Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

BigInsight

How to open the black box

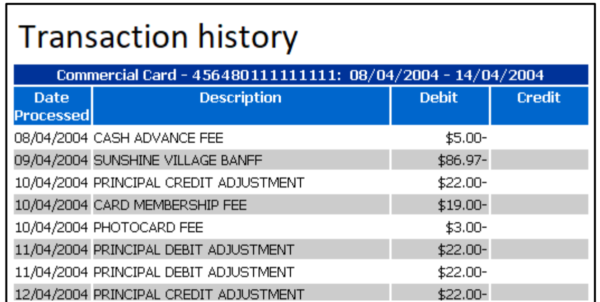
Individual prediction explanation

Martin Jullum

Joint work with Kjersti Aas, Anders Løland and Annabelle Redelmeier

NTNU Statistics Seminar
Trondheim, March 9th 2020

Example: Bank creates mortgage robot



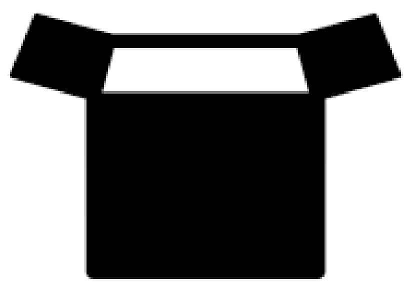
Transaction history

Commercial Card - 456480111111111: 08/04/2004 - 14/04/2004			
Date Processed	Description	Debit	Credit
08/04/2004	CASH ADVANCE FEE	$5.00-	
09/04/2004	SUNSHINE VILLAGE BANFF	$86.97-	
10/04/2004	PRINCIPAL CREDIT ADJUSTMENT	$22.00-	
10/04/2004	CARD MEMBERSHIP FEE	$19.00-	
10/04/2004	PHOTOCARD FEE	$3.00-	
11/04/2004	PRINCIPAL DEBIT ADJUSTMENT	$22.00-	
11/04/2004	PRINCIPAL DEBIT ADJUSTMENT	$22.00-	
12/04/2004	PRINCIPAL CREDIT ADJUSTMENT	$22.00-	

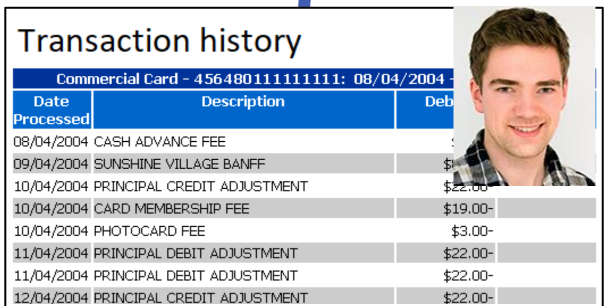
Age/gender/prev. loans…

&

Defaulted loan?



Example: Bank creates mortgage robot



Age/gender/prev. loans…

P(Mortgage default)

Mortgage granted ?

$$\boldsymbol{x} \longrightarrow f(\boldsymbol{x}) \longrightarrow p = 0.7$$

<u>Why</u> was  rejected a loan?

Why is this important?

- Customers may have a “right to an explanation”
- Builds trust to the “robot”

<u>Individual</u> prediction explanation

- **<u>NOT</u>** a general explanation of the black-box model
- $\boldsymbol{x} = \boldsymbol{x}^*$: Transaction history/covariates for 

<u>Explanation for $f(\boldsymbol{x}^*)$ = 70%</u>

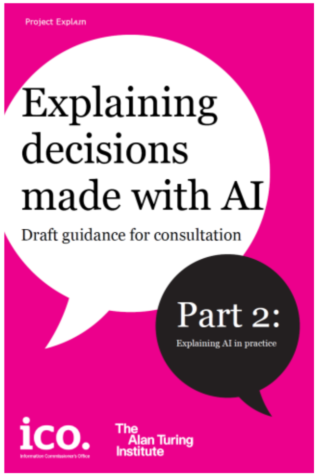
A (mathematical) description/visualization/ characterization of how **each of the covariates** contributed/affected the specific prediction $f(\boldsymbol{x}^*) = 70\%$

Explaining a simple linear model

- Model $y = f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
- How would you explain predictions from this model?
- Explanation depends on:
  - β_1 and β_2
  - β_0
  - x_1^* and x_2^*
  - $E[x_1]$ and $\mathrm{E}[x_2]$
  - $sd(x_1)$ and $sd(x_2)$
  - $\underline{corr(x_1, x_2)}$

- Claim: A simple linear model is only easily interpretable if x_1 and x_2 are independent and standardized!

