Towards counterfactual logics for machine learning

By Paulius Skaisgiris

Supervisor: Dr. Levin Hornischer

A semester project presented for MSc Logic

Institute for Logic, Language and Computation
University of Amsterdam
The Netherlands

August 29, 2023

Abstract

This paper investigates the interplay between generated counterfactual explanations (CEs) in machine learning models and formal counterfactual logics. The connection between counterfactual logic (CL) and CEs offers the opportunity to reason formally about CEs, enhance comprehension of decision boundaries in diverse machine learning models, and interpret logic-based statements within the context of these models. This work clarifies counterfactuals in both logical and machine learning contexts, and presents an attempt to establish a variant of soundness and completeness between a counterfactual explanation generation method and a specific counterfactual semantics. While soundness is demonstrated, completeness remains elusive. The paper concludes by suggesting avenues for further research and potential strategies to attain completeness, contributing to the understanding and development of counterfactual explanations in the realm of machine learning interpretability.

1 Introduction

The main motivation of this project is to explore the connection between generated counterfactual explanations (CEs) of machine learning models and counterfactual logics. One common task of both counterfactual logic (CL) and counterfactual explanations is to shed light about alternatives to the given situation and the consequences that follow. If a version of soundness and completeness result could be demonstrated between the CL and the generated CEs, we would be able to use the well-established rules of reason from symbolic logic to inform ourselves more about the behaviour of the machine learning model. Specifically, we could reason formally about CEs, better understand the resulting decision boundary of arbitrary machine learning models, generate various statements using the deductive system of the logic and directly interpret them in the context of the machine learning model.

Machine learning (ML) is a method for modeling data - automatically finding rules that predict the behaviour of one feature of the data, given some other features of this data. While machine learning algorithms are notoriously performant in solving various real-world problems, they are also opaque black-boxes. In other words, most ML methods are essentially highly complex functions which cannot be easily investigated, "opened up" even by their engineers. ML model interpretation is thus highly desirable since ML methods are deployed in safety-critical situations [Bojarski et al., 2016] [Julian et al., 2016] or in various situations dealing directly with people [O'Neil, 2016] [Sheikh et al., 2020]. The latter is particularly relevant for this project as, according to recent General Data Protection Regulation (GDPR) laws in the European Union, users of ML systems have the right to understand their output [Kaminski, 2018]. Counterfactual explanations may help with that [Wachter et al., 2017].

The contents of this project report are as follows. First, in Section 2 we clearly define counterfactuals occurring in logic as well as counterfactuals occurring in the machine learning literature. Section 3 specifies the different counterfactual logic semantics and their application to the data context. Then, in Section 4, we describe an attempt to show soundness and completeness between the counterfactual explanation generation method and a specific counterfactual semantics. Section 5 addresses the encountered issues in the first attempt and summarizes possible remedies for the issues so that subsequent projects may tackle them. Section 6 summarizes this project.

2 Preliminaries: counterfactuals

2.1 Counterfactuals in philosophical logic

In Philosophical Logic, counterfactuals are a certain kind of conditional statement. A typical conditional statement is of the form "if φ, then ψ". Here, depending on the context, φ and ψ are events described in natural language or some variables describing a propositional claim. Counterfactual statements are sentences of the form "if φ were the case, then ψ would be the case" [Hornischer, 2022]. Counterfactuals are thus useful for formally reasoning about alternative possibilities given the current situation [Starr, 2022].

We will now introduce the language used for counterfactual logics which is an extension to propositional and modal logics. Complex formulas can be built using the vocabulary and rules that follow.

$$\varphi := \top \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi \mid \varphi \mathrel{\Box\!\!\rightarrow} \psi$$

As usual, we make use of the following abbreviations from propositional and modal logic:

- $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi)$
- $\varphi \rightarrow \psi = \neg\varphi \vee \psi$
- $\Diamond\varphi = \neg\Box\neg\varphi$
- Consequently, we also define $\varphi \mathrel{\Diamond\!\!\rightarrow} \psi = \neg(\varphi \mathrel{\Box\!\!\rightarrow} \psi)$.

Two types of counterfactual semantics have been studied the most by philosophers: strict and similarity semantics [Starr, 2022]. Both of them stem from possible world semantics, thus our point of departure will be Kripke models.

Let W be a non-empty set and let $R \subseteq W \times W$ be a binary relation on W. A Kripke frame is then $\mathcal{F} = (W, R)$. A Kripke model is $\mathcal{M} = (W, R, V)$ where V is a valuation function assigning each propositional variable a subset of W [Blackburn et al., 2001].

In strict analysis of the counterfactual modality $\Box\!\!\rightarrow$ is defined on standard Kripke models simply as $\varphi \mathrel{\Box\!\!\rightarrow} \psi = \Box(\varphi \rightarrow \psi)$. This is to mean that

$$M, w \Vdash \varphi \mathrel{\Box\!\!\rightarrow} \psi \text{ iff for all } v \text{ s.t. } wRv,\ M, v \Vdash \varphi \rightarrow \psi$$

In similarity analysis (also sometimes called *variably strict analysis*), the semantics of $\Box\!\!\rightarrow$ are a bit more subtle. Namely, $\varphi \mathrel{\Box\!\!\rightarrow} \psi$ is true on a world w just in case all the φ-worlds most similar to w are ψ-worlds [Starr, 2022]. In order to formally define similarity semantics, we need to know which worlds are similar to each other. To tackle this, we will define a similarity model $\mathcal{M} = (W, R, S, V)$ where $S : W \rightarrow \mathcal{P}(W)$

- $w \in S(w)$, since any $w \in W$ is similar to itself
- $S(w) \setminus \{w\} \subseteq R(w)$

Thus, we define the similarity relation/function in terms of the existing R. And so, in similarity models:

$$\mathcal{M}, w \Vdash \varphi \mathrel{\Box\!\!\rightarrow} \psi \text{ iff for all } v \in S(w),\ \mathcal{M}, v \Vdash \varphi \rightarrow \psi$$

Obviously, this definition is more relaxed than the strict analysis. That is, let $V_{sim} = \{w : M, w \Vdash \varphi \mathrel{\Box\!\!\rightarrow} \psi\}$, then $V_{sim} \subseteq V_{str}$ where V_{str} is the valuation function for strict semantics, and M and w are arbitrary similarity or Kripke models (as is appropriate) and world respectively.

