Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

A (deeper) look at counterfactuals in explainable AI

Annabelle Redelmeier

April 29th, 2021

Outline

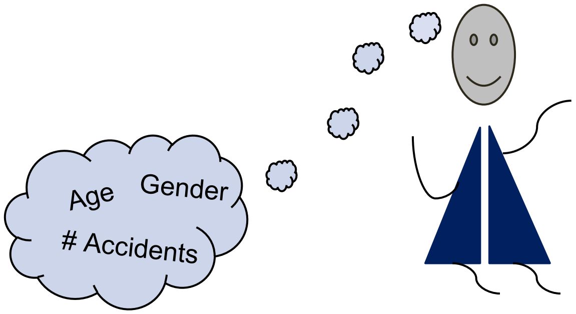
1. Recap (counterfactuals definition, properties, 2 papers).
2. Counterfactuals vs recourse
3. Paper 1: Multi-Objective Counterfactual Explanations (Dandl, Molnar, Binder, Bischl, 2020)
4. Paper 2: FACE: Feasible and Actionable Counterfactual Explanations (Poyiadzi et al., 2020)
5. Conclusion
6. Paper 3: Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making (Joshi et al., 2019)

Recap: Counterfactuals in XAI (*what*)

- A counterfactual explanation takes the form:

  "If Janet had *less car accidents in a year*,
  she would have *cheaper car insurance*".

- If *A*, then *desired outcome.*
- Counterfactuals try to answer the question: *How can we change Janet's features to get a different prediction?*



Recap: Desirable properties of counterfactuals

Sparsity

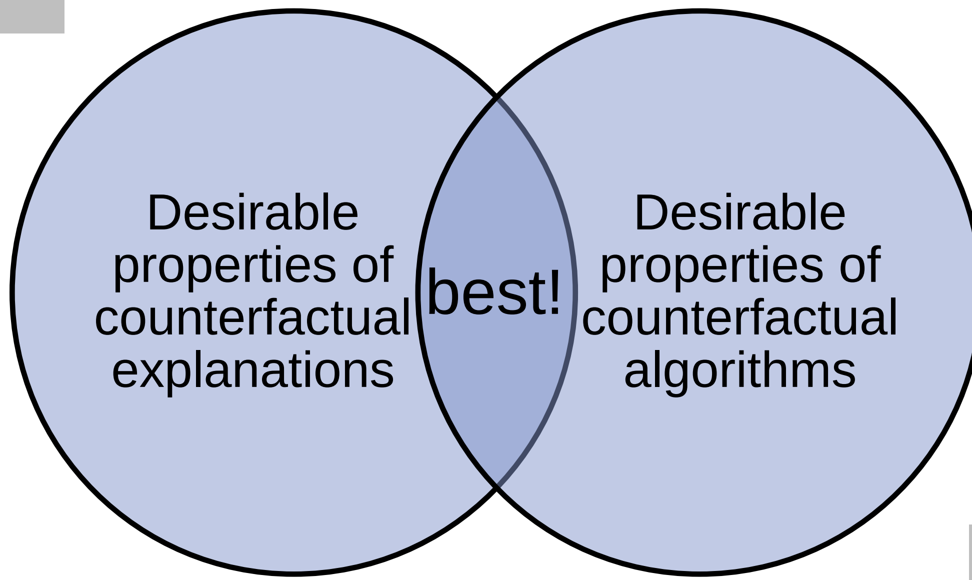
Feasibility

Diversity

Actionability

Proximity

Causality



Flexibility in predictive model $f()$

Flexibility in distance function $d()$

Flexibility in feature types

Speed

Convergence

Recap: 2 papers from last time

- Wachter et al., 2017: first to define counterfactual explanations as solving for the *closest* individual x_i such that $f(x_i) = y'$.

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

- DiCE: extension on *diverse* and *feasible* counterfactuals.

$$L(c_1, \ldots, c_k, x', \lambda_1, \lambda_2) = \frac{1}{k}\sum_{i=1}^{k} \text{yloss}(\hat{f}(\boldsymbol{c}_i), y) + \frac{\lambda_1}{k}\sum_{i=1}^{k} d(c_i, x') - \lambda_2 \det \boldsymbol{K}$$

$$\mathbf{K}_{i,j} = \frac{1}{1+dist(\boldsymbol{c}_i, \boldsymbol{c}_j)}$$

Counterfactuals vs *recourse?*

Improve Wachter by:

- New loss functions;
- Loss → objective function;
- Reformatting the problem → minimizing density-based metric.

Wachter et al., 2017

$$
\begin{aligned}
x^{CFE} \in \operatorname{argmin}_x \quad & d(x, x^F) \\
\text{s.t.} \quad & f(x) \neq f(x^F) \\
& x \in \text{Actionable}
\end{aligned}
$$

"Recourse"

$$
\begin{aligned}
\delta^* \in \operatorname{argmin}_\delta \quad & \operatorname{cost}(\delta, x^F) \\
\text{s.t.} \quad & f(x) \neq f(x^F) \\
& x^{CFE} = x^F + \delta \\
& x \in \text{Actionable} \\
& \delta \in \text{Feasible}
\end{aligned}
$$

Ustun et al., 2019

Counterfactuals vs recourse

	Counterfactuals	**Recourse**
Optimization function	Loss function	Cost function
Algorithm solves for…	Vectors/Individuals (x)	Actions (δ)
Ultimate goal	Explain a model	Solve for actions to achieve “recourse”

Counterfactuals **explain** complex models with the use of **examples…**

... while **recourse** tries to find actions that leads to a better outcome.

A (short) history on recourse

Paper	Description
Ustun et al., 2019: 'Actionable Recourse in Linear Classification'	Counterfactuals → recourse
Joshi et al., 2019: 'Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making'	Recourse with accounting for data distribution
Karimi et al., 2020: 'Algorithmic Recourse: from Counterfactual Explanations to Interventions'	Recourse with causal structural models
Karimi et al., 2020b: 'Algorithmic recourse under imperfect causal knowledge: a probabilistic approach'	Recourse with causal structural models *when structural model is unknown*

Why is the recourse literature so much more limited?

Paper 1: Multi-Objective Counterfactual Explanations (Dandl, Molnar, Binder, Bischl, 2020)

- Loss function → **four-objective** function.
- Each objective satisfies a desirable **counterfactual property**.

