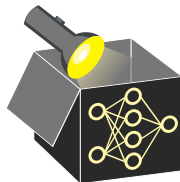
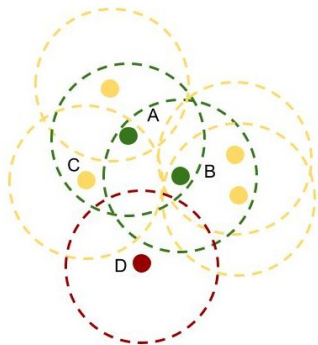


Interpretable Machine Learning



Local Explanations: Increasing Trust in Explanations

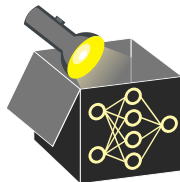


Learning goals

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

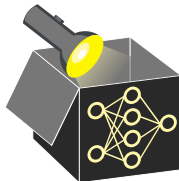
MOTIVATION & IMPORTANT PROPERTIES

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



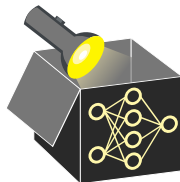
MOTIVATION & IMPORTANT PROPERTIES

- Local explanations should not only make a model interpretable but also reveal if the model is trustworthy
- **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?”
 - ❶ 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)

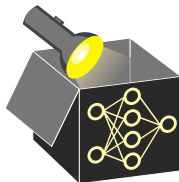


MOTIVATION & IMPORTANT PROPERTIES

- 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

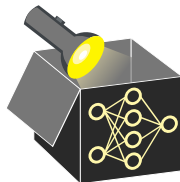
~> measure how robust an IML method is to small changes in the input data or parameters

~> Is an observation out-of-distribution?



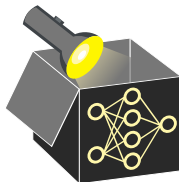
MOTIVATION & IMPORTANT PROPERTIES

- 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?”
 - ❶ 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)
 - ❷ 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?
- Failing in one of these ↪ undermining users' trust in the explanations
 - ↪ undermining trust in the model

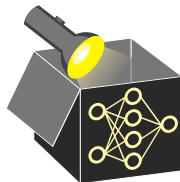


OUT-OF-DISTRIBUTION (OOD) DETECTION

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

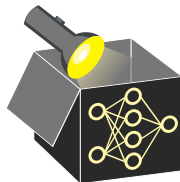


OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
~> 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

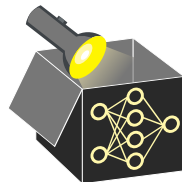
OUT-OF-DISTRIBUTION (OOD) DETECTION



- Models are unreliable in areas with little data support
~> 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

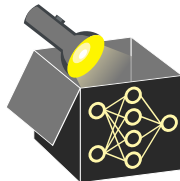
► "Daxberger et al." 2020

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

OOD DETECTION: OOD-CLASSIFIER

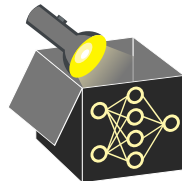


- Problem: we have only in-distribution data
 - Idea: Hallucinate new (ood) data by randomly sampling data points
- ~> Learn a binary classifier to distinguish between the origins of the data
- Study whether an explanation approach can be fooled
- Hide bias in the true (deployed) model, but use an unbiased model for all out-of-distribution samples
- ~> Important way to diagnose an explanation approach

► "Dylan Slack et al." 2020

OOD DETECTION: CLUSTERING VIA DBSCAN

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

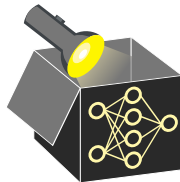


OOD DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ► "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

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

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



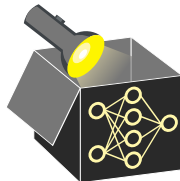
OOD DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ► "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

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

$d(\cdot)$ 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



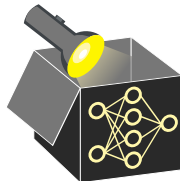
OOD DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ► "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

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

$d(\cdot)$ 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



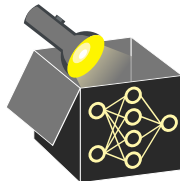
OOD DETECTION: CLUSTERING VIA DBSCAN

- DBSCAN is a data clustering algorithm ► "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