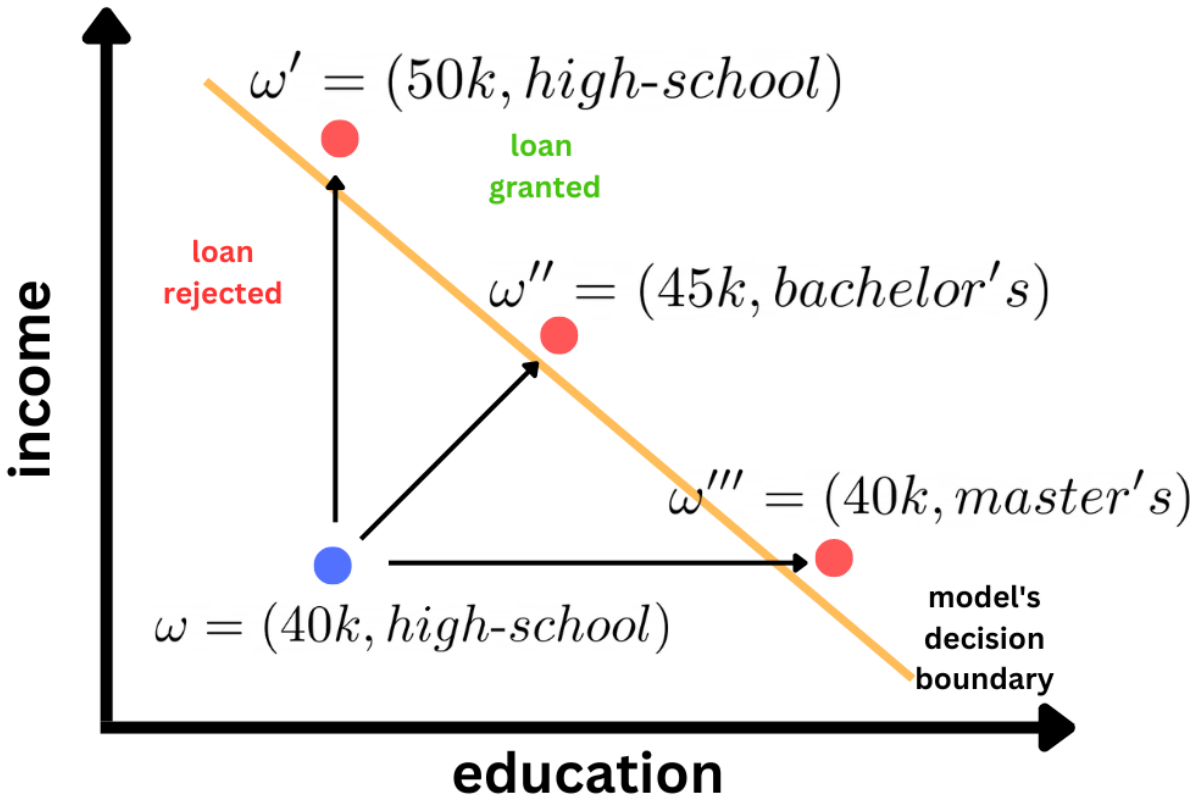

Figure 1: A simple example of a linear ML model predicting whether to give a loan or not depending on two variables: age and education. ω is the original input and was rejected the loan. However, $\omega', \omega'', \omega'''$ are similar variants of ω and were given the loan. They are also examples of possible counterfactual explanations given this ML model. ω'' is the closest point to the original ω (by Euclidean distance).

2.2 Counterfactual explanations in machine learning

As mentioned in the introduction, ML models are often difficult to understand. Owing to their strength, however, they are ubiquitous, and thus ways of interpreting or explaining the model is of great interest to various parties. One such way to explain an ML model is using counterfactuals. Similarly as in logic, these are statements that talk about alternatives, [Mothilal et al., 2020] define them as "hypothetical examples that show people how to obtain a different prediction". More concretely, counterfactual explanations in machine learning are data points which change the output of our trained ML model compared to some other reference point. CEs thus may inform us about "what could have happened" had the original situation been different as it were in the case of the counterfactual. This may provide information as to how a user may want to change their future lives / input data into the model so that an alternative outcome is achieved. Figure 1 illustrates a CE example.

The literature on counterfactual explanations and their generation is moderate. The existing approaches include finding counterfactuals using integer linear programming [Russell, 2019], growing spheres [Laugel et al., 2017], class prototypes [Looveren and Klaise, 2019], SAT solvers [Karimi et al., 2019], and determinantal point processes [Mothilal et al., 2020]. For a more broad overview, consult [Mittelstadt et al., 2018] and [Molnar, 2022, Section 9.3]. In our experiments we used the DiCE generation method [Mothilal et al., 2020], mainly because of its ease of use in Python. While we think our results in this paper should carry over to other counterfactual generation engines, concrete results of this are left for future work.

