Interpretable Machine Learning

Local Explanations: Adversarial Examples

Learning goals

- Understand the definition of ADEs
- Understand first methods that generate ADEs
- Discuss potential causes of ADEs and standard defenses against them

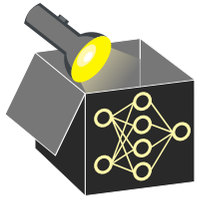
Interpretable Machine Learning

Local Explanations: Adversarial Examples

Learning goals

- Understand the definition of ADEs
- Understand first methods that generate ADEs
- Discuss potential causes of ADEs and standard defenses against them

ADVERSARIAL MACHINE LEARNING

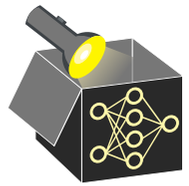


- What happens if a computer system gets an erroneous input?
- Even worse:
  What happens if someone feeds in a malicious input on purpose to attack a system?

⇝ **Robustness** is important to ensure a safe service!

- **Adversarial ML** studies the robustness of machine learning (ML) algorithms to malicious input
- Two different kinds of attacks:
  - **Evasion attacks** mislead an employed ML model with manipulated inputs (our focus)
  - **Data Poisoning**: Malicious inputs to the training dataset

ADVERSARIAL MACHINE LEARNING



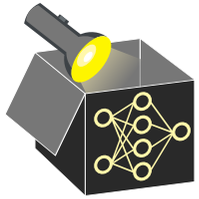
- What happens if a computer system gets an erroneous input?
- Even worse:
  What happens if someone feeds in a malicious input on purpose to attack a system?

⇝ **Robustness** is important to ensure a safe service!

- **Adversarial ML** studies the robustness of machine learning (ML) algorithms to malicious input
- Two different kinds of attacks:
  - **Evasion attacks** mislead an employed ML model with manipulated inputs (our focus)
  - **Data Poisoning**: Malicious inputs to the training dataset

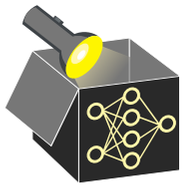
Interpretable Machine Learning – 1 / 14

ADVERSARIAL EXAMPLES



- **Informal Definition**: An ADE is an input to a model that is deliberately designed to "fool" the model into misclassifying it
- Even possible with low generalization error
- Both deep learning models (e.g., CNNs) and classical ML can be vulnerable to such attacks
- ADEs created from a real data observation **x** can be indistinguishable from **x** by a human observer
- Since the model misclassifies this input, it does not seem to have a real understanding of the underlying concepts of the provided inputs

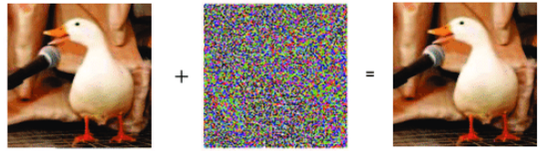
ADVERSARIAL EXAMPLES

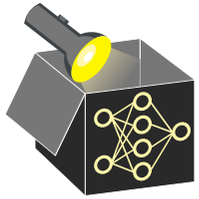


- **Informal Definition**: An ADE is an input to a model that is deliberately designed to "fool" the model into misclassifying it
- Even possible with low generalization error
- Both deep learning models (e.g., CNNs) and classical ML can be vulnerable to such attacks
- ADEs created from a real data observation **x** can be indistinguishable from **x** by a human observer
- Since the model misclassifies this input, it does not seem to have a real understanding of the underlying concepts of the provided inputs

EXAMPLES: MODEL-ATTACKS

Gong & Poellabauer 2018

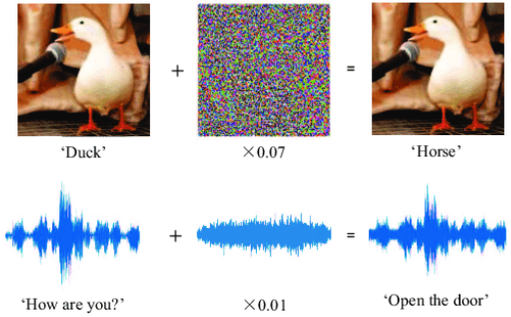


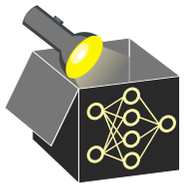


- Is this a duck or a horse?
- Small (hard-to-see) noise can change the prediction

EXAMPLES: MODEL-ATTACKS

POELLABAUER_2018

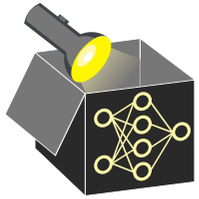




- Is this a duck or a horse?
- Small (hard-to-see) noise can change the prediction

EXAMPLES: IMAGE DATA

Eykholt et al. (2018) | Athalye et al. (2018)

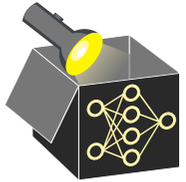


- Stop signs can be missclassified e.g., because of graffiti
- With some well-placed patches, the model identifies it as a “right of way” sign

- 3D-print of a turtle
- Misclassified as a rifle (from every angle)
- Video: MITCSAIL (2017)

EXAMPLES: IMAGE DATA

EYKHOLT_2018 | ATHALYE_2018



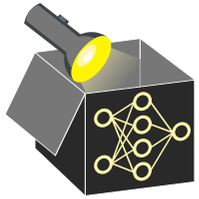
- Stop signs can be missclassified e.g., because of graffiti
- With some well-placed patches, the model identifies it as a “right of way” sign

- 3D-print of a turtle
- Misclassified as a rifle (from every angle)
- Video: MITCSAIL 2017

Interpretable Machine Learning – 4 / 14

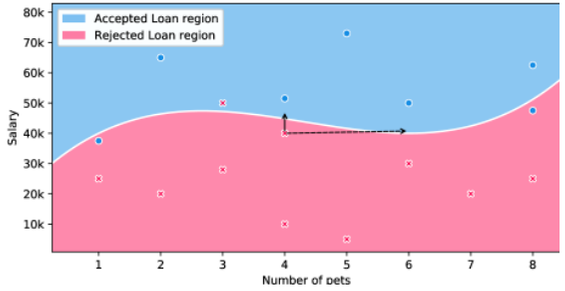
EXAMPLE: TABULAR DATA

Ballet (2019)



What is imperceptibility on tabular data?

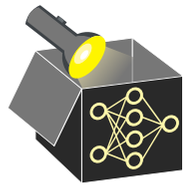
- Idea: experts focus on the most important features in their judgment
- An ADE arises from manipulating features the model deems important but experts do not



Decision boundary of a classifier deciding loan applications. ADE via “number of pets”

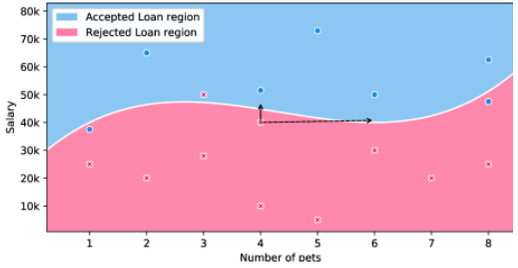
EXAMPLE: TABULAR DATA

BALLET_2019



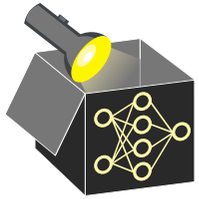
What is imperceptibility on tabular data?

- Idea: experts focus on the most important features in their judgment
- An ADE arises from manipulating features the model deems important but experts do not



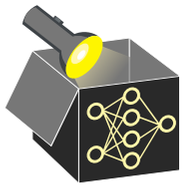
Decision boundary of a classifier deciding loan applications. ADE via “number of pets”

ADE AND INTERPRETABILITY



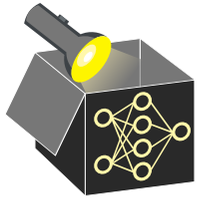
1. ADEs show where models fail $\rightsquigarrow$ improved model understanding
2. Because of ADEs, we need more interpretability
3. Interpretation can lead to robustness against ADEs
4. Explanations can be used to construct ADEs (e.g., see numer of pets on previous slide)

ADE AND INTERPRETABILITY



1. ADEs show where models fail $\rightsquigarrow$ improved model understanding
2. Because of ADEs, we need more interpretability
3. Interpretation can lead to robustness against ADEs
4. Explanations can be used to construct ADEs (e.g., see numer of pets on previous slide)

FORMAL DEFINITION



Adversarial Input

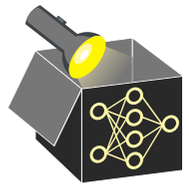
Let $\epsilon > 0$, $f : \mathcal{X} \to \mathcal{Y}$ be an ML model and $\mathbf{x} \in \mathcal{X}$ be a real data point that is correctly classified: $f(\mathbf{x}) = y_{\mathbf{x},true}$.

We call $\mathbf{a_x}$ an **adversarial input** to $\mathbf{x}$ if:

$$\|\mathbf{a_x} - \mathbf{x}\| < \epsilon \text{ and } f(\mathbf{a_x}) \neq y_{\mathbf{a_x},true} = f(\mathbf{x})$$

- $\mathbf{a_x}$ an is a data point close to a real, correctly classified input that is misclassified
- $\mathbf{a_x}$ is called **targeted** if the class it is assigned to is determined $f(\mathbf{a_x}) = y'$ with y' being a desired prediction
- Can be generalized to regression problems

FORMAL DEFINITION



Adversarial Input

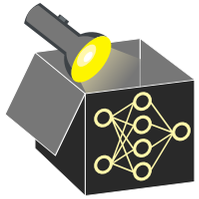
Let $\epsilon > 0$, $f : \mathcal{X} \to \mathcal{Y}$ be an ML model and $\mathbf{x} \in \mathcal{X}$ be a real data point that is correctly classified: $f(\mathbf{x}) = y_{\mathbf{x},true}$.

We call $\mathbf{a_x}$ an **adversarial input** to $\mathbf{x}$ if:

$$\|\mathbf{a_x} - \mathbf{x}\| < \epsilon \text{ and } f(\mathbf{a_x}) \neq y_{\mathbf{a_x},true} = f(\mathbf{x})$$

- $\mathbf{a_x}$ is a nearby point to a real, correctly classified input that is misclassified
- $\mathbf{a_x}$ is called **targeted** if the class it is assigned to is determined $f(\mathbf{a_x}) = y'$ with y' being a desired prediction
- Can be generalized to regression problems

WHY DO ADE EXIST?



Non-exhaustive list of hypotheses:

1. Low-probability spaces hypotheses: ADEs live in low-probability yet dense spaces in the data manifold that are not well represented in the training samples

▸ Szegedy et al. (2013)

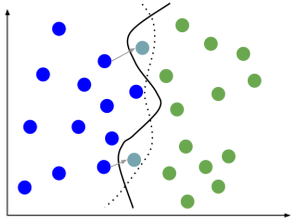
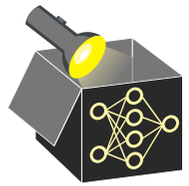


Figure: Binary classification example (dark blue vs. green dots). Dotted line represents the true decision boundary, bold line the trained one. Low probability space close to decision boundary allow for adversarial examples (turquoise dot).

WHY DO ADE EXIST?



Non-exhaustive list of hypotheses:

1. Low-probability spaces hypotheses: ADEs live in low-probability yet dense spaces in the data manifold that are not well represented in the training samples

▸ Szegedy 2013

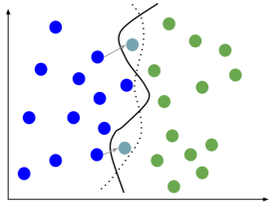
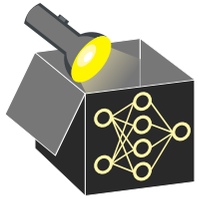


Figure: Binary classification example (dark blue vs. green dots). Dotted line represents the true decision boundary, bold line the trained one. Low probability space close to decision boundary allow for adversarial examples (turquoise dot).

WHY DO ADE EXIST?



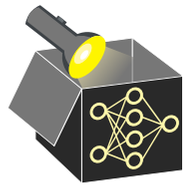
Non-exhaustive list of hypotheses:

2. Linearity hypotheses (most popular):
Adversarial examples are omnipresent in the data manifold
$\rightsquigarrow$ occur, because commonly used models often show linear behavior
$\rightsquigarrow$ small changes of ϵ in every feature cause a change of $\epsilon \|\|\|_1$ in prediction

▸ Goodfellow et al. (2014)

Example: linear model
Original: $f(\mathbf{x}) = \mathbf{x}^T$
Small changes: $f(\mathbf{x} + \epsilon) = (\mathbf{x} + \epsilon)^T$
Difference: $f(\mathbf{x} + \epsilon) - f(\mathbf{x}) = \epsilon \cdot$

WHY DO ADE EXIST?



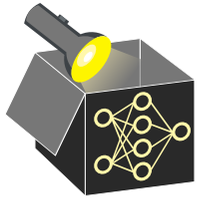
Non-exhaustive list of hypotheses:

2. Linearity hypotheses (most popular):
Adversarial examples are omnipresent in the data manifold
$\rightsquigarrow$ occur, because commonly used models often show linear behavior
$\rightsquigarrow$ small changes of ϵ in every feature cause a change of $\epsilon \|\theta\|_1$ in prediction

▸ Goodfellow 2014

Example: linear model
Original: $f(\mathbf{x}) = \mathbf{x}^T\theta$
Small changes: $f(\mathbf{x} + \epsilon) = (\mathbf{x} + \epsilon)^T\theta$
Difference: $f(\mathbf{x} + \epsilon) - f(\mathbf{x}) = \epsilon \cdot \theta$

WHY DO ADE EXIST?



Non-exhaustive list of hypotheses:

3. The boundary tilting hypothesis: Linearity is neither necessary nor sufficient to explain ADEs
$\rightsquigarrow$ ADEs mostly result from overfitting the sampled manifold ▸ Tanay and Griffin (2016)

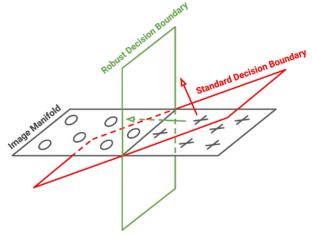
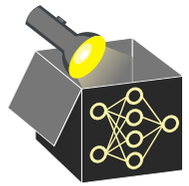


Figure: Linear binary classification example. Due to overfitting the decision boundary (red) is close to the manifold of the training data. Techniques like regularization could help to make the decision boundary more robust (green). ▸ Kim et al. (2019)

WHY DO ADE EXIST?