Prediction explanation frameworks

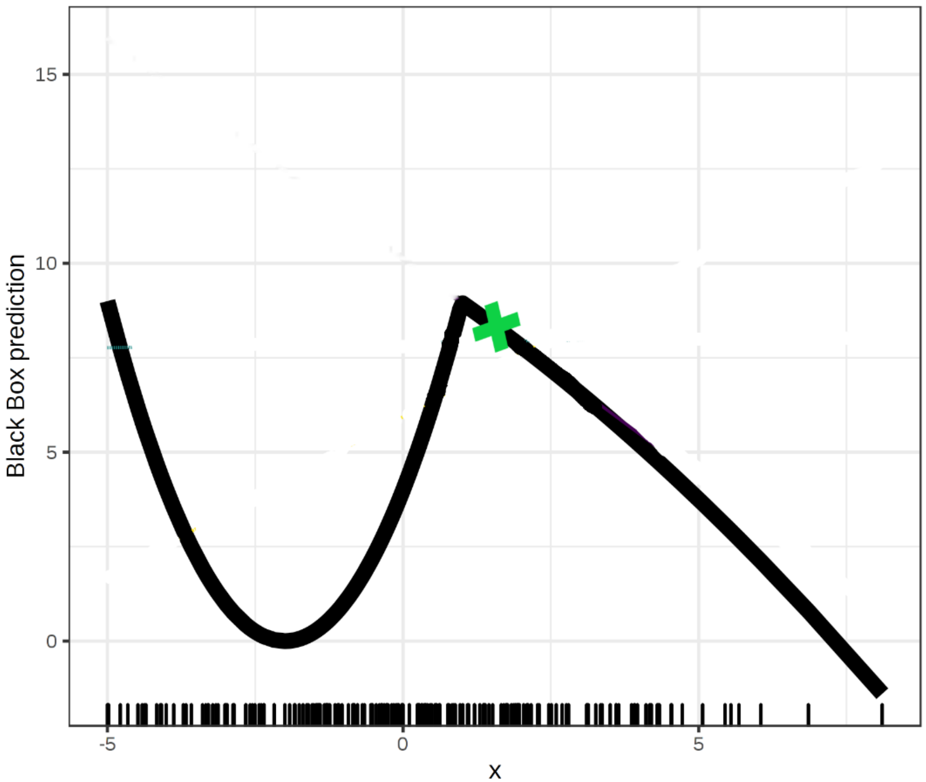


- Model-specific methods:
  - Deep Lift/Relevance propagation: For neural networks
  - TreeSHAP: For tree based methods
- Model-agnostic methods:
  - LIME: Local linear regression
  - Counterfactual explanations: Which covariates should be altered to obtain a different decision?
  - Shapley values: Based on concepts from game theory

LIME

(**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanation)