Let us demonstrate how a CE generated with the DiCE method looks like. Consider the Adult Income dataset [Becker and Kohavi, 1996] often used in CE literature. The dataset is used to train a ML model to predict whether a person is earning more than 50k per year or not based on some of their personal details. Specifically, these are the input features: age, workclass, education, marital status, occupation, race, gender, number of hours they work per week. In this example, we fit a Random Forest model provided by scikit-learn. Let us pick a concrete row of the dataset:

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	27	Private	School	Single	Blue-Collar	White	Male	40	0

As is evident, this person was predicted not to earn 50k or more (indicated by a 0 in the `income` column). Let us generate a counterfactual for this specific instance. The following CE was generated using exponential random method available in the DiCE package:

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	56.0	-	Assoc	-	-	-	-	-	1

That is, if the person was 56 years old instead of 27 and had an associate-level education instead of only a school diploma, the machine learning model would have predicted that he would earn 50k or more per year (indicated by a 1 in the `income` column).

For the purpose of this project, we will only be concerned with supervised learning tasks. That is, we have a training set of input points and, for each point in that training set, we know what the dependent variable should be, we can *supervise* the model to learn the mapping of outputs in a way presented in the dataset. In fact, we will primarily restrict ourselves to classification tasks. That is, given input variables, our ML model will try to predict to which class the point belongs to. Our analysis should be applicable to regression models (predicting a continuous numerical value) as well, but it may be a bit more cumbersome to write out counterfactual logic statements for those.

3 Counterfactual logic semantics for statements concerning data

In order to think about semantics for data, we may start with existing methods first and see whether they fit our scenario. We will soon see that the most popular methods in philosophy

are problematic especially in the data context.

3.1 Attempts with strict and similarity semantics

Let us first examine strict semantics. We must first build a Kripke model which would allow us to reason about data. Recall that in strict semantics the counterfactual statement we care about is simply $\varphi \mathrel{\square\!\!\rightarrow} \psi = \square(\varphi \rightarrow \psi)$. Data points are d-dimensional vectors, so, rather naturally, $W = \mathcal{X}^d$. Notice that in this case we do not pose a restriction on the worlds as technically any point may be a counterfactual and this choice reflects that. However, choosing appropriate relations R between the points is more problematic.

One naive approach would be connecting no points. Clearly, the counterfactual $\square(\varphi \rightarrow \psi)$ would be vacuously true evaluated on any $x \in W$. Conversely, we may connect all points together. In that case, any world at which φ is true would have to have ψ true. The counterfactual would then be a sort of global property which is clearly too strong. We would not be able to distinguish between similar and not similar points to the original point as *all* points would be similar to one another.

Certainly, a middle-ground is required, we must consider a subset of all points that are somehow close to any other point - we require a notion of similarity. In a vector space, a natural way to go about this is picking some distance metric and marking all points within some fixed radius ε as similar. In notation, this set of points is the image of similar relations, $S(\omega)$. Then, $\varphi \mathrel{\square\!\!\rightarrow} \psi$ is true at the original point ω iff for each $x \in S(\omega)$ φ is false or φ is true and ψ is true. This approach, while more nuanced than strict semantics, has a few serious flaws:

- Which distance metric to choose? See Figure 2 showing a few different distance metrics, specificallt L_p norms. Almost each metric chosen will alter the truth value of the counterfactual.
- Which radius to choose? Similarly, almost any choice of a radius will alter the truth value of the counterfactual.

The reasons above point to the fact that neither strict nor similarity semantics are powerful enough to construct Kripke models for vectors so that we could formally reason about data situations and their alternatives.

3.2 Variation Semantics

[Hudetz and Crawford, 2022] introduce novel semantics for evaluating counterfactuals that avoid the above-posed difficulties. Furthermore, similarity semantics are inadequate for evaluating ML model outputs on dependent variables which non-monotonically depend on some input variable. The novel variation semantics addresses this as well as the issue of evaluating statements involving ranges. Lastly, [Hudetz and Crawford, 2022] mention the issue of disjunctive antecedents. It may be the case that a sentence $A \vee B \mathrel{\square\!\!\rightarrow} C$ is true at a point in similarity semantics but at the same time $A \mathrel{\square\!\!\rightarrow} C$ is not true at the same

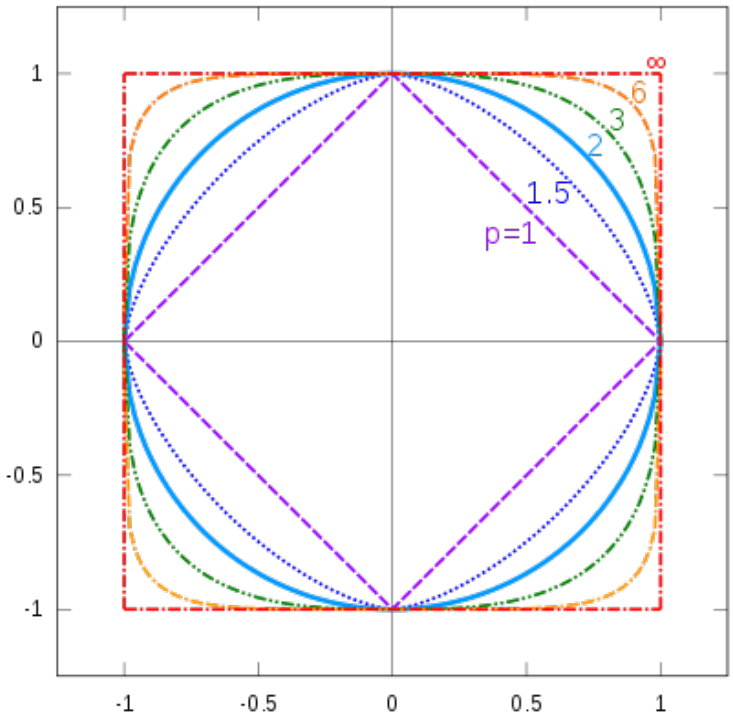

Figure 2: Different L_p norms showing different notions of distance from a single point. Quartl, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons.

point according to similarity semantics. Variation semantics fix this issue by making the first sentence either not true in the first place or entailing the second one.

The definitions that follow are taken from [Hudetz and Crawford, 2022] and describe the main formal semantic framework of this paper.

