Prediction explanation with Shapley values

Martin Jullum (jullum@nr.no)

Imperial College London, 03.02.2022

Prediction explanation

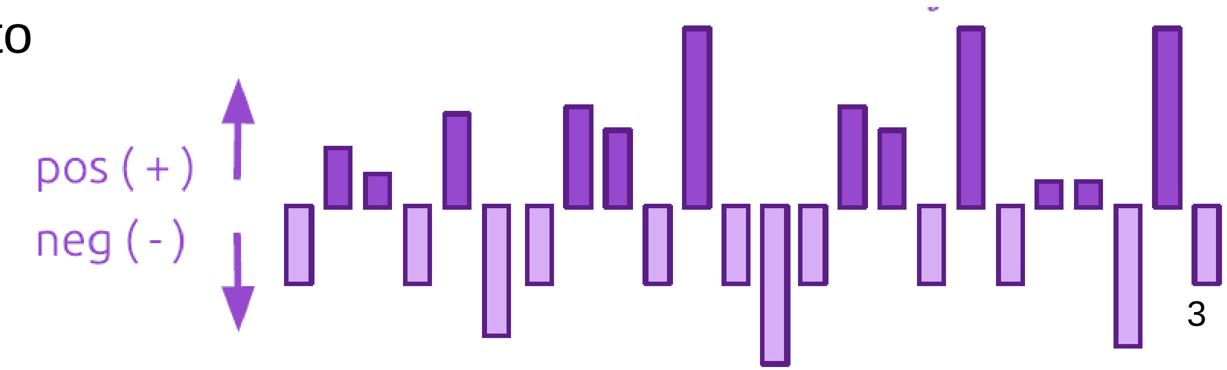
- Assume a model $f(\boldsymbol{x}) \in \mathbb{R}$ that predicts some unknown outcome based on a set of features $\boldsymbol{x} = (x_1, \ldots, x_M)$
- We apply the predictive model for a specific input $\boldsymbol{x} = \boldsymbol{x}^*$, reaching a certain prediction $f(\boldsymbol{x}^*)$
- Individual prediction explanation
  - Want to understand how the different features, or types of features affect this specific prediction value $f(\boldsymbol{x}^*)$
  - I.e. explain the predicted outcome in terms of the input $\boldsymbol{x} = \boldsymbol{x}^*$ (local explanation)

- Frameworks…
  - LIME
  - Anchors
  - Counterfactual explanations
  - Explanation Vectors
  - PredDiff
  - **Shapley values**

Prediction explanation – by example

- Car insurance
  - Response y: Insured crashed or not
  - Features $\boldsymbol{x} = (x_1, \ldots, x_M)$: 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?



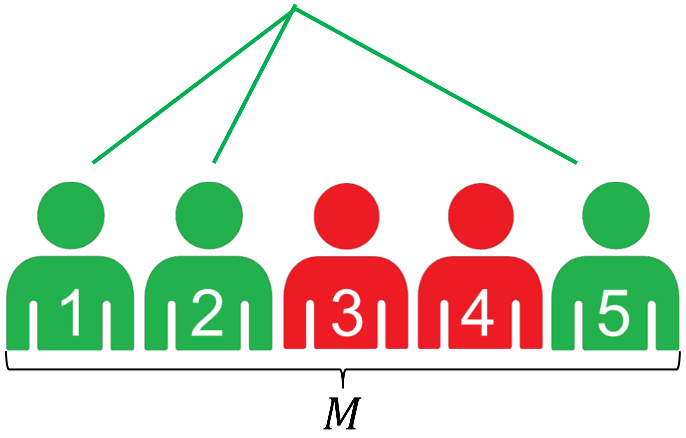
Shapley values

- 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:

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

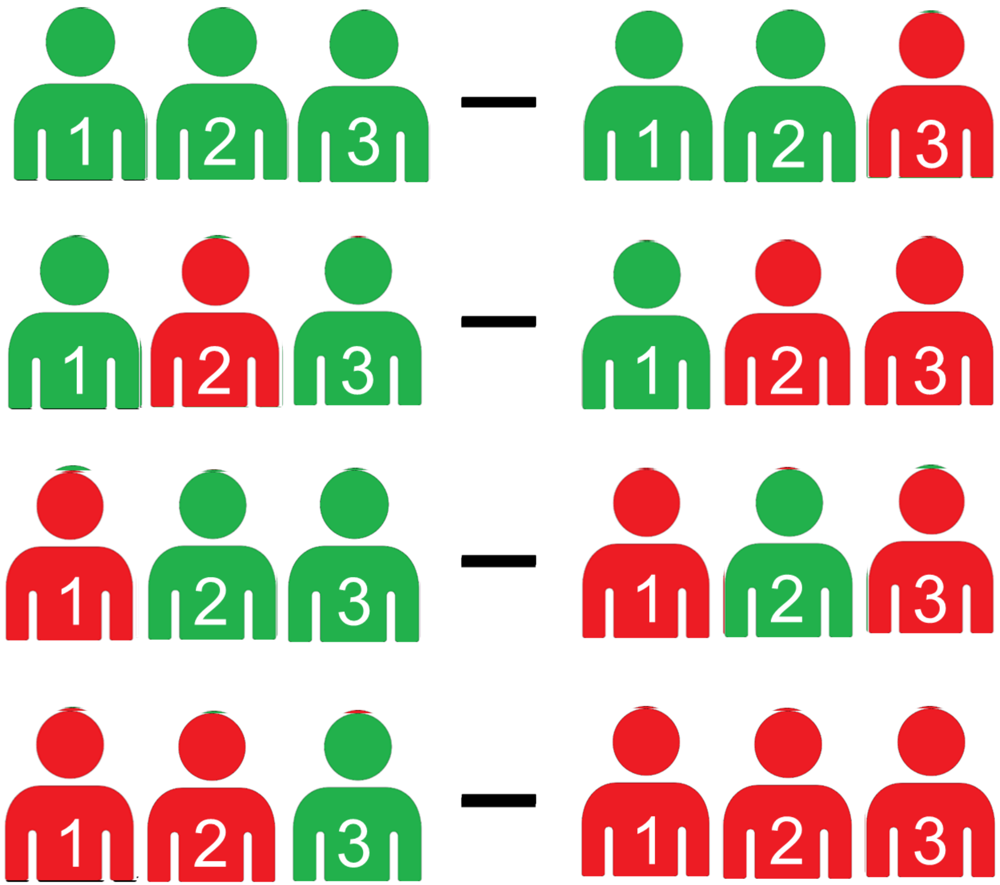
$v(S)$ is the payoff with only players in subset S