Non-exhaustive list of hypotheses:

3. The boundary tilting hypothesis: Linearity is neither necessary nor sufficient to explain ADEs
$\rightsquigarrow$ ADEs mostly result from overfitting the sampled manifold ▸ Griffin 2016

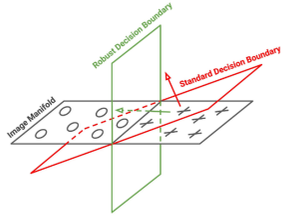
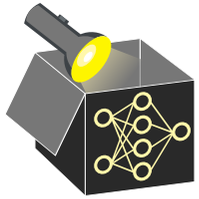


Figure: Linear binary classification example. Due to overfitting the decision boundary (red) is close to the manifold of the training data. Techniques like regularization could help to make the decision boundary more robust (green). ▸ Kim 2019

WHY DO ADE EXIST?

Non-exhaustive list of hypotheses:

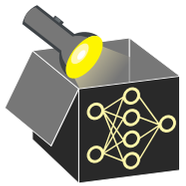
4. Human-centric hypotheses: ML models make use of predictive but non-robust features – meaning they are highly correlated with the prediction target, but not used by humans ▸ Ilyas et al. (2019)



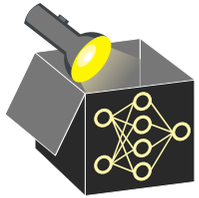
WHY DO ADE EXIST?

Non-exhaustive list of hypotheses:

4. Human-centric hypotheses: ML models make use of predictive but non-robust features – meaning they are highly correlated with the prediction target, but not used by humans ▸ Ilyas 2019



WAYS TO GENERATE ADE



Different ways for constructing ADEs: There exist various ways in the literature to generate ADEs for a given model in feasible time

- Formulate the search for ADEs as an **optimization problem**, e.g.

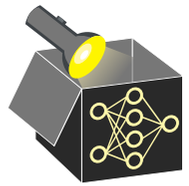
$$\underset{\mathbf{x}' \in \mathcal{X}}{\operatorname{argmin}} \underbrace{\|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}}}_{\text{minmize}} - \lambda \underbrace{\|f(\mathbf{x}') - y'\|_{\mathcal{Y}}}_{\text{maximize}}$$

- Use **sensitivity analysis** to identify features that influence the target class
- Train a generative adversarial network (GAN) ▸ Goodfellow et al. (2014)

Moreover, depending on the attacker's model access, we can distinguish between

- **Full-access attacks**: the attacker has full access to the internals of the model
- **Black-box attacks**: the attacker can only query the model on some inputs and receives the model's outputs

WAYS TO GENERATE ADE



Different ways for constructing ADEs: There exist various ways in the literature to generate ADEs for a given model in feasible time

- Formulate the search for ADEs as an **optimization problem**, e.g.

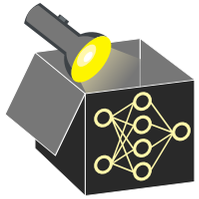
$$\underset{\mathbf{x}' \in \mathcal{X}}{\operatorname{argmin}} \underbrace{\|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}}}_{\text{minmize}} - \lambda \underbrace{\|f(\mathbf{x}') - y'\|_{\mathcal{Y}}}_{\text{maximize}}$$

- Use **sensitivity analysis** to identify feats that influence the target class
- Train a generative adversarial network (GAN) ▸ Goodfellow 2014

Moreover, based on the attacker's model access, we can distinguish between

- **Full-access attacks**: attacker has full access to internals of the model
- **Black-box attacks**: the attacker can only query the model on some inputs and receives the model's outputs

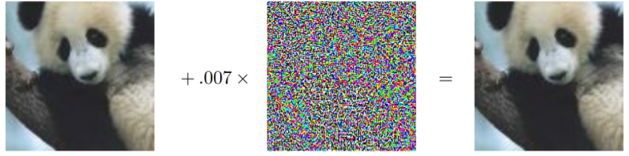
FAST-GRADIENT-SIGN-METHOD (FGSM) ▸ Goodfellow et al. (2015)



- FGSM is based on the linearity hypothesis
- FGSM finds ADEs from:

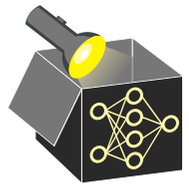
$$a_{\mathbf{x}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y_{\mathbf{x},true}))$$

where $\text{sign}(\nabla_{\mathbf{x}} J(\theta, x, y_{\mathbf{x},true}))$ describes the component-wise signum of the gradient of cost function J in $\mathbf{x}$ with true label $y_{\mathbf{x},true}$



FAST-GRADIENT-SIGN-METHOD (FGSM)

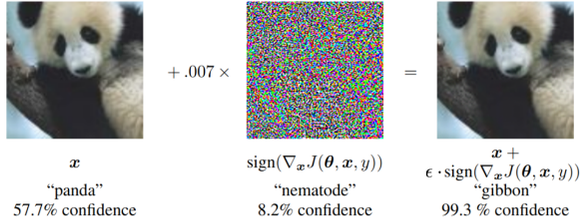
▸ GOODFELLOW_2015



- FGSM is based on the linearity hypothesis
- FGSM finds ADEs from:

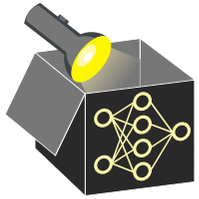
$$a_{\mathbf{x}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y_{\mathbf{x},true}))$$

where $\text{sign}(\nabla_{\mathbf{x}} J(\theta, x, y_{\mathbf{x},true}))$ describes the component-wise signum of the gradient of cost function J in $\mathbf{x}$ with true label $y_{\mathbf{x},true}$



NOTES ON FGSM

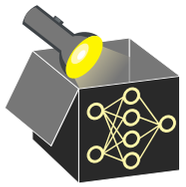
Goodfellow et al. (2015) Lyu et al. 2015



- FGSM works particularly well for linear(-like) models in high-dimensional spaces,
  e.g., LSTMs, logistic regressions or CNNs with ReLU activations
- Not every $\mathbf{a_x}$ generated by FGSM is an ADE, especially if ϵ is too small
- FGSM attacks can be also generated without model access by approximating the gradient,
  e.g. with finite difference methods
- The notion of similarity in FGSM is based on $\|\cdot\|_\infty \rightsquigarrow$ there are generalizations of FGSM to other norms

NOTES ON FGSM

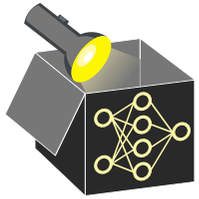
GOODFELLOW_2015 LYU_2015



- FGSM works particularly well for linear(-like) models in high-dim. spaces, e.g., LSTMs, logistic regressions or CNNs with ReLU activations
- Not every $\mathbf{a_x}$ generated by FGSM is an ADE, especially if ϵ is too small
- FGSM attacks can be also generated without model access by approximating the gradient,
  e.g. with finite difference methods
- The notion of similarity in FGSM is based on $\|\cdot\|_\infty \rightsquigarrow$ there are generalizations of FGSM to other norms

BLACK-BOX ATTACKS WITH SURROGATES

Papernot et al. (2016)



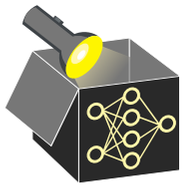
- So far, we assumed full access to the predictive model
- Black-box attacks only assume query-access
- Large risk of attacks since often one can query predictive models many times

1. Query the model you aim to attack as often as allowed on data similar to the training data
2. Use the labeled data you received to train a surrogate model
3. Generate ADEs for the surrogate model
4. Use these ADEs to attack the original model

$\rightsquigarrow$ Known as the **transferability** of ADEs.

BLACK-BOX ATTACKS WITH SURROGATES

PAPERNOT_2016

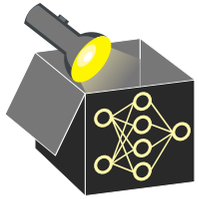


- So far, we assumed full access to the predictive model
- Black-box attacks only assume query-access
- High attack risk since often one can query predictive models many times

1. Query the model you aim to attack as often as allowed on data similar to the training data
2. Use the labeled data you received to train a surrogate model
3. Generate ADEs for the surrogate model
4. Use these ADEs to attack the original model

$\rightsquigarrow$ Known as the **transferability** of ADEs.

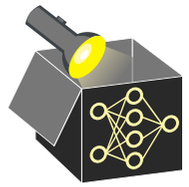
DEFENSES AGAINST ADE



There are several ways to protect your network against such attacks – we distinguish between two broad types of defenses, differing in the position in which they act

- **Guards** act on the inputs a model receives
  - **Detect anomalies:** e.g., statistical testing, or discriminator networks from GANs
  - **Conduct transformations** on inputs (e.g. PCA)
- **Defense by design** act on the model itself
  - **Adversarial training**: train model on adversarials
  - **Architectural defenses**: e.g., removing low predictive features from the model

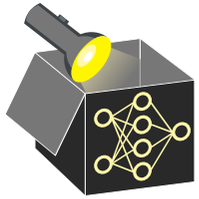
DEFENSES AGAINST ADE



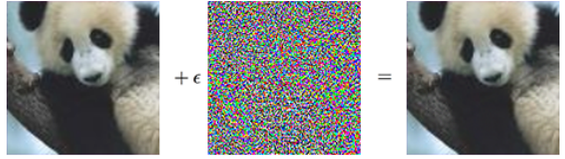
There are several ways to protect your network against such attacks – we distinguish between two broad types of defenses, differing in the position in which they act

- **Guards** act on the inputs a model receives
  - **Detect anomalies:** e.g., statistical testing, or discriminator networks from GANs
  - **Conduct transformations** on inputs (e.g. PCA)
- **Defense by design** act on the model itself
  - **Adversarial training**: train model on adversarials
  - **Architectural defenses**: e.g., removing low predictive features from the model

SUMMARY

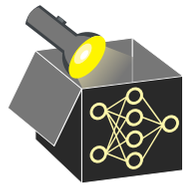


- ADEs are not explanations themselves but are conceptually connected to them
- ADEs can be generated in diverse settings $\rightsquigarrow$ crucial modeling decisions are the distance measure, the local environment, and the target level (model or process)
- There are various hypotheses on the existence of ADEs which also motivate different defense strategies

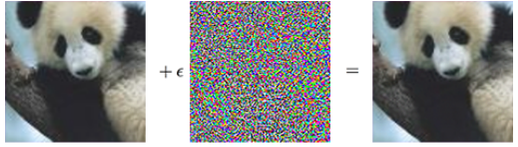


▸ Goodfellow et al. (2017)

SUMMARY



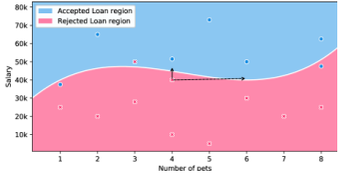
- ADEs are not explanations themselves, but conceptually related to them
- ADEs can be generated in diverse settings $\rightsquigarrow$ crucial modeling decisions are the distance measure, the local environment, and the target level (model or process)
- There are various hypotheses on the existence of ADEs which also motivate different defense strategies



▸ Goodfellow 2017

Interpretable Machine Learning

Adversarial Examples and Counterfactual Explanations

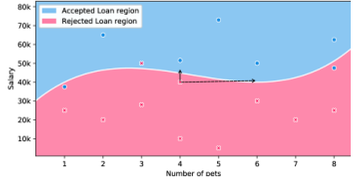


Learning goals

- Compare adversarial examples to counterfactual explanations
- See an example where both coincident

Interpretable Machine Learning

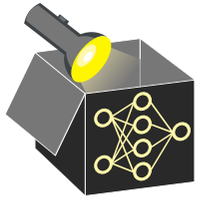
Local explanations: Adversarial Examples and Counterfactual Explanations



Learning goals

- Compare adversarial examples to counterfactual explanations
- See an example where both coincident

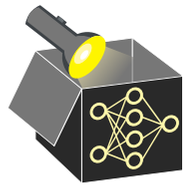
ADE AND COUNTERFACTUAL EXPLANATIONS



It seems as if ADEs and counterfactual explanations (CEs) are defined similarly. Both ADEs and CEs describe inputs close to a given input $\mathbf{x}$ that gets a different assignment. What are their differences?

- Counterfactuals do not have to be misclassified.
- Different notions of distance $\|\cdot\|$ are applied, e.g., $p_{2,\infty}$-norm for ADEs or $p_{0,1}$-norm for CEs.
- Informal difference I: ADEs are mostly considered for high-dimensional data, while CEs are mostly considered in the context of low-dimensional data.
- Informal difference II: ADEs hide changes while CEs highlight them.

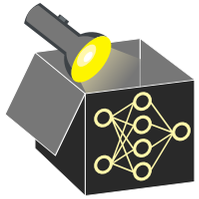
ADE AND COUNTERFACTUAL EXPLANATIONS



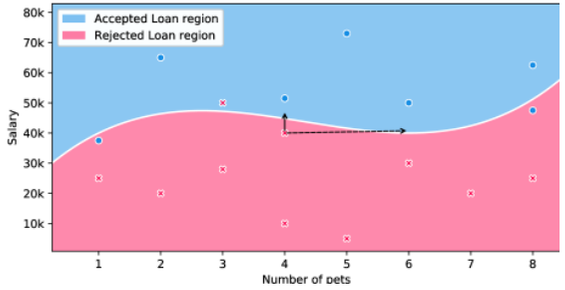
It seems as if ADEs and counterfactual explanations (CEs) are defined similarly. Both ADEs and CEs describe inputs close to a given input $\mathbf{x}$ that gets a different assignment. What are their differences?

- Counterfactuals do not have to be misclassified.
- Different notions of distance $\|\cdot\|$ are applied, e.g., $p_{2,\infty}$-norm for ADEs or $p_{0,1}$-norm for CEs.
- Informal difference I: ADEs are mostly considered for high-dimensional data, while CEs are mostly considered in the context of low-dim. data.
- Informal difference II: ADEs hide changes while CEs highlight them.

SHARED EXAMPLE ▸ Ballet (2019)

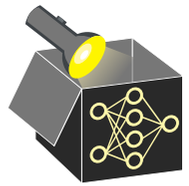


- ‘‘If you had two more pets, your loan application would have been granted" is an example of both ADEs and CEs.

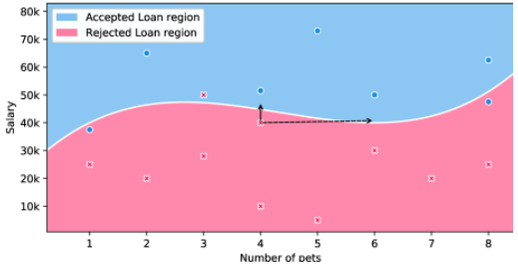


Decision boundary of a classifier deciding loan applications. ADE via “number of pets”

SHARED EXAMPLE ▸ BALLET_2019

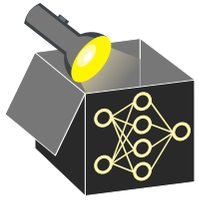


- ‘‘If you had two more pets, your loan application would have been granted" is an example of both ADEs and CEs.