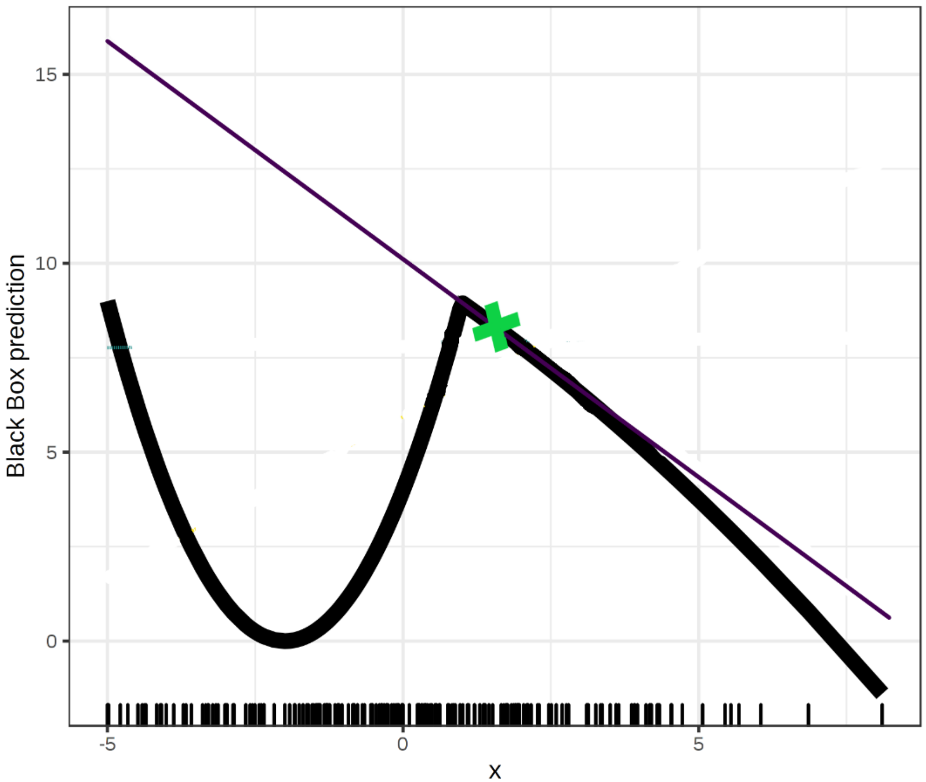
- Fits a (local) weighted linear regression model to $f(\boldsymbol{x})$ based on standardized covariates and weight determined by distance to $\boldsymbol{x}^*$
- Importance score for each covariate: Coefficient from local model



LIME

(**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanation)

- Fits a (local) weighted linear regression model to $f(\boldsymbol{x})$ based on standardized covariates and weight determined by distance to $\boldsymbol{x}^*$
- Importance score for each covariate: Coefficient from local model