- Several mathematical optimality properties



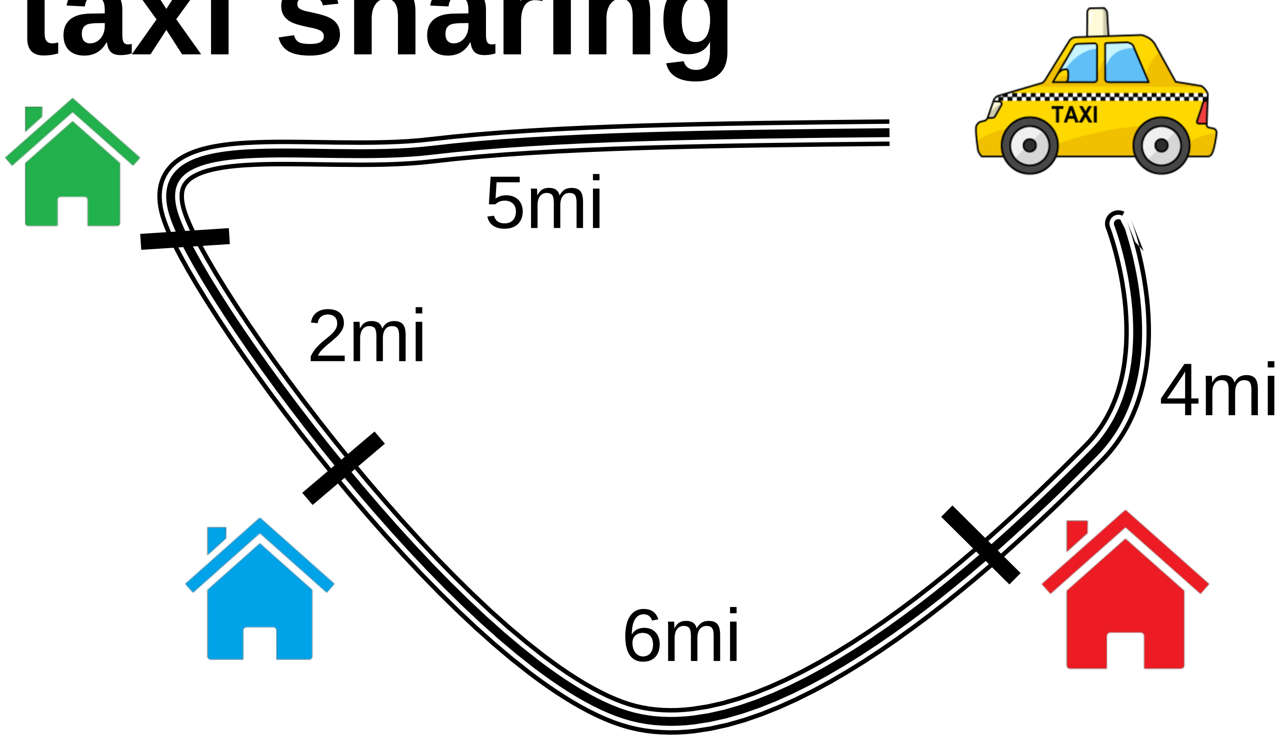
Intuition behind the Shapley formula

Game with 3 players



Shapley values for taxi sharing

Costs: \$3/mi



$v(\{R, B, G\}) = (4 + 6 + 2)mi * \$3 = \$36$

$v(\{\ \}) = \$0$

$v(\{R\}) = 4mi * \$3 = \$12$

$v(\{B\}) = (5 + 2)mi * \$3 = \$21$

$v(\{G\}) = 5mi * \$3 = \$15$

$v(\{R, B\}) = (4 + 6)mi * \$3 = \$30$

$v(\{R, G\}) = (4 + 6 + 2)mi * \$3 = \$36$

$v(\{B, G\}) = (5 + 2)mi * \$3 = \$21$

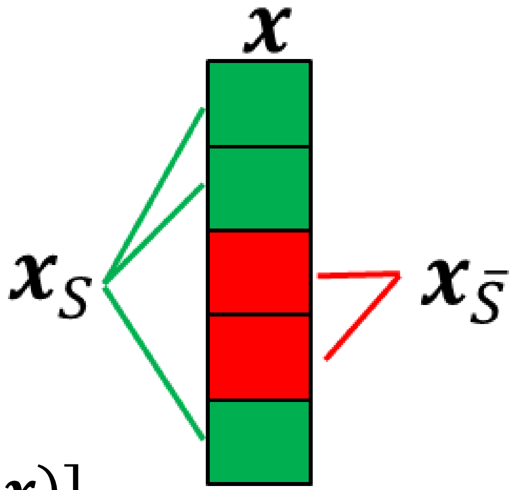
$$\phi_R = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{B,G\})\big) + \frac{1}{6}\big(v(\{R,B\}) - v(\{B\})\big) + \frac{1}{6}\big(v(\{R,G\}) - v(\{G\})\big) + \frac{1}{3}\big(v(\{R\}) - v(\{\ \})\big) = \$14$$

$$\phi_B = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{R,G\})\big) + \frac{1}{6}\big(v(\{R,B\}) - v(\{R\})\big) + \frac{1}{6}\big(v(\{B,G\}) - v(\{G\})\big) + \frac{1}{3}\big(v(\{B\}) - v(\{\ \})\big) = \$11$$

$$\phi_G = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{R,B\})\big) + \frac{1}{6}\big(v(\{R,G\}) - v(\{R\})\big) + \frac{1}{6}\big(v(\{B,G\}) - v(\{B\})\big) + \frac{1}{3}\big(v(\{G\}) - v(\{\ \})\big) = \$11$$

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



$$\phi_0 + \sum_{j=1}^{M} \phi_j = f(\boldsymbol{x}^*)$$

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

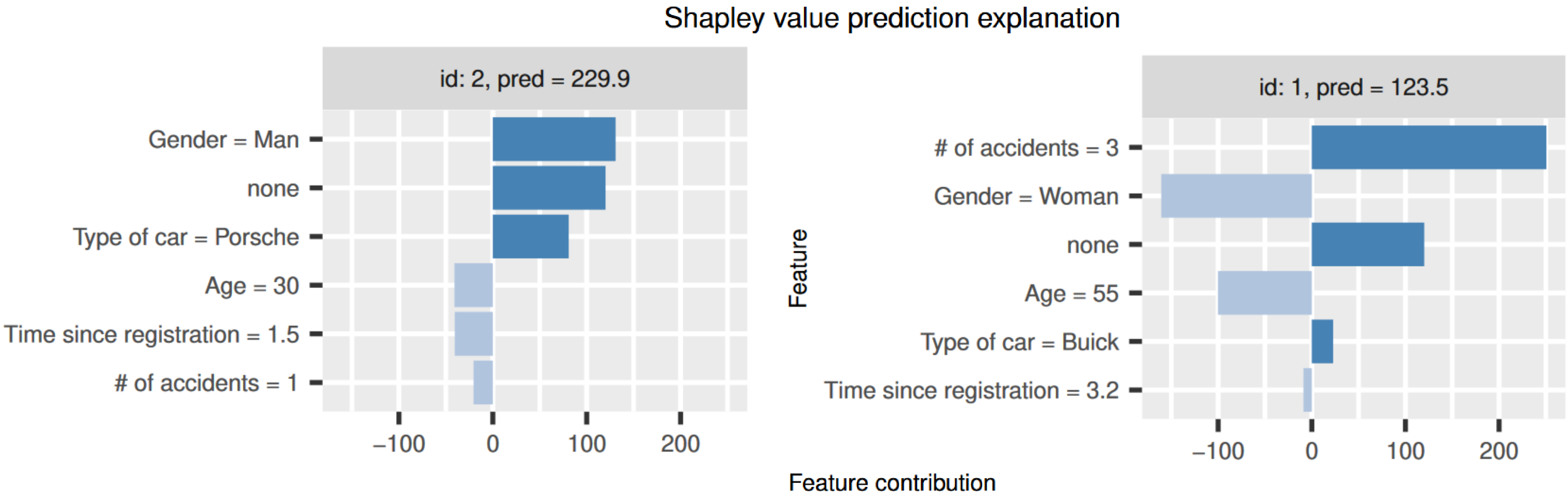
$f(\boldsymbol{x}) \perp\!\!\!\perp x_j$
implies $\phi_j = 0$

