# Explaining individual predictions when features are dependent:

More accurate approximations to Shapley values

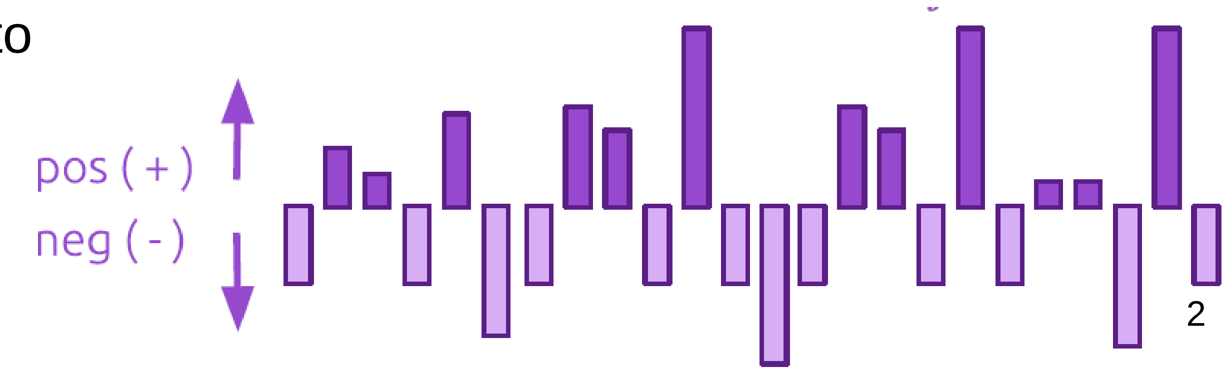
Kjersti Aas,
**Martin Jullum (jullum@nr.no),**
Anders Løland

Paper presentation, Journal track, IJCAI 2021

# Prediction explanation – by example

- Car insurance
  - Response $y$: The insured crashes
  - Features $\boldsymbol{x} = (x_1, \dots, x_M)$: Data about the insured, his/her car and crashing history
  - Predictive model $f$: Model trained to predict probability of crash: $f(\boldsymbol{x}) \approx \Pr(y = yes|\boldsymbol{x})$

- Prediction explanation
  - Why did a guy with features $\boldsymbol{x}^*$ get a predicted probability of crashing equal to $f(\boldsymbol{x}^*)$= 0.3?



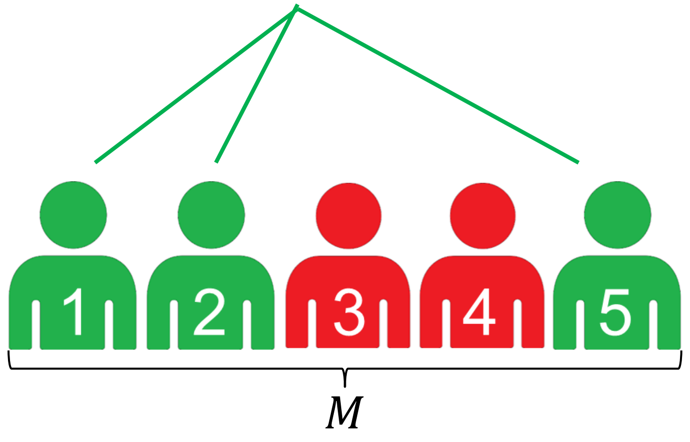
# Shapley values

- 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 $j$:

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(|M| - |S| - 1)}{|M|!} \big(v(S \cup \{j\}) - v(S)\big)$$