1. *Response-proximity:* $f(x')$ is close to the desired outcome y', (objective 1: o_1)

$$o_1(\hat{f}(x'), y') = \begin{cases} 0 & \text{if } \hat{f}(x') \in y' \\ \inf\limits_{y' \in y'} |\hat{f}(x') - y'| & \text{else} \end{cases}$$

2. *Feature-proximity:* x' is close to x^* in the feature space, (objective 2: o_2)

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

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

δ_G is called the Gower distance.

R_j is the range of feature j.

Loss function continuation

- **Two new properties**:
  3. *Sparsity*: better with less changed features, (objective 3: o_3)

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

  4. *Feasibility*: better if counterfactual is **plausible**, (objective 4: o_4)

$$o_4(x', \mathbf{X}^{obs}) = \frac{1}{p} \sum_{j=1}^{p} \delta_G(x'_j, x_j^{[1]})$$

$\mathbf{X}^{obs}$ is the training data

$x^{[1]}$ is the nearest observed data point.

Final loss function

- Loss function:

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

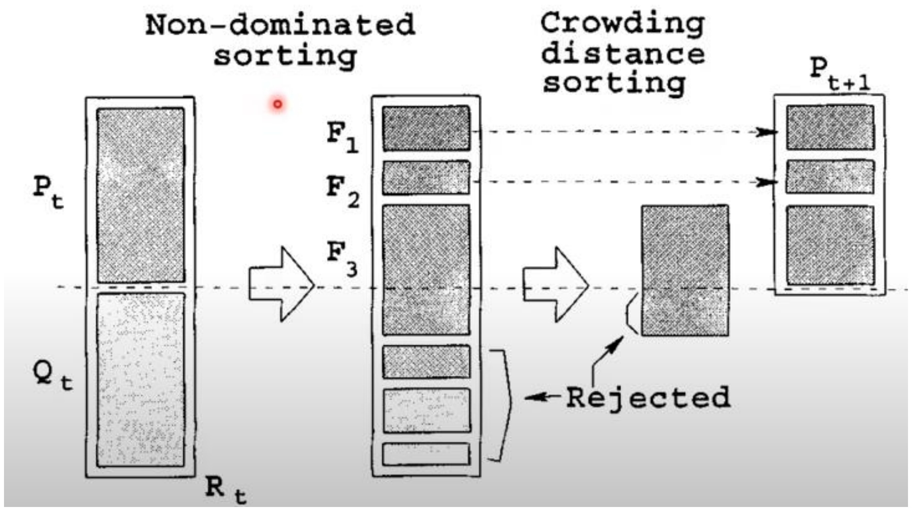
Loss = ([distance to y'], [distance to x], [# features changed], [distance between x and nearest observed data])

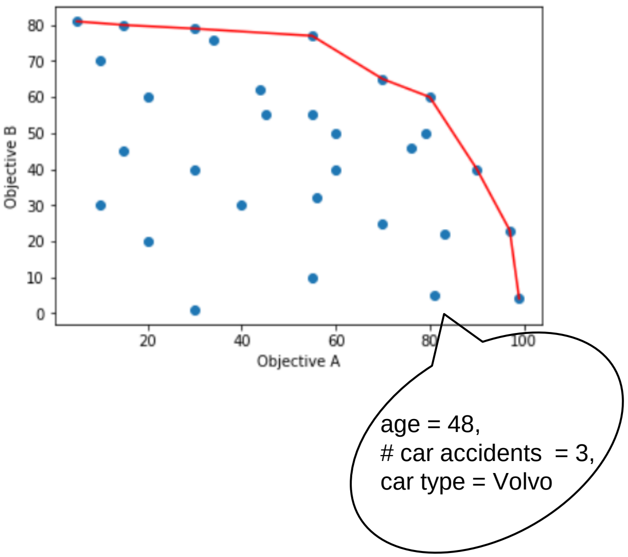
Response-Proximity | Feature-Proximity | Sparsity | Feasibility

- Goal: **Jointly** minimize all four objectives.

How to solve this 4-part optimization?

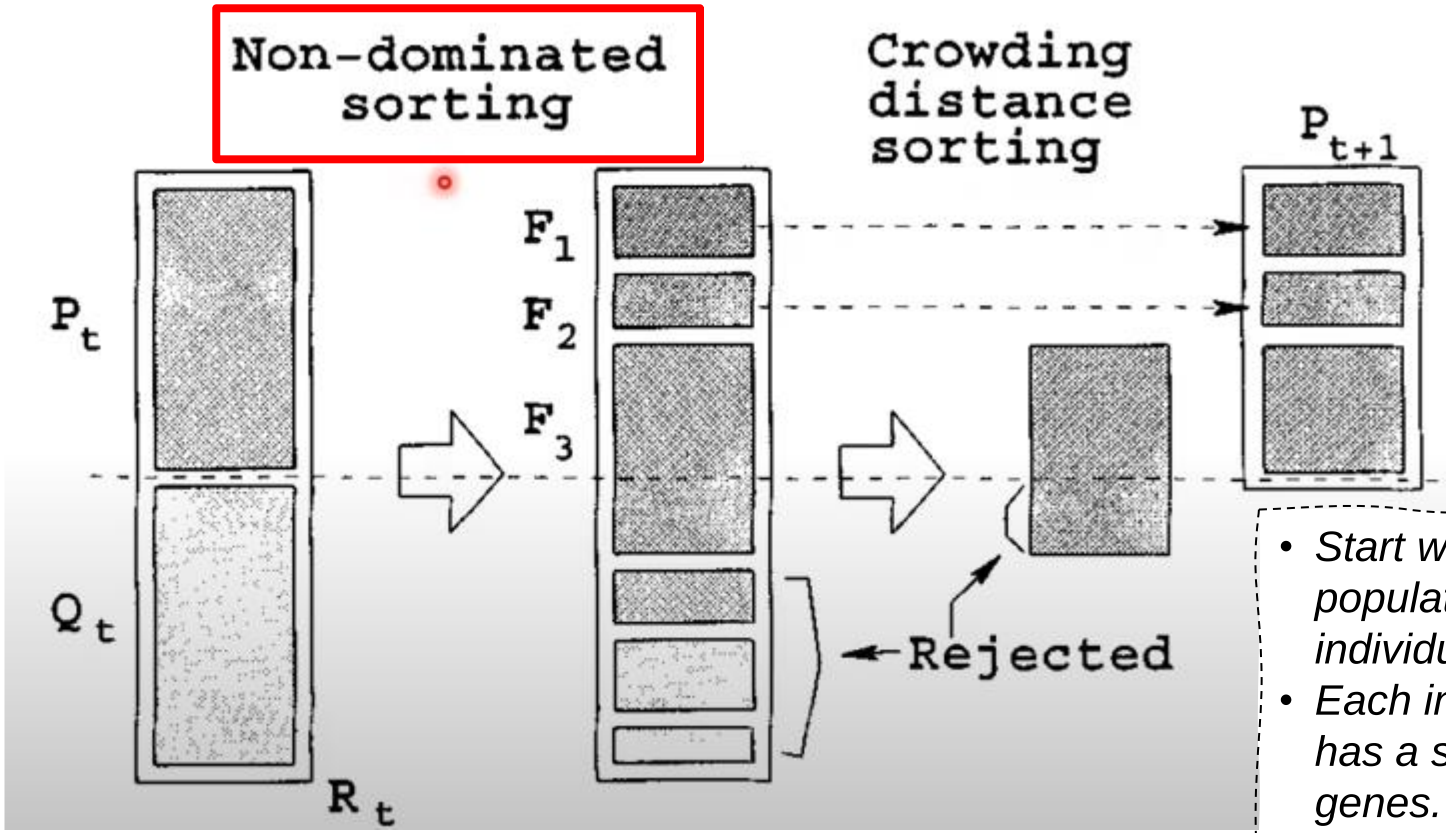
- Nondominated Sorting Genetic Algorithm II (NSGA-II) of course!
- Goal of NSGA-II: **Find the Pareto front for defined objectives** (o_1-o_4)**.**
  - The Pareto front will then be the list of all counterfactuals.





Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation 6.2 (2002): 182-197.

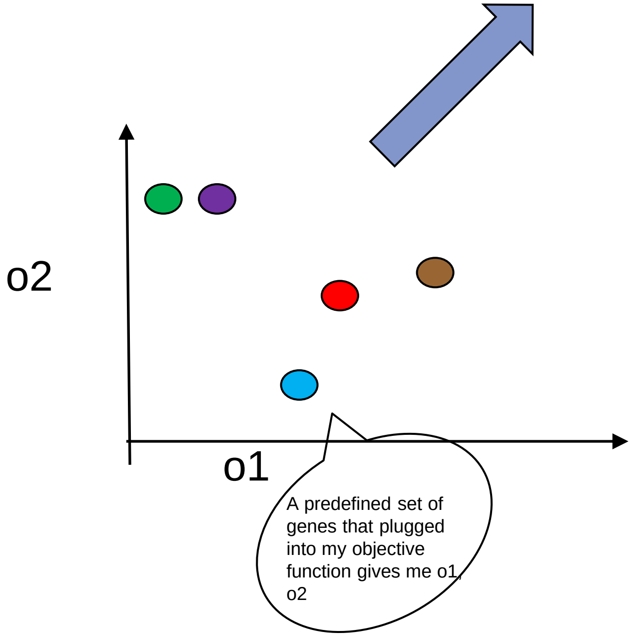
Nondominated Sorting Genetic Algorithm II (NSGA-II)



- *Start with a population of individuals (P_t).*
- *Each individual has a set of genes.*
- *We are maximizing two objectives.*

Nondominated sorting

- Plot the population in terms of the objectives.
- Find which points dominate others:
  - If **at least one** objective is **better**, and no objectives are worse.
  - The point is more **North** and/or **East** than the other.
- For each pair of points, we decide if one point dominates the other. It is possible that no point dominates.

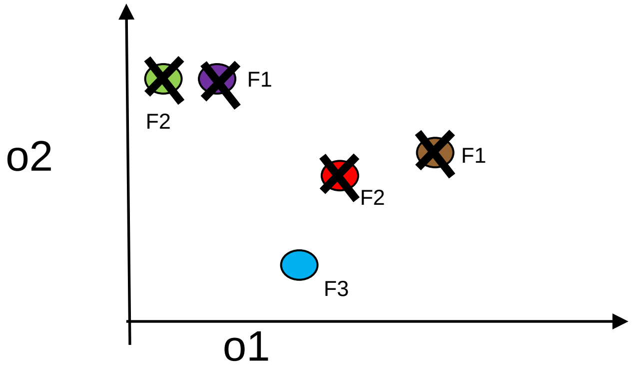


We assume that we want to jointly ***maximize*** objectives.

Point 1	**Point 2**	**Dominates?**
Green	Purple	Purple dominates green
Green	Red	-
Blue	Brown	Brown dominates blue
Purple	Brown	-