Definition 3.1. (Ω, Var) is a *frame* in variation semantics iff

1. Ω is a non-empty set (called the possible data points).
2. Var is a non-empty set of variables on Ω and partitioned into indepedent variable Var_i and dependent variables Var_d. That is, each $X \in Var$ is a function $X : \Omega \to \mathcal{V}_X$, and either $X \in Var_i$ or $X \in Var_d$.
3. Possible data points are characterised by the values variables take in them. In symbols, for all $\omega, \omega' \in \Omega, \omega = \omega'$ iff for all $X \in Var, X(\omega) = X(\omega')$.
4. Independent variables can be varied independently of each other. That is, for every independent variable $X \in Var_i$, every possible value $v \in \mathcal{V}_X$ of X and every possible data points $\omega \in \Omega$, there exists a unique possible data point ω' at which X takes the value v but all other independent varables take the same value as in ω. We denote this data point $\omega_{X:v}$.
5. The values of dependent variables are determined by the values of independent vari-

ables. That is, any possible data points $\omega, \omega' \in \Omega$ that agree regarding the values of all variables in Var_i also agree regarding the values of all variables in Var_d.

Definition 3.2. $(\Omega, Var, Rel, Prop)$ is a *model* in variation semantics iff

1. (Ω, Var) is a frame
2. *Rel* assigns a set of variables $Rel(p)$ to each atomic sentence p. That is, the set of variables relevant to the truth/falsity of p. Can usually be "read off" from the statement.
3. *Prop* assigns a set of possible data points $Prop(p)$, i.e. a proposition/intension to each atomic sentence p. That is, the set of possible data points at which the atomic sentence p is true.

Definition 3.3. For all $\omega \in \Omega$, all $V \subseteq Var$, all atomic sentences in p and all normal sentences (without counterfactual modality) A, B:

1. (a) $\omega|_V$ is a *verifier* of p iff $V = Rel(p)$ and $\omega \in Prop(p)$
   (b) $\omega|_V$ is a *falsifier* of p iff $V = Rel(p)$ and $\omega \notin Prop(p)$
2. (a) $\omega|_V$ is a verifier of $\neg A$ iff $\omega|_V$ is a falsifier of A.
   (b) $\omega|_V$ is a falsifier of $\neg A$ iff $\omega|_V$ is a verifier of A.
3. (a) $\omega|_V$ is a verifier of $A \wedge B$ iff there are $V_1, V_2 \subseteq Var$ such that $V = V_1 \cup V_2$ and $\omega|_{V_1}$ is a verifier of A and $\omega|_{V_2}$ is a verifier of B.
   (b) $\omega|_V$ is a falsifier of $A \wedge B$ iff $\omega|_V$ is a falsifier of A or $\omega|_V$ is a falsifier of B or $\omega|_V$ is a falsifier of $A \vee B$.
4. (a) $\omega|_V$ is a verifier of $A \vee B$ iff $\omega|_V$ is a verifier of A or $\omega|_V$ is a verifier of B or $\omega|_V$ is a verifier of $A \wedge B$.
   (b) $\omega|_V$ is a falsifier of $A \vee B$ iff there are $V_1, V_2 \subseteq Var$ such that $V = V_1 \cup V_2$ and $\omega|_{V_1}$ is a falsifier of A and $\omega|_{V_2}$ is a falsifier of B.

Definition 3.4. Let $X \in Var_i$ and $Y \in Var_d$. Then Y *depends on* X iff there is a possible data point $\omega \in \Omega$ and a possible value $v \in \mathcal{V}_X$ of X such that Y takes different values in $\omega_{X:v}$ and ω.

Definition 3.5. Let $V \subseteq Var$. Then B is the *basis* of V (written $Basis(V)$) iff B is the set of all independetn variables that are in V or that some variables in V depend on.

According to the authors, this semantic framework is a nuanced approach to working with counterfactual statements concerning data which is distinct from similarity as well as interventionist semantics. As [Hudetz and Crawford, 2022] put it "We explicate the idea of a variation that makes a given sentence true while leaving "other things equal" in a novel way. On our account, "leaving other things equal" does not entail "leaving all other things equal". Our explication is neither based on similarity comparisons nor on surgical interventions. Instead, it focuses on upstream changes to relevant independent variables."

Definition 3.6. Let A be a normal sentence and let $\omega, \omega' \in \Omega$. Then we say that ω' is an *A-variant* of ω iff for some variable set $V \subseteq Var$:

1. $\omega'|_V$ is a verifier of A
2. ω' agrees with ω regarding the values of all indepedent variables outside the basis of V, i.e. for all $X \in Var_i$, if $X \notin Basis(V)$, then $X(\omega') = X(\omega)$
3. If A is true at ω, then ω' agrees with ω regarding the values of all variables in the basis of V.

Definition 3.7. Let A and C be normal sentences, and let $\omega \in \Omega$.

1. A is *true at* ω iff for some variable set $V \subseteq Var$, $\omega|_V$ is a verifier of A.
2. $A \mathrel{\square\!\!\rightarrow} C$ is true at ω iff C is true at all A-variants of ω.
3. $A \mathrel{\Diamond\!\!\rightarrow} C$ is true at ω iff C is true at some A-variant of ω.

3.3 Discussion on the deductive system of variation semantics

[Hudetz and Crawford, 2022] do not introduce a deductive system for variation semantics. In fact, it still remains an open question. Because of this reason, there is no ready-made deductive system which we could try to evaluate empirically using CE generation methods.

One possible way of improvement, then, is to devise an empirical approach to gather clues as to how this deductive system should look like. Since we are primarily interested in developing a connection between empirical counterfactual explanation generation methods and formal counterfactual logics, we could try showing a sort of "soundness and completeness" for CE generation and variation semantics. By attempting to do this, and, if we fail, analysing why, we may shed light as to what deductive rules should be part of the deductive system for variation semantics. We first have to specify the notion of soundness and completeness we have in mind which we do in the next section.

