PADERBORN UNIVERSITY

Counterfactual Explanations

Explainable Artificial Intelligence

Dr. Stefan Heindorf

Counterfactual explanations

Example

- "If I hadn't taken a sip of this hot coffee, I wouldn't have burned my tongue"
- Cause X is that I had a hot coffee
- Effect Y is that I burned my tongue

Counterfactuals

- Widely used in psychology and causality
- Require imagining a hypothetical reality that contradicts the observed facts (for example, a world in which I have not drunk the hot coffee)

  → hence the name counterfactual ("counter to the facts")

Counterfactual explanations

- Explain a single prediction of a ML model
- "If X had not occurred, Y would not have occurred"
- Cause X: Feature values of an instance (input of the model)
- Effect Y: Prediction of the model

Counterfactual explanations

Examples

Example 1: Rejection of loan application

- Bank customer Peter would like to know “why” loan was denied
- Features: income, number of credit cards, age, …
- Counterfactual explanation 1: If Peter earned 10,000 more per year, he would get the loan
- Counterfactual explanation 2: If Peter had fewer credit cards and had not defaulted on a loan five years ago, he would get the loan

Example 2: Prediction of apartment rent

- Landlord Anna uses ML model to predict rent of an apartment and would like to know how to increase the rent. Prediction: 900 EUR
- Features: details about size, location, whether pets are allowed, …
- Counterfactual explanation 1: If the apartment were 15 m^2 larger, it could be rented for over 1000 EUR (non-actionable: apartment size cannot be changed)
- Counterfactual explanation 2: If pets allowed and windows had better insulation, rent could be over 1000 EUR (actionable: Anna can change those features)

Counterfactual explanations

Remarks

- Counterfactuals do not have to be actual instances from the training data (can be a new combination of feature values)
- There can be multiple counterfactual explanations for a given instance and a prediction
- How to choose the best explanations? See next slide

Counterfactual explanations…

Desirable criteria

Should make prediction similar to target defined by user
(= alternative reality)

Should be similar to the instance regarding feature values

- Example: Similarity metrics: Manhattan distance, Gower distance, …

Should change few features

- Example: count number of changed features

Should be diverse

- Multiple counterfactual explanations presented to the user should be diverse
- Example: change different features
  (some features might be easier for a user to change than others)

Should have feature values that are likely

- Feature values should be possible in the real world
- Formally: values should be likely according to joint probability distribution
- Example: size of apartment should not be negative
- Example: apartment with 10 rooms and 20 m^2 is unlikely

Generating counterfactual explanations

First, naïve method

Define loss based on criteria mentioned above

1. Input of loss: instance of interest, counterfactual, desired outcome
2. Weighting and formalization of each of the criteria
3. Output: weighted average of criteria

Algorithm

- Input: instance of interest, desired outcome
- Repeat until stopping criterium reached:
  1. Randomly change feature values of the instance of interest (counterfactual)
  2. Compute loss
- Return counterfactual instance with smallest loss

Generating counterfactual explanations

Method by Wachter et al.

Given: Tabular data with numeric feature values $\mathbf{x}$, weight λ

Return: counterfactual $\mathbf{x}'$ that minimizes loss

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

where distance d and median absolute deviation (MAD_j) are defined as

$$d(\mathbf{x}, \mathbf{x}') := \sum_{j=1}^{p} \frac{\left|x_j - x_j'\right|}{MAD_j}$$

$$MAD_j := \text{median}_{i \in \{1,\ldots,n\}}\left(\left|x_j^{(i)} - \text{median}_{l \in \{1,\ldots,n\}}\left(x_j^{(l)}\right)\right|\right)$$

$\mathbf{x}$	instance to be explained
$\mathbf{x}'$	counterfactual of instance x
y'	desired outcome
λ	parameter to balance difference in prediction and feature values

Generating counterfactual explanations

Method by Wachter et al., see also CF in Alibi library

Algorithm in practice

Input: instance $\mathbf{x}$ to be explained, the desired outcome y', a tolerance ϵ and a (low) initial value for λ