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

- Interpretation of ϕ_j: **The prediction change caused by observing the value of** x_j – averaged over whether the other features were observed or not

Example of Shapley value explanation

- Consider a model $f(\boldsymbol{x})$ trained to predict a fair price of a car insurance based on the following features $\boldsymbol{x}$:
  - Owner's age, owner's gender, type of car, time since the car was registered, number of accidents the last 5 years



Two main challenges

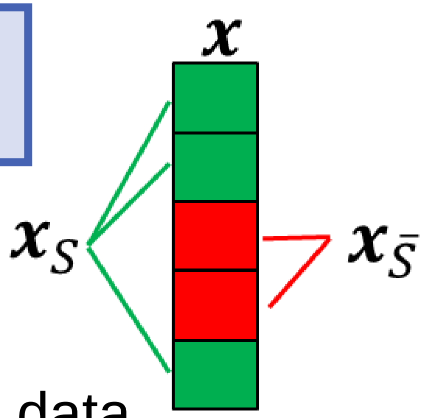
1. The computational complexity in the Shapley formula is of size 2^M

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

  - Approximate solutions may be obtained by using a finite sample of subsets S (KernelSHAP; Lundberg & Lee, 2017)

2. Estimating the contribution function

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



  - 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!

Two main challenges

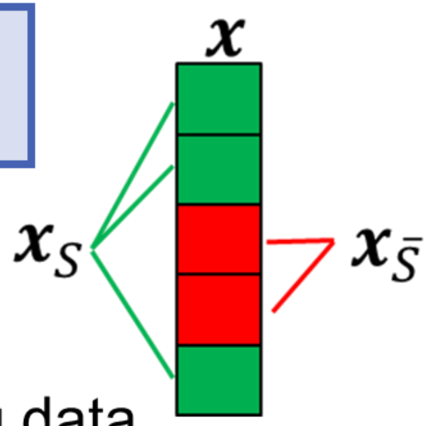
1. The computational complexity in the Shapley formula is of size 2^M

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

   - Approximate solutions may be obtained by using a finite sample of subsets S (KernelSHAP; Lundberg & Lee, 2017)

2. Estimating the contribution function

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



   - 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

Estimating v(S) properly

Artificial Intelligence
Volume 298, September 2021, 103502

ELSEVIER

Explaining individual predictions when features are dependent: More accurate approximations to Shapley values ☆

Kjersti Aas *, Martin Jullum, Anders Løland

Depend. Model. 2021; 9:62–81 DE GRUYTER

Kjersti Aas*, Thomas Nagler, Martin Jullum, and Anders Løland

Explaining predictive models using Shapley values and non-parametric vine copulas

CD-MAKE 2020: Machine Learning and Knowledge Extraction pp 117-137

Explaining Predictive Models with Mixed Features Usin Shapley Values and Conditional Inference Trees

Authors

Annabelle Redelmeier, Martin Jullum, Kjersti Aas

arXiv:2111.13507v1 [stat.ML] 26 Nov 2021

Using Shapley Values and Variational Autoencoders to Explain Predictive Models with Dependent Mixed Features

Lars Henry Berge Olsen*[1], Ingrid Kristine Glad[1], Martin Jullum[2], and Kjersti Aas[2]

[1]Department of Mathematics, University of Oslo
[2]Norwegian Computing Center

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

- Gaussian/empirical/ Gaussian copula
- Pair copulas
- Conditional inference trees
- Variational autoencoders with arbitrary conditioning (VAEAC)

l-agnostic idea:

e $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ in a proper way

\+

nte Carlo integration to approximate integral:

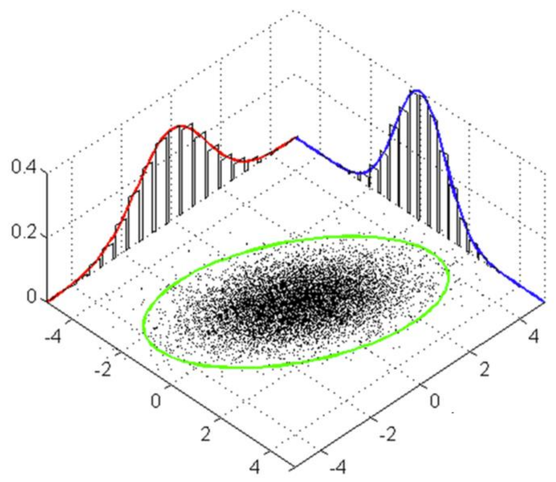
$$\hat{v}(S) = \frac{1}{K}\sum_{k=1}^{K} f(\boldsymbol{x}_{\bar{S}}^k, \boldsymbol{x}_S^*)$$

