed-up in the following algorithm:

When increasing the number of nodes one of the first reachable limits is an increasing number of double causation edges being discovered. This is problematic because the number of graph generated then is tremendous, leading to a dramatic computation time. One approach to considerate is the reduction of the number of graph used. This could be done by clustering the adjency matrixes and keeping those that are closer to the centroids, thus reducing the number of graphs to the number of clusters. While permitting to overcome this first limitation, a increasing number of nodes still leads to a dramatic number of graph being generated when handling the double causation edges.

Another more promising approach is to use a Graph Condensation method. To explain a bit more what graph condensation is we have to introduce some notions:

Algorithm 1 Automatic Counterfactual Processing

Input: D dataset

Discover a graph G using a Causal Discovery Algorithm (e.g. PC)
Let M be the set of causal models
$M = \{G\}$
while $\longleftrightarrow$ edge or — edge in any m in M **do**
  for all m in M such that m contains at least a $\longleftrightarrow$ or — edge **do**
    Remove m from M
    $m_1 \leftarrow m$, $m_2 \leftarrow m$
    Select a $\longleftrightarrow$ or — edge e
    Replace e by $\longleftarrow$ in m_1 and by $\longrightarrow$ in m_2
    Add m_1 and m_2 to M
  end for
end while
for all m in M **do**
  if m is acyclic **then**
    Infer SCM_m
    Compute Counterfactuals w.r.t. SCM_m
  else
    Remove m from M
  end if
end for
Train a model using (13), the dataset and the $\#M$ sets of counterfactuals

Definition 5.1.1 *A directed graph or subgraph is said strongly connected if there is a path between each pair of nodes in each direction.*

Thus having cycles between variables means that those variables are strongly connected.

Definition 5.1.2 *A strongly connected component of a directed Graph is a subgraph that is strongly connected and which cannot be extended by adding nodes or edges without breaking the strongly connected property.*

When increasing the number of nodes, some discovered variables can be found to be linked by double causation edges or more generally by cycles. Those subgraphs of variables being strongly connected are called strongly connected components. The purpose of the graph condensation method is to reduce those strongly connected components to one variable. The resulting graph is a directed acyclic graph that we can then feed to a machine learning model. The main problem of this method is how can we aggregate those variables while keeping as much causal information as possible ? This question is yet to be answered. However we provide a first step approach: we do not aggregate the variables but we instead keep one of the variable as a representative of the strongly connected component. The following algorithm describe the whole process:

Algorithm 2 Automatic Counterfactual Processing by Graph Condensation

Input: D dataset

Discover a graph G using a Causal Discovery Algorithm (e.g. PC)
Condensate G to obtain G'
for all Condensated variable v' in G' **do**
  Replace v' by a variable v from the strongly connected component in G that was condensated to obtain v' in G'
end for
Extract variables of G' from D to obtain D'
Infer $SCM_{G'}$
Compute Counterfactuals w.r.t. $SCM_{G'}$
Train a model using (13), D' and the set of counterfactuals

5.1.2 Results

We re-run the process several times with the same adjency matrix and the same steps described above in the algorithm 1. Results of several trajectories for the graph described before are provided in Appendix. For each we trained the model several times incrementing lambda and tracking the performances of the model using the metrics given above. Except for 12a we increased lambda up to the point where the model became a trivial classifier, i.e. when the model does not perform better than random. 12a corresponds to the situation where the discovered graph

is in fact the real graph. We observe an increase of the Counterfactual Fairness Accuracy: adding the counterfactual penalty term in the loss generally increased the Counterfactual Fairness Accuracy on our experiment. On our experiment increasing lambda does not helped the model to be more fair in the sense of Counterfactual Fairness in a majority of cases. Worse, 12e tends to show that on this experiment the use of the penalty term actually decreased the Counterfactual Fairness Accuracy, that may be due to discovered graphs leading to counterfactual values diferent enough of the real counterfactuals. More experiments are required to evalutate in which contexts adding the penalty term worsen the fairness. We also observe that the Precision plummets from a certain value of lambda while the Counterfactual Fairness Accuracy skyrockets: in our experiences it corresponds to the moment where the penalty is so important that the classifier is unable to learn anything. The trade off Precision/Fairness can also be seen.

