MCCE

Monte Carlo sampling of valid and realistic counterfactual explanations

Martin Jullum (jullum@nr.no)

Internseminar SAMBA 09.03.23

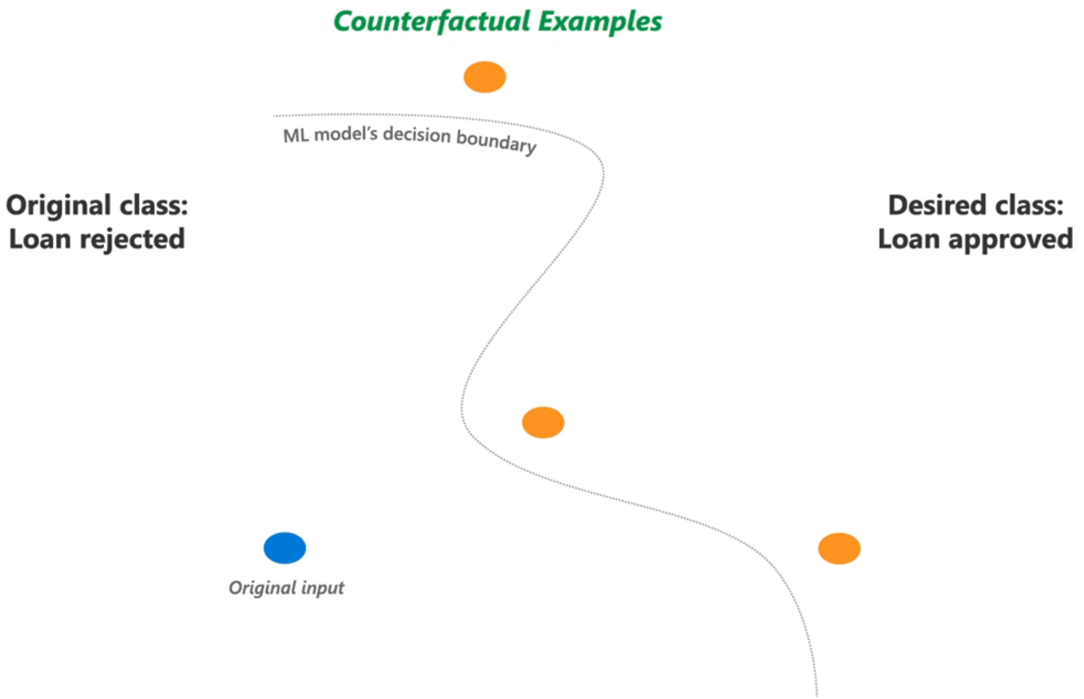
Prediction explanation

- Assume a model $f(\boldsymbol{x}) \in \mathbb{R}$ that predicts some unknown outcome based on a set of features $\boldsymbol{x} = (x_1, \dots, x_M)$
- We apply the predictive model for a specific input $\boldsymbol{x} = \boldsymbol{x}^*$, reaching a certain prediction $f(\boldsymbol{x}^*)$
- Individual prediction explanation
  - Want to understand how the different features, or types of features affect this specific prediction value $f(\boldsymbol{x}^*)$
  - I.e. explain the predicted outcome in terms of the input $\boldsymbol{x} = \boldsymbol{x}^*$ (local explanation)
- Frameworks…
  - LIME
  - Anchors
  - Shapley values
  - PDP/ICE
  - PredDiff
  - **Counterfactual explanations (CE)**

Counterfactual explanations – by example

Default prediction model as a basis for automatic processing of loaning applications

- Response y: Loan defaulted or not
- Features $\boldsymbol{x} = (x_1, \dots, x_M)$: Info about the applicant, income, other loans, previous defaults, transactions history
- Predictive model f: Model trained to predict probability of default: $f(\boldsymbol{x}) \approx \Pr(y = \text{default} | \boldsymbol{x})$
- *Loan approved if* $f(\boldsymbol{x}) < c = 0.1$



CASE: Peter has features x^*, and got his loan application rejected as $f(x^*) = 0.3 > c$

Question: What can Peter do to receive a loan?