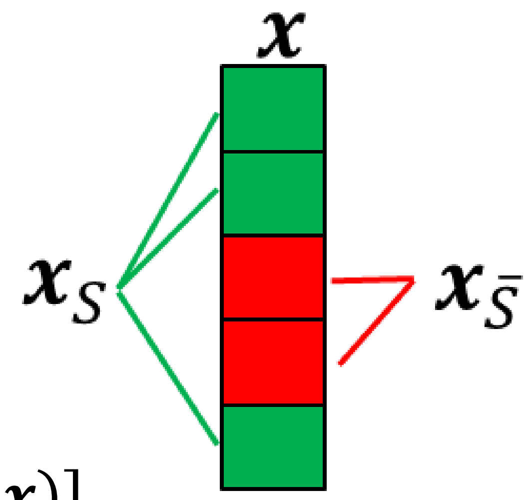
$v(S)$ is the payoff with only players in subset $S$

- Several mathematical optimality properties



# Shapley values for prediction explanation

- Approach popularised by Lundberg & Lee (2017)
  - Players = features ($x_1, \dots, x_M$)
  - Payoff = prediction ($f(\boldsymbol{x}^*)$)
  - Contribution function: $v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$
  - Properties




$$\sum_{j=1}^{M} \phi_j = f(\boldsymbol{x}^*) - \phi_0 \qquad\qquad \phi_0 = E[f(\boldsymbol{x})]$$

$f(\boldsymbol{x}) \perp\!\!\!\perp x_j$
implies $\phi_j = 0$

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

- Rough interpretation of $\phi_j$: **The prediction change when you don't know the value of** $x_j$ – averaged over all features

# Shapley values for prediction explanation

- Two main challenges

1. The computational complexity in the Shapley formula

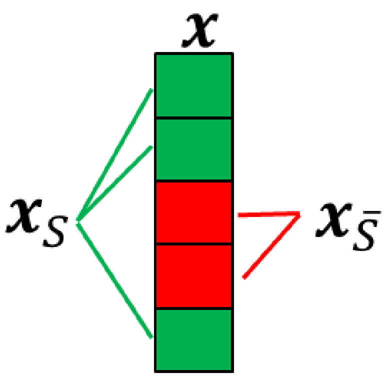
$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(|M| - |S| - 1)}{|M|!} \big(v(S \cup \{j\}) - v(S)\big)$$

   - Approximate solutions may be obtained by cleverly reducing the sum by subset sampling (KernelSHAP; Lundberg & Lee, 2017)

# Shapley values for prediction explanation

- Two main challenges

Recall




2. Estimating the contribution function

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

- Lundberg & Lee (2017)
  - 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 the features 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: 100 €

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

# The idea of the present paper

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

+

Monte Carlo integration to approximate

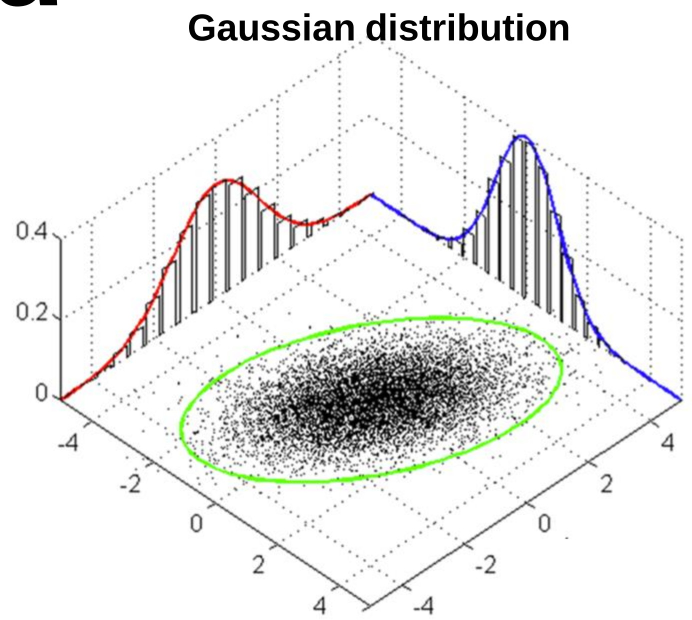
$$\boxed{v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S) p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}}}$$