1. Do
   - Minimize the loss $L(\mathbf{x}, \mathbf{x}', y', \lambda) = \lambda \cdot \left(\hat{f}(\mathbf{x}') - y'\right)^2 + d(\mathbf{x}, \mathbf{x}')$ for $\mathbf{x}'$ and obtain set of counterfactuals X'_{min}
     (e.g., optimization via ADAM with different random initializations for $\mathbf{x}'$)
   - Increase λ (e.g., multiply by 2)
2. Until exists $\mathbf{x}'$ in X'_{min} such that $|\hat{f}(\mathbf{x}') - y'| \leq \epsilon$

Generating counterfactual explanations

Method by Wachter et al.

Method takes following criteria into account:

- Prediction similar to target defined by user
- Explanation similar to the instance regarding feature values

Method does not take following criteria into account

- Few feature changes
  (increasing 10 features by 1 will give the same distance to $\mathbf{x}$ as increasing one feature by 10)
- Diverse counterfactuals
- Likely feature values
  (unrealistic feature combinations are not penalized)

Method does not handle categorical features

Generating counterfactual explanations

Method by Dandl et al.

Let $\hat{f}: \mathcal{X} \to \mathbb{R}$ be a prediction function, $\mathcal{X}$ the feature space, and $Y \subset \mathbb{R}$ a set of desired outcomes

Minimize the multi-objective loss with four objectives:

$$L\left(\mathbf{x}, \mathbf{x}', y', X^{obs}\right) = \left(o_1(\hat{f}(\mathbf{x}'), Y'), o_2(\mathbf{x}, \mathbf{x}'), o_3(\mathbf{x}, \mathbf{x}'), o_4\left(\mathbf{x}', \mathbf{X}^{obs}\right)\right)$$

Distance of counterfactual $\mathbf{x}'$ to desired prediction y'

$$o_1(\hat{f}(\mathbf{x}'), Y') = \begin{cases} 0 & \text{if } \hat{f}(\mathbf{x}') \in Y' \\ inf_{y' \in Y'} \left| \hat{f}(\mathbf{x}') - y' \right| & \text{else} \end{cases}$$

Generating counterfactual explanations

Method by Dandl et al.

Closeness of counterfactual x' to instance x (Gower distance)

$$o_2(\mathbf{x}, \mathbf{x}') = \frac{1}{p}\sum_{j=1}^{p} \delta_G\left(x_j, x_j'\right)$$

$$\delta_G\left(x_j, x_j'\right) = \begin{cases} \frac{1}{\hat{R}_j}\left|x_j - x_j'\right| & \text{if } x_j \text{ numerical} \\ \mathbb{I}_{x_j \neq x_j'} & \text{if } x_j \text{ categorical} \end{cases}$$

$$\hat{R}_j = \max\{x_j \in X_j\} - \min\{x_j \in X_j\}$$

Number of feature changes

$$o_3(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_0 = \sum_{j=1}^{p} \mathbb{I}_{x_j' \neq x_j}$$

Likely feature values/combinations

$$o_4\left(\mathbf{x}', \mathbf{X}^{obs}\right) = \min_{\mathbf{x}^{obs} \in X^{obs}} \frac{1}{p}\sum_{j=1}^{p} \delta_G(x_j', x_j^{obs})$$

Counterfactual search

Method by Dandl et al.

Wachter et al. uses weight λ to balance different criteria

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

Problem

- Setting weights for different criteria in a meaningful way is difficult
- Weights have to be set in advance
- Result: only few counterfactuals (with the same loss)

Goal

- Optimize all four objectives o_1, o_2, o_3, o_4 simultaneously
- Obtain set of nondominated counterfactuals
- Inspect whole set of counterfactuals (and decide for one later)

Solution

- Nondominated Sorting Genetic Algorithm (NSGA-II)
  (evolutionary algorithm applying Darwin's law of "survival of the fittest")
- Fitness: vector of objective values (o_1, o_2, o_3, o_4)