4 Soundness and completeness of CE generation and variation semantics

4.1 Soundness and completeness of an algorithm

The terms *soundness* and *completeness* are most often used in mathematical logic. A logical system is *sound* if every formula that can be proved in the system is logically valid (with respect to the semantics of the system). On the contrary, a logical system is *complete* if every logically valid (with respect to the semantics of the system) formula can be derived using that system.

[Dietrich, 2012] discusses adapting these definitions for algorithms. Soundness of an algorithm means that the algorithm does not yield any results that are untrue. For instance, a

sorting algorithm that sometimes does not return a sorted list is not sound. Completeness, on the other hand, means that the algorithm addresses all possible inputs and does not miss any. For instance, if the sorting algorithm never returned an unsorted list, but simply refused to work on lists that contained the number 7, it would not be complete. An algorithm is complete and sound if it works on all inputs (semantically valid in the world of the program) and always gets the answer right.

In this paper we will draw inspiration from the two notions of soundness and completeness above and define our own (at the risk of overloading the term).

Definition 4.1. A CE generator G is *sound* if every formula that is generated by G is true on a corresponding model of variation semantics.

Definition 4.2. A CE generator G is *complete* if every formula which is true on some model of variation semantics can be generated by a corresponding generator G.

If we establish such a soundness and completeness relationship between the generator and variation semantics, whenever the deduction system for variational semantics is uncovered, we should be able to generate the conclusions of the deduction rules. That is, we will be able to gain new information about the generator without going through the process of generating.

4.2 Soundness

To show soundness, we must show that whatever statements are generated, they are true on some model of variation semantics. We will first show that there is a canonical way to translate each generated CE to a counterfactual statement using the language and vocabulary defined in Sections 2.1 and 3.2. Afterwards, we will prove that such counterfactual statements will always be true when evaluated on Variation Semantics. Thus, we will conclude that the CE generator is sound w.r.t. variation semantics. To build intuition, we will be following a concrete example throughout.

Consider some data point (without its target feature) $\omega \in \mathcal{X}^d$ from the training set *train* where $\mathcal{X}^d$ is the d-dimensional input space. Let $f : \mathcal{X}^d \to \mathcal{Y}$ be the trained machine learning model where $\mathcal{Y}$ is the output space. Let $G : \mathcal{F} \times \mathcal{X}^d \to \mathcal{X}^d$ be a counterfactual explanation generator where $\mathcal{F}$ is the set of functions $\mathcal{X}^d \to \mathcal{Y}$ or machine learning models. Let $\omega' \in G(f, \omega)$ be a counterfactual explanation for ω such that $f(\omega) \neq f(\omega')$. That is, the machine learning model does not predict the labels of the points ω and ω' to be the same.

Claim 4.3. Every generated CE ω' from the original point ω can be canonically converted to a formal counterfactual statement of variation semantics of the form $A \,\square\!\!\rightarrow C$.

Proof. Using the language of conditional logic, set A as the conjunction describing solely the altered features and their current states at the point ω' compared to ω. In symbols, $\mathcal{S} = \{X = X(\omega') : \forall X \in Var_i,\ X(\omega) \neq X(\omega')\}$ and thus set $A = \bigwedge \mathcal{S}$.

Set C as the statement describing $f(\omega')$, so $C = (Y = f(\omega'))$.

It is easy to see that both A and C are then well-formed formulas, and putting the modality $\Box\!\!\rightarrow$ inbetween them leads to a well-formed counterfactual statement. ■

Example 4.4. Let

$$\omega = (\text{age} = 27, \text{education} = \text{high-school}, \text{marital\_status} = \text{single})$$

be the original point from some data set (in this case, the Adult Income dataset with less features) and let

$$\omega' = (\mathbf{age = 50}, \mathbf{education = master's}, \text{marital\_status} = \text{single})$$

be the point generated by some CE generator G. Assume that f is the machine learning model predicting whether the input has the salary of more than 40k a year (income = 1) or not (income = 0). So, $f(\omega) = 0$ and $f(\omega') = 1$. Then, in this case $Var = \{\text{age}, \text{education}, \text{marital\_status}, \text{income}\}$. And, as per our usual assumption, the only dependent variable is income.

Then, $\mathcal{S} = \{\text{age} = 50, \text{education} = \text{master's}\}$ since these are the only two features that differ in ω' from ω and the resulting antecedent would be $A = (\text{age} = 50 \wedge \text{education} = \text{master's})$. The consequent is $C = (\mathit{income} = 1)$ (the result of $f(\omega')$). This yields us the counterfactual statement

$$(\text{age} = 50 \wedge \text{education} = \text{master's}) \Box\!\!\rightarrow (\mathit{income} = 1)$$

Theorem 4.5. *CE generator G is sound w.r.t. variation semantics. That is, every counterfactual explanation ω' generated by G and afterwards converted to a statement CF statement $A \Box\!\!\rightarrow C$ as per Claim 4.3 is true at ω on variation semantics.*

Proof. Assume ω is an arbitrary possible data point and ω' is a counterfactual explanation generated by a generation procedure G. Further, assume $A \Box\!\!\rightarrow C$ is a counterfactual statement converted as per Claim 4.3 from ω'. In order to check that $A \Box\!\!\rightarrow C$ is valid at ω, we must check that C is true at all A-variants of ω.

Lemma 4.6. There is a unique A-variant of ω.

Proof. Assume not. Then, without loss of generality, there are two distinct A-variants of ω, call them σ_1 and σ_2. As per the definition of an A-variant of ω, there exists some variable sets $V \subseteq Var$ such that $\sigma|_V$ is a verifier of A and $\sigma|_V$ agrees with ω regarding the values of all independent variables outside the basis of V.