by sampling from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$

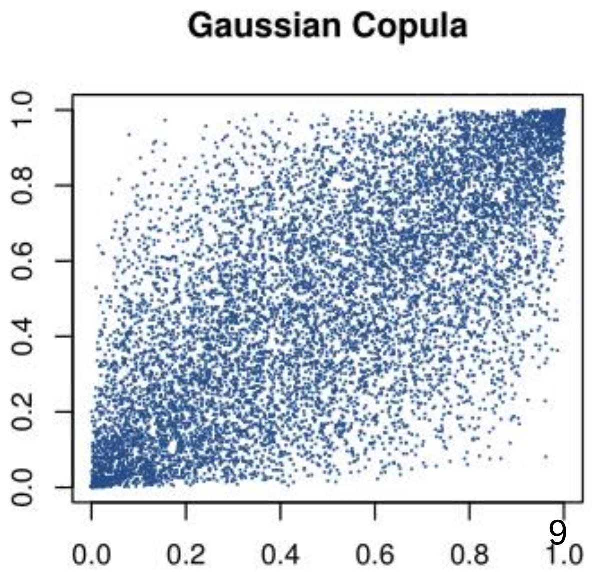
*Following the preprint of the present paper, other papers have used similar approaches

# 3 approaches to estimate and sample from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$

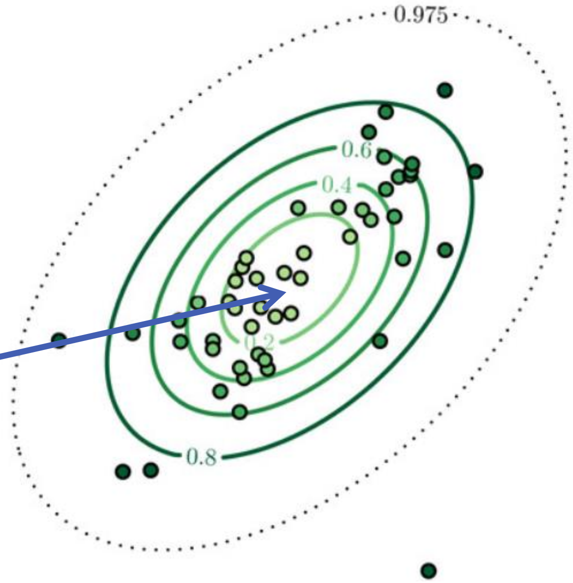
1. Assume $p(\boldsymbol{x})$ is Gaussian $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
    1. Estimate $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ using the training data
    2. Obtain analytical expression for $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ to sample from




2. Assume $p(\boldsymbol{x})$ is a Gaussian copula
    1. Transform all features in the training data to $N(0,1)$: $(v_1, \dots, v_M)$
    2. Estimate the correlation $\boldsymbol{\Sigma}^*$ in $(v_1, \dots, v_M)$
    3. Obtain analytical expression for $p(\boldsymbol{v}_{\bar{S}}|\boldsymbol{v}_S = \boldsymbol{v}_S^*)$ to sample from
    4. Transform the samples back to original scale



# 3 approaches to estimate and sample from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$




3. Use an empirical (conditional) distribution which weights the training observations ($\boldsymbol{x}_{\bar{S}}^{i}$) by their proximity to $\boldsymbol{x}_S^*$:

   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 weights for each training observation:

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

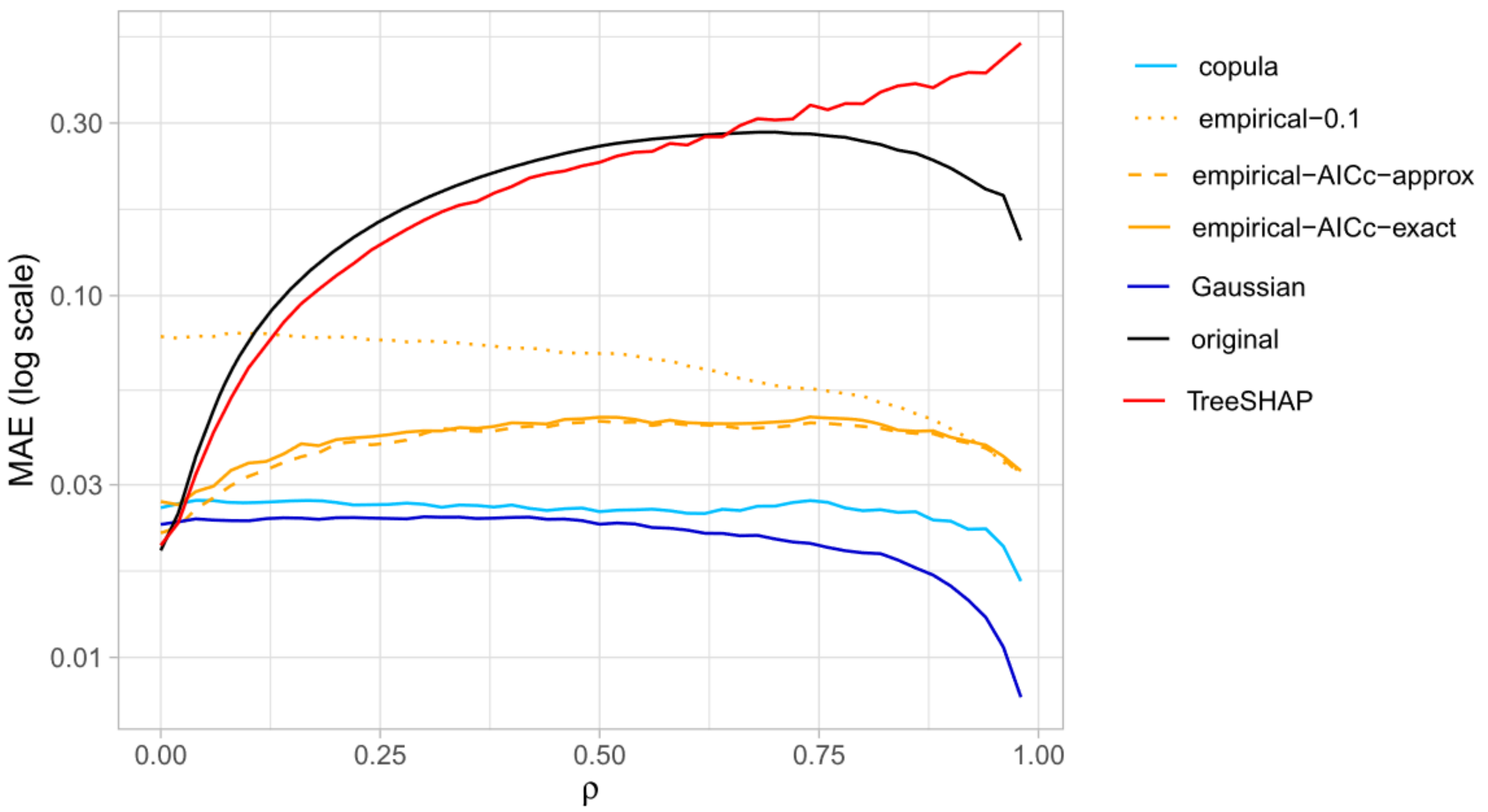
   3. Use the training observations $\boldsymbol{x}_{\bar{S}}^{i}$ *weighted* by $w_S(\boldsymbol{x}^*, \boldsymbol{x}^i)$ as a sample from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$

* $w_S(\boldsymbol{x}^*, \boldsymbol{x}^i)$=1/n corresponds to the independence method of Lundberg & Lee (2017)

