Interpretable models with LIME and SHAP

Chloé-Agathe Azencott

Center for Computational Biology (CBIO)
Mines ParisTech – Institut Curie – INSERM U900
PSL Research University & PR[AI]RIE, Paris, France

September 29, 2021

http://cazencott.info chloe-agathe.azencott@mines-paristech.fr @cazencott

Model interpretability

Why is my model making these predictions?

- Drive scientific hypotheses
- Detect bias
- Acceptance

Doshi-Velez and Kim 2017

Global vs local explanations

- **Global explanations:** How does a **specific part of the model** affect the predictions?

  "part of the model":
  - a **feature** or set of features
  - a **training sample** or set of samples

  Example: coefficient in a linear model, random forest importance (see next slide)

Global vs local explanations

- **Global explanations:** How does a **specific part of the model** affect the predictions?

  "part of the model":
  - a **feature** or set of features
  - a **training sample** or set of samples

  Example: coefficient in a linear model, random forest importance (see next slide)

- **Local explanations:** Why does the model make this prediction for a **specific instance?**

Lipton 2016

Global vs local explanations

- **Global explanations:** How does a **specific part of the model** affect the predictions?

  "part of the model":
  - a **feature** or set of features
  - a **training sample** or set of samples

  Example: coefficient in a linear model, random forest importance (see next slide)

- **Local explanations:** Why does the model make this prediction for a **specific instance?**

By extension: **aggregate** local explanations to understand why the model makes these predictions for the **entire dataset** (or an entire class
van der Linden, Haned, and Kanoulas 2019

Lipton 2016

Random forest feature importance

- **Global explanations**
- **Mean Decrease in Impurity** (`feature_importance` attribute in sklearn):

  Mean decrease in impurity attributed to the feature

Breiman 2001; Louppe 2014

Random forest feature importance

- **Global explanations**
- **Mean Decrease in Impurity** (`feature_importance` attribute in sklearn):

  Mean decrease in impurity attributed to the feature

  ☹ Seem to favor numerical features and categorical features with high cardinality

Breiman 2001; Loupe 2014

Random forest feature importance

- **Global explanations**
- **Mean Decrease in Impurity** (`feature_importance` attribute in sklearn):

  Mean decrease in impurity attributed to the feature

  ☹ Seem to favor numerical features and categorical features with high cardinality

- **Permutation importance** (`inspection.permutation_importance` in sklearn):

  Decrease in model score when the feature is randomly shuffled in the train set

Breiman 2001; Louppe 2014

Random forest feature importance

- **Global explanations**
- **Mean Decrease in Impurity** (`feature_importance` attribute in sklearn):

  Mean decrease in impurity attributed to the feature

  ☹ Seem to favor numerical features and categorical features with high cardinality
- **Permutation importance** (`inspection.permutation_importance` in sklearn):

  Decrease in model score when the feature is randomly shuffled in the train set

  ☺ Can be used with any model!

Breiman 2001; Louppe 2014

Random forest feature importance

- **Global explanations**
- **Mean Decrease in Impurity** (`feature_importance` attribute in sklearn):

  Mean decrease in impurity attributed to the feature

  ☹ Seem to favor numerical features and categorical features with high cardinality

- **Permutation importance** (`inspection.permutation_importance` in sklearn):

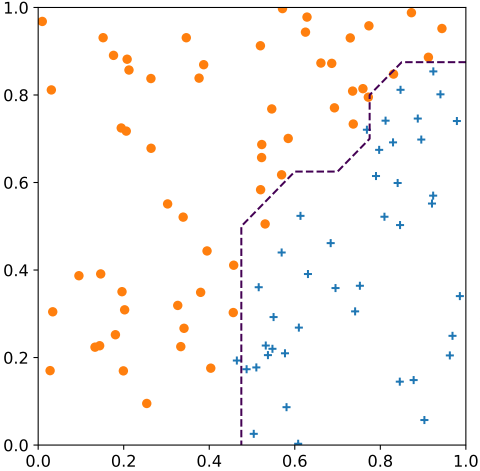
  Decrease in model score when the feature is randomly shuffled in the train set

  ☺ Can be used with any model!