CE solution: Examples of (minimal) changes in features which approves the application

Counterfactual explanations – criteria

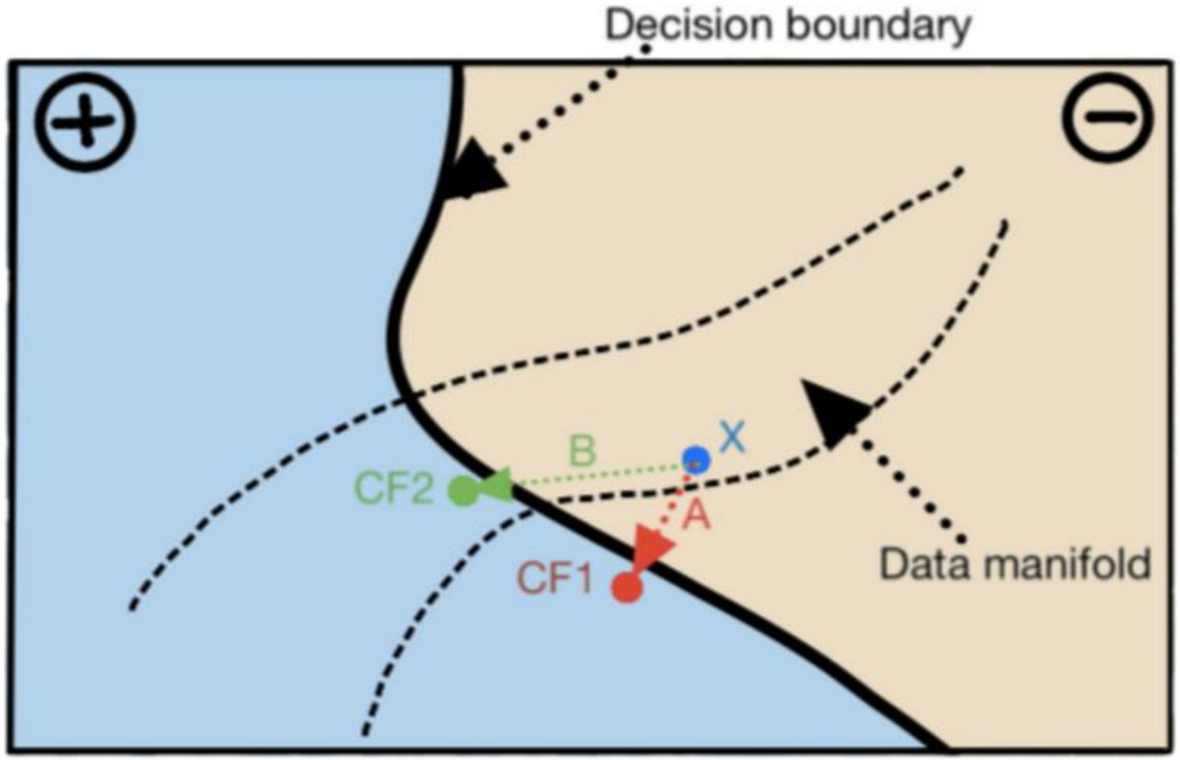
$\boldsymbol{e}$ is a counterfactual explanation of $f(\boldsymbol{x}^*)$

- **Criterion 1**: $\boldsymbol{e}$ is *on-manifold*, i.e., $p(\boldsymbol{X}^m = \boldsymbol{e}^m \mid \boldsymbol{X}^f = \boldsymbol{e}^f) > \epsilon$, for some $\epsilon > 0$;
- **Criterion 2**: $\boldsymbol{e}$ is *actionable*, i.e., does not violate any of the fixed features;
- **Criterion 3**: $\boldsymbol{e}$ is *valid*, i.e., $f(\boldsymbol{e}) \geq c$, for the chosen cutoff c;
- **Criterion 4**: $\boldsymbol{e}$ is *low cost*, i.e., close to the factual, $\boldsymbol{x}^*$

We measure "cost" by
1. # features changed
2. Gower distance

$$\text{Gower distance} = \frac{1}{p}\sum_{j=1}^{p}\delta_G(d_j, x_j) \in [0,1],$$

$$\delta_G(d_j, x_j) = \begin{cases} \frac{1}{R_j} \mid d_j - x_j \mid & \text{if } x_j \text{ is numerical,} \\ \mathbb{1}_{d_j \neq x_j} & \text{if } x_j \text{ is categorical,} \end{cases}$$



Existing CE methods

Optimization based methods

- Minimize loss functions (wrt ***e***) of type
  - Often require differentiable *f*
  - Not necessarily on-manifold
  - Categorical features more troublesome

$$L_{\boldsymbol{x}^*}(\boldsymbol{e}) = \mathrm{dist}_1(f(\boldsymbol{e}), c) + \lambda \cdot \mathrm{dist}_2(\boldsymbol{x}^*, \boldsymbol{e})$$