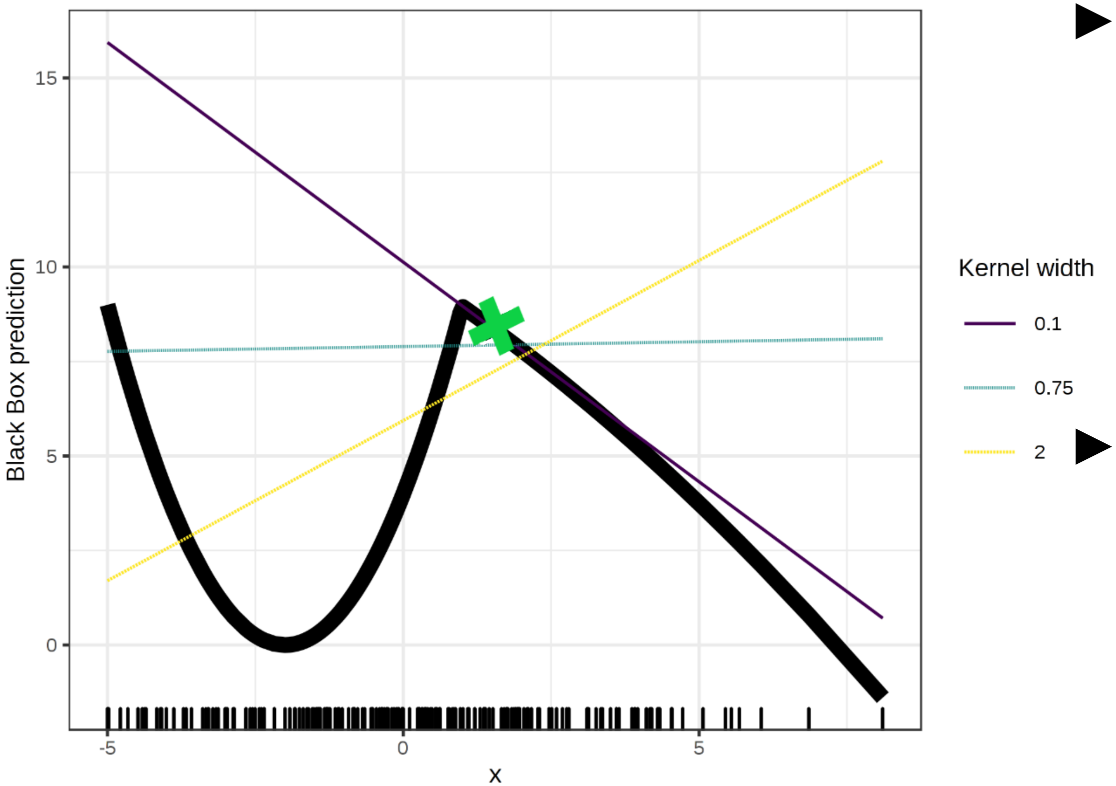


- Challenges
  - Defining the distance and weight functions

LIME

(**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanation)

- Fits a (local) weighted linear regression model to $f(\boldsymbol{x})$ based on standardized covariates and weight determined by distance to $\boldsymbol{x}^*$
- Importance score for each covariate: Coefficient from local model



- Challenges
  - Defining the distance and weight functions
  - Direct use of local model coefficients
- Advantages
  - Simple idea
  - Easy to use

Counterfactual explanations

- What is the smallest covariate change necessary to change the prediction “significantly”?
- Optimization problem:

$$(\mathrm{Ex}) \qquad \arg\min_{x'} d(\boldsymbol{x}^*, \boldsymbol{x}'), \quad \text{subject to } |f(\boldsymbol{x}') - [f(\boldsymbol{x}^*) + \lambda]| \leq \varepsilon$$

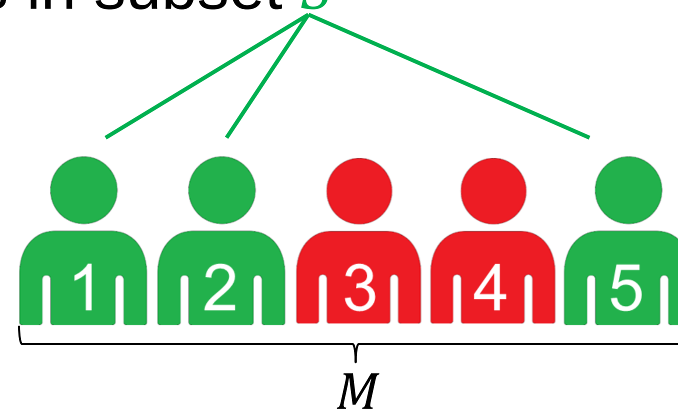
- Explanation: Minimizers of (Ex)
- Challenges:
  - Choosing d, λ and ε
  - May lead to many sub-explanations
  - Need to interpret the explanations yourself
- Advantages
  - Cannot be wrong
  - Guides user on how to change prediction

Shapley values

- Concept from (cooperative) game theory in the 1950s
- Used to distribute the total payoff to the players
- Explicit formula for the “fair” payment to every player j:

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} w(S) \left(v(S \cup \{j\}) - v(S) \right), \qquad w(S) \text{ is a weight function}$$