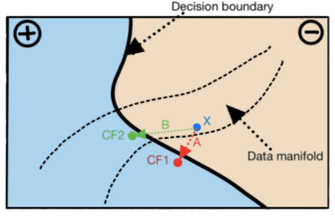


Decision boundary of a classifier deciding loan applications. ADE via “number of pets”

Interpretable Machine Learning



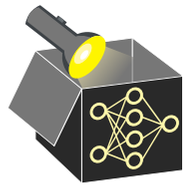
Counterfactual Explanations (CEs): Motivation



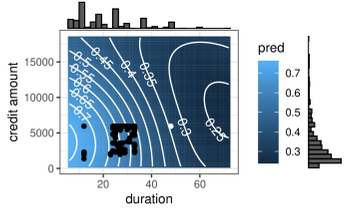
Learning goals

- Understand the motivation behind CEs
- Know why and how CEs are used
- Recognize the philosophical foundations of counterfactual reasoning

Interpretable Machine Learning



Counterfactual Explanations: Methods & Discussion of CEs



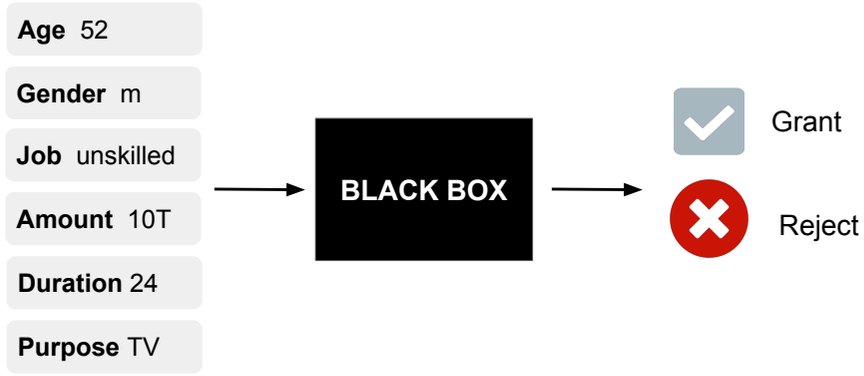
Learning goals

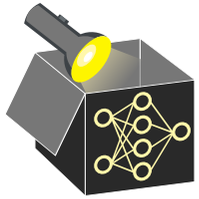
- See two strategies to generate CEs
- Know problems and limitations of CEs

MOTIVATING EXAMPLE: CREDIT RISK & CE

x: customer and credit information

y: grant or reject credit





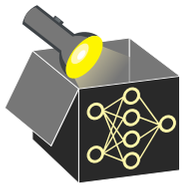
Potential questions:

- Why was the credit rejected?
- Is this decision fair compared with similar applicants?
- **How should x be changed so that the credit is accepted?**

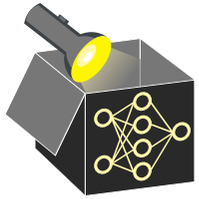
OVERVIEW OF COUNTERFACTUAL METHODS

Many methods exist to generate counterfactuals, they mainly differ in:

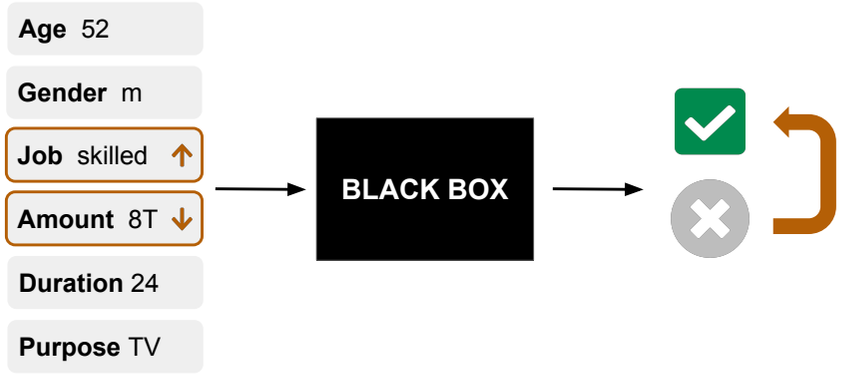
- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)



MOTIVATING EXAMPLE: CREDIT RISK & CE

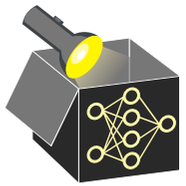


Counterfactual Explanations provide answers in the form of "What-If"-scenarios.



“If the applicant had higher skills and the credit amount had been reduced to $8.000, the loan would have been granted."

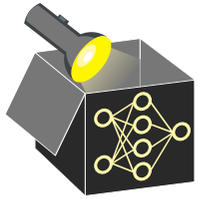
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

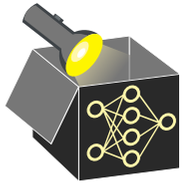
- **Target:** Most support classification; few extend to regression
  ⇝ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio

CORE DEFINITION AND PURPOSE OF CE



- **Counterfactual explanation (CE):** Hypothetical input $\mathbf{x}'$ close to the data point of interest $\mathbf{x}$ whose prediction equals a user-defined desired outcome y'

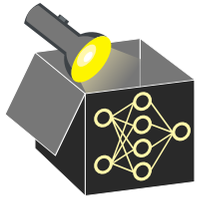
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed data types

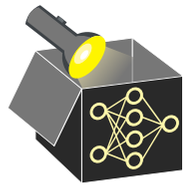
CORE DEFINITION AND PURPOSE OF CE



- **Counterfactual explanation (CE):** Hypothetical input $\mathbf{x}'$ close to the data point of interest $\mathbf{x}$ whose prediction equals a user-defined desired outcome y'
- **Proximity constraint:**

$$\text{Find} \quad \mathbf{x}' \approx \mathbf{x} \quad \text{such that} \quad f(\mathbf{x}') = y' \quad \text{and distance} \quad d(\mathbf{x}, \mathbf{x}') \quad \text{is minimal}$$

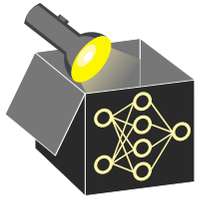
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed data types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness

CORE DEFINITION AND PURPOSE OF CE

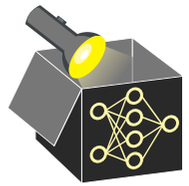


- **Counterfactual explanation (CE):** Hypothetical input $\mathbf{x}'$ close to the data point of interest $\mathbf{x}$ whose prediction equals a user-defined desired outcome y'
- **Proximity constraint:**

$$\text{Find} \quad \mathbf{x}' \approx \mathbf{x} \quad \text{such that} \quad f(\mathbf{x}') = y' \quad \text{and distance} \quad d(\mathbf{x}, \mathbf{x}') \quad \text{is minimal}$$

- **Minimal actionable changes:** Difference $\mathbf{x}' - \mathbf{x}$ shows the smallest feature change a user could realize in practice

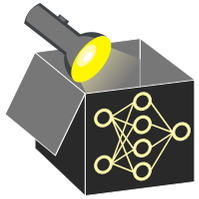
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed data types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring access to model internals/gradients) to model-agnostic (using only prediction funcs)

CORE DEFINITION AND PURPOSE OF CE

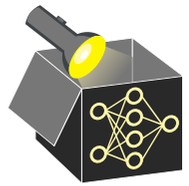


- **Counterfactual explanation (CE):** Hypothetical input $\mathbf{x}'$ close to the data point of interest $\mathbf{x}$ whose prediction equals a user-defined desired outcome y'
- **Proximity constraint:**

$$\text{Find} \quad \mathbf{x}' \approx \mathbf{x} \quad \text{such that} \quad f(\mathbf{x}') = y' \quad \text{and distance} \quad d(\mathbf{x}, \mathbf{x}') \quad \text{is minimal}$$

- **Minimal actionable changes:** Difference $\mathbf{x}' - \mathbf{x}$ shows the smallest feature change a user could realize in practice
- **Primary audience:**
  - Individuals aiming to alter model predictions
  - ML engineers exploring model behavior under adversarial conditions
    $\rightsquigarrow$ how small text changes in email flip prediction from "spam" to "no spam"

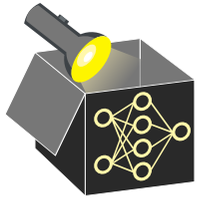
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed data types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring access to model internals/gradients) to model-agnostic (using only prediction funcs)
- **Optimization:** From gradient-based (differentiable models) and mixed-integer programming (linear models) to gradient-free methods (e.g., genetic algorithms)

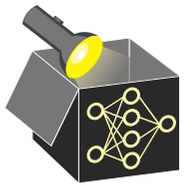
INTERPRETIVE AIMS & ROLES



CEs can serve various purposes; the user can decide what to learn from them, e.g.:

“If the person had been **one year older** and the **credit amount had been increased** to $12.000, the credit would have been granted."

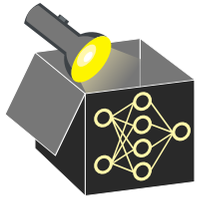
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed data types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring access to model internals/gradients) to model-agnostic (using only prediction funcs)
- **Optimization:** From gradient-based (differentiable models) and mixed-integer programming (linear models) to gradient-free methods (e.g., genetic algorithms)
- **Rashomon Effect:** Many methods return one CE, some diverse sets of CEs, others prioritize CEs, or let the user choose

INTERPRETIVE AIMS & ROLES

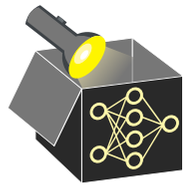


CEs can serve various purposes; the user can decide what to learn from them, e.g.:

"If the person had been **one year older** and the **credit amount had been increased** to $12.000, the credit would have been granted."

- **Guidance for future actions:**
  Ok, I will apply again next year for the higher amount.

FIRST OPTIMIZATION-BASED CE METHOD



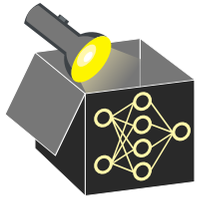
WACHTER_2018

Introduced CEs in context of ML predictions by solving

$$\arg\min_{\mathbf{x}'} \max_{\lambda} \lambda \underbrace{(\hat{f}(\mathbf{x}') - y')^2}_{o_{target}(\hat{f}(\mathbf{x}'), y')} + \underbrace{\sum_{j=1}^{p} \frac{|x_j' - x_j|}{MAD_j}}_{o_{proximity}(\mathbf{x}', \mathbf{x})}$$

- o_{target} ensures prediction flips to y' (by increasing weight λ)
- $o_{proximity}$ penalizes deviations from $\mathbf{x}$, rescaled by median abs. deviation:
  $MAD_j = \text{med}_{i \in \{1,\ldots,n\}}(|x_j^{(i)} - \text{med}_{k \in \{1,\ldots,n\}}(x_j^{(k)})|)$

INTERPRETIVE AIMS & ROLES



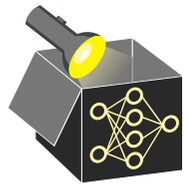
CEs can serve various purposes; the user can decide what to learn from them, e.g.:

"If the person had been **one year older** and the **credit amount had been increased** to $12.000, the credit would have been granted."

- **Guidance for future actions:**
  Ok, I will apply again next year for the higher amount.

- **Provide reasons:**
  Interesting, I did not know that age plays a role in loan applications.

FIRST OPTIMIZATION-BASED CE METHOD



▸ WACHTER_2018

Introduced CEs in context of ML predictions by solving

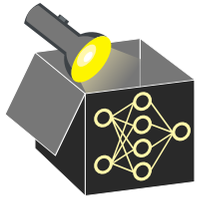
$$\arg\min_{\mathbf{x}'} \max_{\lambda} \lambda \underbrace{(\hat{f}(\mathbf{x}') - y')^2}_{o_{target}(\hat{f}(\mathbf{x}'), y')} + \underbrace{\sum_{j=1}^{p} \frac{|x_j' - x_j|}{MAD_j}}_{o_{proximity}(\mathbf{x}', \mathbf{x})}$$

- o_{target} ensures prediction flips to y' (by increasing weight λ)
- $o_{proximity}$ penalizes deviations from $\mathbf{x}$, rescaled by median abs. deviation: $MAD_j = \text{med}_{i \in \{1,\ldots,n\}}(|x_j^{(i)} - \text{med}_{k \in \{1,\ldots,n\}}(x_j^{(k)})|)$

Approach: Alternating optimization over $\mathbf{x}'$ and λ

- Start with an initial λ (controls emphasis on o_{target} vs. $o_{proximity}$)
- Use a gradient-free optimizer (e.g., Nelder-Mead) to minimize over $\mathbf{x}'$
- If prediction constraint not satisfied ($\hat{f}(\mathbf{x}') \neq y'$), increase λ and repeat
  $\rightsquigarrow$ λ serves as soft constraint, gradually enforcing prediction validity $\hat{f}(\mathbf{x}') = y'$
- Iteratively shift focus: 1. achieve prediction validity, 2. minimize proximity

INTERPRETIVE AIMS & ROLES

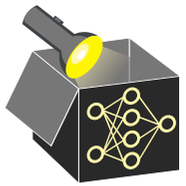


CEs can serve various purposes; the user can decide what to learn from them, e.g.:

“If the person had been **one year older** and the **credit amount had been increased** to $12.000, the credit would have been granted."

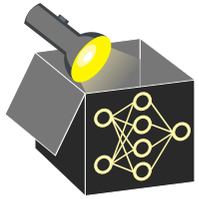
- **Guidance for future actions:**
  Ok, I will apply again next year for the higher amount.
- **Provide reasons:**
  Interesting, I did not know that age plays a role in loan applications.
- **Provide grounds to contest the decision:**
  How dare you, I do not want to be discriminated for my age in an application.

LIMITATIONS OF WACHTER’S APPROACH



- **Manual tuning:** No principled way to set λ; requires iterative increase
- **Asymmetric focus:** Early iterations dominated by minimizing target loss
- **Limited feature support:** Proximity term defined only for numerical feats
- **No additional objectives:** Ignores sparsity, plausibility, fairness, diversity
- **Single solution:** Returns one CE; no support for diverse or ranked CEs

INTERPRETIVE AIMS & ROLES



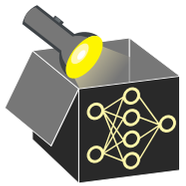
CEs can serve various purposes; the user can decide what to learn from them, e.g.:

"If the person had been **one year older** and the **credit amount had been increased** to $12.000, the credit would have been granted."

- **Guidance for future actions:**
  Ok, I will apply again next year for the higher amount.
- **Provide reasons:**
  Interesting, I did not know that age plays a role in loan applications.
- **Provide grounds to contest the decision:**
  How dare you, I do not want to be discriminated for my age in an application.
- **Detect model biases:**
  There is a bug, an increase in amount should not increase approval rates.