*Frye et. al (2020) briefly outline a VAEAC approach

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

1. Continuous features: Assume $p(\boldsymbol{x})$ is Gaussian $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Estimate $\boldsymbol{\mu}, \boldsymbol{\Sigma}$



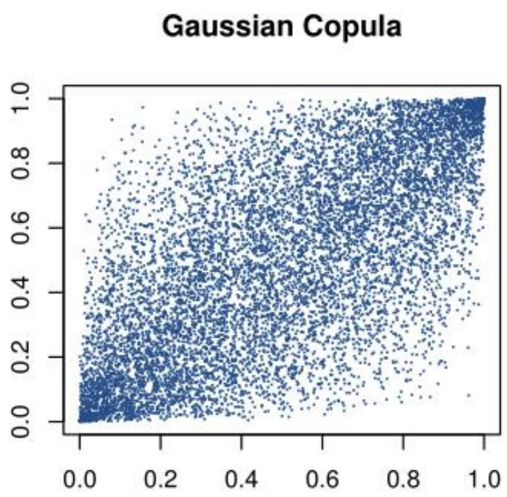
Gaussian distribution

Compute conditional means and covariances to obtain analytical expression for

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

Sample from that Gaussian distribution

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



2. Continuous features: Assume $p(\boldsymbol{x})$ is a Gaussian copula

Transform each $\boldsymbol{x}_i$ to $u_i \sim U[0,1]$ with inverse empirical CDF

Transform each u_i to $v_i \sim N(0,1)$

Estimate the correlation $\boldsymbol{\Sigma}^{\mathbf{v}}$ of $(v_1, \dots, v_M)$

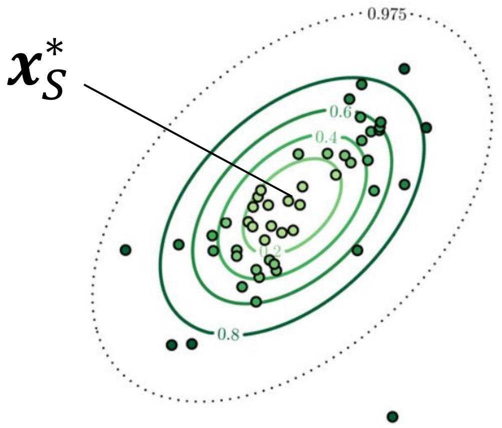
Obtain analytical expression for the Gaussian distribution $p_v(\boldsymbol{v}_{\bar{S}} | \boldsymbol{v}_S = \boldsymbol{v}_S^*)$

Sample from $p_v(\boldsymbol{v}_{\bar{S}} | \boldsymbol{v}_S = \boldsymbol{v}_S^*)$ + transform back to original scale

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

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

Mahalanobis distance to $\boldsymbol{x}_S^*$



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

Gaussian kernel on distance as weight

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

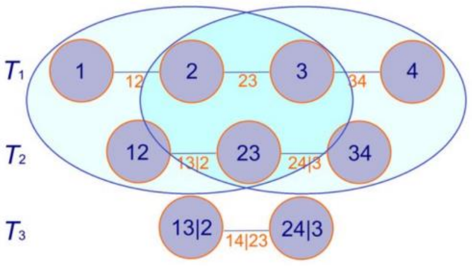
Approx $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ by the empirical distribution with weight $\frac{w_S(\boldsymbol{x}_S^*, \boldsymbol{x}_{\bar{S}}^i)}{\sum_{i=1}^{n_{\text{train}}} w_S(\boldsymbol{x}_S^*, \boldsymbol{x}_{\bar{S}}^i)}$ on each training observation $\boldsymbol{x}_{\bar{S}}^{\boldsymbol{i}}$

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

4. Continuous features: Estimate the dependence structure with a pair copula and weight the training observations using this construction

Transform each $\boldsymbol{x}_i$ to $u_i \sim U[0,1]$ with inverse empirical CDF

For each S, estimate the dependence structure $\hat{c}(u_{\bar{S}}, u_S)$ with a non-parametric pair copula



Define $w_S(x_S, x_{\bar{S}}) = \hat{c}(u_{\bar{S}}, u_S)/\hat{c}(u_{\bar{S}})$ as the dependence structure when conditioning on x_S

Approx $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ by the empirical distribution with weight