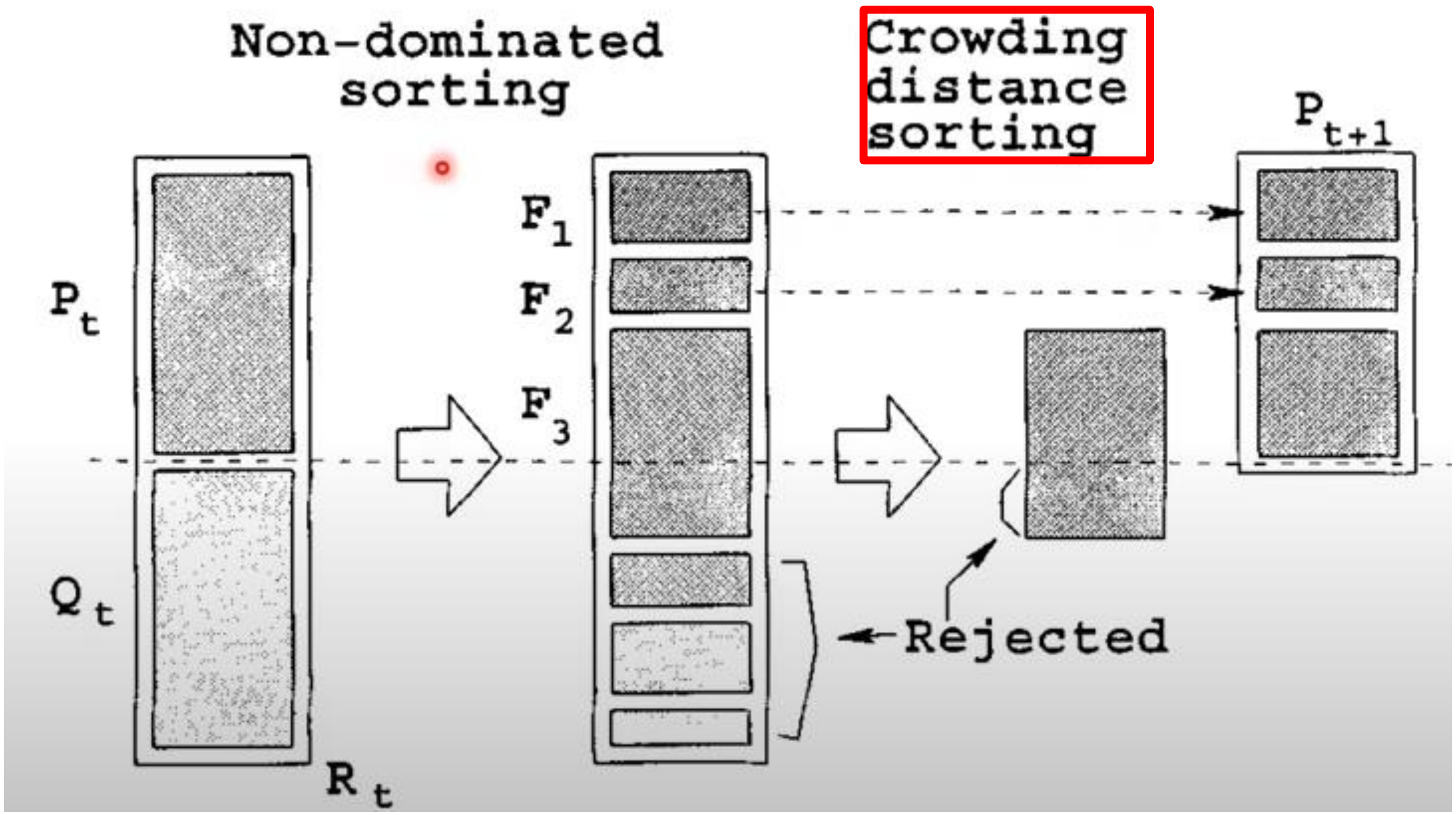
Nondominated sorting

- Front 1: All those not dominated.
- Front 2: Remove those in Front 1 (purple, brown).
  - Find all those that are not dominated (green, red).
  - Those are Front 2.
- Front 3: Remove Front 2 (green, red).
  - Find all those that are not dominated (blue).



Point	How many dominate it?	Front	Front	Front
Green	1		2	2
Purple	0	1	1	1
Red	1		2	2
Blue	2			3
Brown	0	1	1	1

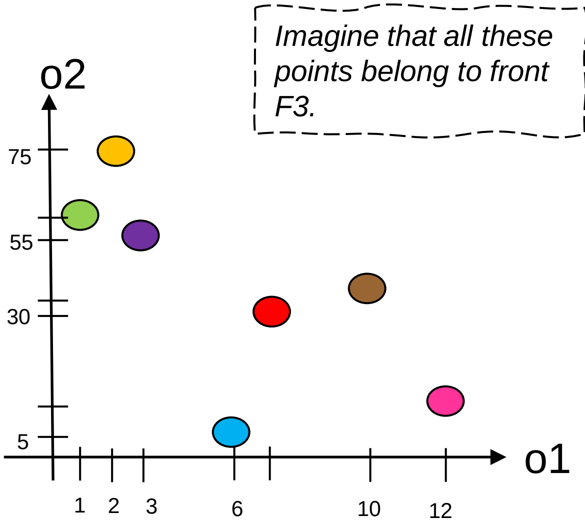
Nondominated Sorting Genetic Algorithm II (NSGA-II)



Crowding Distance Sorting

Crowding distance for objective o and individual i :

$$\frac{[o(i+1)-o(i-1)]}{o(max)-o(min)}$$



For objective **o1**:

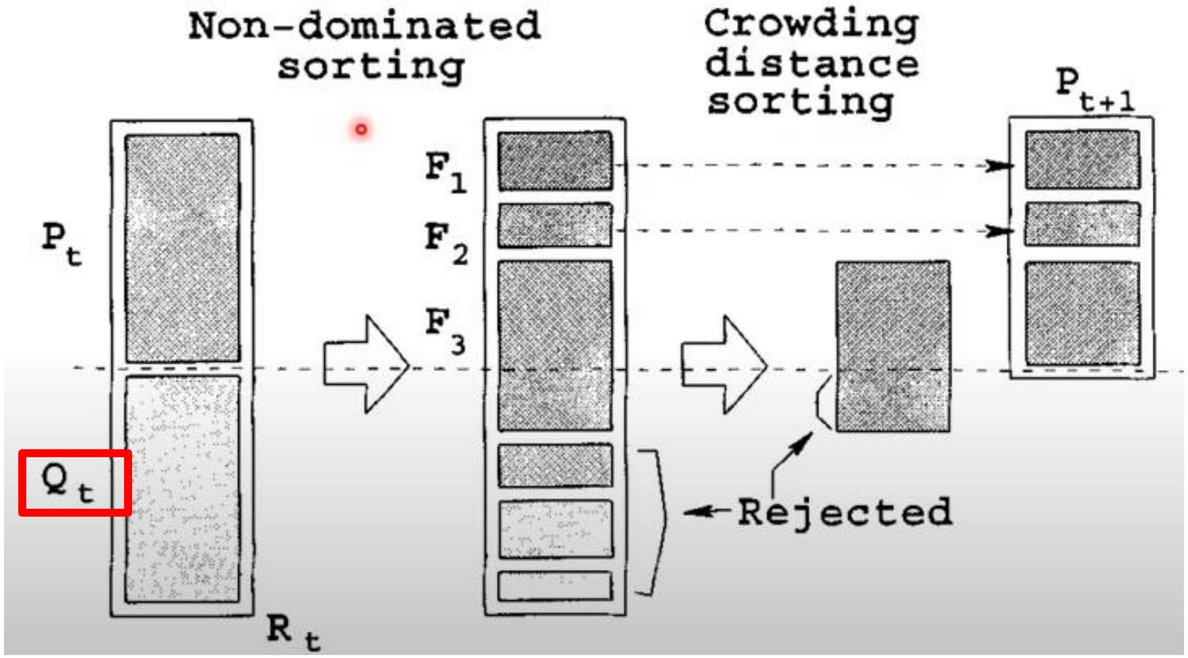
- Max (pink) = 12; min (green) = 1.
- crowding distance(red) = [o1(brown) – o1(blue)] / [o1(pink) – o1(green)]
  = [10 – 6] / [12 – 1] = 0.36
- crowding distance(orange) = [o1(purple) – o1(green)] / [o1(pink) – o1(green)]
  = [3 – 1] / [12 – 1] = 0.18

Repeat for objectives o2, o3, o4. Add all together.