So, if both σ_1 and σ_2 are such, they will both have variable sets V_1 and V_2 such that $\sigma_1|_{V_1}$ and $\sigma_2|_{V_2}$ are verifiers of A. However, σ_1 and σ_2 are from the same set Ω, so they are be described in terms of the same features and since they are verifiers of the same statement A, $V_1 = V_2 = \mathit{Rel}(A)$. In addition, both $\sigma_1, \sigma_2 \in \mathit{Prop}(A)$ (skipping the fact that A may not necessarily be an atomic sentence). Recall that A is a statement which is a conjunction of equalities, describing specific values of variables. This means that it must necessarily be the case that $\forall X \in V_1(X(\sigma_1) = X(\sigma_2))$.

Because both σ_1 and σ_2 are A-variants of ω, they also share the same values for independent variables of ω oustide the basis of V_1 (or V_2), leading to the fact that

$$\forall X \in Var_i(X \notin V_1 \rightarrow X(\omega) = X(\sigma_1) = X(\sigma_2)).$$

Thus far we have proven that all independent variable values of σ_1 and σ_2 are the same. Lastly, observe that by the fifth clause of Definition 3.1, all dependent variable values of σ_1 and σ_2 are also the same.

We conclude that all values of all variables of σ_1 and σ_2 are the same, meaning they are the same points. This contradicts our initial assumption that they were distinct. □

We are left to show that C is true at the unique A-variant σ of ω. This is trivial since C is simply the only dependent variable which takes on the value of $f(\omega')$ where ω' is the counterfactual explanation and f is the machine learning model. And in variation semantics the dependent variables take on the values depending on independent variables. Since it is the case that machine learning models describe dependent variables in this case, we can simply query $f(\sigma)$. But since we have established that it is unique and satisfies the same values as ω', $f(\sigma) = f(\omega')$ and thus C is true at σ. Meaning that $A \boxdot\!\!\rightarrow C$ is valid at ω.

■

Example 4.7. Recall ω and ω' from Example 4.4. We will show that the CF statement in question there is true on variation semantics.

$\omega'|_{age,education}$ is a verifier of $A \wedge B = (age = 50.0 \wedge education = master's)$ since:

- $\{age\}, \{education\} \subseteq Var$ are variables such that $V = \{age\} \cup \{education\} = \{age, education\}$
- $\omega'|_{age}$ is a verifier of $A = (age = 50.0)$ and $\omega'|_{education}$ is a verifier of $B = (education = master's)$

Since $\omega'|_{age,education}$ is a verifier of $A \wedge B$ and it agrees with all independent variables of ω outside of the basis of $V = \{age, education\}$, ω' is an $A \wedge B$-variant of ω.

Lastly, since $f(\omega') = 1$, we get that the statement $income = 1$ is true at ω' and so

$$(\text{age} = 50 \wedge \text{education} = \text{master's}) \boxdot\!\!\rightarrow (income = 1)$$

is true at ω.

As could be understood from Section 2.2, our counterfactual generation procedures may generate many counterfactuals for a single original point. Each such generated CE can be converted to a CF statement as mentioned in Claim 4.3. In fact, they can be put together by combining their antecedents using $\vee$ and creating a CF statement of the form

$$A_1 \vee A_2 \vee A_3 \vee \ldots \boxdot\!\!\rightarrow C$$

where A_i are antecedents for each converted CF statement and C is the statement describing the dependent variable. It is easy to see that this statement will still be valid as per the definitions of Section 3.2.

4.3 Logical consequence in generated statements

It would be useful to investigate whether the inference rules outlined, for example, in [Egré and Rott, 2021] are respected by the CE generator. This would partly allow us to tackle the completeness part as to show completeness we must show whatever is true on variation semantics can be generated using the CE generator. To illustrate that this problem is quite thorny, consider the cautious transitivity rule:

$(A \wedge B) \boxdot\!\!\rightarrow C$ and $A \boxdot\!\!\rightarrow B$ are valid. Then $A \boxdot\!\!\rightarrow C$ is valid.

To check that our generator respects this rule, we would have to generate the first two statements and then query the generator further (or check manually) that the third one holds as well. Since we are dealing with data, almost surely none of these deduction rules hold globally (are valid), nevertheless, we could at least attempt to check them case-by-case, devise a sort of metric for a dataset and ML model combination.

However, in this case the problem is the second of the three statements. This is because currently we only have a canonical scheme to convert from CEs to formal counterfactual statements as described in Claim 4.3. That is, the statements A and B in the antecedent would always have to be about independent variables, and the C in the consequent always concerns a dependent variable. However, the statement $A \boxdot\!\!\rightarrow B$ would then have independent variables in both the antecedent and consequent. Thus, essentially, we need a way to generate statements of the form $A \boxdot\!\!\rightarrow C$ where both A and C have independent variables. As a matter of fact, depending on the situation, the sentences A and C may have any combination of different variables in the antecedent and consequent. So, we need ways to generate statements of the form $A \boxdot\!\!\rightarrow C$ that have:

1. The same independent variable in the consequent as in the antecedent.
2. The same dependent variable in the consequent as in the antecedent.
3. Both independent variables in the antecedent and consequent.
4. Both dependent variables in the antecedent and consequent.
5. Independent variable in the antecedent, dependent in the consequent.
6. Dependent variable in the antecedent, independent in the consequent.
7. Multiple independent variable in the antecedent, one dependent in the consequent.
8. Multiple dependent variable in the antecedent, one independent in the consequent.
9. Multiple independent variable in the antecedent, multiple dependent in the consequent.
10. Multiple dependent variable in the antecedent, multiple independent in the consequent.

Claim 4.3 covers only cases 5 and 7 if our ML model is predicting a single variable (as is most often the case), or it may cover case 9 also if multiple dependent variables are being predicted. The following sections investigate a few attempts to tackle this problem whilst keeping in mind the cautious transitivity rule example.