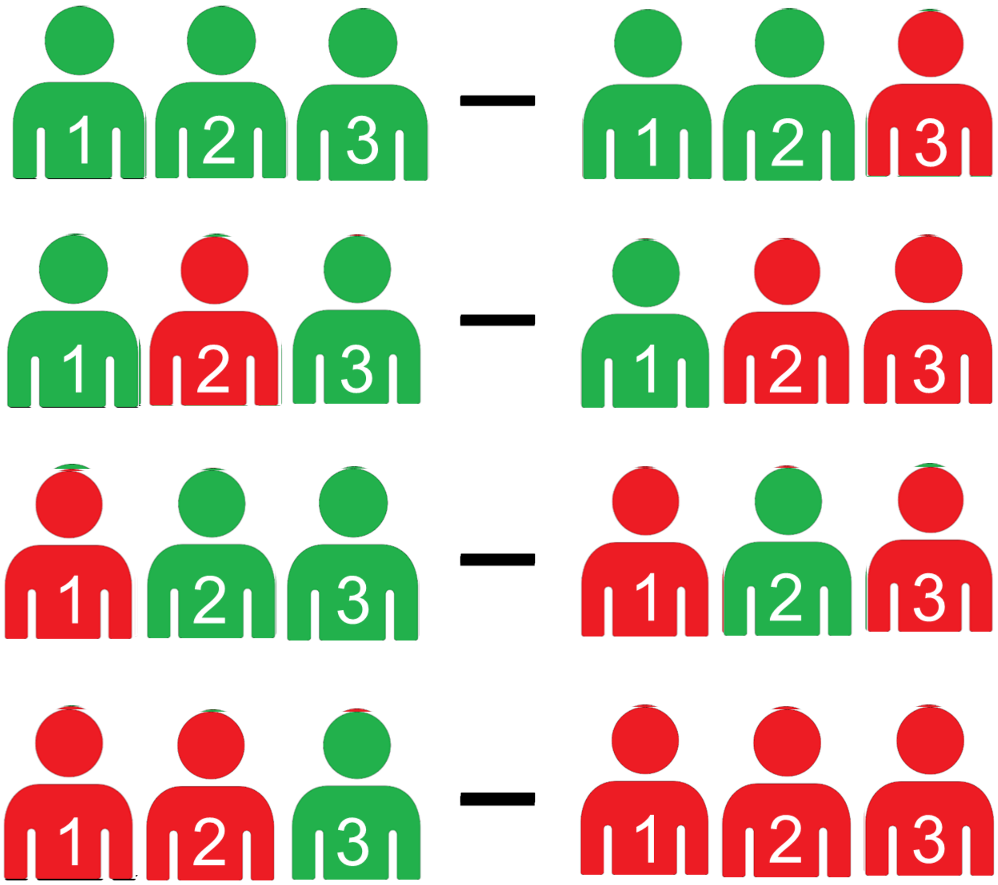
$v(S)$ is the payoff with only players in subset S



- Several mathematical optimality properties

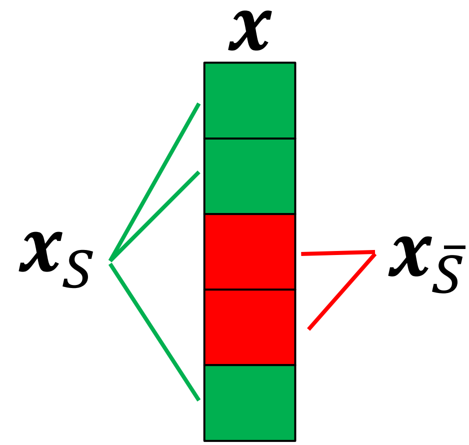
Intuition behind the Shapley formula

Game with 3 players



Shapley values for prediction explanation

- Players = covariates ($x_1, \dots, x_M$)
- Payoff = prediction ($f(\boldsymbol{x}^*)$)
- Contribution function: $v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$
- Properties



$f(\boldsymbol{x}^*) = \sum_{j=0}^{M} \phi_j$

$\phi_0 = E[f(\boldsymbol{x})]$

$E[f(\boldsymbol{x})] = E[f(\boldsymbol{x})|x_j]$ implies $\phi_j = 0$

x_i, x_j same contribution implies $\phi_i = \phi_j$

- Rough interpretation of ϕ_j: **The prediction change when you don't know the value of** x_j -- averaged over all covariates