The individuals with the ***larger crowding distance*** *are put into Front 3 first.*

Produce offspring

- To create new set of offspring Q_{t+1}:
- Each **offspring** is created with 3 steps:
  1. Tournament selection
  2. Crossover
  3. Mutation



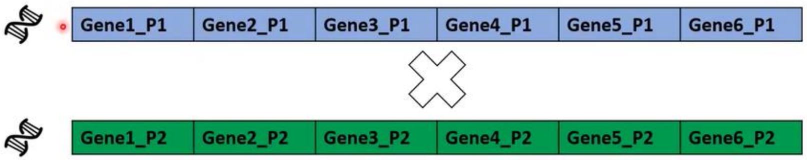
Produce offspring

- **Tournament selection**:
  1. Sample two observations.
  2. Choose the parent with the higher front (or higher crowding distance).
  3. Repeat twice.

- **Crossover**:

  Each parent (P1 and P2) has a vector of **genes**.

  4. Combine **half** genes of P1 and P2 to make the **offspring**:



- **Mutation**:
  4. Randomly change x% of the genes to something else.

= one offspring

Repeat until Q_{t+1} is same size as P_{t+1}.

How does this help us with counterfactuals?

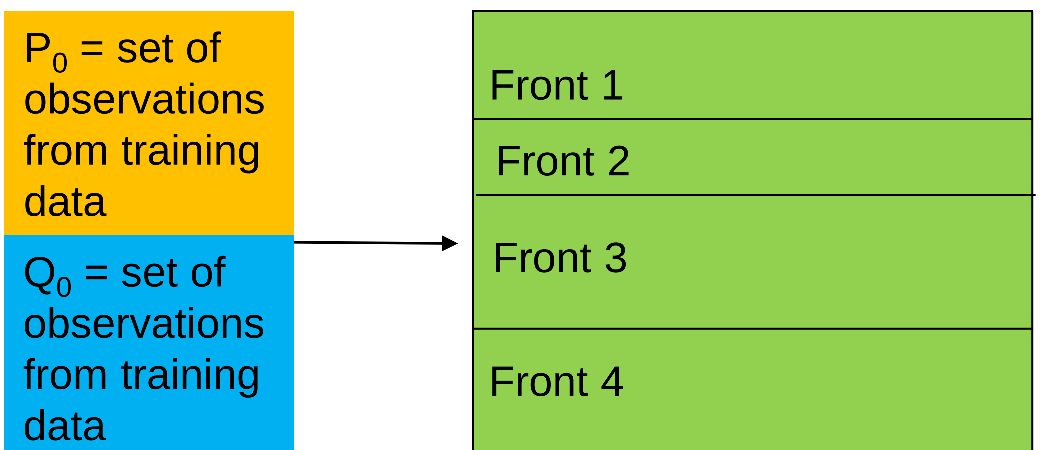
Choose observation: x*

1. Initialize P_0 and Q_0
   1. Measure the feature importance of each feature in x*.
   2. Higher influence → higher probability it is initialized with a different value than that of x*.

P_0 = set of observations from training data

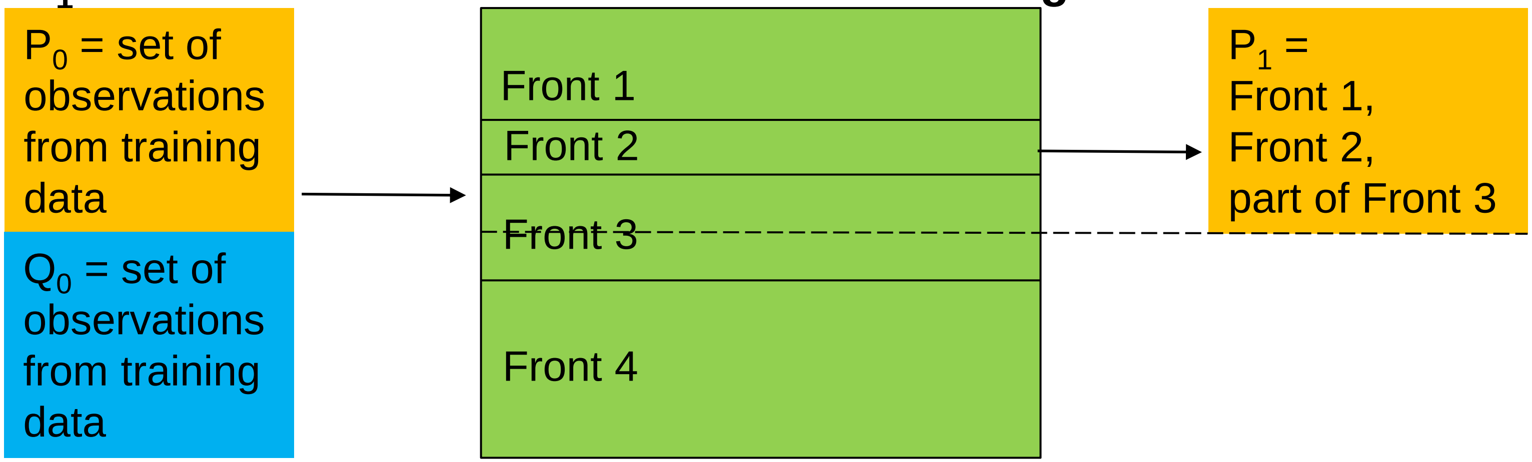
Q_0 = set of observations from training data

2. Sort population into **fronts** based on objectives o1-o4.

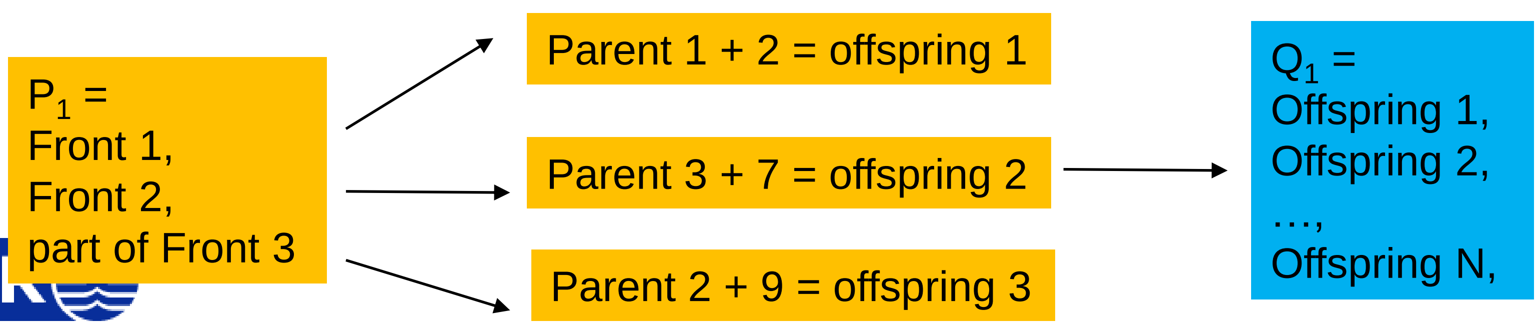


How does this help us with counterfactuals?