As said before, the experimental process was used with 4 different and random situations, that is to say 4 different true graphs as a total. We reproduced the same steps each time, using the graph clustering version when needed reducing the number of graphs to a sufficiently small number. Results are shown below, the first figure on the top being the results obtained when lauching the experiments a hundred time with the specific graph shown above.

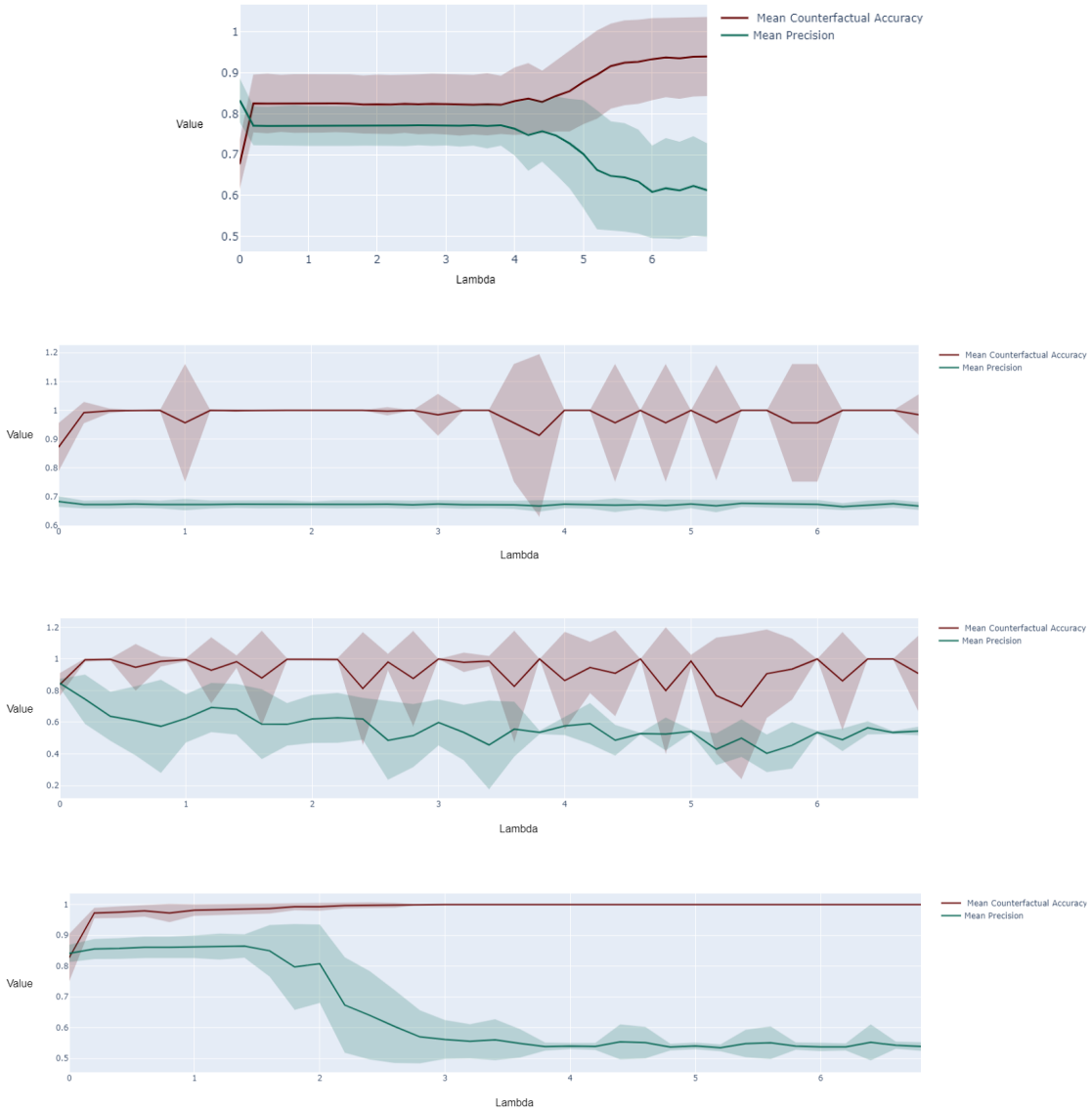


Figure 9: Plot of the curves obtained for four different situations. For each situations we re-run the whole process sufficiently to get representative statistics of the metrics behavior. The curve corresponds to the mean value obtained over the several runs and the envelops corresponds to the standard deviation. The x-axis is the value of lambda and the y-axis is a percentage.

6 Future Work

A lot of work is yet to be done. Firstly a greater number of experiments should be done using the algorithm 2. The algorithm should be tested a great number of time with a huge variety of graph size in order to get confident with the results that could be expected with this method. A first result obtained with algorithm 2 is the following:

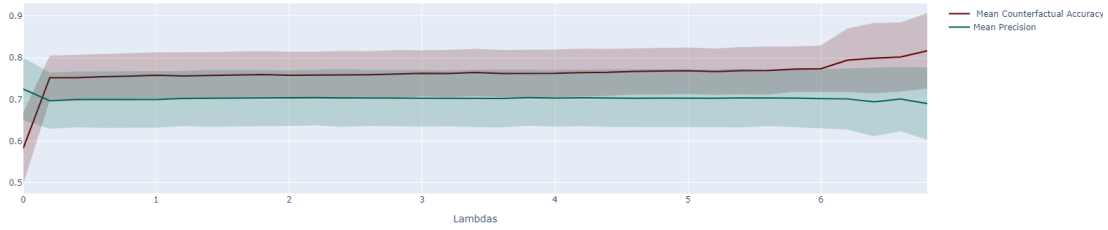


Figure 10: The results obtained using the algorithm 2

Thoses curves were obtained when dealing with a graph composed of 61 nodes. The results obtained seems to be the same as the previous algorithm but more work are needed to really assess the effectiveness of this method. More generally, causal notions should be also tested on a large variety of situations, modifying the causal influences between variables or by extracting a dataset with a lower number of variables than the Real World Data Distribution for instance.
A possible way of development could be to test to use it upside down when considering real use case. Given a dataset, one could discover a graph using a causal discovery step, then fit a SCM corresponding to the graph discovered. The SCM permits then to generate the Real World Data Distribution corresponding to it. By doing so, one could then measure to what extent does the dataset is a representative sample of the Real World Data Distribution. Thus not only could it be used to strenghten the causal discovery step, but also it could provide a simulation twin on which the measure of counterfactual is possible thus providing an experimental insight on the behavior of the methods on this use case.

Finally the tool is not complete yet. Data geneterated is only binary, it could be interesting to develop a way to test for continuous and categorical variables. Also the SCM are postulated linear along the whole process: during the generation and the estimation of the SCM. In reality SCM are not all linear depending on the situation, it could then be interesting to try modifying the structure of the equations.