Pareto fronts: sets of non-dominated points

A point is dominated if there is another point that is at least as good in every aspect and strictly better in at least one aspect

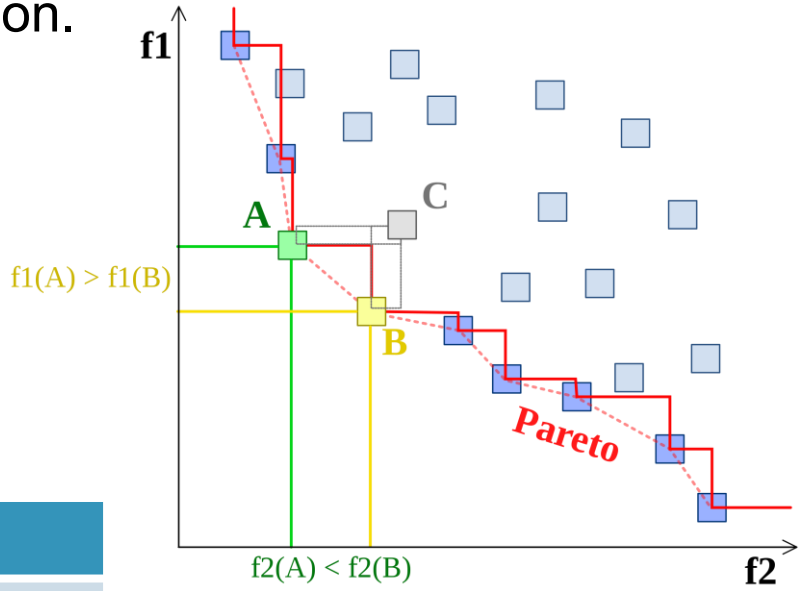
Domination rank: Nondominated points have rank 1 (belong to front F_1). Points that are not dominated anymore after removing all rank 1 points, have rank 2 (and belong to front F_2). And so on.

Example 1 (lower values are better)

- C is dominated both by A and B
  → not on Pareto front
- Points A and B are not dominated
  → both on Pareto front

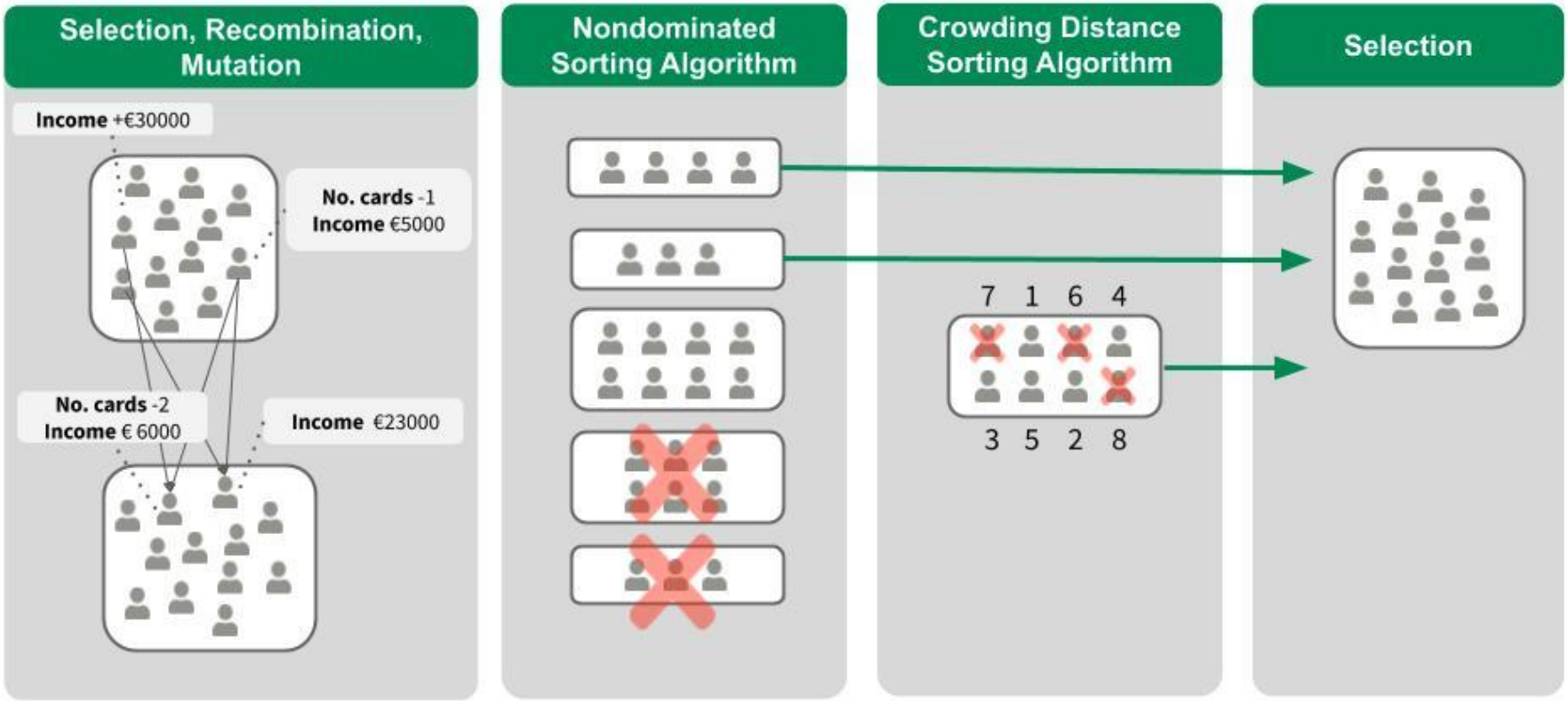
Example 2 (lower values for o_1, o_2, o_3, o_4 are better)