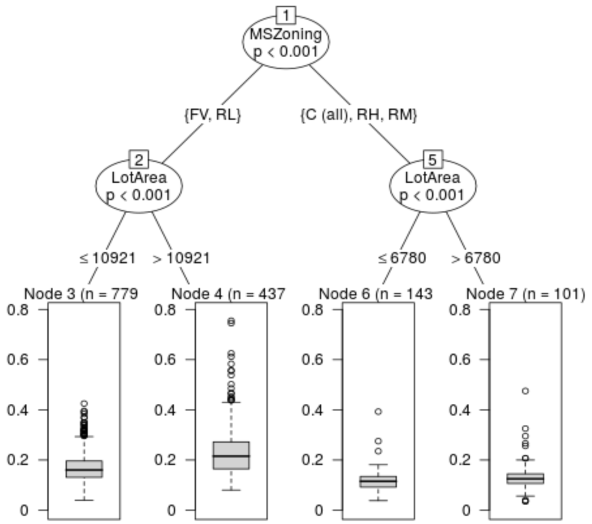
$$\frac{w_S(\boldsymbol{x}_S^*, \boldsymbol{x}_{\bar{S}}^i)}{\sum_{i=1}^{n_{\text{train}}} w_S(\boldsymbol{x}_S^*, \boldsymbol{x}_{\bar{S}}^i)}$$

on each training observation $\boldsymbol{x}_{\bar{S}}^i$

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

5. Mixed data: Use a multivariate decision trees as empirical distribution

For each S, fit a multivariate decision tree* to response $\boldsymbol{y} = \boldsymbol{x}_{\bar{S}}$ based on $\boldsymbol{x}_S$