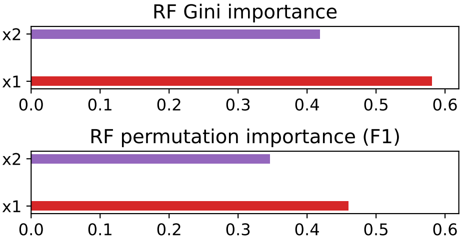
☹ Not robust to correlations between features

Breiman 2001; Louppe 2014

Example



1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



RF Gini importance
x2
x1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
RF permutation importance (F1)
x2
x1
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Outline

Objective: Given

- training data $\mathcal{D} = \{\vec{x}_i, y_i\}_{i=1,\dots,n}$, with $\vec{x}_i \in \mathbb{R}^p$,
- a model $\boldsymbol{f}$ that has been learned on $\mathcal{D}$,
- an instance $\vec{\boldsymbol{x}} \in \mathbb{R}^p$,

find a **local explanation** for $\boldsymbol{f}(\vec{\boldsymbol{x}})$

Outline

Objective: Given

- training data $\mathcal{D} = \{\vec{x}_i, y_i\}_{i=1,\dots,n}$, with $\vec{x}_i \in \mathbb{R}^p$,
- a model $\boldsymbol{f}$ that has been learned on $\mathcal{D}$,
- an instance $\vec{\boldsymbol{x}} \in \mathbb{R}^p$,

find a **local explanation** for $\boldsymbol{f}(\vec{\boldsymbol{x}})$

1. LIME: Local Interpretable Model-agnostic Explanations

2. Shapley values

3. SHAP

LIME: Local Interpretable Model-agnostic Explanations

- **Local surrogate** model: an interpretable model $g \in \mathcal{G}$ that approximates the trained model
- Algorithm:
  - Generate a labeled data set $\mathcal{Z}$ of m **perturbed samples:**
    for $i = 1, \ldots, m$
    for $j = 1, \ldots, p$: sample z_j from $\mathcal{N}(\mu_j, \sigma_j^2)$ — μ_j, σ_j^2 computed on $\mathcal{D}$
    label $\vec{z}_i$ by $\boldsymbol{f}(\vec{z}_i)$
  - Compute **weights** w_i inversely proportional to $||\vec{z}_i - \vec{\boldsymbol{x}}||_2$ — $w_i = \sqrt{\frac{\exp(-||\vec{z}_i - \vec{\boldsymbol{x}}||_2^2)}{0.75^2\, p}}$
  - Train a model from $\mathcal{G}$ on $\mathcal{Z}$, weighting the loss of sample i by w_i

$$\arg\min_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} w_i L(\underbrace{\boldsymbol{f}(\vec{z}_i)}_{\text{true label}}, \underbrace{g(\vec{z}_i)}_{\text{prediction}}) + \lambda \underbrace{\Omega(g)}_{\text{model complexity}}$$

Ribeiro, Singh, and Guestrin 2016

LIME with linear surrogate models

$$\arg\min_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\,\Omega(g)$$

becomes

$$\arg\min_{\vec{\beta}\in\mathbb{R}^p} \frac{1}{m}\sum_{i=1}^{m} w_i \left(\boldsymbol{f}(\vec{z}_i) - \langle\vec{\beta}, \vec{z}_i\rangle\right)^2 + \lambda \left|\left|\vec{\beta}\right|\right|_1$$

– Set λ so as to select a user-defined number of features/explanations.

LIME with linear surrogate models

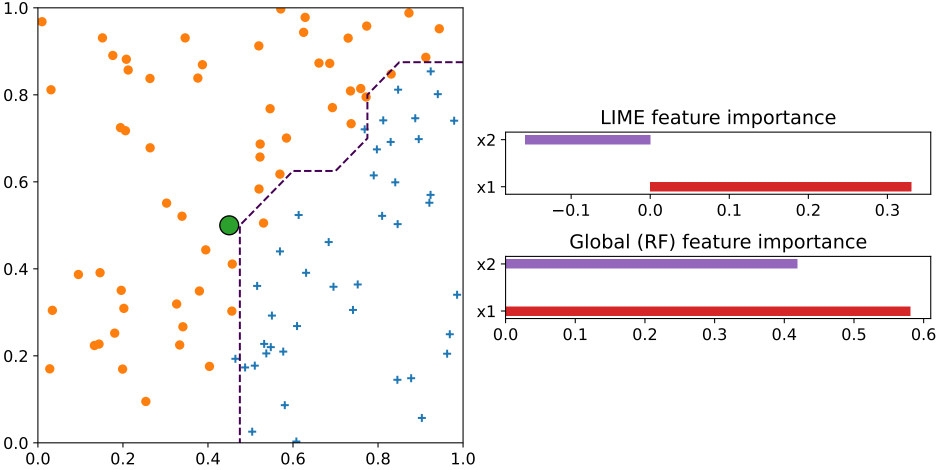
$$\arg\min_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\, \Omega(g)$$