No	o_1	o_2	o_3	o_4	Front	Comment
1	0.7	0.5	0.3	0.4	F_1	Not dominated
2	0.7	0.6	0.9	0.4	F_2	Dominated by No. 1
3	0.1	0.9	0.2	0.6	F_1	Not dominated



Counterfactual search

One generation of the NSGA-II algorithm



"Visualization of one generation of the NSGA-II algorithm" (https://github.com/christophM/interpretable-ml-book/blob/master/manuscript/images/cfexp-nsgaII.jpg) by Susanne Dandl, https://creativecommons.org/licenses/by-nc-sa/4.0/

Counterfactual search

NSGA-II algorithm [Dandl et al. 2020]

Initialization: Population P_0 of N counterfactual candidates is initialized by randomly changing some of the features compared to our instance $\mathbf{x}$ to be explained

- Example: $\mathbf{x} = (58, f, unskilled), N = 3$
- $P_0 = \{(53, f, skilled), (69, m, unskilled), (62, f, skilled)\}$

Selection: Use binary tournament selection to choose candidates for recombination step (for each candidate: pick two individuals from population, select the one with lower domination rank)

- $P_0 = \{(53, f, skilled), (69, m, unskilled), (62, f, skilled)\}$
- Hypothetical domination ranks: 1, 2, 3
- Candidate 1: From $\{(53, f, skilled), (62, f, skilled)\}$ choose $(53, f, skilled)$
- Candidate 2: From $\{(69, m, unskilled), (62, f, skilled)\}$ choose $(69, m, unskilled)$

Counterfactual search

NSGA-II algorithm [Dandl et al. 2020]

Recombination: The candidates are pairwise recombined to produce children that are similar to them by averaging their numerical feature values or by crossing over their categorical features (two children per two parents)

- Candidate 1: $(53, f, skilled)$
- Candidate 2: $(69, m, unskilled)$
- Features used for recombination: age, gender
- Child 1: $(61, m, skilled)$
- Child 2: $(61, f, unskilled)$
- $Q_t = \{(61, m, skilled), (61, f, unskilled), (62, f, skilled)\}$