MULTI-OBJECTIVE CE

▸ DANDL_2020

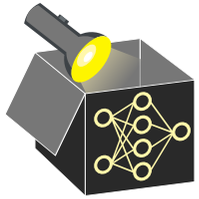


- **Multi-Objective Counterfactual Explanations (MOC):** Instead of collapsing objectives into a single obj., optimize all 4 obj. simultaneously

$$\arg\min_{\mathbf{x}'} \left(o_{target}(\hat{f}(\mathbf{x}'), y'), o_{proximity}(\mathbf{x}', \mathbf{x}), o_{sparse}(\mathbf{x}', \mathbf{x}), o_{plausible}(\mathbf{x}', \mathbf{X}) \right).$$

- Avoids using/tuning of weights (e.g., λ); returns Pareto-optimal set
- Uses an adjusted multi-objective genetic algo. (NSGA-II) for mixed feats
- Outputs diverse CEs representing different trade-offs between objectives

PHILOSOPHICAL FOUNDATIONS ▸ Lewis (1973)

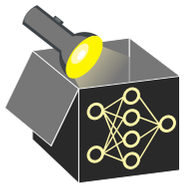


Counterfactuals have a long tradition in analytic philosophy
$\rightsquigarrow$ A **counterfactual conditional** takes the form:

"If S had occurred, Q would have occurred."

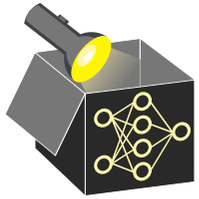
- *S*: past event that never happened $\rightsquigarrow$ CE run contrary to fact
- Statement is true iff *Q* holds in all **closest** worlds where *S* is true
- Closest worlds preserve laws and change as few facts as possible (related to *S*)

EXAMPLE: CREDIT DATA



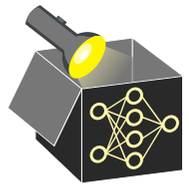
- Model: SVM with RBF kernel
- **x**: First data point of credit data with $\mathbb{P}(y = good) = 0.34$
- Goal: Increase the probability to desired outcome $[0.5, 1]$
- MOC (with default parameters) returned 69 valid CEs after 200 iterations
- All CEs modified credit duration; many also adjusted credit amount

PHILOSOPHICAL FOUNDATIONS

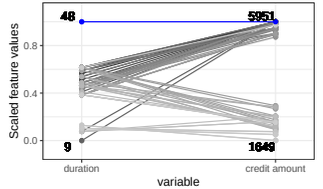


- CEs have largely been studied to explain causal dependence
- **Causal dependence**: Q depends on $S \Leftrightarrow$ without S, no Q
  $\rightsquigarrow$ Good CEs point to critical causal factors that drove the algorithmic decision
  $\rightsquigarrow$ **CE objective**: find $\mathbf{x}' \approx \mathbf{x}$ with $f(\mathbf{x}') = y'$ to expose causal features

EXAMPLE: CREDIT DATA ▸ DANDL_2020

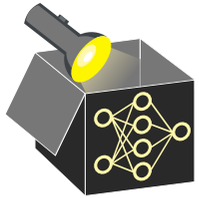


- Feature changes can be visualized using parallel and 2D surface plots
- Parallel plot: All CEs had values equal to or smaller than the values of **x**



Parallel plot: Grey lines = CEs $\mathbf{x}'$, blue line = **x**.
Features without changes omitted.
Bold numbers denote numeric ranges.

PHILOSOPHICAL FOUNDATIONS

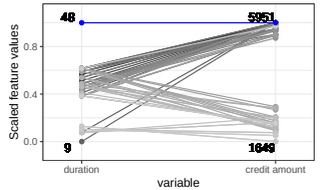


- CEs have largely been studied to explain causal dependence
- **Causal dependence**: Q depends on $S \Leftrightarrow$ without S, no Q
  $\rightsquigarrow$ Good CEs point to critical causal factors that drove the algorithmic decision
  $\rightsquigarrow$ **CE objective**: find $\mathbf{x}' \approx \mathbf{x}$ with $f(\mathbf{x}') = y'$ to expose causal features
- Relaxing closeness may add causally irrelevant edits to the explanation
  $\rightsquigarrow$ e.g., suggest to lower loan *and* increase age (but only loan matters)

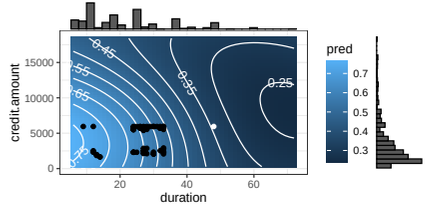
EXAMPLE: CREDIT DATA

▸ DANDL_2020

- Feature changes can be visualized using parallel and 2D surface plots
- Parallel plot: All CEs had values equal to or smaller than the values of **x**
- Surface plot: CEs in lower-left appear distant, but lie in high-density regions near training data (as shown by histograms)

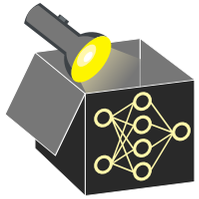


Parallel plot: Grey lines = CEs $\mathbf{x}'$, blue line = $\mathbf{x}$.
Features without changes omitted.
Bold numbers denote numeric ranges.



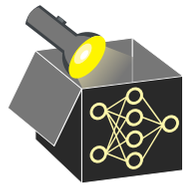
Surface plot: White dot = $\mathbf{x}$, black dots = CEs $\mathbf{x}'$.
Histograms: Marginal distribution of training data $\mathbf{X}$.

PHILOSOPHICAL FOUNDATIONS



- CEs have largely been studied to explain causal dependence
- **Causal dependence**: Q depends on $S \Leftrightarrow$ without S, no Q
  $\rightsquigarrow$ Good CEs point to critical causal factors that drove the algorithmic decision
  $\rightsquigarrow$ **CE objective**: find $\mathbf{x}' \approx \mathbf{x}$ with $f(\mathbf{x}') = y'$ to expose causal features
- Relaxing closeness may add causally irrelevant edits to the explanation
  $\rightsquigarrow$ e.g., suggest to lower loan *and* increase age (but only loan matters)
- CEs are contrastive: Explain a decision by comparing it to a different outcome
  $\rightsquigarrow$ If age were 30 instead of 60, loan would have been \$9k instead of rejected
  $\rightsquigarrow$ Answers contrastive question: "Why Q' instead of Q?" (preferred by humans)

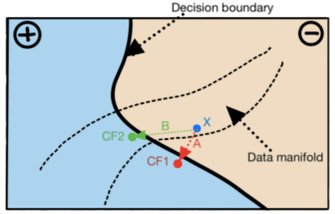
PROBLEMS, PITFALLS, & LIMITATIONS



- **Illusion of model understanding:** CEs explain ML decisions by pointing to few specific alternatives, reducing complexity but offering limited explanatory power
  $\rightsquigarrow$ Psychologists have shown that although perceived model understanding of end-users increases, the objective model understanding remains unchanged

Interpretable Machine Learning

CE: Optimization Problem and Objectives



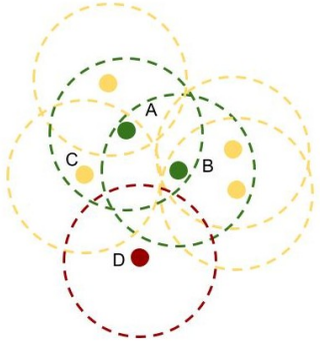
Learning goals

- Formulate CEs as optimization problem
- Identify key objectives (proximity, sparsity)
- Understand trade-offs in CE generation

magentaDaxberger et al. 2020

Interpretable Machine Learning

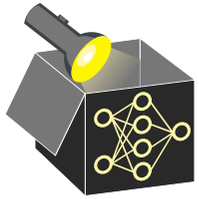
Local Explanations: Increasing Trust in Explanations



Learning goals

- Understand the aspects that undermine users' trust in an explanation
- Learn diagnostic tools that could increase trust

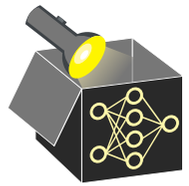
MATHEMATICAL PERSPECTIVE



Terminology:

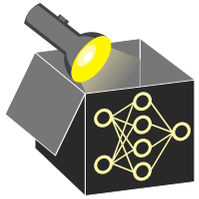
- **x**: original/factual datapoint whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired prediction ($y' =$ “grant credit") or interval ($y' = [1000, \infty[$)

MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy

MATHEMATICAL PERSPECTIVE



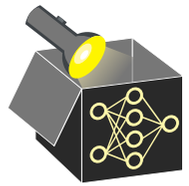
Terminology:

- **x**: original/factual datapoint whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired prediction ($y' =$ "grant credit") or interval ($y' = [1000, \infty[$)

A **valid** counterfactual $\mathbf{x}'$ satisfies two criteria:

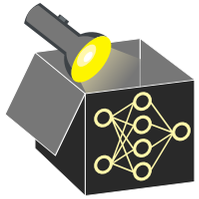
1. **Prediction validity:** CE's prediction $\hat{f}(\mathbf{x}')$ is equal to the desired prediction y'
2. **Proximity:** CE $\mathbf{x}'$ is as close as possible to the original input **x**

MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: "Why did the model come up with this decision?"

MATHEMATICAL PERSPECTIVE



Terminology:

- $\mathbf{x}$: original/factual datapoint whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired prediction ($y' =$ "grant credit") or interval ($y' = [1000, \infty[$)

A **valid** counterfactual $\mathbf{x}'$ satisfies two criteria:

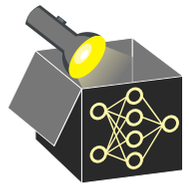
1. **Prediction validity:** CE's prediction $\hat{f}(\mathbf{x}')$ is equal to the desired prediction y'
2. **Proximity:** CE $\mathbf{x}'$ is as close as possible to the original input $\mathbf{x}$

Reformulate these two objectives as optimization problem:

$$\arg\min_{\mathbf{x}'} \lambda_1 o_{target}(\hat{f}(\mathbf{x}'), y') + \lambda_2 o_{proximity}(\mathbf{x}', \mathbf{x})$$

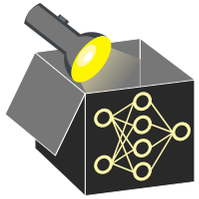
- λ_1 and λ_2 balance the two objectives
- o_{target}: distance in target space
- $o_{proximity}$: distance in feature space

MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: "Why did the model come up with this decision?"
- **Trustworthy**: "How certain is this explanation?"
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)

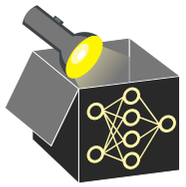
OBJECTIVE FUNCTIONS ▸ Dandl et al. (2020)



Distance in target space o_{target}:

- **Regression:** L_1 distance $o_{target}(\hat{f}(\mathbf{x}'), y') = |\hat{f}(\mathbf{x}') - y'|$
- **Classification:**
  - For predicted probabilities: $o_{target} = |\hat{f}(\mathbf{x}') - y'|$
  - For predicted hard labels: $o_{target} = \mathbb{I}\{\hat{f}(\mathbf{x}') \neq y'\}$

MOTIVATION & IMPORTANT PROPERTIES



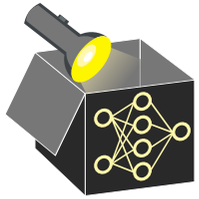
- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: "Why did the model come up with this decision?"
- **Trustworthy**: "How certain is this explanation?"
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist

     ⇝ measure how robust an IML method is to small changes in the input data or parameters

     ⇝ Is an observation out-of-distribution?

OBJECTIVE FUNCTIONS

Dandl et al. (2020)



Distance in target space o_{target}:

- **Regression:** L_1 distance $o_{target}(\hat{f}(\mathbf{x}'), y') = |\hat{f}(\mathbf{x}') - y'|$
- **Classification:**
  - For predicted probabilities: $o_{target} = |\hat{f}(\mathbf{x}') - y'|$
  - For predicted hard labels: $o_{target} = \mathbb{I}\{\hat{f}(\mathbf{x}') \neq y'\}$

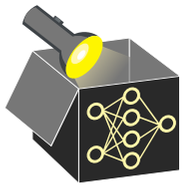
Distance in input space $o_{proximity}$: Gower distance (mixed feature types)

$$o_{proximity}(\mathbf{x}', \mathbf{x}) = d_G(\mathbf{x}', \mathbf{x}) = \frac{1}{p} \sum_{j=1}^{p} \delta_G(x'_j, x_j) \in [0, 1], \text{ where}$$

- $\delta_G(x'_j, x_j) = \mathbb{I}\left\{x'_j \neq x_j\right\}$ if x_j is categorical
- $\delta_G(x'_j, x_j) = \frac{1}{\widehat{R}_j} |x'_j - x_j|$ if x_j is numerical

  $\rightsquigarrow$ $\widehat{R}_j$ is the range of feature j in the training set to ensure $\delta_G(x'_j, x_j) \in [0, 1]$

MOTIVATION & IMPORTANT PROPERTIES



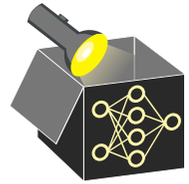
- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: "Why did the model come up with this decision?"
- **Trustworthy**: "How certain is this explanation?"
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist

     $\rightsquigarrow$ measure how robust an IML method is to small changes in the input data or parameters

     $\rightsquigarrow$ Is an observation out-of-distribution?
- Failing in one of these $\rightsquigarrow$ undermining users' trust in the explanations

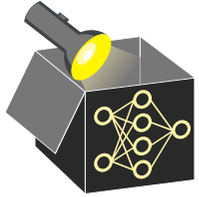
  $\rightsquigarrow$ undermining trust in the model

OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
  ⇝ explanations from local explanation methods are unreliable

FURTHER OBJECTIVES: SPARSITY

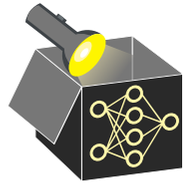


Additional constraints can improve the explanation quality of the corresponding CEs
⇝ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

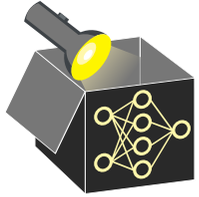
- End-users often prefer short over long explanations

OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted obs. to calculate the marginal contribs
  - ICE curves grid data points

FURTHER OBJECTIVES: SPARSITY

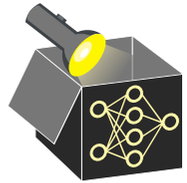


Additional constraints can improve the explanation quality of the corresponding CEs
$\rightsquigarrow$ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

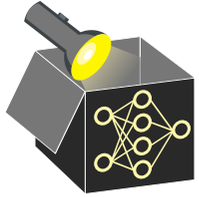
- End-users often prefer short over long explanations
- Sparsity could be integrated into $o_{proximity}$
  e.g., using L_0-norm (number of changed features) or L_1-norm (LASSO)

OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted obs. to calculate the marginal contribs
  - ICE curves grid data points
- Two very simple and intuitive approaches
  - Classifier for out-of-distribution
  - Clustering
- More complicated also possible, e.g., variational autoencoders
  ▸ Daxberger 2020