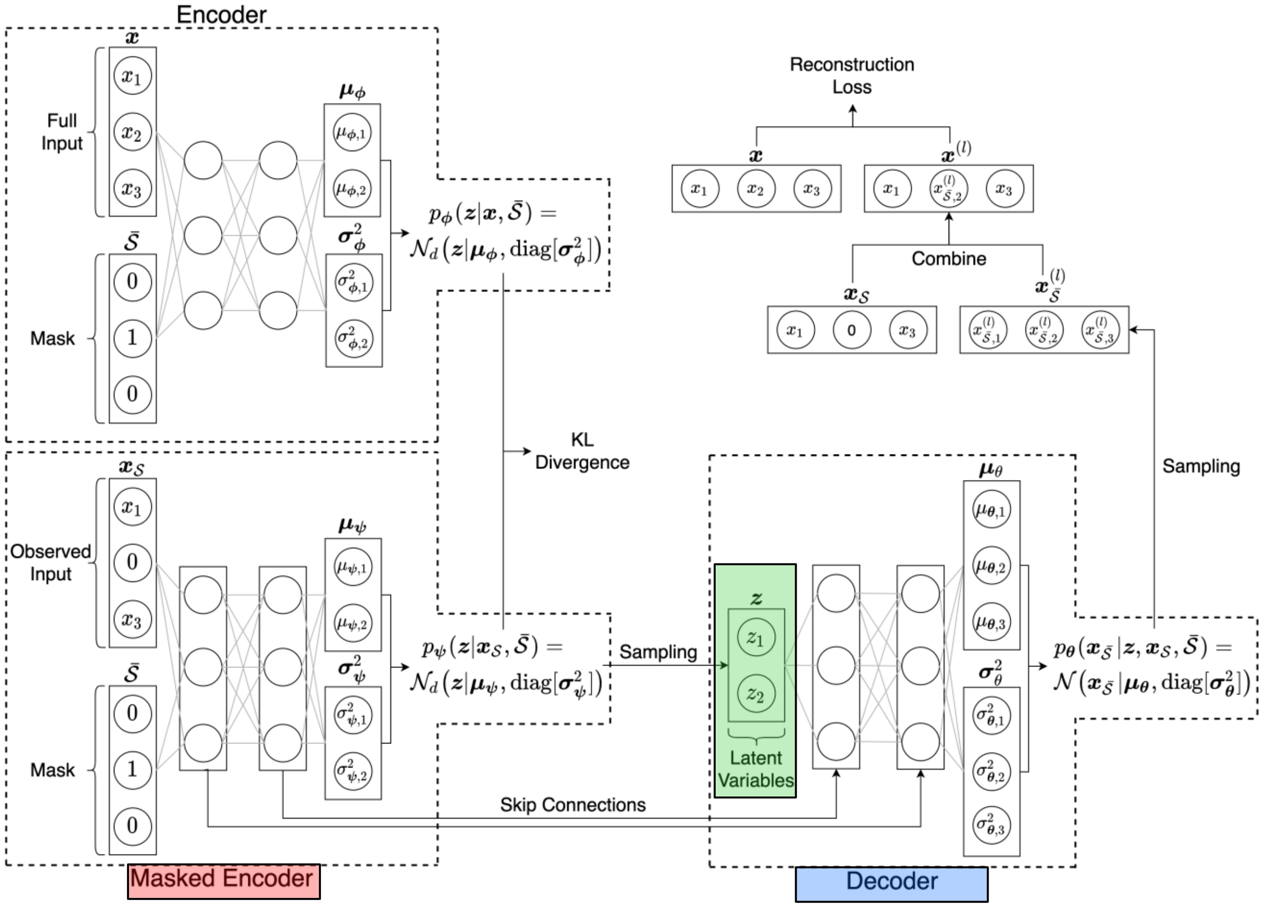
*We use conditional inference trees

Approx $p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*)$ by the empirical distribution of the training observations $\boldsymbol{x}_{\bar{S}}^i$ in the terminal node of $\boldsymbol{x}_S = \boldsymbol{x}_S^*$

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

6. Mixed data: Use an variational autoencoder with arbitrary conditioning (VAEAC)

Fit a VAEAC to all conditional distributions



For each S:

1. Feed the masked encoder with $\boldsymbol{x}_S^*$ (masking $\bar{S}$)
2. Generate latent variables $\boldsymbol{z}$
3. Feed $\boldsymbol{z}$ to the decoder and use it to generate samples $\boldsymbol{x}_{\bar{S}}^k$
4. Use $\boldsymbol{x}_{\bar{S}}^k$ as samples from samples $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$

When to use the different approaches

- Trade-offs between speed and accuracy
- Performance depends on data type and dependence structure
- General advice
  - Continuous data: Empirical approach
    - Gaussian if large n_train or M
    - Pair-copula if very heavy tails
  - Categorical/mixed data: Ctree
  - Future: Maybe VAEAC when properly implemented

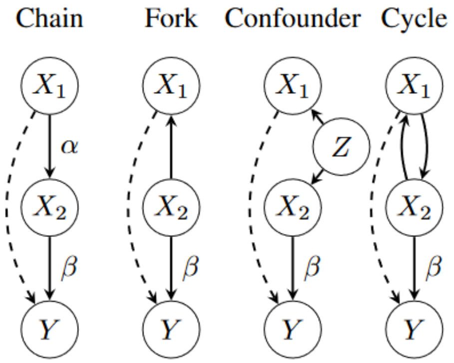
- TreeSHAP and KernelSHAP available from the shap Python library do NOT give useable estimates of $v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$ unless all features are close to independent

Different prediction explanation games

- Observational/conditional Shapley values uses
  - $v_C(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}}$

Janzing et al. (2019)

- Interventional Shapley values uses
  - $v_{do}(S) = E[f(\boldsymbol{x})|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*)] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*)) \mathrm{d}\boldsymbol{x}_{\bar{S}} = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}) \mathrm{d}\boldsymbol{x}_{\bar{S}} = v_I(S)$
- Chen et al. (2020) states that whether $v_C(S)$ or $v_I(S)$ is most appropriate depends on the application
  - $v_C(S)$ is most approriate if you want to learn about the actual relationship between features and modelled response
  - $v_I(S)$ is appropriate if you are debugging your model
- Heskes et al. (2020): Causal Shapley values
  - $v_{do}(S) = E[f(\boldsymbol{x})|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*)] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*)) \mathrm{d}\boldsymbol{x}_{\bar{S}}$, but $p(\boldsymbol{x}_{\bar{S}}|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*)) \neq p(\boldsymbol{x}_{\bar{S}})$, so $v_{do}(S) \neq v_I(S)$!
  - Rather *model* $p(\boldsymbol{x}_{\bar{S}}|do(\boldsymbol{x}_S = \boldsymbol{x}_S^*))$ with assumed causal ordering
  - Requires estimate conditional distributions, but not all combinations



Different prediction explanation games
My viewpoint

- Independence/Interventional Shapley values ($v_I(S)$) is only appropriate if
  - all features close to independent
  - or all dependence between features are due to a common confounder
  - You are debugging/testing robustness of your model

- Use Causal Shapley values
  - when you have confident knowledge about causal dependence between features

- All other cases: Use observational/conditional Shapley values
  - Observational/conditional Shapley values $\Leftrightarrow$ Causal Shapley values if no features are assumed to causally affect other features

Two main challenges

1. The computational complexity in the Shapley formula is of size 2^M

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

   - Approximate solutions may be obtained by using a finite sample of subsets S (KernelSHAP; Lundberg & Lee, 2017)