3. Fill $\mathbf{P_1}$ with the lowest fronts. Calculate the **crowding number** for last front.



4. Combine 2 two observations. Choose the best parent in terms of front/crowding number. Create an **offspring** based on the parents. **Mutate** some of the features. Repeat until Q_1 is finished.



How does this help us with counterfactuals?

MOC stops after 60 or the performance no longer improves.

P_t =
F_1,
F_2,
and part of
F_3.

Q_t =
Offspring 1,
Offspring 2,
…,
Offspring N,

=

Pareto front = list of counterfactuals

Final thoughts on MOC

- Features can be fixed (for example age, sex).
- Offspring can be **penalized** if far from target prediction (put at bottom of front list).
- **Mutations**: Generate plausible feature values conditional on values of other features (ctree).

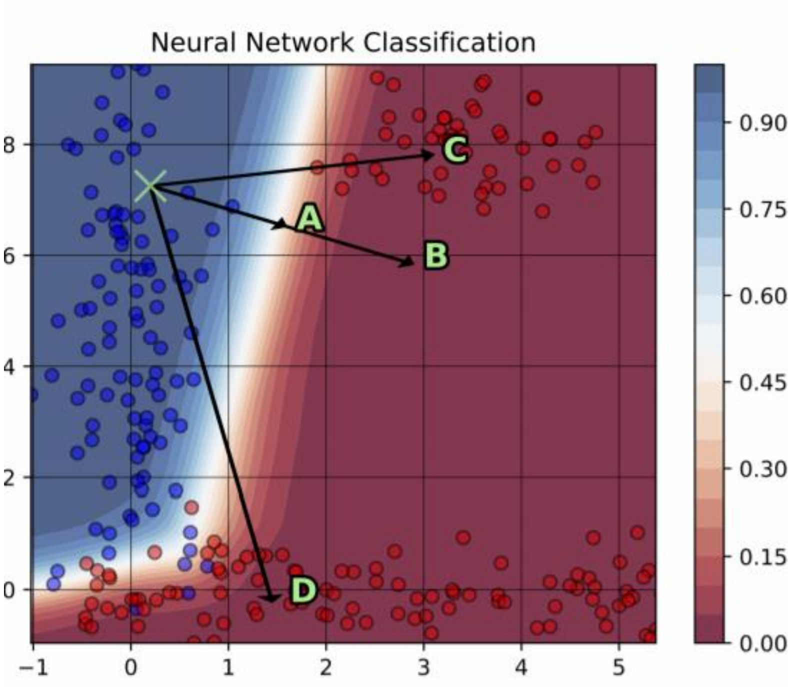
- Slow algorithm!
- Lots of parameters! (e.g., size of generations, probabilities an offspring is mutated, probability a pair of parents recombines, how to initialize population…)
- Can result in **thousands** of counterfactuals! How to display all of them?

NR

Summary of 3 counterfactual algorithms

	MOC	
Properties	• Response-proximity • Feature-proximity • Sparsity • Feasibility • Actionability	
Loss function	$\min_{\mathbf{x}} \left(o_1(\hat{f}(\mathbf{x}), Y'),\ o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs})\right)$	
Optimization	NSGA-II	
Advantages	• NSGA-II could work for additional objectives. • Predictive model doesn't have to be differentiable.	

Paper 2: FACE: Feasible and Actionable Counterfactual Explanations (Poyiadzi et al., 2020)



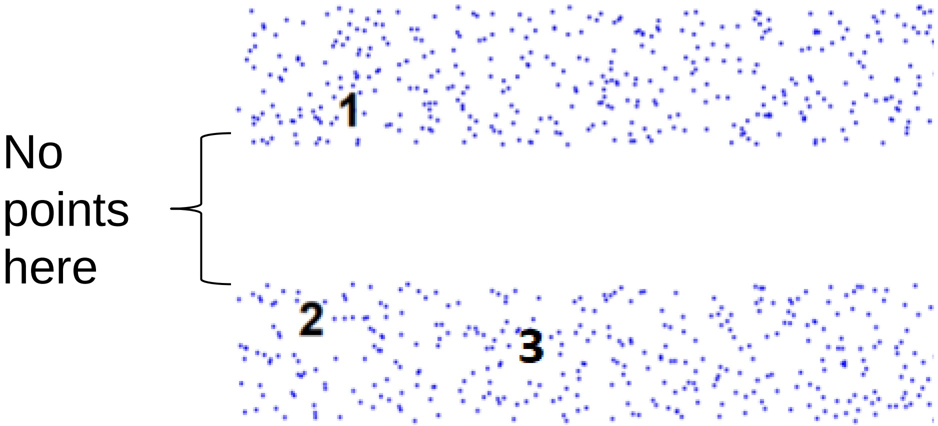
Find set of counterfactuals that are:

- In a **high-density** region (C, D);
  - Representative of the underlying data distribution.
  - "Feasible"
- Can be "accessed" via a **path along the distribution** (D);
  - Give feasible actions to individuals.
  - "Actionable"

FACE: Idea behind the algorithm

- To find observations along “high density paths”, we need a density-based distance (DBD).
- Then:
  - Calculate DBD between the given individual and all other observations in our data set;
  - Return obs with **the smallest DBD**.
- The DBD is based on Sajama and Orlitsky, 2005.

Sajama and Orlitsky, 2005 ideas



The goal is to create a metric such that points with a high-density path between them are **closer**.