Heuristic search-based methods

- Optimization with heuristic search strategies

Instance-based methods

- Finds counterfactuals by searching for instances in a reference distribution/dataset

MCCE – the method

A 3-step procedure

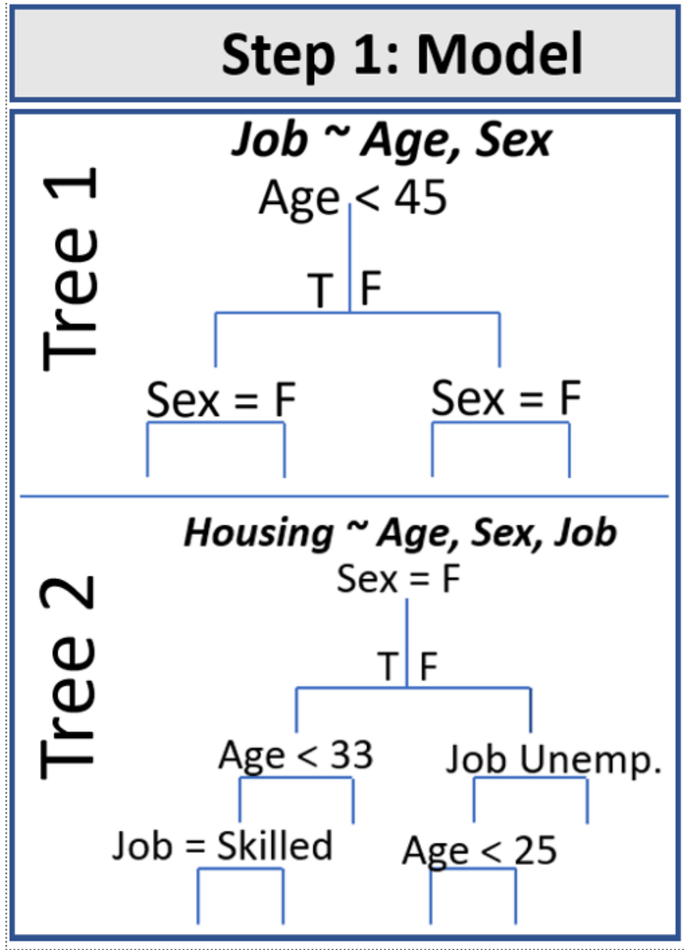
1. Model the distribution of mutable features, given the immutable features and the decision
2. Generate a large number of samples from the modelled distribution with the specified fixed features x^{*f}
3. Discard the invalid samples, and choose the one “nearest” to x^*

MCCE – step 1: Model

We utilize

$$p(\boldsymbol{X}^m \mid \boldsymbol{X}^f, Y') = p(X_1^m \mid \boldsymbol{X}^f, Y') \prod_{i=2}^{q} p(X_i^m \mid \boldsymbol{X}^f, Y', X_1^m, \dots, X_{i-1}^m)$$

- Then fit $q-1$ decision trees to $\boldsymbol{X}_i^m \sim \left(\boldsymbol{X}^f, Y', \boldsymbol{X}_1^m, \dots, \boldsymbol{X}_{i-1}^m\right), i = 2, \dots, q,$ using CART or Conditional Inference Trees (ctree), where the observations in the end nodes are stored