Mutation: In addition, we slightly mutate the feature values of the children to explore the whole feature space. After the mutation step, we have N parents P_t and N children Q_t

- $P_t = \{(53, f, skilled), (69, m, unskilled), (62, f, skilled)\}$
- $Q_t = \{(59, m, skilled), (61, m, unskilled), (62, m, skilled)\}$

Counterfactual search

NSGA-II algorithm [Dandl et al. 2020]

Nondominated Sorting Algorithm: The nondominated sorting algorithm sorts the candidates according to their domination rank

- $P_t = \{(53, f, skilled), (69, m, unskilled), (62, f, skilled)\}$
- $Q_t = \{(59, m, skilled), (61, m, unskilled), (62, m, skilled)\}$
- Hypothetical domination ranks 1, 2, 2, 2, 1, 2
- Sorting: $(53, f, skilled), (61, m, unskilled), (69, m, unskilled), (62, f, skilled)$ $(59, m, skilled), (62, m, skilled)\}$

Crowding Distance Sorting Algorithm: If candidates have the same domination rank (=are on the same nondominated front), the crowding distance sorting algorithm sorts the candidates according to their diversity (=average distance to the two closest points for each objective)

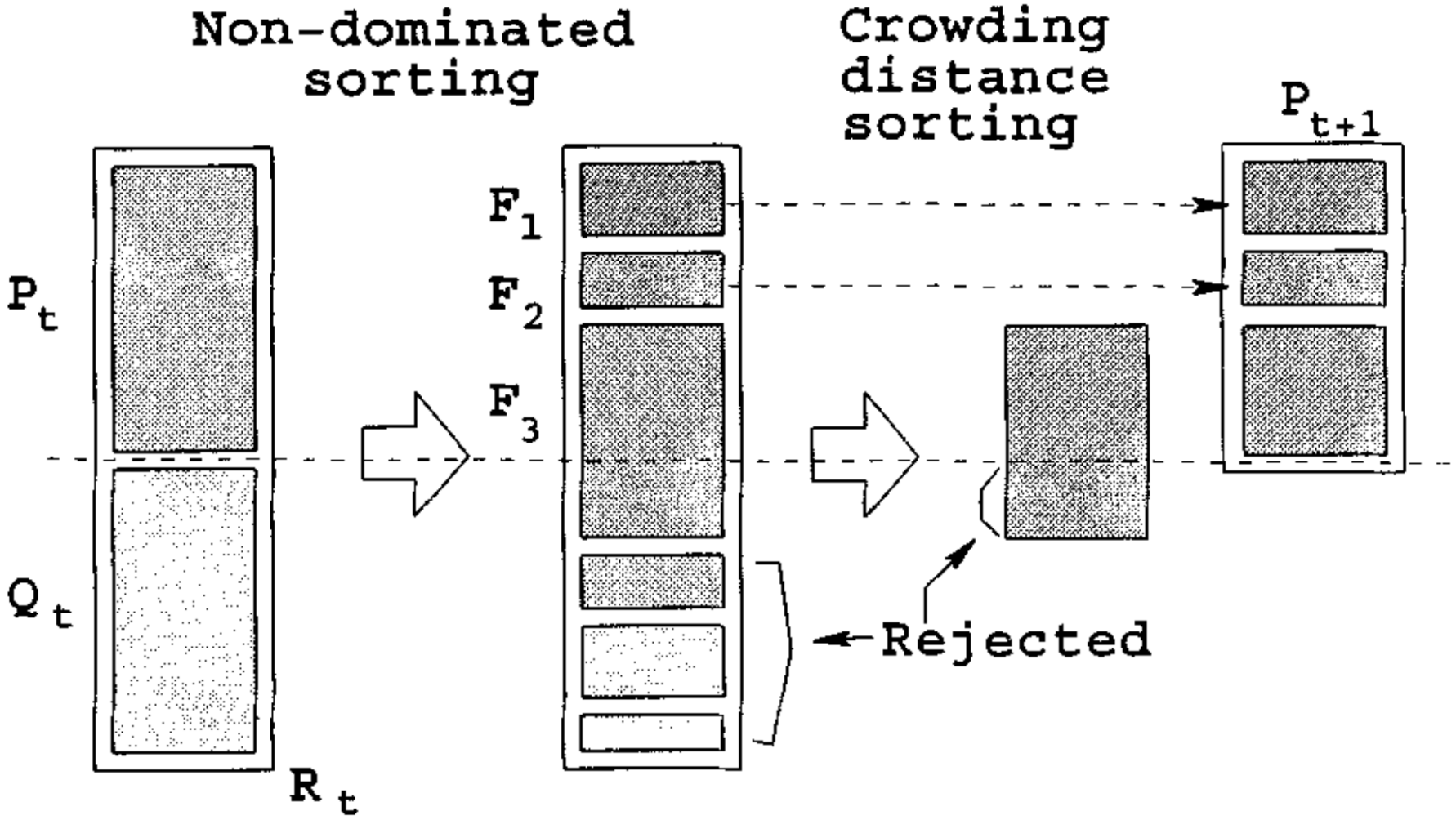
- Candidates with domination rank 2: $(69, m, unskilled), (62, f, skilled), (59, m, skilled), (62, m, skilled)$
- Their crowding distances (see next slides): ∞, ∞, 2, 3