Sajama and Orlitsky, 2005 ideas

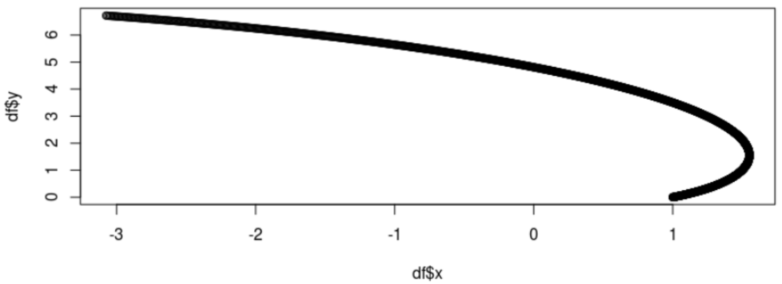
- Let γ be a smooth parametric curve with $x = f(t),\ y = g(t),\ a \le t \le b$.
- Length of the curve is given by:

$$\int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt = \int_a^b \left|\frac{d\gamma(t)}{dt}\right|_2 dt$$

where $|\boldsymbol{x}|_2 = \sqrt{x_1^2 + \ldots, +x_n^2}$

- Example:

$$x = e^t \cos(t), \quad y = e^t \sin(t), \quad 0 \le t \le 2$$



Sajama and Orlitsky, 2005 ideas

- iid points $\{x_1, \ldots, x_n\}$ with probability density function $f(x)$.
- Path length of γ that ***depends on density*** $f(x)$:

$$\int_a^b \left| \frac{d\gamma(t)}{dt} \right|_2 dt \Longrightarrow \Gamma(\gamma, f) = \int_a^b g(f(x)) \left| \frac{d\gamma(t}{dt} \right|_2 dt,$$

$g()$ is a specific function (monotonically decreasing, bounded, etc).

- The DBD metric between two points $\boldsymbol{x}'$ to $\boldsymbol{x}''$ is:

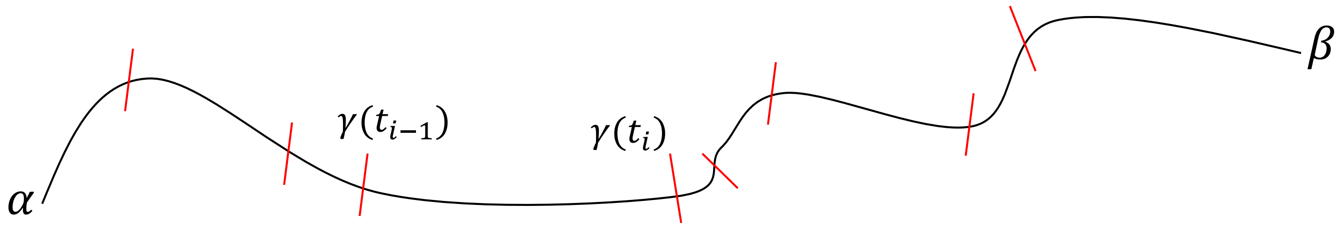
$$d(x', x'', f) = \inf_f \{\Gamma(\gamma, f)\}$$

where γ varies over the set of all paths from x' to x''

Sajama and Orlitsky, 2005 ideas

We can simplify three ways:

1. **Break up** our path into segments**:**



And estimate the path length by summing the segments.

Sajama and Orlitsky, 2005 ideas

2. Estimate the density $f(\boldsymbol{x})$ using a **Kernel Density Estimator,** $K()$.

Estimate the DBD of a path γ between $t_0 = \alpha$ and $t_N = \beta$ with:

$$\Gamma(\gamma; f) = \int_{t=0}^{LE(\gamma)} g(f(\boldsymbol{x})) \left| \frac{d\gamma(t)}{dt} \right|_2 dt, \Rightarrow \hat{\Gamma}(\gamma, K) = \sum_{i=1}^{N} g\left(K\left(\frac{\gamma(t_{i-1}) + \gamma(t_i)}{2}\right)\right) |\gamma(t_{i-1}) - \gamma(t_i)|_2$$

3. Represent the data points as a **graph** with specific **edge weights** and find paths along the graph.

Sajama and Orlitsky, 2005 ideas

- **Construction of graph**:
  - The vertices are: $x_1, \ldots, x_n$.
  - Two nodes are connected by an edge if the distance between them $< \varepsilon$.
  - The weight of the edge is $w(x_i, x_j) = g\left(K\left(\frac{x_i + x_j}{2}\right)\right)\left|x_i - x_j\right|_2$.
- **Estimate DBD metric:**
  - Find all paths from x to y through the graph.
  - Sum the weights of each path.
  - Choose path with smallest sum.

How does this help counterfactuals?

Construct a graph (V, E, W):

Distance can be MAD, Gower distance, L_2, …

For every pair of data points in the data set:

- If the distance is $d\left(x_i, x_j\right) < \varepsilon$:
  - Draw an edge between them;
  - Estimate the weight between them: $g\left(\widehat{K}\left(\frac{x_i + x_j}{2}\right)\right) d(x_i, x_j)$.

$\widehat{K}()$ *is the estimated Kernel Density Estimator.*

$g()$ *is a pre-determined function chosen (see paper for 3 different choices).*

How does this help us with counterfactuals?

To create the set of counterfactuals:

1. Compute N shortest paths based on this **graph** and Dijkstra's algorithm (1956).

closest obs x_1
closest obs x_2
closest obs x_3
closest obs x_N

2. For each x_i:
   1. If $f(x_i) > t_p$
   2. And $K(x_i) > t_d$
      - Add x_i to the list of counterfactuals.

Summary of 3 counterfactual algorithms

	MOC	FACE

Properties	• Response-proximity
	• Feature-proximity
	• Sparsity
	• Feasibility
	• Actionability
		• Response-proximity
		• Feature-proximity
		• Feasibility
		    • *High dense path*
		    • *High dense area*
		• Actionability

Loss function	$\min_{\mathbf{x}} \left(o_1(\hat{f}(\mathbf{x}), Y'),\ o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs})\right)$	
Optimization	NSGA-II	Graph with estimated weights
Advantages	• NSGA-II could work for additional objectives. • Predictive model doesn't have to be differentiable.	• Seems to be the only method focusing on these "high dimensional paths"

Conclusion & Summary

- Two algorithms to solve for counterfactuals:
  - MOC: Jointly minimize a set of objective.
    - Easy to add objectives but slow…
  - FACE: Use density-based distance to find counterfactuals that are “accessible” and “feasible”.
- Counterfactuals vs recourse – Paper 3 in extra slides ☺

List of papers metioned

- Wachter, Sandra and Mittelstadt, Brent and Russell, Chris (2017) Counterfactual explanations without opening the black box: Automated decisions and the GDPRHarv. JL & Tech.31, 841
- Mothilal, Ramaravind K., Amit Sharma, and Chenhao Tan. "Explaining machine learning classifiers through diverse counterfactual explanations." *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020.
- Dandl, Susanne and Molnar, Christoph and Binder, Martin and Bischl, Bernd (2020) Multi-objective counterfactual explanations International Conference on Parallel Problem Solving from Nature
- Barocas, Solon and Selbst, Andrew D and Raghavan, Manish (2020)
- Ch 6.1 Interpretable ML book by Dandl and Molnar
- Karimi, Amir-Hossein, et al. "Model-agnostic counterfactual explanations for consequential decisions." *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020.