FURTHER OBJECTIVES: SPARSITY



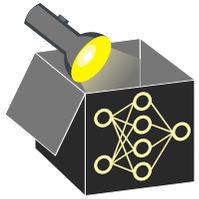
Additional constraints can improve the explanation quality of the corresponding CEs
$\rightsquigarrow$ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

- End-users often prefer short over long explanations
- Sparsity could be integrated into $o_{proximity}$
  e.g., using L_0-norm (number of changed features) or L_1-norm (LASSO)
- Alternative: Include separate objective measuring sparsity, e.g., via L_0-norm

$$o_{sparse}(\mathbf{x}', \mathbf{x}) = \sum_{j=1}^{p} \mathcal{I}_{\{x_j' \neq x_j\}}$$

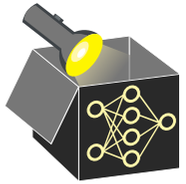
FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed

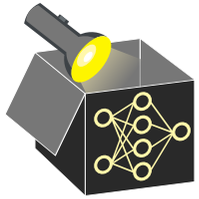
OOD DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- Idea: Hallucinate new (ood) data by randomly sampling data points

$\rightsquigarrow$ Learn a binary classifier to distinguish between the origins of the data

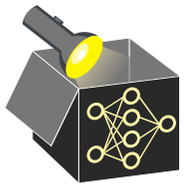
FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  $\rightsquigarrow$ Avoid unrealistic combinations of feature values

OOD DETECTION: OOD-CLASSIFIER



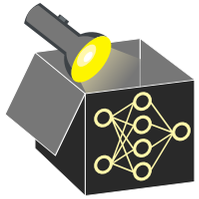
- Problem: we have only in-distribution data
- Idea: Hallucinate new (ood) data by randomly sampling data points

$\rightsquigarrow$ Learn a binary classifier to distinguish between the origins of the data

- Study whether an explanation approach can be fooled ▸ Slack 2020
  - Hide bias in the true (deployed) model, but use an unbiased model for all out-of-distribution samples

$\rightsquigarrow$ Important way to diagnose an explanation approach

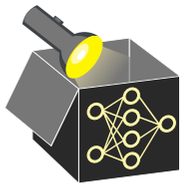
FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

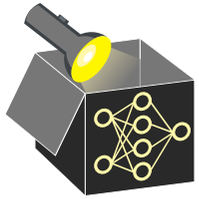
- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  $\rightsquigarrow$ Avoid unrealistic combinations of feature values
- Estimating joint distribution is hard, especially for mixed feature spaces
  $\rightsquigarrow$ Common proxy: ensure that $\mathbf{x}'$ is close to training data $\mathbf{X}$

OOD DETECTION: CLUSTERING VIA DBSCAN



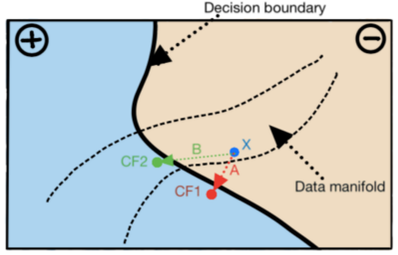
- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)

FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

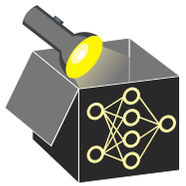
- CEs should suggest realistic (i.e., plausible) alternatives
  ⇝ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  ⇝ Avoid unrealistic combinations of feature values
- Estimating joint distribution is hard, especially for mixed feature spaces
  ⇝ Common proxy: ensure that $\mathbf{x}'$ is close to training data $\mathbf{X}$



Example from Verma et al. (2020)

- Input **x** originally classified as ⊖
- Two valid CEs in class ⊕: CF1 and CF2
- Path **A** (CF1) is shorter (but unrealistic)
- Path **B** (CF2) is longer but in data manifold

OOD DETECTION: CLUSTERING VIA DBSCAN

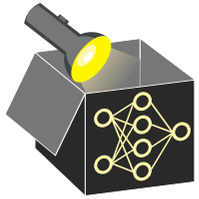


- DBSCAN is a data clustering algorithm Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)

FURTHER OBJECTIVES

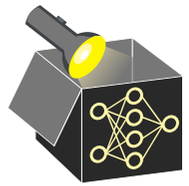


Plausibility term: Encourage counterfactuals close to observed data.

- Define $\mathbf{x}^{[1]}$ as the nearest neighbor of $\mathbf{x}'$ in the training set $\mathbf{X}$
- Use Gower distance between $\mathbf{x}'$ and $\mathbf{x}^{[1]}$ to define plausibility objective:

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

OOD DETECTION: CLUSTERING VIA DBSCAN

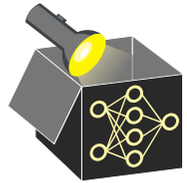


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points

OOD DETECTION: CLUSTERING VIA DBSCAN

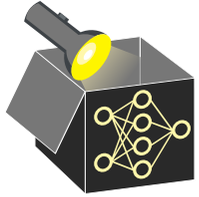


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point

FURTHER OBJECTIVES



Plausibility term: Encourage counterfactuals close to observed data.

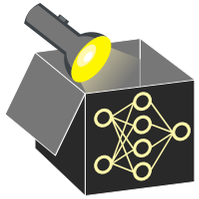
- Define $\mathbf{x}^{[1]}$ as the nearest neighbor of $\mathbf{x}'$ in the training set $\mathbf{X}$
- Use Gower distance between $\mathbf{x}'$ and $\mathbf{x}^{[1]}$ to define plausibility objective:

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

Extended optimization: Add sparsity and plausibility terms to the objective

$$\arg\min_{\mathbf{x}'} \lambda_1 o_{\text{target}}(\hat{f}(\mathbf{x}'), y') + \lambda_2 o_{\text{proximity}}(\mathbf{x}', \mathbf{x}) + \lambda_3 o_{\text{sparse}}(\mathbf{x}', \mathbf{x}) + \lambda_4 o_{\text{plausible}}(\mathbf{x}', \mathbf{X})$$

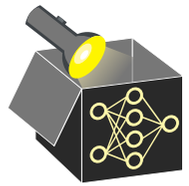
REMARKS: THE RASHOMON EFFECT



Issue (Rashomon effect):

- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

OOD DETECTION: CLUSTERING VIA DBSCAN

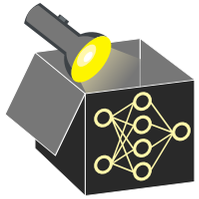


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point
- Noise points
  - Are not within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Not part of any cluster

REMARKS: THE RASHOMON EFFECT



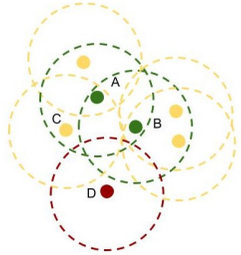
Issue (Rashomon effect):

- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

Possible solutions:

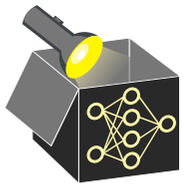
- Present all CEs for $\mathbf{x}$ (but: time and human processing capacity is limited)
- Focus on one or few CEs (but: by which criterion should guide this choice?)

OUT-OF-DISTRIBUTION DETECTION

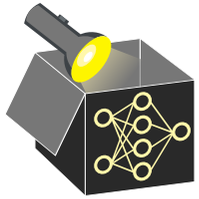


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster



REMARKS: THE RASHOMON EFFECT



Issue (Rashomon effect):

- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

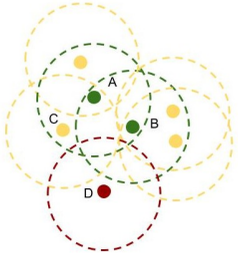
Possible solutions:

- Present all CEs for $\mathbf{x}$ (but: time and human processing capacity is limited)
- Focus on one or few CEs (but: by which criterion should guide this choice?)

Note:

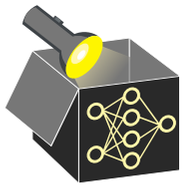
- Nonlinear models can produce diverse and inconsistent CEs
  $\rightsquigarrow$ suggest both increasing and decreasing credit duration (confusing for users)
- Handling this **Rashomon effect** remains an open problem in interpretable ML

OUT-OF-DISTRIBUTION DETECTION

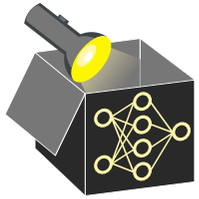


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point

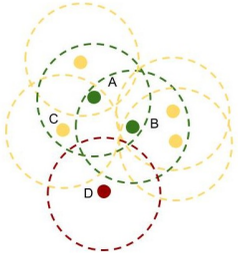


REMARKS: MODEL OR REAL-WORLD



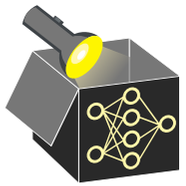
- CEs explain model predictions, but may appear to explain the real-world users
  $\rightsquigarrow$ Transfer of model explanations to explain real-world is generally not permitted
- **Example:** CE suggests increasing age by 5 to receive a loan
  $\rightsquigarrow$ The applicant waits 5 years and reapplies

OUT-OF-DISTRIBUTION DETECTION

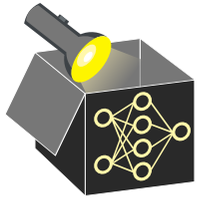


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster

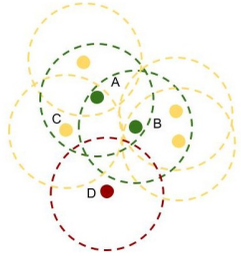


REMARKS: MODEL OR REAL-WORLD



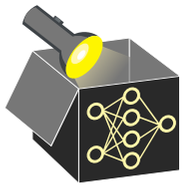
- CEs explain model predictions, but may appear to explain the real-world users
  $\rightsquigarrow$ Transfer of model explanations to explain real-world is generally not permitted
- **Example:** CE suggests increasing age by 5 to receive a loan
  $\rightsquigarrow$ The applicant waits 5 years and reapplies
- **Problem:** Other features may change in the meantime (e.g., job status, income)
  $\rightsquigarrow$ ▸ Karimi et al. (2020) propose CEs that respect causal structure
- **Model drift:** Bank's algorithm itself may change over time
  $\rightsquigarrow$ Past CEs may become invalid

OUT-OF-DISTRIBUTION DETECTION



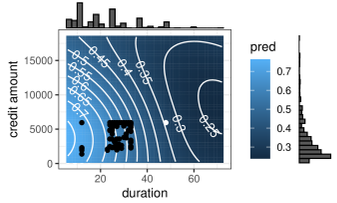
Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters



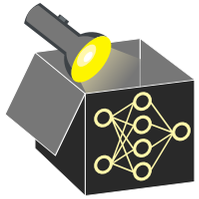
Interpretable Machine Learning

Methods & Discussion of CEs



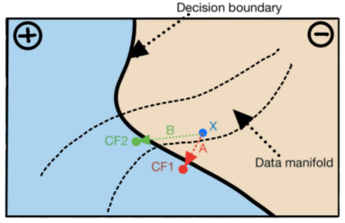
Learning goals

- See two strategies to generate CEs
- Know problems and limitations of CEs



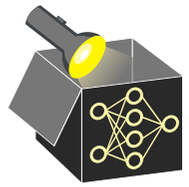
Interpretable Machine Learning

Counterfactual Explanations: Optimization Problem and Objectives

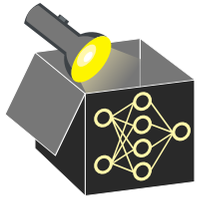


Learning goals

- Formulate CEs as optimization problem
- Identify key objectives (proximity, sparsity)
- Understand trade-offs in CE generation



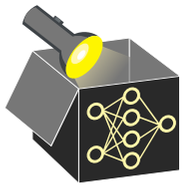
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)

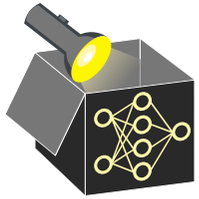
MATHEMATICAL PERSPECTIVE



Terminology:

- $\mathbf{x}$: original/factual data point whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired predi. ($y' =$ "grant credit") or interval ($y' = [1000, \infty[$)

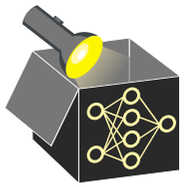
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio

MATHEMATICAL PERSPECTIVE



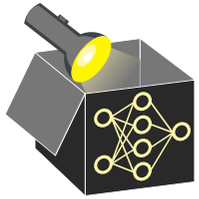
Terminology:

- $\mathbf{x}$: original/factual data point whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired predi. ($y' =$ "grant credit") or interval ($y' = [1000, \infty[$)

A **valid** counterfactual $\mathbf{x}'$ satisfies two criteria:

1. **Prediction validity:** CE's prediction $\hat{f}(\mathbf{x}')$ is equal to the desired pred. y'
2. **Proximity:** CE $\mathbf{x}'$ is as close as possible to the original input $\mathbf{x}$

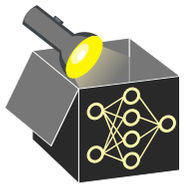
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed types

MATHEMATICAL PERSPECTIVE



Terminology:

- $\mathbf{x}$: original/factual data point whose prediction we want to explain
- $y' \subset \mathbb{R}^g$: desired predi. ($y' =$ "grant credit") or interval ($y' = [1000, \infty[$)

A **valid** counterfactual $\mathbf{x}'$ satisfies two criteria:

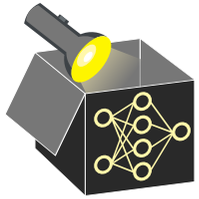
1. **Prediction validity:** CE's prediction $\hat{f}(\mathbf{x}')$ is equal to the desired pred. y'
2. **Proximity:** CE $\mathbf{x}'$ is as close as possible to the original input $\mathbf{x}$

Reformulate these two objectives as optimization problem:

$$\arg\min_{\mathbf{x}'} \lambda_1 o_{target}(\hat{f}(\mathbf{x}'), y') + \lambda_2 o_{proximity}(\mathbf{x}', \mathbf{x})$$

- λ_1 and λ_2 balance the two objectives
- o_{target}: distance in target space
- $o_{proximity}$: distance in feature space

OVERVIEW OF COUNTERFACTUAL METHODS

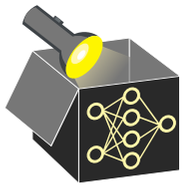


Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness

OBJECTIVE FUNCTIONS

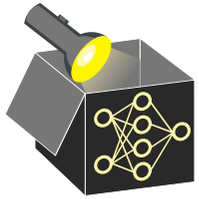
DANDL_2020



Distance in target space o_{target}:

- **Regression:** L_1 distance $o_{target}(\hat{f}(\mathbf{x}'), y') = |\hat{f}(\mathbf{x}') - y'|$
- **Classification:**
  - For predicted probabilities: $o_{target} = |\hat{f}(\mathbf{x}') - y'|$
  - For predicted hard labels: $o_{target} = \mathbb{I}\{\hat{f}(\mathbf{x}') \neq y'\}$

OVERVIEW OF COUNTERFACTUAL METHODS

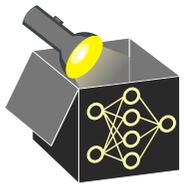


Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring model internals or access to gradients) to model-agnostic (using only prediction functions)

OBJECTIVE FUNCTIONS

DANDL_2020



Distance in target space o_{target}:

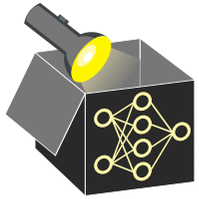
- **Regression:** L_1 distance $o_{target}(\hat{f}(\mathbf{x}'), y') = |\hat{f}(\mathbf{x}') - y'|$
- **Classification:**
  - For predicted probabilities: $o_{target} = |\hat{f}(\mathbf{x}') - y'|$
  - For predicted hard labels: $o_{target} = \mathbb{I}\{\hat{f}(\mathbf{x}') \neq y'\}$

Distance in input space $o_{proximity}$: Gower distance (mixed feature types)

$$o_{proximity}(\mathbf{x}', \mathbf{x}) = (\mathbf{x}', \mathbf{x}) = \frac{1}{p} \sum_{j=1}^{p} \delta_G(x_j', x_j) \in [0, 1], \text{ where}$$

- $\delta_G(x_j', x_j) = \mathbb{I}\left\{x_j' \neq x_j\right\}$ if x_j is categorical
- $\delta_G(x_j', x_j) = \frac{1}{\widehat{R}_j} |x_j' - x_j|$ if x_j is numerical
  $\rightsquigarrow$ $\widehat{R}_j$: range of feature j in the training set to ensure $\delta_G(x_j', x_j) \in [0, 1]$

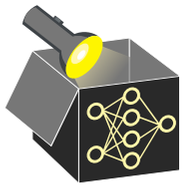
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring model internals or access to gradients) to model-agnostic (using only prediction functions)
- **Optimization:** From gradient-based (differentiable models) and mixed-integer programming (linear models) to gradient-free methods (e.g., genetic algorithms)

FURTHER OBJECTIVES: SPARSITY

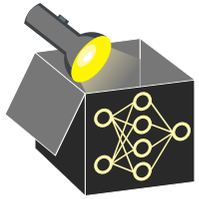


Additional constraints can improve the explanation quality of the corresponding CEs
$\rightsquigarrow$ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

- End-users often prefer short over long explanations

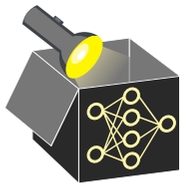
OVERVIEW OF COUNTERFACTUAL METHODS



Many methods exist to generate counterfactuals, they mainly differ in:

- **Target:** Most support classification; few extend to regression
  $\rightsquigarrow$ Recent work extends CEs to other ML tasks (un-, semi-, self-supervised)
- **Data type:** Focus is on tabular data; little work on text, vision, audio
- **Feature space:** Some handle only numerical features; few support mixed types
- **Objectives:** From core goals like sparsity and plausibility to emerging aims such as fairness, personalization, and robustness
- **Model access:** Methods range from model-specific (requiring model internals or access to gradients) to model-agnostic (using only prediction functions)
- **Optimization:** From gradient-based (differentiable models) and mixed-integer programming (linear models) to gradient-free methods (e.g., genetic algorithms)
- **Rashomon Effect:** Many methods return one CE, some diverse sets of CEs, others prioritize CEs, or let the user choose

FURTHER OBJECTIVES: SPARSITY

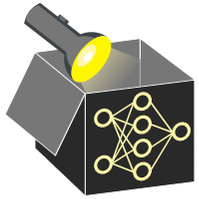


Additional constraints can improve the explanation quality of the corresponding CEs
$\rightsquigarrow$ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

- End-users often prefer short over long explanations
- Sparsity could be integrated into $o_{proximity}$
  e.g., using L_0-norm (number of changed features) or L_1-norm (LASSO)

FIRST OPTIMIZATION-BASED CE METHOD ▸ Wachter et. al (2018)

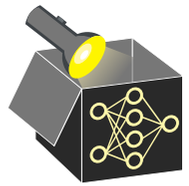


Introduced CEs in context of ML predictions by solving

$$\arg\min_{\mathbf{x}'} \max_{\lambda} \lambda \underbrace{(\hat{f}(\mathbf{x}') - y')^2}_{o_{target}(\hat{f}(\mathbf{x}'), y')} + \underbrace{\sum_{j=1}^{p} \frac{|x'_j - x_j|}{MAD_j}}_{o_{proximity}(\mathbf{x}', \mathbf{x})}$$

- o_{target} ensures prediction flips to y' (by increasing weight λ)
- $o_{proximity}$ penalizes deviations from $\mathbf{x}$, rescaled by median absolute deviation: $MAD_j = \text{med}_{i \in \{1,\ldots,n\}}(|x_j^{(i)} - \text{med}_{k \in \{1,\ldots,n\}}(x_j^{(k)})|)$

FURTHER OBJECTIVES: SPARSITY



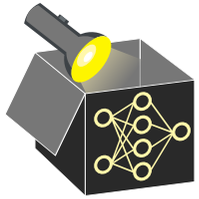
Additional constraints can improve the explanation quality of the corresponding CEs
⇝ popular constraints include **sparsity** and **plausibility**

Sparsity Favor explanations that change few features

- End-users often prefer short over long explanations
- Sparsity could be integrated into $o_{proximity}$
  e.g., using L_0-norm (number of changed features) or L_1-norm (LASSO)
- Alternative: Include separate objective measuring sparsity, e.g., via L_0-norm

$$o_{sparse}(\mathbf{x}', \mathbf{x}) = \sum_{j=1}^{p} \mathcal{I}_{\{x'_j \neq x_j\}}$$

FIRST OPTIMIZATION-BASED CE METHOD ▸ Wachter et. al (2018)



Introduced CEs in context of ML predictions by solving

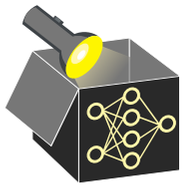
$$\arg\min_{\mathbf{x}'} \max_{\lambda} \lambda \underbrace{(\hat{f}(\mathbf{x}') - y')^2}_{o_{target}(\hat{f}(\mathbf{x}'), y')} + \underbrace{\sum_{j=1}^{p} \frac{|x_j' - x_j|}{MAD_j}}_{o_{proximity}(\mathbf{x}', \mathbf{x})}$$

- o_{target} ensures prediction flips to y' (by increasing weight λ)
- $o_{proximity}$ penalizes deviations from $\mathbf{x}$, rescaled by median absolute deviation: $MAD_j = \text{med}_{i \in \{1,\ldots,n\}}(|x_j^{(i)} - \text{med}_{k \in \{1,\ldots,n\}}(x_j^{(k)})|)$

Approach: Alternating optimization over $\mathbf{x}'$ and λ

- Start with an initial λ (controls emphasis on o_{target} vs. $o_{proximity}$)
- Use a gradient-free optimizer (e.g., Nelder-Mead) to minimize over $\mathbf{x}'$
- If prediction constraint not satisfied ($\hat{f}(\mathbf{x}') \neq y'$), increase λ and repeat
  $\rightsquigarrow$ λ serves as soft constraint, gradually enforcing prediction validity $\hat{f}(\mathbf{x}') = y'$
- Iteratively shift focus: first achieve prediction validity, then minimize proximity

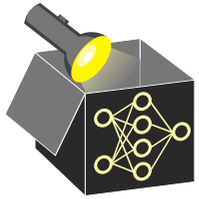
FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

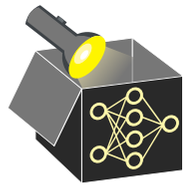
- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed

LIMITATIONS OF WACHTER'S APPROACH



- **Manual tuning:** No principled way to set λ; requires iterative increase
- **Asymmetric focus:** Early iterations dominated by minimizing target loss
- **Limited feature support:** Proximity term defined only for numerical features
- **No additional objectives:** Ignores sparsity, plausibility, fairness, diversity
- **Single solution:** Returns one CE; no support for diverse or ranked CEs

FURTHER OBJECTIVES: PLAUSIBILITY

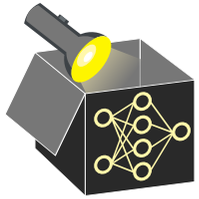


Plausibility:

- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  $\rightsquigarrow$ Avoid unrealistic combinations of feature values

MULTI-OBJECTIVE CE

Dandl et al. (2020)

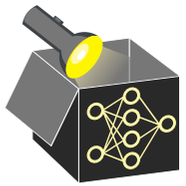


- **Multi-Objective Counterfactual Explanations (MOC):** Instead of collapsing objectives into a single objective, optimize all four objectives simultaneously

$$\arg\min_{\mathbf{x}'} \left(o_{target}(\hat{f}(\mathbf{x}'), y'), o_{proximity}(\mathbf{x}', \mathbf{x}), o_{sparse}(\mathbf{x}', \mathbf{x}), o_{plausible}(\mathbf{x}', \mathbf{X}) \right).$$

- Avoids using/tuning of weights (e.g., λ); returns Pareto-optimal set
- Uses an adjusted multi-objective genetic algorithm (NSGA-II) for mixed features
- Outputs diverse CEs representing different trade-offs between objectives

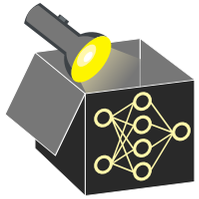
FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

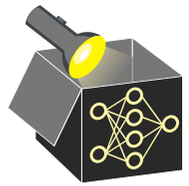
- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  $\rightsquigarrow$ Avoid unrealistic combinations of feature values
- Estimating joint distribution is hard, especially for mixed feature spaces
  $\rightsquigarrow$ Common proxy: ensure that $\mathbf{x}'$ is close to training data $\mathbf{X}$

EXAMPLE: CREDIT DATA



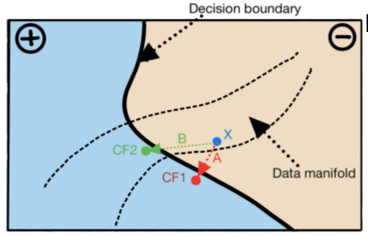
- Model: SVM with RBF kernel
- **x**: First data point of credit data with $\mathbb{P}(y = \textit{good}) = 0.34$
- Goal: Increase the probability to desired outcome $[0.5, 1]$
- MOC (with default parameters) returned 69 valid CEs after 200 iterations
- All CEs modified credit duration; many also adjusted credit amount

FURTHER OBJECTIVES: PLAUSIBILITY



Plausibility:

- CEs should suggest realistic (i.e., plausible) alternatives
  $\rightsquigarrow$ Implausible: increase income *and* become unemployed
- CEs should adhere to data manifold or originate from distribution of $\mathcal{X}$
  $\rightsquigarrow$ Avoid unrealistic combinations of feature values
- Estimating joint distribution is hard, especially for mixed feature spaces
  $\rightsquigarrow$ Common proxy: ensure that $\mathbf{x}'$ is close to training data $\mathbf{X}$



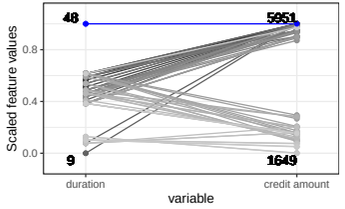
Example from ▸ Verma 2020

- Input **x** originally classified as ⊖
- Two valid CEs in class ⊕: CF1 and CF2
- Path **A** (CF1) is shorter (but unrealistic)
- Path **B** (CF2) is longer but in data manifold

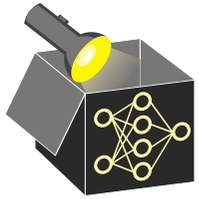
EXAMPLE: CREDIT DATA

▸ Dandl et al. (2020)

- Feature changes can be visualized using parallel and 2D surface plots
- Parallel plot: All CEs had values equal to or smaller than the values of **x**



Parallel plot: Grey lines = CEs $\mathbf{x}'$, blue line = $\mathbf{x}$.
Features without changes omitted.
Bold numbers denote numeric ranges.

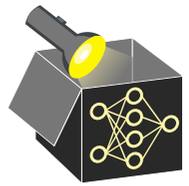


FURTHER OBJECTIVES

Plausibility term: Encourage counterfactuals close to observed data.

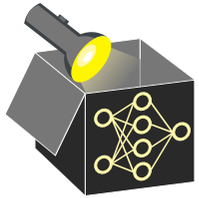
- Define $\mathbf{x}^{[1]}$ as the nearest neighbor of $\mathbf{x}'$ in the training set $\mathbf{X}$
- Use Gower distance between $\mathbf{x}'$ and $\mathbf{x}^{[1]}$ to define plausibility objective:

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

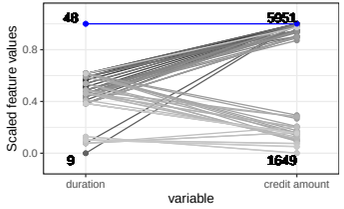


EXAMPLE: CREDIT DATA

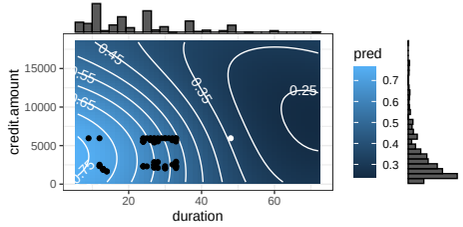
Dandl et al. (2020)



- Feature changes can be visualized using parallel and 2D surface plots
- Parallel plot: All CEs had values equal to or smaller than the values of **x**
- Surface plot: CEs in lower-left appear distant, but lie in high-density regions near training data (as shown by histograms)

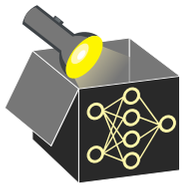


Parallel plot: Grey lines = CEs $\mathbf{x}'$, blue line = $\mathbf{x}$.
Features without changes omitted.
Bold numbers denote numeric ranges.



Surface plot: White dot = $\mathbf{x}$, black dots = CEs $\mathbf{x}'$.
Histograms: Marginal distribution of training data $\mathbf{X}$.

FURTHER OBJECTIVES



Plausibility term: Encourage counterfactuals close to observed data.