becomes

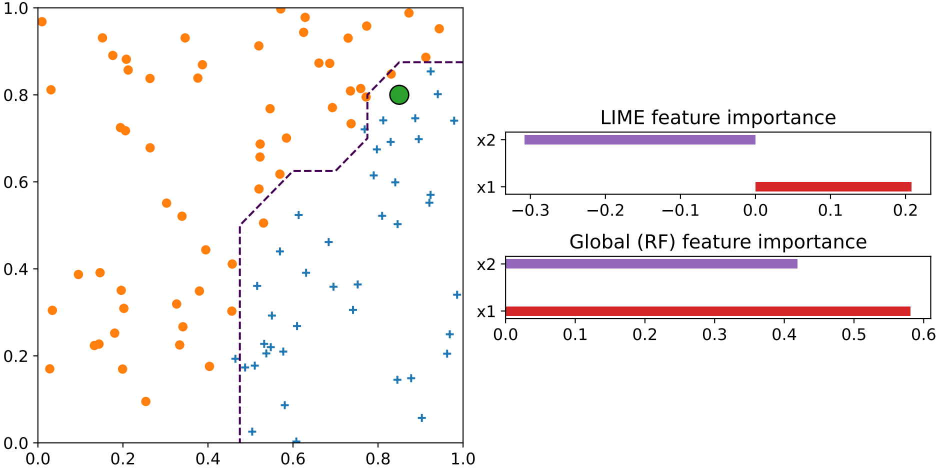
$$\arg\min_{\vec{\beta} \in \mathbb{R}^p} \frac{1}{m} \sum_{i=1}^{m} w_i \left(\boldsymbol{f}(\vec{z}_i) - \langle \vec{\beta}, \vec{z}_i \rangle\right)^2 + \lambda \left|\left|\vec{\beta}\right|\right|_1$$

- Set λ so as to select a user-defined number of features/explanations.
- Alternatives include **decision trees** (and then $\Omega(g)$ is the number of features used + tree depth).

LIME Example 1



LIME Example 2



Global explanations from LIME

- Features that explain many different instances are more important
- Given a budget of B instances to look at:

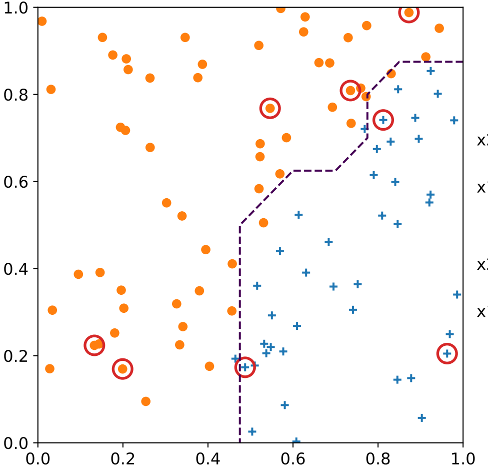
$$I_j = \sqrt{\sum_{i=1}^{B} |\beta_j^i|}$$

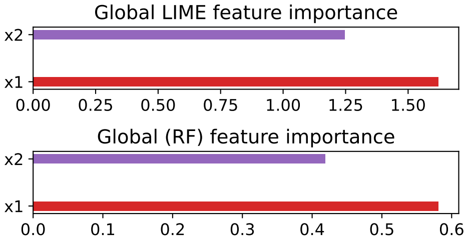
- For visualization: find a subset $\mathcal{V}$ of instances with greater **coverage**

$$c(\mathcal{V}) = \sum_{j=1}^{p} I_j \; \mathbb{1}_{\exists i \in \mathcal{V} : |\beta_j^i| > 0}$$

Voir aussi van der Linden, Haned, and Kanoulas 2019

LIME Example





Advantages and limitations of LIME

- ☺ Explanations are relatively **human-friendly** (few features, use an interpretable model)
- ☺ Variants specific to **text** and **images**
- ☹ Sensitive to the definition of the **neighborhood**
- ☹ **Instable:** explanations vary significantly in small neighborhoods) Alvarez-Melis and Jaakkola 2018

Shapley values

- In **game theory:** how to assign **payouts** to cooperative **players** depending on their **contribution** to the global payout
  - **game** $\equiv$ making a prediction
  - **players** $\equiv$ features
  - **global payout** $\equiv$ (prediction - average prediction)
  - **payouts** $\equiv$ feature importance

Shapley values

- In **game theory:** how to assign **payouts** to cooperative **players** depending on their **contribution** to the global payout
  - **game** $\equiv$ making a prediction
  - **players** $\equiv$ features
  - **global payout** $\equiv$ (prediction - average prediction)
  - **payouts** $\equiv$ feature importance
- **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ of feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$: average **contribution** of a feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$ in different **coalitions** (= sets of features)