NSGA-II Selection: We select the most promising candidates according to the nondominated sorting algorithm (smaller domination ranks). In case of ties, we select the most diverse individuals (higher crowding distance)

- $P_{t+1} = (53, f, skilled), (61, m, unskilled), (69, m, unskilled)\}$

Counterfactual search

NSGA-II algorithm [Deb et al. 2002]



Counterfactual search

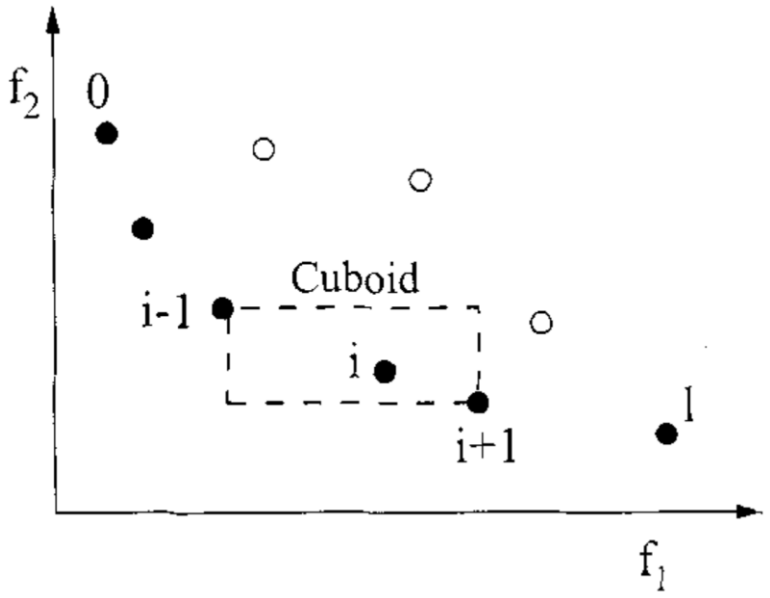
Crowding distance [Deb et al. 2002]

For each objective m

- Sort individuals according to this objective yielding the list $\mathcal{I}$
- Compute the distance

$$\mathcal{I}[i+1].m \; - \mathcal{I}[i-1].m$$

- Normalize this distance by division by the range between the max and min value of this objective



Crowding distance for individual i

- Sum of normalized distances across all objectives

crowding-distance-assignment($\mathcal{I}$)

$l = |\mathcal{I}|$ — number of solutions in $\mathcal{I}$

for each i, set $\mathcal{I}[i]_{\text{distance}} = 0$ — initialize distance

for each objective m

$\mathcal{I} = \text{sort}(\mathcal{I}, m)$ — sort using each objective value

$\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$ — so that boundary points are always selected

for $i = 2$ to $(l-1)$ — for all other points

$$\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)/(f_m^{\max} - f_m^{\min})$$

Crowding distance: example

age	sex	job	o_1	o_2	o_3	o_4
69	m	unskilled	5	5	5	5
62	f	skilled	1	1	1	1
59	m	skilled	2	2	2	2
62	m	skilled	3	3	3	3

(the values for o_1, o_2, o_3, o_4 are arbitrarily made up. To simplify the example o_1, o_2, o_3, o_4 take on the same values.)