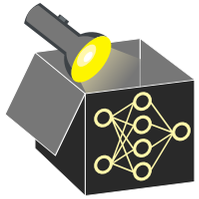
- Define $\mathbf{x}^{[1]}$ as the nearest neighbor of $\mathbf{x}'$ in the training set $\mathbf{X}$
- Use Gower distance between $\mathbf{x}'$ and $\mathbf{x}^{[1]}$ to define plausibility objective:

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

Extended optimization: Add sparsity and plausibility terms to the objective

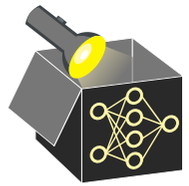
$$\arg\min_{\mathbf{x}'} \lambda_1 o_{\text{target}}(\hat{f}(\mathbf{x}'), y') + \lambda_2 o_{\text{proximity}}(\mathbf{x}', \mathbf{x}) + \lambda_3 o_{\text{sparse}}(\mathbf{x}', \mathbf{x}) + \lambda_4 o_{\text{plausible}}(\mathbf{x}', \mathbf{X})$$

PROBLEMS, PITFALLS, & LIMITATIONS



- **Illusion of model understanding:** CEs explain ML decisions by pointing to few specific alternatives, reducing complexity but offering limited explanatory power
  $\rightsquigarrow$ Psychologists have shown that although perceived model understanding of end-users increases, the objective model understanding remains unchanged

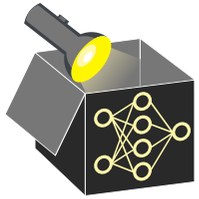
REMARKS: THE RASHOMON EFFECT



Issue (Rashomon effect):

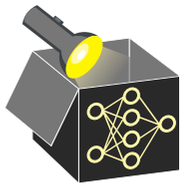
- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

PROBLEMS, PITFALLS, & LIMITATIONS



- **Illusion of model understanding:** CEs explain ML decisions by pointing to few specific alternatives, reducing complexity but offering limited explanatory power
  $\rightsquigarrow$ Psychologists have shown that although perceived model understanding of end-users increases, the objective model understanding remains unchanged
- **Right metric:** Similarity measures are crucial to find good CEs (depends on context/domain)
  $\rightsquigarrow$ e.g., L_1 can be reasonable for tabular data but not for image data
  $\rightsquigarrow$ sparsity desirable for end-users but not for auditors searching for model bias

REMARKS: THE RASHOMON EFFECT



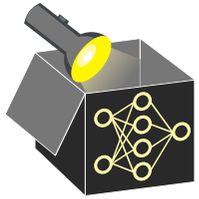
Issue (Rashomon effect):

- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

Possible solutions:

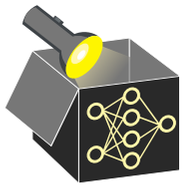
- Present all CEs for $\mathbf{x}$ (but: time and human processing capacity is limited)
- Focus on one/few CEs (but: by which criterion should guide this choice?)

PROBLEMS, PITFALLS, & LIMITATIONS



- **Illusion of model understanding:** CEs explain ML decisions by pointing to few specific alternatives, reducing complexity but offering limited explanatory power
  $\rightsquigarrow$ Psychologists have shown that although perceived model understanding of end-users increases, the objective model understanding remains unchanged
- **Right metric:** Similarity measures are crucial to find good CEs (depends on context/domain)
  $\rightsquigarrow$ e.g., L_1 can be reasonable for tabular data but not for image data
  $\rightsquigarrow$ sparsity desirable for end-users but not for auditors searching for model bias
- **Confusing Model and Real-World:** Model explanations are not easily transferable to reality
  $\rightsquigarrow$ End-users must know that CEs provide insights into a model, not real world

REMARKS: THE RASHOMON EFFECT



Issue (Rashomon effect):

- Solution to the optimization problem might not be unique
- Many equally close CE might exist that obtain the desired prediction
  $\Rightarrow$ Many different equally good explanations for the same decision exist

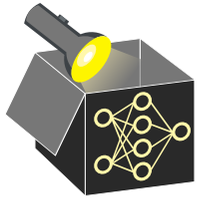
Possible solutions:

- Present all CEs for $\mathbf{x}$ (but: time and human processing capacity is limited)
- Focus on one/few CEs (but: by which criterion should guide this choice?)

Note:

- Nonlinear models can produce diverse and inconsistent CEs
  $\rightsquigarrow$ suggest both increasing and decreasing credit duration (confusing for users)
- Handling this **Rashomon effect** remains an open problem in interpretable ML

PROBLEMS, PITFALLS, & LIMITATIONS

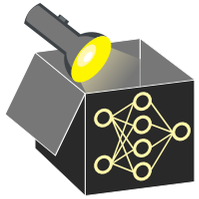


- **Illusion of model understanding:** CEs explain ML decisions by pointing to few specific alternatives, reducing complexity but offering limited explanatory power
  $\rightsquigarrow$ Psychologists have shown that although perceived model understanding of end-users increases, the objective model understanding remains unchanged
- **Right metric:** Similarity measures are crucial to find good CEs (depends on context/domain)
  $\rightsquigarrow$ e.g., L_1 can be reasonable for tabular data but not for image data
  $\rightsquigarrow$ sparsity desirable for end-users but not for auditors searching for model bias
- **Confusing Model and Real-World:** Model explanations are not easily transferable to reality
  $\rightsquigarrow$ End-users must know that CEs provide insights into a model, not real world
- **Disclosing too much information:** CEs can reveal too much information about the model and help potential attackers

REMARKS: MODEL OR REAL-WORLD

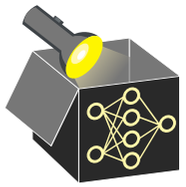
- CEs explain model predictions, but may seem to explain real-world users
  $\rightsquigarrow$ Transfer of model explanations to explain real-world is generally not permitted
- **Example:** CE suggests increasing age by 5 to receive a loan
  $\rightsquigarrow$ The applicant waits 5 years and reapplies

PROBLEMS, PITFALLS, & LIMITATIONS



- **Rashomon effect:** One, few, all? Which CEs should be shown to the end-user?
  ⇝ No universal answer; depends on user goals, cognitive load, and resources

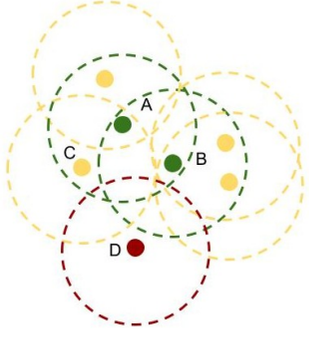
REMARKS: MODEL OR REAL-WORLD



- CEs explain model predictions, but may seem to explain real-world users
  ⇝ Transfer of model explanations to explain real-world is generally not permitted
- **Example:** CE suggests increasing age by 5 to receive a loan
  ⇝ The applicant waits 5 years and reapplies
- **Problem:** Other features may change in the meantime (e.g., job status, income) ⇝ ▸ Karimi 2020 propose CEs that respect causal structure
- **Model drift:** Bank's algorithm itself may change over time
  ⇝ Past CEs may become invalid

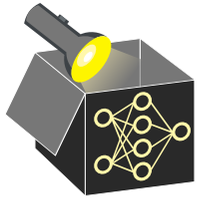
Interpretable Machine Learning

Increasing Trust in Explanations



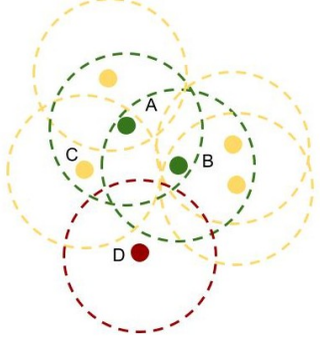
Learning goals

- Understand the aspects that undermine users' trust in an explanation
- Learn diagnostic tools that could increase trust



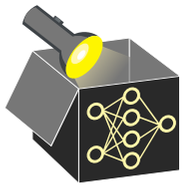
Interpretable Machine Learning

Local Explanations: Increasing Trust in Explanations



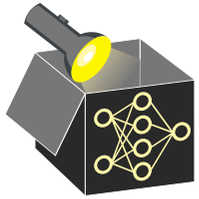
Learning goals

- Understand the aspects that undermine users' trust in an explanation
- Learn diagnostic tools that could increase trust



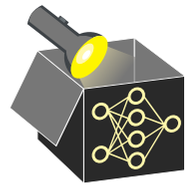
MOTIVATION & IMPORTANT PROPERTIES

- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy



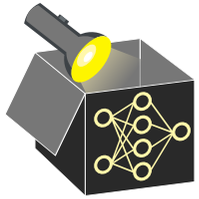
MOTIVATION & IMPORTANT PROPERTIES

- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy



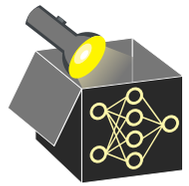
MOTIVATION & IMPORTANT PROPERTIES

- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”

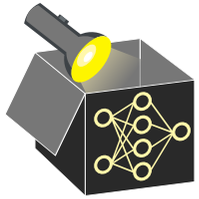


MOTIVATION & IMPORTANT PROPERTIES

- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”

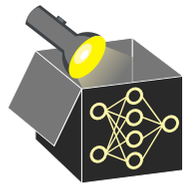


MOTIVATION & IMPORTANT PROPERTIES



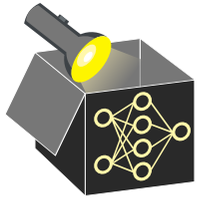
- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)

MOTIVATION & IMPORTANT PROPERTIES



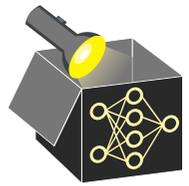
- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)

MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist
     - ⇝ measure how robust an IML method is to small changes in the input data or parameters
     - ⇝ Is an observation out-of-distribution?

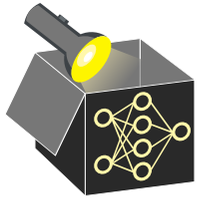
MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist
     - ⇝ measure how robust an IML method is to small changes in the input data or parameters
     - ⇝ Is an observation out-of-distribution?

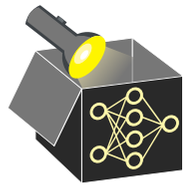
Interpretable Machine Learning – 1 / 7

MOTIVATION & IMPORTANT PROPERTIES



- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist
     - $\rightsquigarrow$ measure how robust an IML method is to small changes in the input data or parameters
     - $\rightsquigarrow$ Is an observation out-of-distribution?
- Failing in one of these $\rightsquigarrow$ undermining users’ trust in the explanations
  $\rightsquigarrow$ undermining trust in the model

MOTIVATION & IMPORTANT PROPERTIES

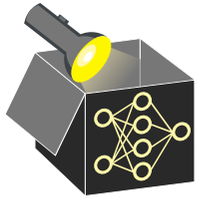


- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **Interpretable**: “Why did the model come up with this decision?”
- **Trustworthy**: “How certain is this explanation?”
  1. accurate insights into the inner workings of our model
     - Failure case: generation is based on inputs in areas where the model was trained with little or no training data (extrapolation)
  2. robust (i.e. low variance)
     - Expectation: similar explanations for similar data points with similar predictions
     - However, multiple sources of uncertainty exist
     - $\rightsquigarrow$ measure how robust an IML method is to small changes in the input data or parameters
     - $\rightsquigarrow$ Is an observation out-of-distribution?
- Failing in one of these $\rightsquigarrow$ undermining users’ trust in the explanations
  $\rightsquigarrow$ undermining trust in the model

Interpretable Machine Learning – 1 / 7

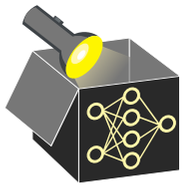
OUT-OF-DISTRIBUTION DETECTION

- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable

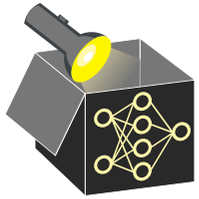


OUT-OF-DISTRIBUTION (OOD) DETECTION

- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable

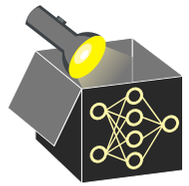


OUT-OF-DISTRIBUTION DETECTION



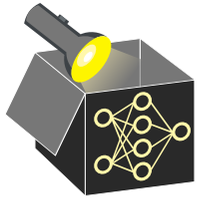
- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted observations to calculate the marginal contributions
  - ICE curves grid data points

OUT-OF-DISTRIBUTION (OOD) DETECTION



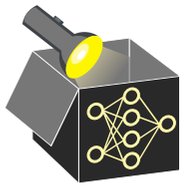
- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted obs. to calculate the marginal contribs
  - ICE curves grid data points

OUT-OF-DISTRIBUTION DETECTION



- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted observations to calculate the marginal contributions
  - ICE curves grid data points
- Two very simple and intuitive approaches
  - Classifier for out-of-distribution
  - Clustering
- More complicated also possible, e.g., variational autoencoders [Daxberger et al. 2020]

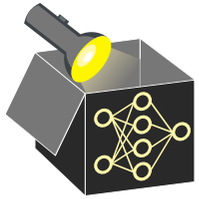
OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
  $\rightsquigarrow$ explanations from local explanation methods are unreliable
- For local explanation methods, the following components could be out-of-distribution (OOD):
  - The data for LIME's surrogate model
  - Counterfactuals themselves
  - Shapley value's permuted obs. to calculate the marginal contribs
  - ICE curves grid data points
- Two very simple and intuitive approaches
  - Classifier for out-of-distribution
  - Clustering
- More complicated also possible, e.g., variational autoencoders

▸ Daxberger 2020

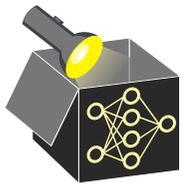
OUT-OF-DISTRIBUTION DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- Idea: Hallucinate new (out-of-distribution) data by randomly sample data points

$\rightsquigarrow$ Learn a binary classifier to distinguish between the origins of the data

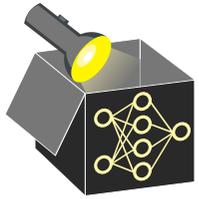
OOD DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- Idea: Hallucinate new (ood) data by randomly sampling data points

$\rightsquigarrow$ Learn a binary classifier to distinguish between the origins of the data

OUT-OF-DISTRIBUTION DETECTION: OOD-CLASSIFIER



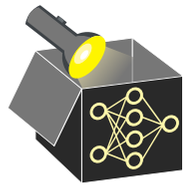
- Problem: we have only in-distribution data
- Idea: Hallucinate new (out-of-distribution) data by randomly sample data points

⇝ Learn a binary classifier to distinguish between the origins of the data

- Study whether an explanation approach can be fooled ▸ Dylan Slack et al. 2020
  - Hide bias in the true (deployed) model, but use an unbiased model for all out-of-distribution samples

⇝ Important way to diagnose an explanation approach

OOD DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- Idea: Hallucinate new (ood) data by randomly sampling data points

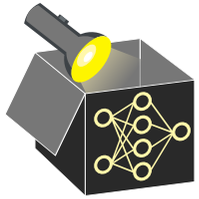
⇝ Learn a binary classifier to distinguish between the origins of the data

- Study whether an explanation approach can be fooled ▸ Slack 2020
  - Hide bias in the true (deployed) model, but use an unbiased model for all out-of-distribution samples