List of papers (cont.)

- Poyiadzi, Rafael, et al. "FACE: feasible and actionable counterfactual explanations." *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020.
- Orlitsky, Alon. "Estimating and computing density based distance metrics." *Proceedings of the 22nd international conference on Machine learning*. 2005.
- Ustun, Berk, Alexander Spangher, and Yang Liu. "Actionable recourse in linear classification." *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019.
- Joshi, Shalmali, et al. "Towards realistic individual recourse and actionable explanations in black-box decision making systems." *arXiv preprint arXiv:1907.09615* (2019).
- Karimi, Amir-Hossein, Bernhard Schölkopf, and Isabel Valera. "Algorithmic recourse: from counterfactual explanations to interventions." *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021.
- Karimi, Amir-Hossein, et al. "Algorithmic recourse under imperfect causal knowledge: a probabilistic approach." *arXiv preprint arXiv:2006.06831* (2020).

Paper 3: Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making (Joshi et al., 2019)

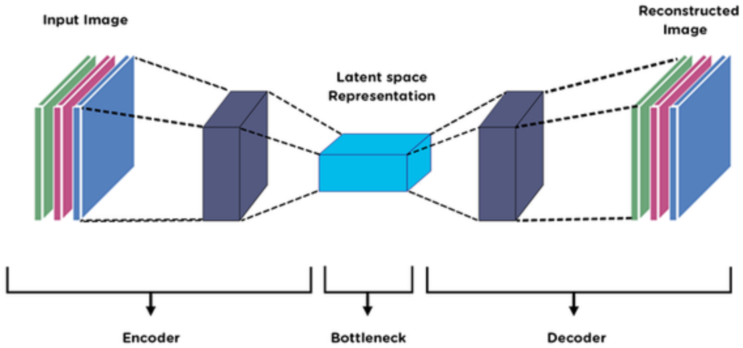
Goal:

1. Characterize the data distribution:
    - Variational Auto-Encoders (VAEs)
    - Generative Adversarial Networks (GANs)

2. Find actions leading to **recourse**:
    - Find the shortest path along the data manifold.

Autoencoders

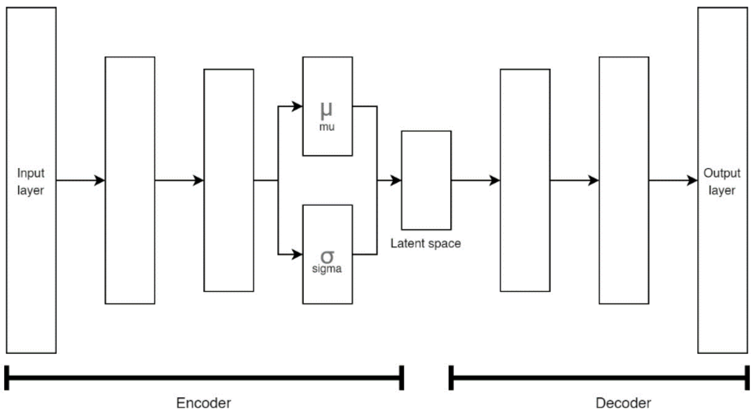
- Encoder:
  - Run through a NN to compress the data.
- Decoder:
  - Reconstructs data
- Loss function:
  - Compares the output to the input.



$$\mathcal{L}\left(x, \hat{x}\right) + regularizer$$

Variational Auto-Encoders (VAEs)

- Encoder:
  - Run through a NN to compress the data.
  - Map to **mean** and **sd** vector.
- Decoder:
  - Take a sample from a multivariate Gaussian with **mean** and **sd**.
  - Pass through the decoder.



- Loss function:
  - Includes the **reconstruction** loss and the **KL divergence:**

$$\left(\mathbb{E}_{z\sim q_x}\left(-\frac{||x-f(z)||^2}{2c}\right)-KL(q_x(z),p(z))\right)$$

How does this help with counterfactuals?

1. Estimate a variational autoencoder.
   - Denote the encoder $\mathcal{F}_{\psi} : R^d \rightarrow R^k$
   - Denote the decoder $\mathcal{G}_{\theta} : R^k \rightarrow R^d$

2. Minimize the loss function with respect to x:

$$L(x, x', \lambda) = \text{yloss}(\hat{f}(\mathcal{G}_{\theta}(\mathcal{F}(x)), y)) + \lambda \cdot \text{cost}(x', \mathcal{G}_{\theta}(\mathcal{F}(x)))$$

3. The set of actions is:
   - $\{(\delta_i,\ x_i^* - x_i')\ \forall\ \delta\ s.t. |x_i^* - x_i'| > 0\}$

Summary of 3 counterfactual algorithms

	MOC	FACE	Recourse with VAE

Properties	• Response-proximity
	• Feature-proximity
	• Sparsity
	• Feasibility
	• Actionability
		• Response-proximity
		• Feature-proximity
		• Feasibility
		  • *High dense path*
		  • *High dense area*
		• Actionability
			• Response-proximity
			• Feature-proximity
			• Feasibility

Loss function	$\min_{\mathbf{x}} \left(o_1(\hat{f}(\mathbf{x}), Y'), \, o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs}) \right)$		$\mathrm{yloss}(\hat{f}(\mathcal{G}_\theta(\mathcal{F}(x)), y)) + \lambda \cdot \mathrm{cost}(x', \mathcal{G}_\theta(\mathcal{F}(x)))$
Optimization	NSGA-II	Graph with estimated weights	Gradient descent along manifold
Advantages	• NSGA-II could work for additional objectives. • Predictive model doesn't have to be differentiable.	• Focuses on these "high dimensional paths"	• Takes into account data distribution.