Shapley values

- In **game theory:** how to assign **payouts** to cooperative **players** depending on their **contribution** to the global payout
  - **game** $\equiv$ making a prediction
  - **players** $\equiv$ features
  - **global payout** $\equiv$ (prediction - average prediction)
  - **payouts** $\equiv$ feature importance
- **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ of feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$: average **contribution** of a feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$ in different **coalitions** (= sets of features)
- **Contribution** of coalition $\mathcal{S} \subseteq \{1, \dots, p\}$ to $\boldsymbol{f}(\vec{\boldsymbol{x}})$ =
  (average prediction when the features in $\mathcal{S}$ are set to their values in $\vec{\boldsymbol{x}}$ – average prediction)

Shapley values

- In **game theory:** how to assign **payouts** to cooperative **players** depending on their **contribution** to the global payout
  - **game** $\equiv$ making a prediction
  - **players** $\equiv$ features
  - **global payout** $\equiv$ (prediction - average prediction)
  - **payouts** $\equiv$ feature importance
- **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ of feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$: average **contribution** of a feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$ in different **coalitions** (= sets of features)
- **Contribution** of coalition $\mathcal{S} \subseteq \{1, \dots, p\}$ to $\boldsymbol{f}(\vec{\boldsymbol{x}})$ =
(average prediction when the features in $\mathcal{S}$ are set to their values in $\vec{\boldsymbol{x}}$ – average prediction)

$$\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S}) = \underbrace{\mathbb{E}[\boldsymbol{f}(X_1, \dots, X_p) | X_k = \boldsymbol{x_k} \text{ for } k \in \mathcal{S}]}_{\text{marginalize over features not in } \mathcal{S}} - \underbrace{\mathbb{E}[\boldsymbol{f}(X_1, \dots, X_p)]}_{\text{average prediction}}$$

Shapley 1952; Owen and Prieur 2017

Shapley values

- **Contribution** of coalition $\mathcal{S} \subseteq \{1, \ldots, p\}$ to $\boldsymbol{f}(\vec{\boldsymbol{x}})$ = average prediction when the features in $\mathcal{S}$ are set to their values in $\vec{\boldsymbol{x}}$ – average prediction

$$\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S}) = \underbrace{\mathbb{E}[\boldsymbol{f}(X_1, \ldots, X_p) | X_k = \boldsymbol{x_k} \text{ for } k \in \mathcal{S}]}_{\text{marginalize over features not in } \mathcal{S}} - \underbrace{\mathbb{E}[\boldsymbol{f}(X_1, \ldots, X_p)]}_{\text{average prediction}}$$

- **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ of feature j to the prediction $\boldsymbol{f}(\vec{\boldsymbol{x}})$:

$$\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \sum_{\mathcal{S} \subseteq \{1, \ldots, p\} \setminus \{j\}} \frac{|\mathcal{S}|!(p - |\mathcal{S}| - 1)!}{p!} \left(\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{j\}) - \psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S})\right)$$

Shapley 1952; Owen and Prieur 2017

Properties of Shapley values

- **Efficiency:** the sum of payouts is the global payout $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$

Properties of Shapley values

- **Efficiency:** the sum of payouts is the global payout $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$
- **Symmetry:** two features that contribute equally to all possible coalitions should have the same Shapley value
  if for all $\mathcal{S} \in \{1, \ldots, p\} \setminus \{j, k\}$, $\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{j\}) = \psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{k\})$, then
  $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \varphi(k, \boldsymbol{f}, \vec{\boldsymbol{x}})$

Properties of Shapley values

- **Efficiency:** the sum of payouts is the global payout $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$
- **Symmetry:** two features that contribute equally to all possible coalitions should have the same Shapley value
  if for all $\mathcal{S} \in \{1, \ldots, p\} \setminus \{j, k\}$, $\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{j\}) = \psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{k\})$, then
  $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \varphi(k, \boldsymbol{f}, \vec{\boldsymbol{x}})$
- **Dummy:** a feature that does not affect predictions has a Shapley value of 0.

Properties of Shapley values

- **Efficiency:** the sum of payouts is the global payout $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$
- **Symmetry:** two features that contribute equally to all possible coalitions should have the same Shapley value
  if for all $\mathcal{S} \in \{1, \ldots, p\} \setminus \{j, k\}$, $\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{j\}) = \psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{k\})$, then
  $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \varphi(k, \boldsymbol{f}, \vec{\boldsymbol{x}})$
- **Dummy:** a feature that does not affect predictions has a Shapley value of 0.
- **Additivity:** if the prediction can be decomposed in $\boldsymbol{f} = f_1 + f_2$, then for all j and $\vec{x}$,
  $\varphi(j, \boldsymbol{f}, \vec{x}) = \varphi(j, f_1, \vec{x}) + \varphi(j, f_2, \vec{x})$