# Simulation experiments

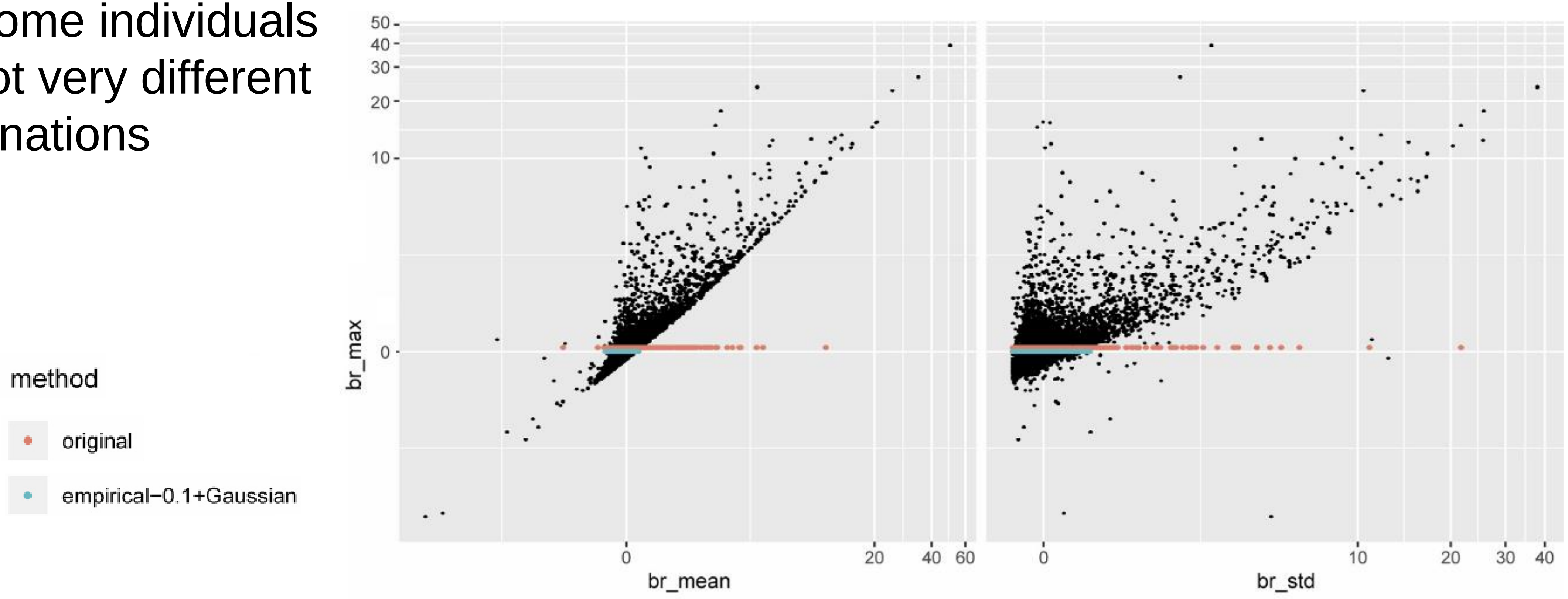
- Generally outperform original (independence) and TreeSHAP approaches
- Often the empirical approach is best for small $S$, and Gaussian/copula better for largest $S$

Sampling model: Piecewise constant, feature distribution: Gaussian, dimension: 3



# Real data example from finance

- 28 features extracted from financial time series used to predict mortgage default
- Used a combination of our empirical and Gaussian method + original (independence) approach to explain predictions
- For some individuals we got very different explanations



# Conclusion

- We explain individual predictions using the Shapley value framework
- We improve upon the original KernelSHAP approach (assuming feature independence) of Lundberg & Lee (2017) by accounting for the dependence
  - 3 methods: Gaussian, Gaussian copula and empirical (conditional) approach
- We outperform the independence approach and TreeSHAP in simulations
- Our method is implemented in the R-package ***shapr***, available on CRAN and GitHub

**References**

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4768-4777).

**Our paper**

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, *298*, 103502.