Crowding distance for o1

- ∞ (boundary point)
- ∞ (boundary point)
- $(3-1)/(5-1) = 0.5$
- $(5-2)/(5-1) = 0.75$

Crowding distance for all 4 objectives

∞

∞

2

3

Example for counterfactuals

Probability of good credit risk

age	sex	job	housing	savings	amount	duration	purpose
58	f	unskilled	free	little	6143	48	car

Prediction: $f(\mathbf{x}) = 24.2\%$

Target prediction: $f(\mathbf{x}') > 50\%$

age	sex	job	amount	duration	o_1	o_2	o_3	o_4	$\hat{f}(\mathbf{x}')$
		skilled		-20	0	0.108	2	0.036	0.501
		skilled		-24	0	0.114	2	0.029	0.525
		skilled		-22	0	0.111	2	0.033	0.513
-6		skilled		-24	0	0.126	3	0.018	0.505
-3		skilled		-24	0	0.120	3	0.024	0.515
-1		skilled		-24	0	0.116	3	0.027	0.522
-3	m			-24	0	0.195	3	0.012	0.501
-6	m			-25	0	0.202	3	0.011	0.501
-30	m	skilled		-24	0	0.285	4	0.005	0.590
-4	m		-1254	-24	0	0.204	4	0.002	0.506

Example

Interpretation of results

- Credit duration needs to be reduced by 20-25 months to 23-25 months (instead of 48)
- (Slightly) reducing age helps
- Being male helps
- Being skilled helps
- Most likely example according to o4: -4, m, -1254, -24

age	sex	job	amount	duration	o_1	o_2	o_3	o_4	$\hat{f}(\mathbf{x}')$
		skilled		-20	0	0.108	2	0.036	0.501
		skilled		-24	0	0.114	2	0.029	0.525
		skilled		-22	0	0.111	2	0.033	0.513
-6		skilled		-24	0	0.126	3	0.018	0.505
-3		skilled		-24	0	0.120	3	0.024	0.515
-1		skilled		-24	0	0.116	3	0.027	0.522
-3	m			-24	0	0.195	3	0.012	0.501
-6	m			-25	0	0.202	3	0.011	0.501
-30	m	skilled		-24	0	0.285	4	0.005	0.590
-4	m		-1254	-24	0	0.204	4	0.002	0.506

Counterfactual explanations

Advantages

Clear interpretation

- If features change then prediction changes as specified by counterfactuals
- No assumptions like LIME (unclear how far we can extrapolate the local model)
- No approximations like SHAP

Two options for reporting results

- Absolute feature values
- Change in feature values (see example above)

Model-agnostic and data independent

- Only requires access to the model predictions (e.g., via a web API)
- No access to training data required

Counterfactual explanations

Disadvantages

Usually multiple counterfactuals per instance (Rashomon effect)

- This increases complexity, most people prefer simple explanations
- Present all counterfactuals to the user?
- How to select the ones presented to the user?

References

- **Alibi CF.** https://docs.seldon.io/projects/alibi/en/stable/methods/CF.html
- **Dandl**, Susanne, Christoph Molnar, Martin Binder, and Bernd Bischl. "Multi-objective counterfactual explanations." In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*, pp. 448-469. Cham: Springer International Publishing, **2020**.
- **Deb**, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6, no. 2 (**2002**): 182-197.
- **Wachter**, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harv. JL & Tech.* 31 (**2017**): 841.