Properties of Shapley values

- **Efficiency:** the sum of payouts is the global payout $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$
- **Symmetry:** two features that contribute equally to all possible coalitions should have the same Shapley value
  if for all $\mathcal{S} \in \{1, \ldots, p\} \setminus \{j, k\}$, $\psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{j\}) = \psi(\boldsymbol{f}, \vec{\boldsymbol{x}}, \mathcal{S} \cup \{k\})$, then
  $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \varphi(k, \boldsymbol{f}, \vec{\boldsymbol{x}})$
- **Dummy:** a feature that does not affect predictions has a Shapley value of 0.
- **Additivity:** if the prediction can be decomposed in $\boldsymbol{f} = f_1 + f_2$, then for all j and $\vec{x}$,
  $\varphi(j, \boldsymbol{f}, \vec{x}) = \varphi(j, f_1, \vec{x}) + \varphi(j, f_2, \vec{x})$

→ For random forests, Shapley values are averages of the Shapley values of the individual trees.

Shapley 1952; Owen and Prieur 2017

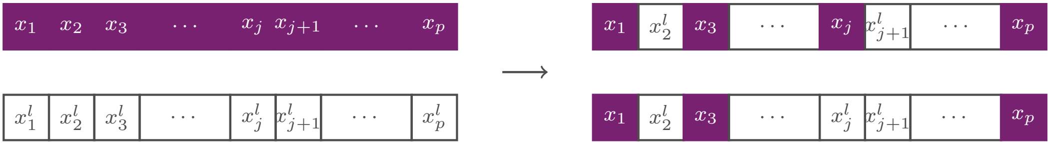
Computing Shapley values

$$\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \sum_{\mathcal{S} \subseteq \{1,\dots,p\} \setminus \{j\}} \frac{|\mathcal{S}|!(p - |\mathcal{S}| - 1)!}{p!} \left(\mathbb{E}[\boldsymbol{f}(X) | X_k = \boldsymbol{x_k}, k \in \mathcal{S} \cup \{j\}] - \mathbb{E}[\boldsymbol{f}(X) | X_k = \boldsymbol{x_k}, k \in \mathcal{S}]\right)$$

– Approximate with **Monte-Carlo sampling**

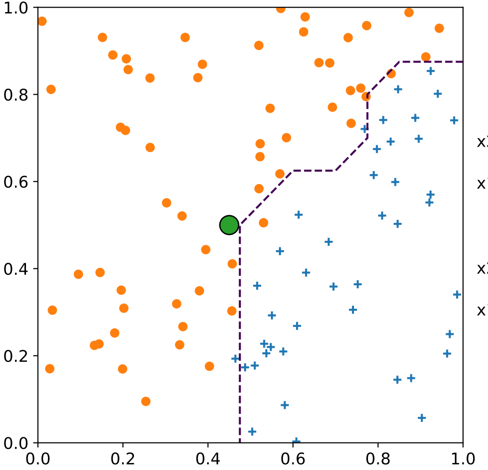
$$\hat{\varphi}(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{f}(\vec{x}^{i}_{+j}) - \boldsymbol{f}(\vec{x}^{i}_{-j})$$

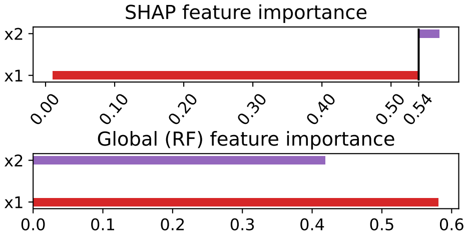
– $\vec{x}^{i}_{+j} = \vec{\boldsymbol{x}}$ but with p' features, **except** x_j, replaced with their values in another instance of $\mathcal{D}$
– $\vec{x}^{i}_{-j} = \vec{\boldsymbol{x}}$ but with p' features, **including** x_j, replaced with their values in another instance of $\mathcal{D}$