Shapley values for prediction explanation

- 2 main challenges

1. The computational complexity in the Shapley formula

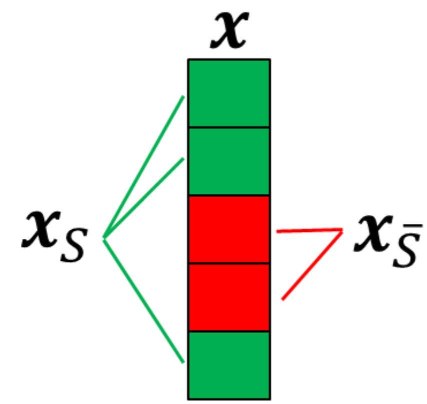
$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} w(S) \left(v(S \cup \{j\}) - v(S) \right)$$

   - Partly solved by cleverly reducing the sum by subset sampling (KernelSHAP; Lundberg & Lee, 2017)

Shapley values for prediction explanation

- ► 2 main challenges

Recall



2. Estimating the contribution function

$$v(S) = E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S) p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}}$$

- Previous methods
  - Approximates $v(S) \approx \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}) \mathrm{d}\boldsymbol{x}_{\bar{S}}$,
  - Estimates $p(\boldsymbol{x}_{\bar{S}})$ using the empirical distribution of the training data
  - Monte Carlo integration to solve the integral

This assumes covariates are independent!

Consequences of the independence assumption

- Requires evaluating $f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S)$ at potentially **unlikely or illegal** combinations of $\boldsymbol{x}_{\bar{S}}$ and $\boldsymbol{x}_S$

- Example 1
  - Number of transactions to Switzerland: 0
  - Average transaction amount to Switzerland: 1000 NOK

- Example 2
  - Age: 17
  - Marital status: Widow
  - Profession: Professor

Shapley values for prediction explanation

- Explicit formula for a linear model $f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{M} \beta_j x_j$ with **independent** covariates:

$$\phi_0 = \beta_0 + \sum_{j=1}^{M} \beta_j E[x_j], \qquad \phi_j = \beta_j\left(x_j^* - E\left[x_j\right]\right),\ j = 1, \ldots, M$$

- Advantages
  - Proper mathematical foundation
  - Desirable set of properties
- Challanges
  - Computationally heavy
  - Requires good estimates of a difficult estimation problem: $E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$

- From our perspective the method with greatest potential – what we have work with the last two years

Our idea

Estimate $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ properly

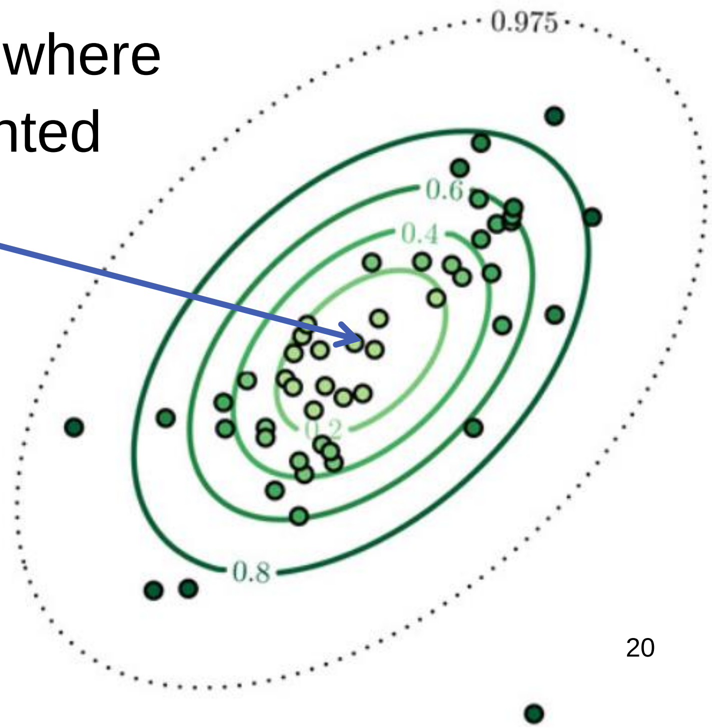
+

Monte Carlo integration to approximate

$$v(S) = E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S) p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}}$$