4.3.1 Predicting the consequent of each CF statement

By keeping the conversion schema of Claim 4.3 in mind and given a generated (and converted) counterfactual $(A \wedge B) \boxdot\!\!\rightarrow C$, we could be tempted to simply predict the consequent B of the CF $A \boxdot\!\!\rightarrow B$ using some machine learning model. However, this is not straightforward to do. If we are given the first CF, our supervised ML model by definition is bound to predict the variable appearing in the statement C. Thus, if we were to predict a variable $V \in Rel(B)$ s.t. $V \notin Rel(C)$, we would have to make an altogether different model, most likely using the variable(s) $Rel(C)$. However, in the step of converting the generated CE based on our new ML model, we would necessarily end up with a different partition of the set Var of independent and dependent variables. Thus, such a decision would prevent us from comparing the two resulting counterfactuals.

4.3.2 Correlation, hypothesis testing, and feature interaction

One could attempt to generate a (version of the) $A \boxdot\!\!\rightarrow B$ statement using statistical correlation, statistical hypothesis testing (if the variables are categorical), or feature interaction [Molnar, 2022, Section 8.3]. There are two problems with all of these approaches. Firstly, the results of correlation, hypothesis testing, or feature interaction are not direct substitutes for the CF statement $A \boxdot\!\!\rightarrow B$. For example, positive correlation of 0.8 between the age and income amount variables does not determine the truth value of a counterfactual like $age \geq 35 \boxdot\!\!\rightarrow income \geq 40k$. Secondly, these statistical relationship results are global statements, talking about properties of whole features, not specific truth values of statements at points.

4.3.3 How, then?

Section 5 will discuss ideas as to how the current approaches may be improved. It seems that primarily we need to either restrict the type of logical statements we consider or invent a new way to transform informal counterfactual explanations to formal counterfactual statements. Even so, the afore-mentioned results point to the fact that the generator is not complete w.r.t. variation semantics, so let us conclude it formally.

4.4 Completeness

To show completeness, we must show that whatever CF statement is true on some model of variation semantics, can be generated by a CE generator corresponding to the model.

Theorem 4.8. *CE generator G is not complete w.r.t. variation semantics.*

Proof. Assume the generator G is complete w.r.t. variation semantics. Even putting aside the conversion scheme of Claim 4.3 and its possible drawbacks, there are plenty of counterexamples to choose from. For instance, let $A \boxdot\!\!\rightarrow B$ be a counterfactual statement true at a point ω on some model. Specifically, let A be true at ω already. This is a valid counterfactual statement as per definitions 3.6 and 3.7. Since G is complete, we expect to be able to generate a counterfactual explanation such that it is no different from the original point ω. However, as per the definition of counterfactual explanations outlined in Section

2.2, a CE must change the output of the ML model. In this case, however, the formal counterfactual does not change the output of a machine learning model meaning that the generator will never output a result which does not change the output of the underlying ML model. In other words, the generator G cannot generate a statement corresponding to $A \boxtimes\!\!\rightarrow B$. Contradiction. ■

While G is not complete w.r.t. variation semantics, this is not necessarily an altogether bad thing. This result reveals to us how CE generation is meaningfully different from its formal counterpart. First of all, the counter-example used above would not be a useful result for data subjects that use the counterfactual explanation generator for practical purposes. In addition, there are many "uninteresting" logical statements that, while true, are of little explanatory value in practical situations. For example, $A \wedge \underbrace{A \wedge \ldots \wedge A}_{\text{1 billion times}} \boxtimes\!\!\rightarrow C$.

The way we could remedy this is one of three things: reduce the space of formulas made true by variation semantics to mimic the space of the counterfactual generator, enable the generator to produce counterfactual explanations more akin to the formal counterfactual statements, or a combination of both. We believe the CE generator is already developed to be useful for practical purposes whereas the formal framework is not and thus the first of the three options seems most reasonable to attempt. The next section will offer recommendations to tackling this and other shortcomings of our paper.

5 Future work

This section investigates improvements over the methods employed in this paper and suggests venues for future work.

5.1 Finding relevant logical statements to generate

As previously indicated, there exist numerous logical statements that do not contribute valuable new information about a given scenario. Thus, in order to establish completeness between variation semantics and the CE generator, we must reduce the space of all logical statements to only the relevant ones. Here are a few ways to achieve that:

- Maximally reduce logical redundancy in statements. For example, in the case of statements such as $A \wedge B \boxtimes\!\!\rightarrow C$ and $\underbrace{A \wedge B \wedge A \wedge B \wedge \ldots}_{\text{1 billion times}} \boxtimes\!\!\rightarrow C$, only the former is relevant. One method which gets rid of logical redundancy is that of Karnaugh maps [Karnaugh, 1953], so we may consider all logical statements after they were treated by Karnaugh maps.
- Do not consider statements which have same variables occurring in the antecedent as in the consequent.
- Do not consider statements which have independent variables both in the antecedent as well as consequent. Alternatively, find a better scheme from formal counterfactual statements to CEs regarding such statements.

- Do not consider statements which have more than one dependent variable anywhere in the counterfactual statement.

Employing such restrictions would leave only cases 5, 7, and 9 of the ones outlined in Section 4.3. Naturally, a formal proof will still be necessary to prove that any restricted set of logical statements makes the CE generator complete w.r.t. variation semantics.

5.2 Logics for Vector Spaces

As an alternative to reducing the space of logical statements considered, we could consider a re-working of the semantics at hand. Combination of statements about data using logical connectives is a lot akin to vector arithmetic. There already have been a few attempts [Mizraji, 1992] [Mizraji, 2008] [Leitgeb, 2020] to combine logic and matrix algebra. Perhaps semantic frameworks for counterfactuals could borrow from these attempts to further improve their systems. Alternatively, some new semantics stemming from vector (space) logics could be developed that more naturally fit the CE generator's goal.