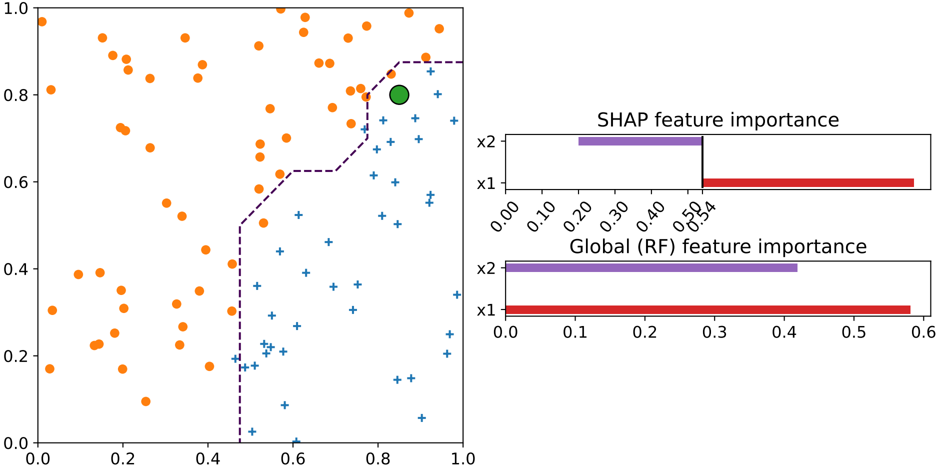
Štrumbelj and Kononenko 2014

Shapley values Example 1





Shapley values Example 2



Advantages and limitations of Shapley values

- ☺ **Good theoretical properties**
- ☺ Possibility of **contrastive explanations** – comparing to the average prediction over a certain subset rather than over all data points
- ☹ **Computationally intensive**
- ☹ **Interpretation** is less straightforward ("the contribution of x_j to the difference between the actual prediction and the average prediction")
- ☹ Need **access to $\mathcal{D}$** (unless you can draw realistic values for $\vec{x}^l, l = 1, \ldots, m$)

SHAP: SHapley Additive exPlanations

- **LIME:** look for a simple model g that approximates $\boldsymbol{f}$ in a neighborhood of $\vec{\boldsymbol{x}}$

$$\arg\min_{g \in \mathcal{G}} \frac{1}{|\mathcal{Z}|} \sum_{\vec{z}_i \in \mathcal{Z}} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\, \Omega(g)$$

SHAP: SHapley Additive exPlanations

- **LIME:** look for a simple model g that approximates $\boldsymbol{f}$ in a neighborhood of $\vec{\boldsymbol{x}}$

$$\arg\min_{g\in\mathcal{G}} \frac{1}{|\mathcal{Z}|}\sum_{\vec{z}_i\in\mathcal{Z}} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\,\Omega(g)$$

Set

- $\mathcal{Z} = \{$vectors of $\mathbb{R}^p$ obtained by setting some of the features of $\vec{\boldsymbol{x}}$ to $0\}$
- $w_i = \frac{(p-1)}{\binom{p}{||\vec{z}_i||_0}||\vec{z}_i||_0(p-||\vec{z}_i||_0)}$ $\quad ||\vec{z}||_0 = \mathcal{S}_{\vec{z}} =$ number of non-zero entries of $\vec{z}$
- $L(\boldsymbol{f}(\vec{z}), g(\vec{z})) = (\mathbb{E}[\boldsymbol{f}(X)|X_k = \boldsymbol{x_k}$ for $k \in \mathcal{S}_{\vec{z}}] - g(\vec{z}))^2$
- $\Omega(g) = 0$
- $g(\vec{z}) = \sum_{j\in\mathcal{S}_{\vec{z}}} \phi_j(\vec{x}) + \phi_0(\vec{x})$ (g is **additive**)

SHAP: SHapley Additive exPlanations

- **LIME:** look for a simple model g that approximates $\boldsymbol{f}$ in a neighborhood of $\vec{\boldsymbol{x}}$

$$\arg\min_{g\in\mathcal{G}} \frac{1}{|\mathcal{Z}|}\sum_{\vec{z}_i\in\mathcal{Z}} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\,\Omega(g)$$

Set

- $\mathcal{Z} = \{$vectors of $\mathbb{R}^p$ obtained by setting some of the features of $\vec{\boldsymbol{x}}$ to $0\}$
- $w_i = \frac{(p-1)}{\binom{p}{||\vec{z}_i||_0}||\vec{z}_i||_0(p-||\vec{z}_i||_0)}$ $\quad ||\vec{z}||_0 = \mathcal{S}_{\vec{z}} =$ number of non-zero entries of $\vec{z}$
- $L(\boldsymbol{f}(\vec{z}), g(\vec{z})) = (\mathbb{E}[\boldsymbol{f}(X)|X_k = \boldsymbol{x_k}$ for $k \in \mathcal{S}_{\vec{z}}] - g(\vec{z}))^2$
- $\Omega(g) = 0$
- $g(\vec{z}) = \sum_{j\in\mathcal{S}_{\vec{z}}} \phi_j(\vec{x}) + \phi_0(\vec{x})$ (g is **additive**)

Then $\phi_j(\vec{x})$ coincides with the **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$