Continuous covariates

- How to estimate $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ when $\boldsymbol{x}$ is continuous?
- 3 approaches
  - Assume $p(\boldsymbol{x})$ Gaussian => analytical $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$
  - Assume Gaussian copula => transformation + analytical expression
  - An empirical (conditional) approach where training observations at $\boldsymbol{x}_{\bar{S}}^{\boldsymbol{i}}$ are weighted based on proximity of $\boldsymbol{x}_S^{\boldsymbol{i}}$ to $\boldsymbol{x}_S^*$



Empirical conditional approach

1. Compute the scaled Mahalanobis distance between $\boldsymbol{x}_S^*$ and the columns S of the training data $\boldsymbol{x}^1, \dots \boldsymbol{x}^n$

$$D_S(\boldsymbol{x}^*, \boldsymbol{x}^i) = \sqrt{\frac{(\boldsymbol{x}_S^* - \boldsymbol{x}_S^i)^T \Sigma_S^{-1} (\boldsymbol{x}_S^* - \boldsymbol{x}_S^i)}{|S|}}$$

2. Use Gaussian kernel to get weight of each training observation:

$$w_S(\boldsymbol{x}^*, \boldsymbol{x}^i) = \exp\left(-\frac{D_S(\boldsymbol{x}^*, \boldsymbol{x}^i)^2}{2\sigma^2}\right)$$

3. Approximate $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ by the probability mass function where $p\left(\boldsymbol{x}_{\bar{S}} = x^i | \boldsymbol{x}_S = \boldsymbol{x}_S^*\right) = \frac{w_S(\boldsymbol{x}^*, \boldsymbol{x}^i)}{\sum_{k=1}^n w_S(\boldsymbol{x}^*, \boldsymbol{x}^k)}$

Empirical conditional approach II

- This gives an estimator of $E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$:

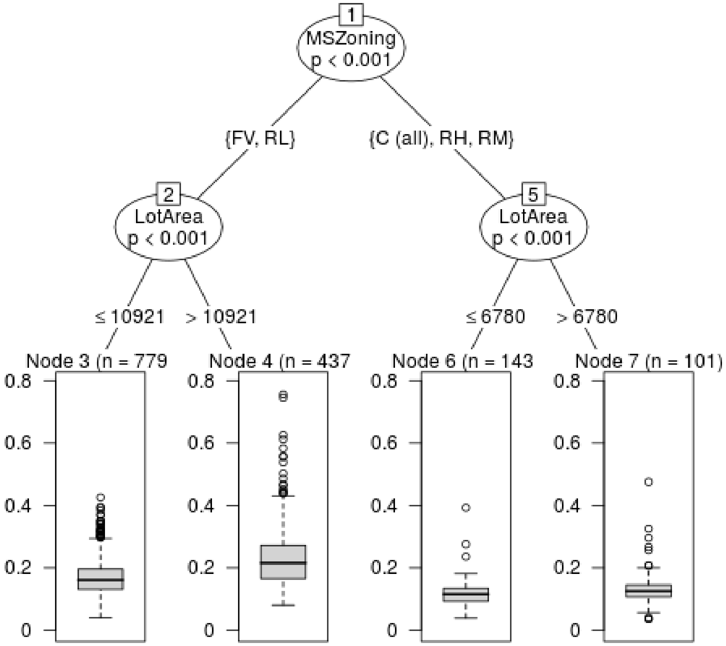
$$\hat{v}(S) = \frac{\sum_{k=1}^{n} w_S\left(\boldsymbol{x}^*, \boldsymbol{x}^k\right) f(\boldsymbol{x}_{\bar{S}}^k, \boldsymbol{x}_S^*)}{\sum_{k=1}^{n} w_S(\boldsymbol{x}^*, \boldsymbol{x}^k)}$$

- It turns out that we re-invented the Nadaraya-Watson estimator (locally constant kernel estimator) aiming at estimating $E[U|V{=}\mathrm{v}]$ for responses $u_i = f\left(\boldsymbol{x}_{\bar{S}}^i, \boldsymbol{x}_S^*\right)$, and covariates $v_i = \boldsymbol{x}_S^i, i = 1, \ldots, n$

- May then use a corrected AIC-criterion by Hurvich and Tsai (JRSS-B, 1998) to select the bandwidth parameter σ.