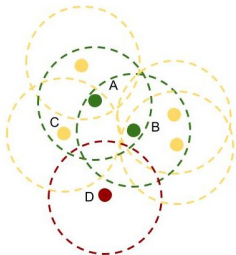
$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} \mid d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

$d(\cdot)$ 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

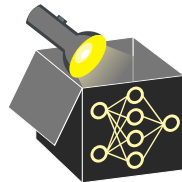


OUT-OF-DISTRIBUTION DETECTION

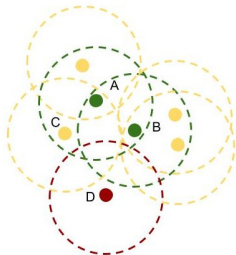


Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

- 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

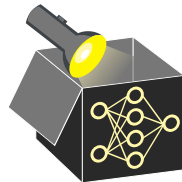


OUT-OF-DISTRIBUTION DETECTION

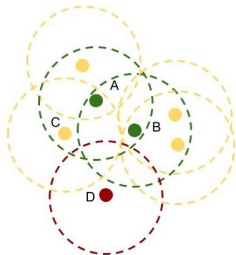


Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

- 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
- Since D is not part of the neighborhood of core points, it is a noise point

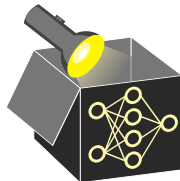


OUT-OF-DISTRIBUTION DETECTION

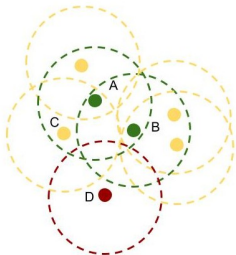


Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

- 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

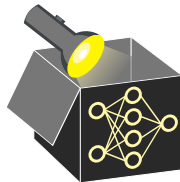


OUT-OF-DISTRIBUTION DETECTION

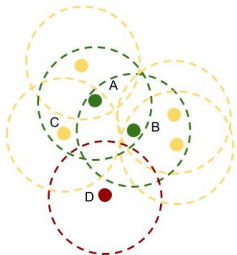


Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

- 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: new point lies outside the clusters



OUT-OF-DISTRIBUTION DETECTION

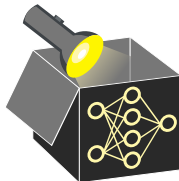


Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

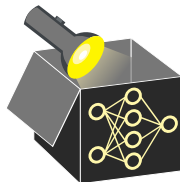
- Green points A and B are core points and form one cluster since they lie in each others neighborhood, all yellow points are border points of this cluster
- Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
- Out-of-distribution: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

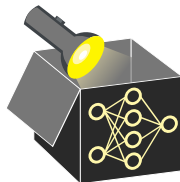


ROBUSTNESS



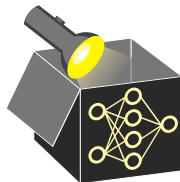
- Differentiate between different kinds of uncertainty:
 - ❶ **Explanation uncertainty:** Change of explanation if we repeat the process, e.g., the explanation could differ depending on which subset of data we use for the expl. method and which hyperparams

ROBUSTNESS



- Differentiate between different kinds of uncertainty:
 - ❶ **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
 - ❷ **Process uncertainty:** Change of explanation if the underlying model is changed
~> are ML models non-robust, e.g., because they are trained on noisy data?

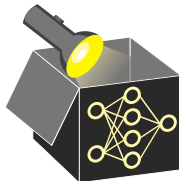
ROBUSTNESS



- Differentiate between different kinds of uncertainty:
 - ❶ **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
 - ❷ **Process uncertainty:** Change of explanation if the underlying model is changed
 - ↪ 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)



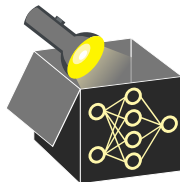
ROBUSTNESS MEASURE FOR LIME AND SHAP

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

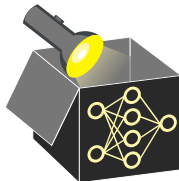
An explanation method $g : \mathcal{X} \rightarrow \mathbb{R}^m$ is locally Lipschitz if

- 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:

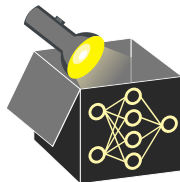
An explanation method $g : \mathcal{X} \rightarrow \mathbb{R}^m$ is locally Lipschitz if

- 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 :
 - ↪ 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 $g : \mathcal{X} \rightarrow \mathbb{R}^m$ 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 :
 - ↪ 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:

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

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