SHAP: SHapley Additive exPlanations

- **LIME:** look for a simple model g that approximates $\boldsymbol{f}$ in a neighborhood of $\vec{\boldsymbol{x}}$

$$\arg\min_{g\in\mathcal{G}} \frac{1}{|\mathcal{Z}|}\sum_{\vec{z}_i\in\mathcal{Z}} w_i L(\boldsymbol{f}(\vec{z}_i), g(\vec{z}_i)) + \lambda\,\Omega(g)$$

Set

- $\mathcal{Z} = \{$vectors of $\mathbb{R}^p$ obtained by setting some of the features of $\vec{\boldsymbol{x}}$ to $0\}$
- $w_i = \frac{(p-1)}{\binom{p}{||\vec{z}_i||_0}||\vec{z}_i||_0(p-||\vec{z}_i||_0)}$ $\quad ||\vec{z}||_0 = \mathcal{S}_{\vec{z}} =$ number of non-zero entries of $\vec{z}$
- $L(\boldsymbol{f}(\vec{z}), g(\vec{z})) = (\mathbb{E}[\boldsymbol{f}(X)|X_k = \boldsymbol{x_k}$ for $k \in \mathcal{S}_{\vec{z}}] - g(\vec{z}))^2$
- $\Omega(g) = 0$
- $g(\vec{z}) = \sum_{j\in\mathcal{S}_{\vec{z}}} \phi_j(\vec{x}) + \phi_0(\vec{x})$ (g is **additive**)

Then $\phi_j(\vec{x})$ coincides with the **Shapley value** $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ **LIME+kernelSHAP**

Lundberg and Lee 2017

SHAP: SHapley Additive exPlanations

- **SHAP explanations:** surrogate models built additively from Shapley values

$$g(\vec{z}) = \sum_{j=1}^{p} \mathbb{1}_{\vec{z}_j \neq 0}\, \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) + \varphi_0$$

where $\vec{z}$ is $\vec{\boldsymbol{x}}$ with some features at 0.

SHAP: SHapley Additive exPlanations

- **SHAP explanations:** surrogate models built additively from Shapley values

$$g(\vec{z}) = \sum_{j=1}^{p} \mathbb{1}_{\vec{z}_j \neq 0}\, \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) + \varphi_0 \qquad \text{where } \vec{z} \text{ is } \vec{\boldsymbol{x}} \text{ with some features at } 0.$$

- Recall the efficiency property of Shapley values: $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$

  Hence if no feature is set to 0, g and $\boldsymbol{f}$ coincide, with $\varphi_0 = \mathbb{E}[\boldsymbol{f}(X)]$

SHAP: SHapley Additive exPlanations

- **SHAP explanations:** surrogate models built additively from Shapley values

$$g(\vec{z}) = \sum_{j=1}^{p} \mathbb{1}_{\vec{z}_j \neq 0}\, \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) + \varphi_0$$

where $\vec{z}$ is $\vec{\boldsymbol{x}}$ with some features at 0.

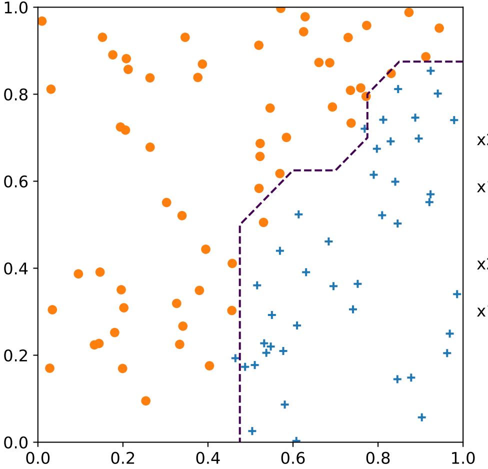
- Recall the efficiency property of Shapley values: $\sum_{j=1}^{p} \varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}}) = \boldsymbol{f}(\vec{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{f}(X)]$

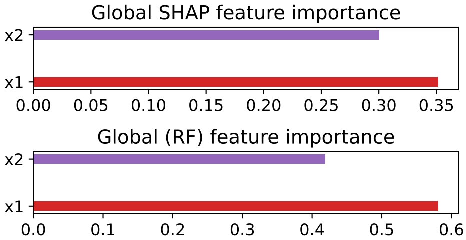
  Hence if no feature is set to 0, g and $\boldsymbol{f}$ coincide, with $\varphi_0 = \mathbb{E}[\boldsymbol{f}(X)]$

- **Interpretation:**
  - With no information the prediction is $\mathbb{E}[\boldsymbol{f}(X)]$
  - Each feature j adds $\mathbb{E}[\boldsymbol{f}(X)|X_j = \boldsymbol{x_j}]$
  - $\varphi(j, \boldsymbol{f}, \vec{\boldsymbol{x}})$ averages this contribution over all possible orderings of the features

