COUNTERFACTUAL EXPLANATION

- At its core, counterfactuals allows us to take action in order to cause a certain outcome.
- In terms of machine learning, the actions are the changes in the features of the model while the outcome is the desired target response.
- The data is essentially perturbed until new instances are returned that correspond to a model prediction class away from the original. Since there are various ways to reach the same outcome, there can be multiple counterfactuals.

Assessing human decision-making

Sandra Bauer Meryem Öztürk

Counterfactual reasoning has been used the social sciences to assess different aspects of huma decision-making [Bertrand and Mullainathan 2003, Weichselbaumer 2019]

Sandra Bauer Meryem Öztürk

Meryem Öztürk Meryem Öztürk

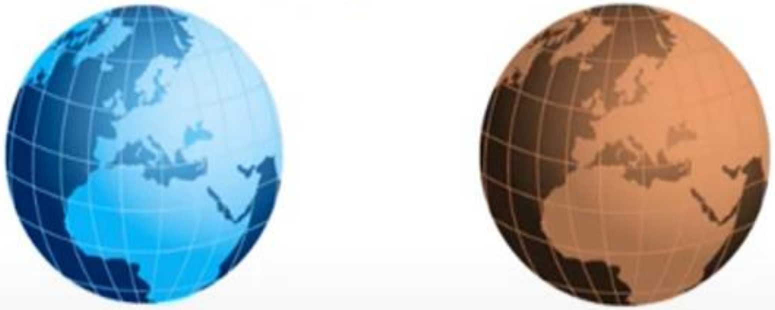
Why does counterfactual reasoning work?

Because only the specific input is varied, provides the **causal effect** of the input, specific to the current context.

Also known as individual causal effect.

What is a counterfactual?

Given a system output y,
a counterfactual $y_{X_i=x'}$ is the output of the system had some input X_i changed but everything else unaffected by X_i remained the same. [Pearl 2009]



REAL WORLD
$(X_i = x)$

COUNTERFACTUAL WORLD
$(X_i = x')$

Counterfactual: $P\left(Y_{X_i=x'} \middle| \boldsymbol{X} = \boldsymbol{x}, Y = y\right)$

→ Since a ML model f is a deterministic model, counterfactual simplifies to $f\left(\boldsymbol{X}_{X_i=x'}\right)$

The many uses of a model counterfactual

Individual Effect of Input Feature $\mathrm{X_i}$

$$=E\left(Y_{X_i=x'} \middle| \boldsymbol{X} = \boldsymbol{x}, Y = y\right) - E(Y|\boldsymbol{X} = \boldsymbol{x})$$

$f\left(\boldsymbol{X}_{X_i=x'}\right) - f(\boldsymbol{X})$ can provide:

1. Explanation of how important X_i feature is.
2. Bias in the model if X_i is a sensitive feature.
3. More generally, provides a natural way to debug ML models (*ala fuzz testing*).

Why use counterfactuals when there are many established methods of ML model explanation?

Explaining machine learning predictions

Techniques to explain machine predictions

LIME (Ribeiro et al., 2016); **Local Rule-based** (Guidotti et al., 2018);
SHAP (Lundberg et al., 2017); **Intelligible Models** (Lou et al., 2012); …..

Feature importance-based methods are widely used in many practical applications

In many cases, feature importance is not enough

Bank → Loan distribution model

Loan granted

Loan denied

Suppose model predicts that the person should not get the loan.

Decision-maker: Why should this person not get the loan?

Person: What should I do to get the loan in the future?

Feature importance-based explanations

Counterfactual explanations (CF)

("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

Feature importance-based explanations

Interpretable,
but not high-fidelity

Counterfactual explanations (CF)

("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

Interpretable,
and high-fidelity

Catch: How to generate the right examples that are useful to end-user?

Wachter et al. suggest minimizing the following loss:

$$L(x,x',y',\lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x,x')$$

- The first term is the quadratic distance between the model prediction for the counterfactual x’ and the desired outcome y’, which the user must define in advance.
- The second term is the distance d between the instance x to be explained and the counterfactual x’.
- The loss measures how far the predicted outcome of the counterfactual is from the predefined outcome and how far the counterfactual is from the instance of interest.
- The distance function d is defined as the Manhattan distance weighted with the inverse median absolute deviation (MAD) of each feature.

```python
# Using sklearn backend
m = dice_ml.Model(model=model, backend="sklearn")
# Using method=random for generating CFs
exp = dice_ml.Dice(d, m, method="random")
```

```python
e1 = exp.generate_counterfactuals(x_train[0:1], total_CFs=2, desired_class="opposite")
e1.visualize_as_dataframe(show_only_changes=True)
```

```
Query instance (original outcome : 0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

```
Diverse Counterfactual set (new outcome: 1.0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	67.0	-	Masters	-	-	Other	-	-	1
1	66.0	-	Prof-school	-	-	Other	-	-	1

```python
# Restricting age to be between [20,30] and Education to be either {'Doctorate', 'Prof-school'}.
e3 = exp.generate_counterfactuals(x_train[0:1],
                                  total_CFs=2,
                                  desired_class="opposite",
                                  permitted_range={'age':[20,30], 'education':['Doctorate', 'Prof-school']})
e3.visualize_as_dataframe(show_only_changes=True)
```

Query instance (original outcome : 0)

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

Diverse Counterfactual set (new outcome: 1.0)

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	28.0	Self-Employed	Doctorate	-	Professional	-	Female	21.0	1
1	27.0	Self-Employed	Doctorate	-	Professional	-	Female	50.0	1

Thank you