7 Appendix

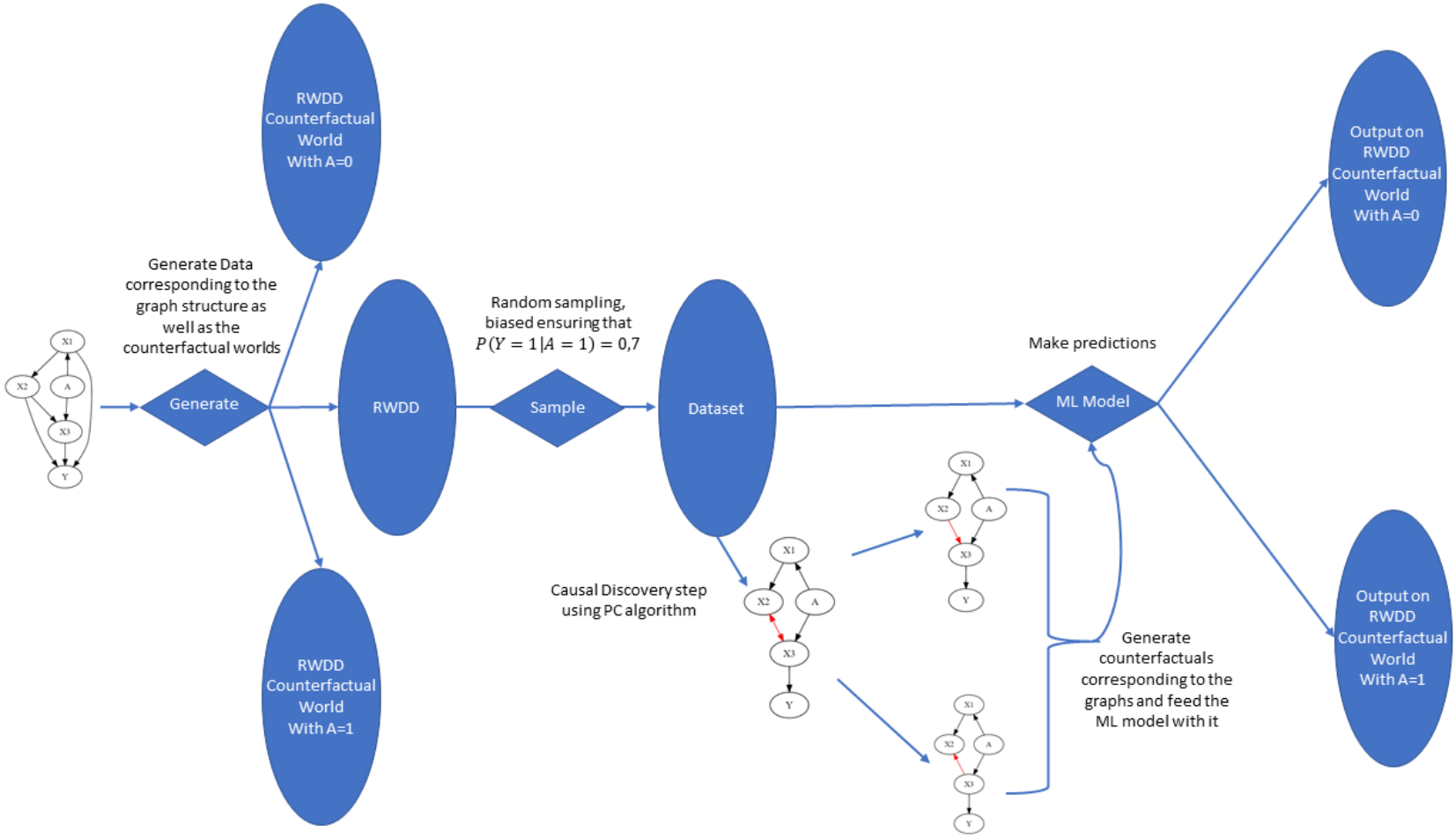


Figure 11: Experimental Process for the Automatic Counterfactual Processing experiment. We start by giving a graph to the tool, we then generate the Real World Data Distribution and its counterfactual values. We observe a sample of that distribution that we call Dataset and we use it for the experiment.

