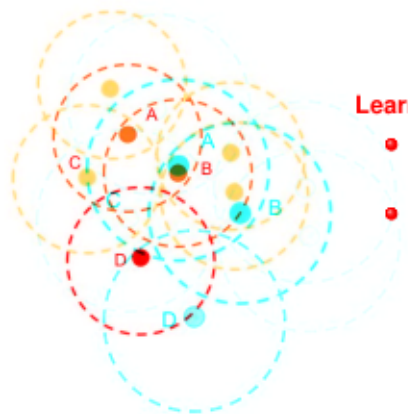


Interpretable Machine Learning

Increasing Trust in Explanations

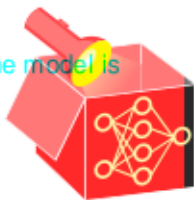


Learning goals

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

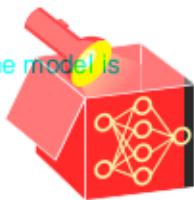
MOTIVATION & IMPORTANT PROPERTIES

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



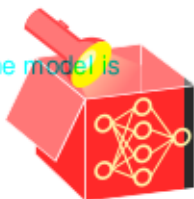
MOTIVATION & IMPORTANT PROPERTIES

- 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?"

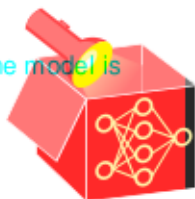


MOTIVATION & IMPORTANT PROPERTIES

- 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?"
 - 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)

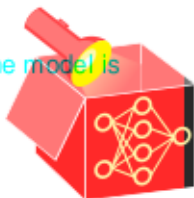


MOTIVATION & IMPORTANT PROPERTIES



- 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?"
 - 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)
 - robust (i.e. low variance)
 - Expectation: similar explanations for similar data points with similar predictions
 - multiple sources of uncertainty exist
 - However, multiple sources of uncertainty exist
- ~> measure how robust an IML method is to small changes in the input
- ~> data or parameters out-of-distribution?
- ~> 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?"
 - 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)
 - robust (i.e. low variance)
 - Expectation: similar explanations for similar data points with similar predictions
 - multiple sources of uncertainty exist
 - However, multiple sources of uncertainty exist
 - small changes in the input data or data or parameters
 - measure how robust an IML method is to small changes in the input
 - data or parameters out-of-distribution?
 - Is an observation out-of-distribution?
- Failing in one of these ~> undermining users' trust in the explanations
- Failing in one of these ~> undermining users' trust in the explanations
~> undermining trust in the model

OUT-OF-DISTRIBUTION DETECTION

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



OUT-OF-DISTRIBUTION DETECTION



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

OUT-OF-DISTRIBUTION DETECTION



- Models are unreliable in areas with little data support
 - Models are unreliable in areas with little data support
 - ~> explanations from local explanation methods are unreliable
 - ~> 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]
- More complicated also possible, e.g., variational autoencoders [Daxberger et al. 2020]

OUT-OF-DISTRIBUTION DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- 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

OUT-OF-DISTRIBUTION DETECTION: OOD-CLASSIFIER



- Problem: we have only in-distribution data
- 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 Sack et al. 2020
- Study whether an explanation approach can be fooled • Dylan Sack 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
- 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
- DBSCAN is a data clustering algorithm (Applications with Noise)
(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)
- DBSCAN is a data clustering algorithm (Applications with Noise)
- (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

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

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

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

$d(\cdot)$ 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

- DBSCAN is a data clustering algorithm (Applications with Noise)

- (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

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

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

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

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

- Core observations \mathbf{x} have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$

- Form an own cluster with all its neighborhood points

- Have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$
- Form an own cluster with all its neighborhood points

OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN



• Martin Ester et al. 1996

- DBSCAN is a data clustering algorithm (Applications with Noise)

- (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

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

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

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

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

- Core observations \mathbf{x} have at least m data points within $\mathcal{N}_\epsilon(\mathbf{x})$

- Form an own cluster with all its neighborhood points

- Border points
- Form 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

- Part of a cluster defined by a core point

OUT-OF-DISTRIBUTION DETECTION: CLUSTERING VIA DBSCAN



• Martin Ester et al. 1996

- DBSCAN is a data clustering algorithm (Applications with Noise)
- (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