5.3 Continuum truth values or introducing probability

Currently, variation semantics employs a boolean notion of truth when evaluating statements. However, this may not be the best approach when dealing with data. Statements in real life are very rarely fully true or not, their truth value lies somewhere inbetween. Thus, a probabilistic version of variation semantics would be desirable to develop as it would apply to the context of data much more easily. For the same reasons a probabilistic entailment framework could accompany the semantics. There are some attempts for a probabilistic semantics for the traditional counterfactuals [Leitgeb, 2012a] [Leitgeb, 2012b] [Hájek, 2014].

5.4 Pushing CEs to obey the inference rules?

[Freiesleben, 2022] explores the tight connection between adversarial examples and counterfactuals. The paper posits that the difference is very small and almost a matter of wording and perspective on objects - positive for counterfactuals, negative for adversarial examples. This is indeed an intriguing observation since the literature on adversarial examples in the machine learning community is booming, meaning that we may perhaps be able to adapt findings from that literature to counterfactuals. Specifically, it would be interesting to investigate how adversarial training [Miyato et al., 2016] would affect the underlying machine learning models used for counterfactual generation. Would then the generated counterfactuals follow logical inference rules of counterfactual logics and/or vector logics more? Less? It remains to be investigated.

6 Conclusion

In this project, we have explored the connections between the informal, but practical, counterfactual explanation generation method and the formal semantic frameworks for counterfactuals. Specifically, we described a restricted notion of soundness and completeness

and attempted to prove it for CE generation and variation semantics. We have succeeded in proving soundness but not completeness. Lastly, this paper provides ideas for further investigations in this area, and intuitions how completeness may be still achieved.

References

[Becker and Kohavi, 1996] Becker, B. and Kohavi, R. (1996). Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

[Blackburn et al., 2001] Blackburn, P., Rijke, M. d., and Venema, Y. (2001). *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

[Bojarski et al., 2016] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *CoRR*, abs/1604.07316.

[Dietrich, 2012] Dietrich, E. (2012). What does it mean to say an algorithm is sound and complete? Software Engineering. https://softwareengineering.stackexchange.com/a/140716 accessed on 2023-08-26.

[Egré and Rott, 2021] Egré, P. and Rott, H. (2021). The Logic of Conditionals. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2021 edition.

[Freiesleben, 2022] Freiesleben, T. (2022). The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds and Machines*, 32(1):77–109.

[Hájek, 2014] Hájek, A. (2014). Probabilities of counterfactuals and counterfactual probabilities. *Journal of Applied Logic*, 12(3):235–251.

[Hornischer, 2022] Hornischer, L. (2022). Philosophical logic lecture notes.

[Hudetz and Crawford, 2022] Hudetz, L. and Crawford, N. (2022). Variation semantics: when counterfactuals in explanations of algorithmic decisions are true.

[Julian et al., 2016] Julian, K. D., Lopez, J., Brush, J. S., Owen, M. P., and Kochenderfer, M. J. (2016). Policy compression for aircraft collision avoidance systems. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10.

[Kaminski, 2018] Kaminski, M. E. (2018). The right to explanation, explained. *SSRN Electronic Journal*.

[Karimi et al., 2019] Karimi, A., Barthe, G., Balle, B., and Valera, I. (2019). Model-agnostic counterfactual explanations for consequential decisions. *CoRR*, abs/1905.11190.

[Karnaugh, 1953] Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72(5):593–599.

[Laugel et al., 2017] Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., and Detyniecki, M. (2017). Inverse classification for comparison-based interpretability in machine learning.

[Leitgeb, 2012a] Leitgeb, H. (2012a). A PROBABILISTIC SEMANTICS FOR COUNTERFACTUALS. PART A. *The Review of Symbolic Logic*, 5(1):26–84.

[Leitgeb, 2012b] Leitgeb, H. (2012b). A PROBABILISTIC SEMANTICS FOR COUNTERFACTUALS. PART B. *The Review of Symbolic Logic*, 5(1):85–121.

[Leitgeb, 2020] Leitgeb, H. (2020). On the logic of vector space models. https://www.youtube.com/watch?v=QHtRiQJpSyU accessed on 2023-08-28.

[Looveren and Klaise, 2019] Looveren, A. V. and Klaise, J. (2019). Interpretable counterfactual explanations guided by prototypes. *CoRR*, abs/1907.02584.

[Mittelstadt et al., 2018] Mittelstadt, B. D., Russell, C., and Wachter, S. (2018). Explaining explanations in AI. *CoRR*, abs/1811.01439.

[Miyato et al., 2016] Miyato, T., Dai, A. M., and Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

[Mizraji, 1992] Mizraji, E. (1992). Vector logics: The matrix-vector representation of logical calculus. *Fuzzy Sets and Systems*, 50(2):179–185.

[Mizraji, 2008] Mizraji, E. (2008). Vector logic: A natural algebraic representation of the fundamental logical gates. *Journal of Logic and Computation*, 18(1):97–121.

[Molnar, 2022] Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.

[Mothilal et al., 2020] Mothilal, R. K., Sharma, A., and Tan, C. (2020). Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

[O'Neil, 2016] O'Neil, C. (2016). *Weapons of math destruction*. Crown Publishing Group.

[Russell, 2019] Russell, C. (2019). Efficient search for diverse coherent explanations. *CoRR*, abs/1901.04909.

[Sheikh et al., 2020] Sheikh, M. A., Goel, A. K., and Kumar, T. (2020). An approach for prediction of loan approval using machine learning algorithm. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 490–494.

[Starr, 2022] Starr, W. (2022). Counterfactuals. In Zalta, E. N. and Nodelman, U., editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edition.

[Wachter et al., 2017] Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *SSRN Electronic Journal*.