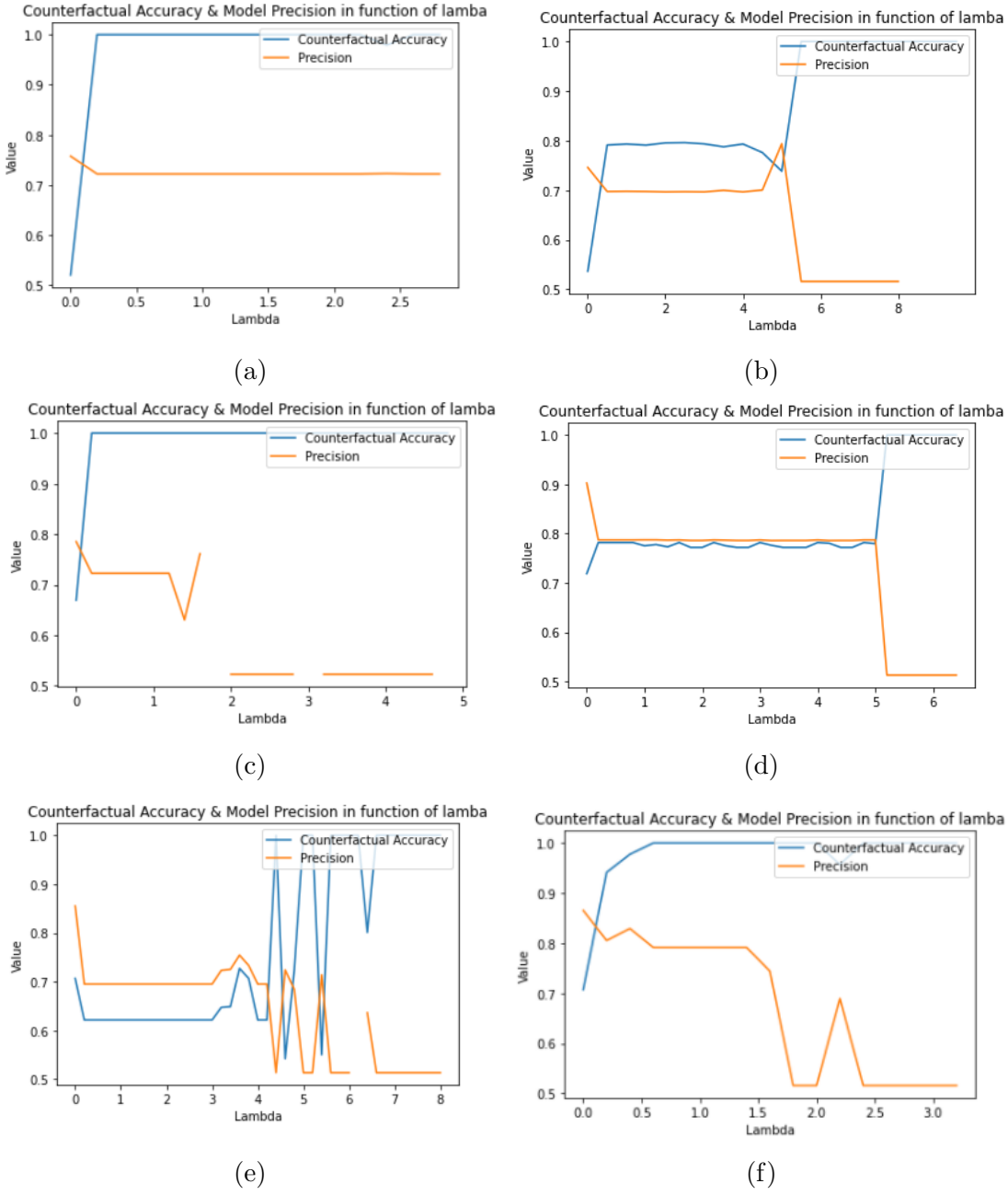


Figure 12: Plots for several run of the process with the same adjency matrix. Curve 12a corresponds to the results obtained when the discovered graph is the real graph. The discontinuities as in 12c occure when the model classifies all instances in the 0 category.

References

[1] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, Jan. 2006. URL: https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/.

[2] Alessandro Castelnovo et al. "BeFair: Addressing Fairness in the Banking Sector". In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020, pp. 3652–3661. DOI: 10.1109/BigData50022.2020.9377894.