Categorical/mixed covariates

- How to estimate $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ when $\boldsymbol{x}$ is categorical, or mixed continuous/categorical
  - Fit a **multivariate decision tree** to $\boldsymbol{U} = \boldsymbol{x}_{\bar{S}}$ based on $\boldsymbol{V} = \boldsymbol{x}_S$ using the training data
  - Approximate $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ by the empirical distribution of the training observations ($\boldsymbol{x}_{\bar{S}}$) within the terminal node of $\boldsymbol{x}_S = \boldsymbol{x}_S^*$



Multivariate decision tree

- Classical decision tree algorithms like CART work only for univariate responses
  - Multivariate generalizations exits
  - CARTs are known to be biased towards splitting on categorical covariates with many levels
- Instead, we rely on Recursive partitioning/**conditional inference trees** (Hothorn et al., 2006)
  - Decide which covariate to split on first
  - Then decides on the splitting point for that covariate

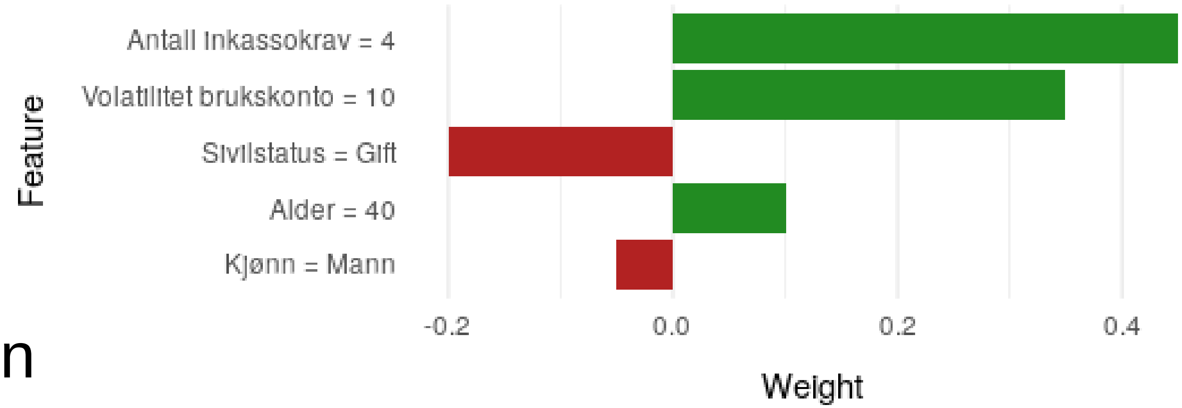
Conditional inference tree algorithm

- Multivariate response $\boldsymbol{U}$, covariates $V_1, \dots, V_p$
- Step 1: Decide whether or not to split by hypothesis testing:

$$H_0\!: p\left(\boldsymbol{U} \middle| V_j\right) = p(\boldsymbol{U})\ \forall\ j \quad \text{vs } H_A\!: p\left(\boldsymbol{U} \middle| V_j\right) \neq p(U) \text{ for some } j$$

  - Hypothesis test performed by permutation test using a summary statistic for the dependence between $\boldsymbol{U}$ and V_j
  - Stop tree building if not rejecting H_0 at a level α
  - If rejecting H_0, pick the covariate with the smallest *p*-value.
- Step 2: Splitting criteria
  - Maximize a two-sample discrepancy statistic
- Implemented in the R-packages *party* and *partykit*

Conclusion



- Individual prediction explanation, i.e. explaining $f(\boldsymbol{x}^*)$ for specific covariate $\boldsymbol{x}^*$
- Not straightforward to explain even a simple linear model
- Mainly three model-agnostic methods in the literature:
  - LIME, counterfactual explanations, Shapley values
- No grand truth when explaining predictions!
- Ignoring dependence between covariates can give completely wrong explanations

Want to know more?

Read our paper on arXiv
arxiv.org/abs/1903.10464

Check out our R-package
shapr on GitHub (soon CRAN) + JOSS
github.com/NorskRegnesentral/shapr