Global SHAP explainer

$$I_j = \frac{1}{n} \sum_{i=1}^{n} \varphi(j, \boldsymbol{f}, \vec{x}_i)$$





Advantages and limitations of Shapley values

- ☺ **Good theoretical properties**
- ☹ **Computationally intensive**
- ☺ but not for tree-based models! (see TreeSHAP)
- ☹ Ignores **dependence** between features
- ☺ but not for tree-based models! (see TreeSHAP)
- ☹ Need **access to $\mathcal{D}$** (unless you can draw realistic values for $\vec{x}^l, l = 1, \ldots, m$)
- ☺ but not for tree-based models! (see TreeSHAP)

Conclusion

- LIME and SHAP provide **model-agnostic**, **local** explanations
- SHAP enjoys nice theoretical properties but is slower (except for tree-based models)
- SHAP is more stable than LIME but neither is very robust for non-linear model
  Alvarez-Melis and Jaakkola 2018; Lakkaraju, Arsov, and Bastani 2020

Conclusion

- LIME and SHAP provide **model-agnostic**, **local** explanations
- SHAP enjoys nice theoretical properties but is slower (except for tree-based models)
- SHAP is more stable than LIME but neither is very robust for non-linear model
  Alvarez-Melis and Jaakkola 2018; Lakkaraju, Arsov, and Bastani 2020
- **Minimal sufficient subsets**
  Chen et al. 2018; Camburu et al. 2021
- How do you **evaluate** interpretability?
  Robnik-Šikonja and Bohanec 2018; Molnar, Casalicchio, and Bischl 2019
- Statistical significance? Causality?

Acknowledgments

- Slides based on the cited papers as well as the online book **Interpretable machine learning. A Guide for Making Black Box Models Explainable**, Molnar, Christoph (2019) `https://christophm.github.io/interpretable-ml-book/`
- Discussions with **Ndèye Maguette Mbaye** and **Charles Vesteghem**
- Python librairies `lime` and `shap` (and, obviously, `numpy`, `scikit-learn`, and `matplotlib`)

References I

Alvarez-Melis, David and Tommi S Jaakkola (2018). "On the robustness of interpretability methods". In: *arXiv preprint*. URL: `https://arxiv.org/abs/1806.08049`.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Camburu, Oana-Maria et al. (2021). "The struggles of feature-based explanations: Shapley values vs. minimal sufficient subsets". In: *Explainable Agency in Artificial Intelligence Workshop at AAAI 2021*. URL: `http://arxiv.org/abs/2009.11023v2`.

Chen, Jianbo et al. (2018). "Learning to explain: An information-theoretic perspective on model interpretation". In: *International Conference on Machine Learning*. PMLR, pp. 883–892.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint*. URL: `https://arxiv.org/abs/1702.08608`.

Lakkaraju, Himabindu, Nino Arsov, and Osbert Bastani (2020). "Robust and stable black box explanations". In: *International Conference on Machine Learning*. PMLR, pp. 5628–5638.

Lipton, Zachary C (2016). "The mythos of model interpretability". In: *arXiv preprint*. URL: `https://arxiv.org/abs/1606.03490`.

Louppe, Gilles (2014). "Random Forests: From Theory to Practice". Doctoral dissertation. University of Liège. URL: `https://orbi.uliege.be/handle/2268/170309`.

References II

Lundberg, Scott M and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Molnar, Christoph, Giuseppe Casalicchio, and Bernd Bischl (2019). "Quantifying model complexity via functional decomposition for better post-hoc interpretability". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. URL: https://arxiv.org/abs/1904.03867v1.

Owen, Art B and Clémentine Prieur (2017). "On Shapley value for measuring importance of dependent inputs". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1, pp. 986–1002.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. URL: https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf.

Robnik-Šikonja, Marko and Marko Bohanec (2018). "Perturbation-based explanations of prediction models". In: *Human and machine learning*. Springer, pp. 159–175.

Shapley, Lloyd S (1952). "17. A value for n-person games". In: *Contributions to the Theory of Games, Volume II*. Vol. 28. Annals of Mathematics Studies. Princeton University Press, pp. 307–313.

References III

Štrumbelj, Erik and Igor Kononenko (2014). "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and information systems* 41.3, pp. 647–665.

van der Linden, Ilse, Hinda Haned, and Evangelos Kanoulas (2019). "Global aggregations of local explanations for black box models". In: *Proceedings of the Workshop on Fairness, Accountability, Confidentiality, Transparency, and Safety in Information Retrieval at SIGIR*. URL: https://arxiv.org/abs/1907.03039.