[3] Ulrike Grömping. "South German Credit Data: Correcting a Widely Used Data Set". In: (Nov. 2019).

[4] Alex Hanna et al. "Towards a Critical Race Methodology in Algorithmic Fairness". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Jan. 2020). arXiv: 1912.03593, pp. 501–512. DOI: 10.1145/3351095.3372826. URL: http://arxiv.org/abs/1912.03593 (visited on 04/27/2022).

[5] Lily Hu and Issa Kohler-Hausmann. "What's Sex Got To Do With Fair Machine Learning?" In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Jan. 2020). arXiv: 2006.01770, pp. 513–513. DOI: 10.1145/3351095.3375674. URL: http://arxiv.org/abs/2006.01770 (visited on 04/27/2022).

[6] Anna Jobin, Marcello Ienca, and Effy Vayena. "The global landscape of AI ethics guidelines". en. In: *Nature Machine Intelligence* 1.9 (Sept. 2019), pp. 389–399. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0088-2. URL: http://www.nature.com/articles/s42256-019-0088-2 (visited on 04/27/2022).

[7] Atoosa Kasirzadeh and Andrew Smart. "The Use and Misuse of Counterfactuals in Ethical Machine Learning". In: *arXiv:2102.05085 [cs]* (Feb. 2021). arXiv: 2102.05085. URL: http://arxiv.org/abs/2102.05085 (visited on 04/27/2022).

[8] Niki Kilbertus et al. "The Sensitivity of Counterfactual Fairness to Unmeasured Confounding". In: *arXiv:1907.01040 [cs, stat]* (July 2019). arXiv: 1907.01040. URL: http://arxiv.org/abs/1907.01040 (visited on 04/27/2022).

[9] Matt J. Kusner et al. "Counterfactual Fairness". In: *arXiv:1703.06856 [cs, stat]* (Mar. 2018). arXiv: 1703.06856. URL: http://arxiv.org/abs/1703.06856 (visited on 04/27/2022).

[10] Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. "Machine learning fairness notions: Bridging the gap with real-world applications". en. In: *Information Processing & Management* 58.5 (Sept. 2021), p. 102642. ISSN: 03064573. DOI: 10.1016/j.ipm.2021.102642. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306457321001321 (visited on 04/27/2022).

[11] Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. "Survey on Causal-based Machine Learning Fairness Notions". In: *CoRR* abs/2010.09553 (2020). arXiv: 2010.09553. URL: https://arxiv.org/abs/2010.09553.

[12] Judea Pearl. *Causal inference in statistics: An Overview*. 2009.

[13] Aida Rahmattalabi and Alice Xiang. "Promises and Challenges of Causality for Ethical Machine Learning". In: *arXiv:2201.10683 [cs]* (Jan. 2022). arXiv: 2201.10683. URL: http://arxiv.org/abs/2201.10683 (visited on 04/27/2022).

[14] Chris Russell et al. "When Worlds Collide: Integrating Different Counterfactual Assumptions in Fairness". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/1271a7029c9df08643b631b02cf9e116-Paper.pdf.

[15] Maya Sen and Omar Wasow. "Race as a Bundle of Sticks: Designs that Estimate Effects of Seemingly Immutable Characteristics". en. In: *Annual Review of Political Science* 19.1 (May 2016), pp. 499–522. ISSN: 1094-2939, 1545-1577. DOI: 10.1146/annurev-polisci-032015-010015. URL: https://www.annualreviews.org/doi/10.1146/annurev-polisci-032015-010015 (visited on 04/27/2022).

[16] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Bias Preservation in Machine Learning: The Legality of Fairness Metrics Under EU Non-Discrimination Law". en. In: *SSRN Electronic Journal* (2021). ISSN: 1556-5068. DOI: 10.2139/ssrn.3792772. URL: https://www.ssrn.com/abstract=3792772 (visited on 04/27/2022).