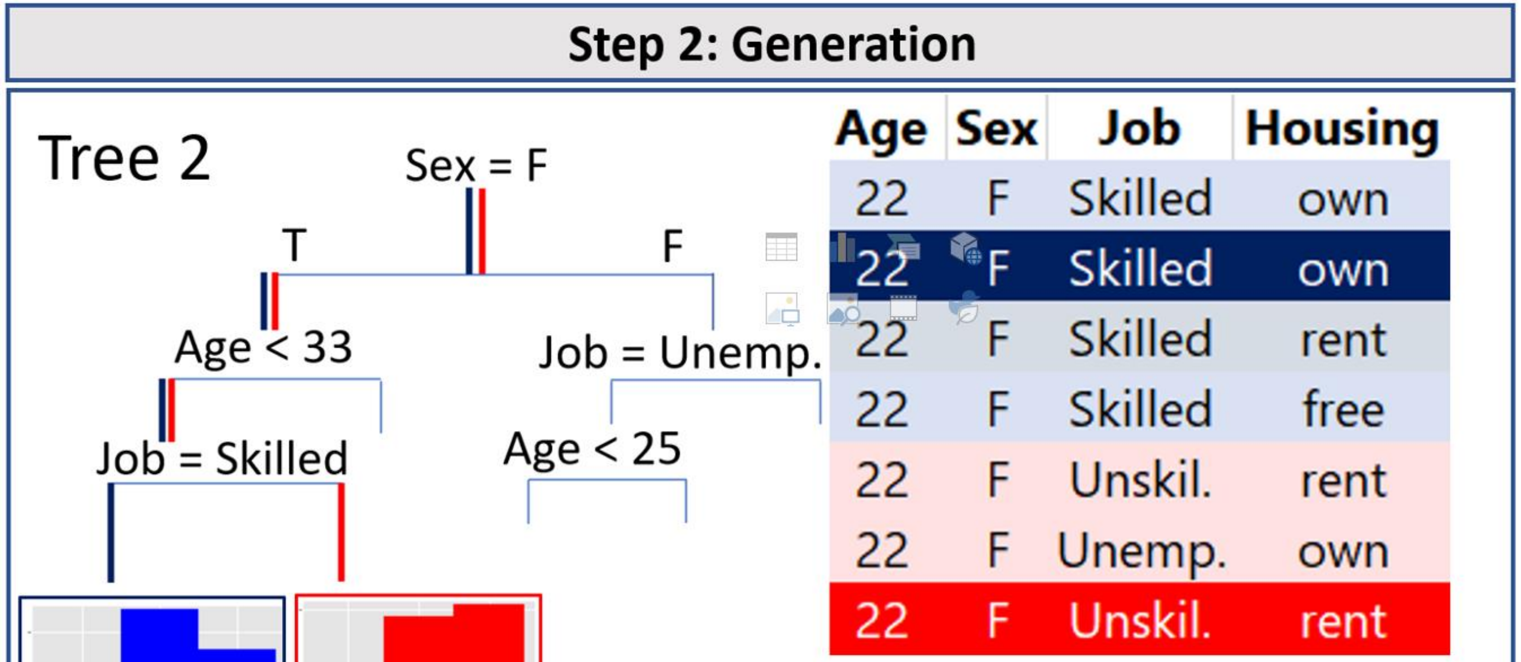


MCCE – step 2: Generation

To generate one sample from $X^m | X^f = x^{*f}, Y' = 1$, we:

1. Follow x^{*f} down the first tree and make one sample $\tilde{x}_1^m$ from the observations in the end node
2. For $i = 2, \dots, q$:
   - Follow $x^{*f}, \tilde{x}_1^m, \dots, \tilde{x}_i^m$down the i-th tree, and make one sample $\tilde{x}_1^m$ from the observations in the end node

Repeat the procedure K times do obtain a synthetic dataset $\boldsymbol{D}$ with K samples



Obs. from each histogram. Dark blue/red shows sample added to $\boldsymbol{D}_h$.

MCCE – step 3: Post-processing

Filter the data set $\boldsymbol{D}$ to obey our four criteria

$\boldsymbol{e}$ is a counterfactual explanation of $f(\boldsymbol{x}^*)$

- **Criterion 1**: $\boldsymbol{e}$ is *on-manifold*, i.e., $p(\boldsymbol{X}^m = \boldsymbol{e}^m \mid \boldsymbol{X}^f = \boldsymbol{e}^f) > \epsilon$, for some $\epsilon > 0$;
- **Criterion 2**: $\boldsymbol{e}$ is *actionable*, i.e., does not violate any of the fixed features;
- **Criterion 3**: $\boldsymbol{e}$ is *valid*, i.e., $f(\boldsymbol{e}) \geq c$, for the chosen cutoff c;
- **Criterion 4**: $\boldsymbol{e}$ is *low cost*, i.e., close to the factual, $\boldsymbol{x}^*$.