⇝ Important way to diagnose an explanation approach

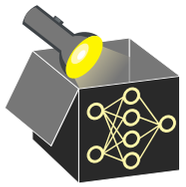
OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ▸ Martin Ester et al. 1996
  (Density-Based Spatial Clustering of Applications with Noise)

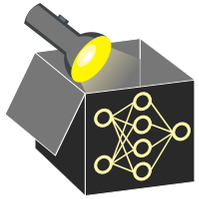


OOD DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)



OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN

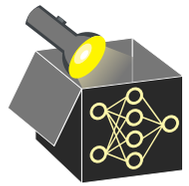


- DBSCAN is a data clustering algorithm ▸ Martin Ester et al. 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)

OOD DETECTION: CLUSTERING VIA DBSCAN

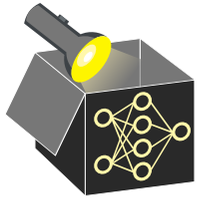


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)

OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN

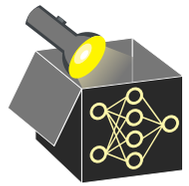


- DBSCAN is a data clustering algorithm ▸ Martin Ester et al. 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points

OOD DETECTION: CLUSTERING VIA DBSCAN

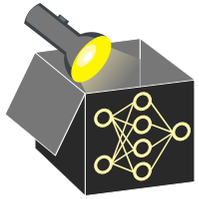


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points

OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN

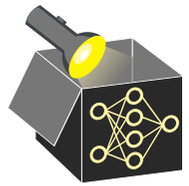


- DBSCAN is a data clustering algorithm ▸ Martin Ester et al. 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point

OOD DETECTION: CLUSTERING VIA DBSCAN

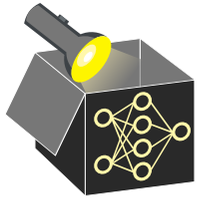


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point

OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN

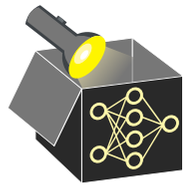


- DBSCAN is a data clustering algorithm ▸ Martin Ester et al. 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point
- Noise points
  - Are not within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Not part of any cluster

OOD DETECTION: CLUSTERING VIA DBSCAN

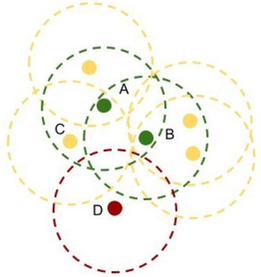


- DBSCAN is a data clustering algorithm ▸ Ester 1996
  (Density-Based Spatial Clustering of Applications with Noise)
- For this method, we define an ϵ-neighborhood:
  Given a dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, an ϵ-neighborhood for $\mathbf{x} \in \mathcal{X}$ is defined as

$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

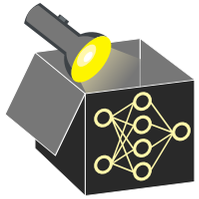
  $d(\cdot)$ is a distance measure (e.g., Euclidean or Gower distance)
- Core observations $\mathbf{x}$
  - Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Forms an own cluster with all its neighborhood points
- Border points
  - Within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Part of a cluster defined by a core point
- Noise points
  - Are not within $\mathcal{N}_\epsilon(\mathbf{x})$
  - Not part of any cluster

OUT-OF-DISTRIBUTION DETECTION

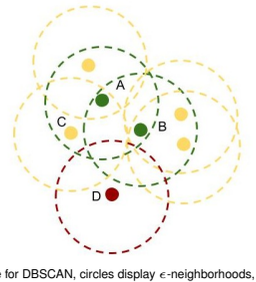


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster

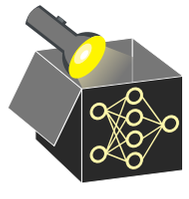


OUT-OF-DISTRIBUTION DETECTION

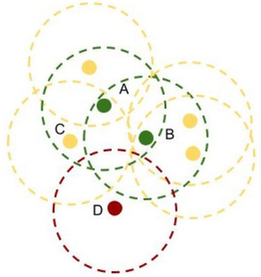


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster

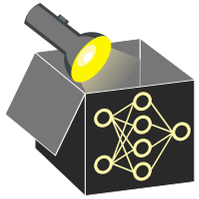


OUT-OF-DISTRIBUTION DETECTION

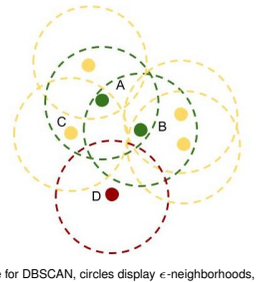


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point

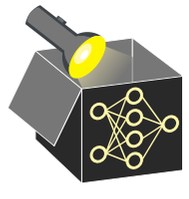


OUT-OF-DISTRIBUTION DETECTION

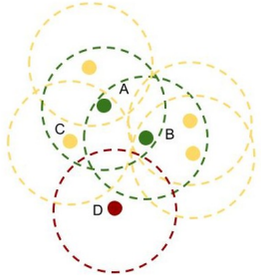


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point

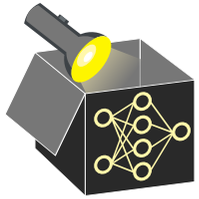


OUT-OF-DISTRIBUTION DETECTION

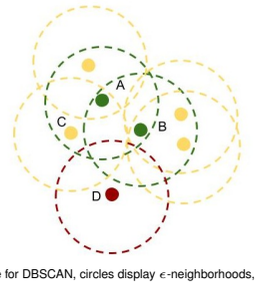


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster

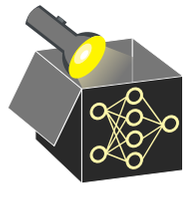


OUT-OF-DISTRIBUTION DETECTION

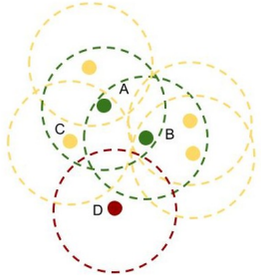


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster

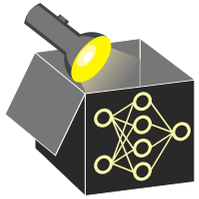


OUT-OF-DISTRIBUTION DETECTION

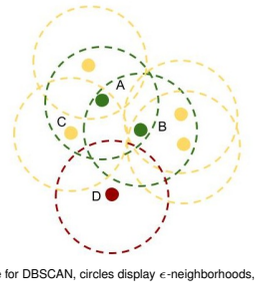


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters

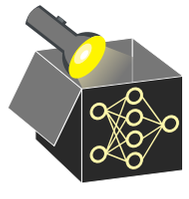


OUT-OF-DISTRIBUTION DETECTION

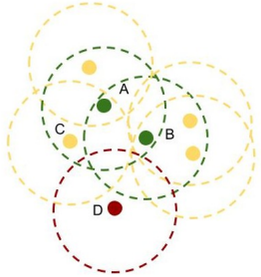


Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$

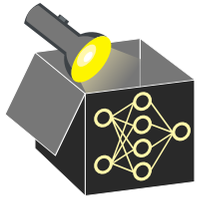
- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters



OUT-OF-DISTRIBUTION DETECTION



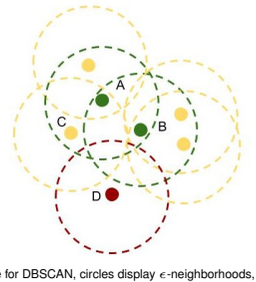
Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$



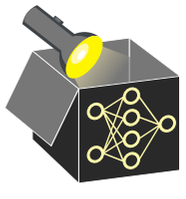
- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters

- Disadvantages:
  - Depending on the distance metric $d(\cdot)$, DBSCAN could suffer from the "curse of dimensionality"
  - The choice of ϵ and m is not clear a-priori

OUT-OF-DISTRIBUTION DETECTION



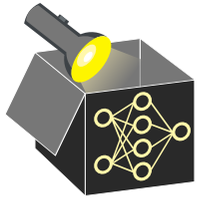
Example for DBSCAN, circles display ϵ-neighborhoods, $m = 4$



- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters

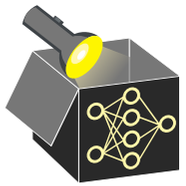
- Disadvantages:
  - Depending on the distance metric $d(\cdot)$, DBSCAN could suffer from the "curse of dimensionality"
  - The choice of ϵ and m is not clear a-priori

ROBUSTNESS



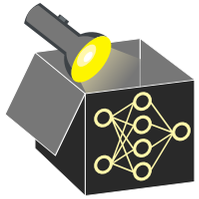
- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the explanation method and which hyperparameters

ROBUSTNESS



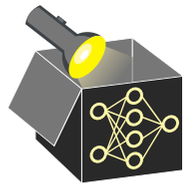
- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the expl. method and which hyperparams

ROBUSTNESS



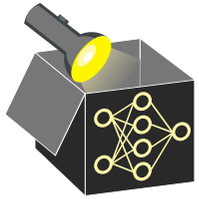
- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the explanation method and which hyperparameters
  2. **Process uncertainty**: Change of explanation if the underlying model is changed
     $\rightsquigarrow$ are ML models non-robust, e.g., because they are trained on noisy data?

ROBUSTNESS



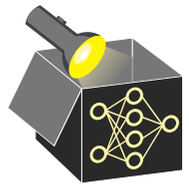
- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the expl. method and which hyperparams
  2. **Process uncertainty**: Change of explanation if the underlying model is changed
     $\rightsquigarrow$ are ML models non-robust, e.g., because they are trained on noisy data?

ROBUSTNESS



- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the explanation method and which hyperparameters
  2. **Process uncertainty**: Change of explanation if the underlying model is changed
     $\rightsquigarrow$ are ML models non-robust, e.g., because they are trained on noisy data?
- We focus on explanation uncertainty
  - Even with the same model and same (or similar) data points, we can receive different explanations

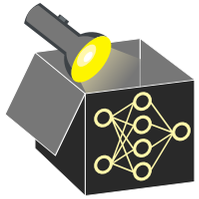
ROBUSTNESS



- Differentiate between different kinds of uncertainty:
  1. **Explanation uncertainty**: Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the expl. method and which hyperparams
  2. **Process uncertainty**: Change of explanation if the underlying model is changed
     $\rightsquigarrow$ are ML models non-robust, e.g., because they are trained on noisy data?
- We focus on explanation uncertainty
  - Even with the same model and same (or similar) data points, we can receive different explanations

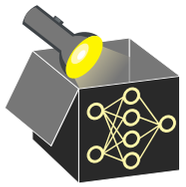
ROBUSTNESS MEASURE FOR LIME AND SHAP

- Objective: Similar explanations for similar inputs (in a neighborhood)

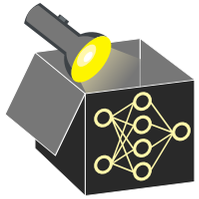


ROBUSTNESS MEASURE FOR LIME AND SHAP

- Objective: Similar explanations for similar inputs (in a neighborhood)



ROBUSTNESS MEASURE FOR LIME AND SHAP

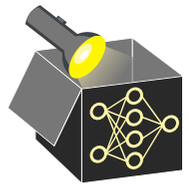


- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Alvarez-Melis and Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

  Note that, for LIME, g returns the m coefficients of the surrogate model

ROBUSTNESS MEASURE FOR LIME AND SHAP

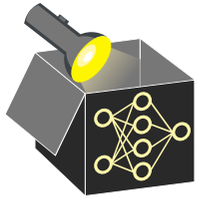


- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

  Note that, for LIME, g returns the m coefficients of the surrogate model

ROBUSTNESS MEASURE FOR LIME AND SHAP



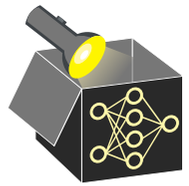
- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Alvarez-Melis and Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

  Note that, for LIME, g returns the m coefficients of the surrogate model
- According to this, we can quantify the robustness of explanation models in terms of ω:

  $\rightsquigarrow$ The closer ω is to 0, the more robust our explanation method is

ROBUSTNESS MEASURE FOR LIME AND SHAP



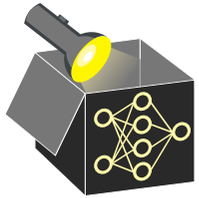
- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

  Note that, for LIME, g returns the m coefficients of the surrogate model
- According to this, we can quantify the robustness of explanation models in terms of ω:

  $\rightsquigarrow$ The closer ω is to 0, the more robust our explanation method is

ROBUSTNESS MEASURE FOR LIME AND SHAP



- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Alvarez-Melis and Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

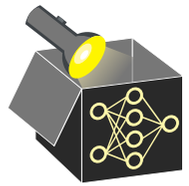
  Note that, for LIME, g returns the m coefficients of the surrogate model
- According to this, we can quantify the robustness of explanation models in terms of ω:

  $\rightsquigarrow$ The closer ω is to 0, the more robust our explanation method is
- ω is rarely known a-priori but it could be estimated as follows:

$$\hat{\omega}_X(\mathbf{x}) \in \underset{\mathbf{x}^{(i)} \in \mathcal{N}_\epsilon(\mathbf{x})}{\arg\max} \frac{||g(\mathbf{x}) - g(\mathbf{x}^{(i)})||_2}{d(\mathbf{x}, \mathbf{x}^{(i)})},$$

  where $\mathcal{N}_\epsilon(\mathbf{x})$ is the ϵ-neighborhood of $\mathbf{x}$

ROBUSTNESS MEASURE FOR LIME AND SHAP



- Objective: Similar explanations for similar inputs (in a neighborhood)
- For LIME and SHAP, notion of stability based on **locally Lipschitz continuity** ▸ Jaakkola 2018:

  An explanation method $g : \mathcal{X} \to \mathbb{R}^m$ is locally Lipschitz if
  - for every $\mathbf{x}_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
  - such that $||\mathbf{x} - \mathbf{x}_0|| < \delta$ implies $||g(\mathbf{x}) - g(\mathbf{x}_0)|| < \omega||\mathbf{x} - \mathbf{x}_0||$

  Note that, for LIME, g returns the m coefficients of the surrogate model
- According to this, we can quantify the robustness of explanation models in terms of ω:

  $\rightsquigarrow$ The closer ω is to 0, the more robust our explanation method is
- ω is rarely known a-priori but it could be estimated as follows:

$$\hat{\omega}_X(\mathbf{x}) \in \underset{\mathbf{x}^{(i)} \in \mathcal{N}_\epsilon(\mathbf{x})}{\arg\max} \frac{||g(\mathbf{x}) - g(\mathbf{x}^{(i)})||_2}{d(\mathbf{x}, \mathbf{x}^{(i)})},$$

  where $\mathcal{N}_\epsilon(\mathbf{x})$ is the ϵ-neighborhood of $\mathbf{x}$