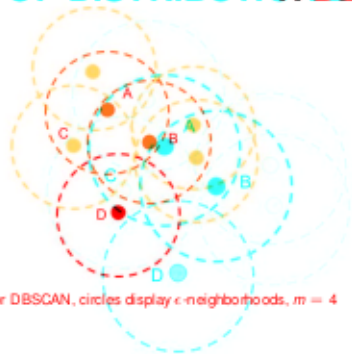
$$\mathcal{N}_\epsilon(\mathbf{x}) = \{\mathbf{x}^{(i)} \in \mathcal{X} | d(\mathbf{x}, \mathbf{x}^{(i)}) \leq \epsilon\}.$$

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

$d(\cdot)$ 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



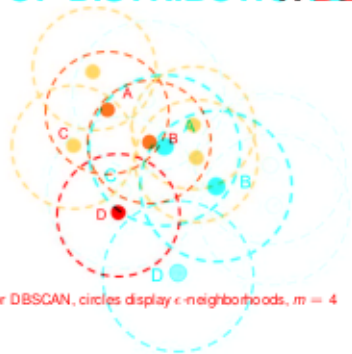
Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

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
- 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



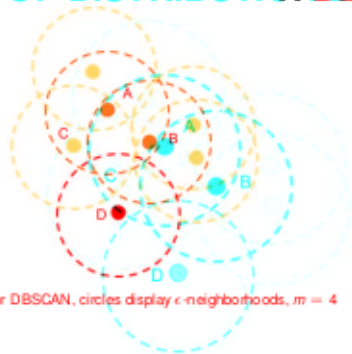
Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

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



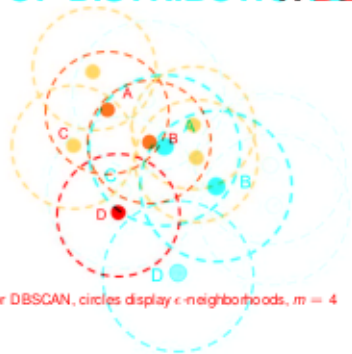
Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

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
 - 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
 - Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
 - In-distribution: new point lies within a cluster



OUT-OF-DISTRIBUTION DETECTION



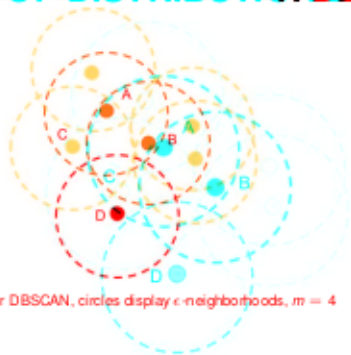
Example for DBSCAN, circles display ϵ -neighborhoods, $m = 4$

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: new point lies outside the clusters



OUT-OF-DISTRIBUTION DETECTION



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

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
 - 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
 - Since D is not part of the neighborhood of core points, it is a noise point
- In-distribution: new point lies within a cluster
 - In-distribution: new point lies within a cluster
- Out-of-distribution: new point lies outside the clusters
 - 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"
- 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



- Differentiate between different kinds of uncertainty:
 - **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



- 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
 - are ML models non-robust, e.g., because they are trained on noisy data?
 - ↪ are ML models non-robust, e.g., because they are trained on noisy data?



- Differentiate between different kinds of uncertainty:
 - **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
 - **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?
 - ↪ 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
- 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 $x_0 \in \mathcal{X}$ there exist $\delta > 0$ and $\omega \in \mathbb{R}$
- such that $\|x - x_0\| < \delta$ implies $\|g(x) - g(x_0)\| \leq \omega \|x - x_0\|$

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 $\|g(\mathbf{x}) - g(\mathbf{x}_0)\| \leq \omega \|\mathbf{x} - \mathbf{x}_0\|$

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
 \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 $g: \mathcal{X} \rightarrow \mathbb{R}^m$ 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 $\|g(\mathbf{x}) - g(\mathbf{x}_0)\| \leq \omega \|\mathbf{x} - \mathbf{x}_0\|$

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

- $\omega \rightarrow 0$ 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:

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

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

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