▶ C1 & C2 already satisfied

▶ Most samples satisfies C3, remove the others

▶ Choose the sample closest to $\boldsymbol{x}^*$. We do this by

- Determine the smallest number of samples being changed, and remove those with more changes (L0)
- Amongst the remaining, chose the one minimizing the Gower distance (L1)

Step 3: Post-processing

Age	Sex	Job	House	Saving	Y	L0	L2
22	*F*	*Unskil.*	*Own*	*Little*	*0*		
22	F	Skilled	own	rich	0	5	2.67
22	F	Unskil.	rent	little	1	5	2.19
22	F	Skilled	own	rich	1	5	2
22	F	Unskil.	rent	little	1	3	0.74
22	F	Unempl.	rent	little	0	7	3.22
22	F	Skilled	rent	little	0	5	2.72
22	F	Skilled	rent	moderate	1	6	1

Counterfactual is chosen as row(s) with smallest L0/L1 and Y=1.

Experiments – setup

- Real data sets
- Generate CE to explain predictions from a test set
  - Use MCCE + 6 other on-manifold methods
- Compare the methods in terms of performance measures
  - L0, L1, feasibility, violation, success, computation time

$$\text{feasibility} = \sum_{i=1}^{k} w^{[i]} \frac{1}{p} \sum_{j=1}^{p} \text{dist}(e_j, x_j^{[i]})$$

Experiments – Give me some credit

- Binary classification of financial distress or not
- 10 cont features
- 150 000 obs
- Use 3-layer ANN for modelling

Data set: Give Me Some Credit, $n_{\text{test}} = 1000$, $K = 1000$

Method	$L_0 \downarrow$	$L_1 \downarrow$	feasibility↓	violation↓	success↑	$N_{\text{CE}} \uparrow$	t(s) all↓
C-CHVAE	8.98 (0.13)	0.95 (0.28)	**0.26** (0.04)	**0.00** (0.00)	**1.00**	**1000**	151.81
CEM-VAE	8.62 (1.08)	1.61 (0.57)	0.27 (0.07)	0.96 (0.19)	0.93	**1000**	813.99
CLUE	10.00 (0.04)	1.41 (0.32)	0.37 (0.06)	1.00 (0.03)	**1.00**	**1000**	3600.35
CRUDS	9.00 (0.00)	1.68 (0.36)	0.42 (0.02)	**0.00** (0.00)	**1.00**	**1000**	11823.25
FACE	8.59 (1.08)	1.66 (0.53)	0.32 (0.09)	0.98 (0.16)	**1.00**	**1000**	32308.78
REViSE	8.36 (1.06)	0.70 (0.27)	0.32 (0.05)	**0.00** (0.00)	**1.00**	**1000**	8286.04
MCCE	**4.52** (0.97)	**0.61** (0.32)	0.27 (0.07)	**0.00** (0.00)	**1.00**	**1000**	**32.18**

Experiments – Adult

- Binary classification of income >= $50 000
- 4 cont + 8 cat features
- 49 000 obs
- Use 3-layer ANN for modelling

Data set: Adult, $n_{\text{test}} = 1000$, $K = 1000$							
Method	$L_0 \downarrow$	$L_1 \downarrow$	feasibility↓	violation↓	success↑	$N_{\text{CE}} \uparrow$	t(s) all↓
C-CHVAE	7.76 (1.02)	3.13 (1.10)	0.27 (0.17)	**0.00** (0.00)	**1.00**	**1000**	140.33
CEM-VAE	6.92 (2.06)	3.18 (1.65)	0.21 (0.15)	1.38 (0.59)	0.49	**1000**	768.76
CLUE	13.00 (0.00)	7.83 (0.31)	0.93 (0.12)	1.36 (0.48)	**1.00**	**1000**	3578.00
CRUDS	7.87 (1.08)	4.55 (1.09)	1.10 (0.16)	**0.00** (0.00)	**1.00**	**1000**	15013.56
FACE	6.98 (1.56)	3.3 (1.50)	**0.24** (0.20)	1.42 (0.51)	**1.00**	**1000**	10280.69
REViSE	5.91 (1.23)	1.62 (1.23)	0.46 (0.33)	**0.00** (0.00)	**1.00**	**1000**	11806.86
MCCE	**2.70** (0.73)	**0.56** (0.45)	0.32 (0.25)	**0.00** (0.00)	**1.00**	**1000**	**24.97**

Conclusion

MCCE

- Models both features and the decision to ensure on-manifold and valid CE
- Conditional sampling guarantees to not violate immutable features
- Relies on trees which handle continuous/discrete/categorical features
- Breaks up tasks into 3 step – each step can easily be altered to specific needs
- Easy to implement
- Outperforms competing methods in terms of both accuracy and speed