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

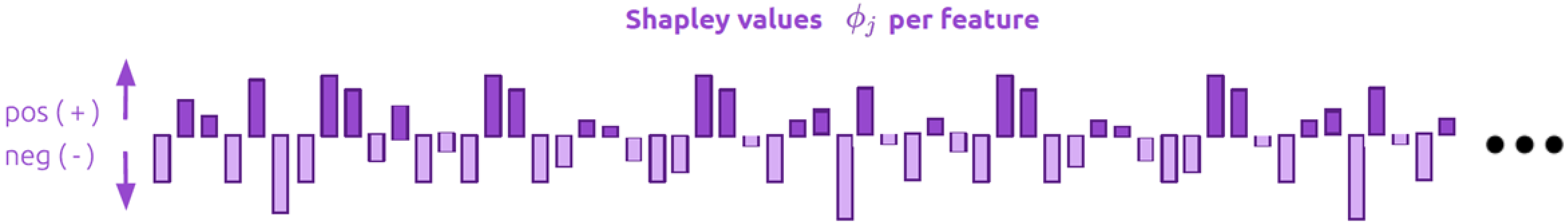
This assumes the features are independent!

Computational bottlenecks

$$
\begin{aligned}
M = 5 &\Rightarrow 2^M = 32 \\
M = 10 &\Rightarrow 2^M = 1024 \\
M = 20 &\Rightarrow 2^M = 1048676 \\
M = 40 &\Rightarrow 2^M > 10^{12} \\
M = 100 &\Rightarrow 2^M > 10^{30} \\
M = 1000 &\Rightarrow 2^M > 10^{301}
\end{aligned}
$$

1. The sum in the Shapley value formula is of size 2^M, growing exponentially in the number of features

2. How can we visualize, interpret and extract knowledge from 100s or 1000s of Shapley values?



- Typically: the sum of many small ϕ_j > sum of the few large ones
- Many highly dependent features complicates the interpretation

groupShapley

XAI.it 2021 - Italian Workshop on Explainable Artificial Intelligence

Efficient and simple prediction explanations with groupShapley: A practical perspective

Martin Jullum[1], Annabelle Redelmeier[1] and Kjersti Aas[1]

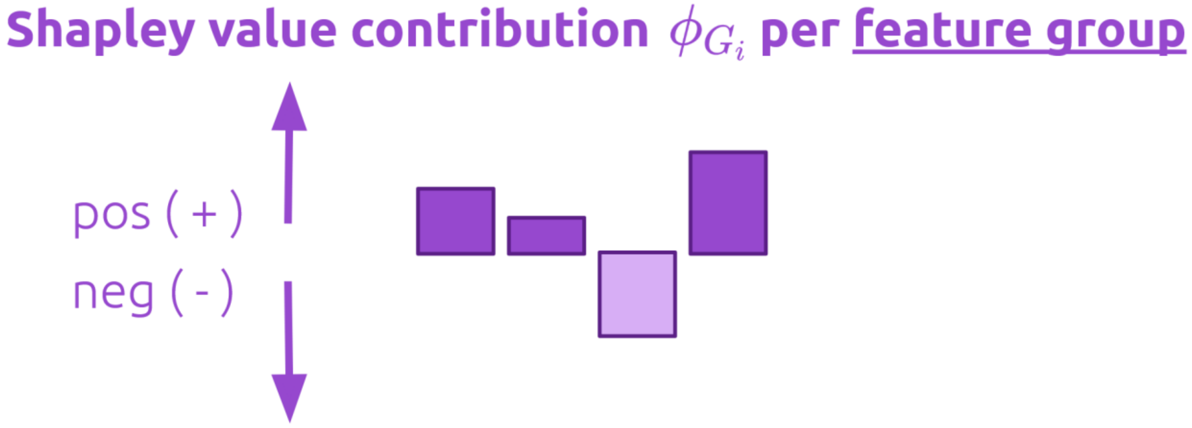
[1]*Norwegian Computing Center, P.O. Box 114, Blindern, N-0314 Oslo, Norway*

- Fundamentally very simple approach
  - Divide the M features into a small number of $\boldsymbol{G}$ disjoint groups $\{G_1, \ldots, G_G\}$.
  - Replace the feature subsets S in the Shapley formula by group subsets T:

$$\phi_{G_i} = \sum_{T \subseteq \boldsymbol{G} \setminus \{G_i\}} w(|T|)\left(v(T \cup G_i) - v(T)\right)$$

  - The scores are still Shapley values, so all mathematical properties are kept (on group level)

- What about the bottlenecks?
  - $2^G \ll 2^M \Rightarrow$ computationally tractable
  - G small $\Rightarrow$ easy to visualize



How to group the features?

- Crucial to group features based on the desired explanation
- Grouping based on feature dependence
  - Highly dependent features grouped together, using e.g. a clustering method.
  - Easier to study theoretically
  - Often difficult to extract knowledge from in practice
- Grouping based on application/feature knowledge
  - Group features of similar type or general category
  - Gives directly meaningful interpretations of computed groupShapley values
  - May perform multiple explanations with different groupings for increased understanding
- We advocate grouping based on feature knowledge in practical applications

Practical example 1: Car insurance

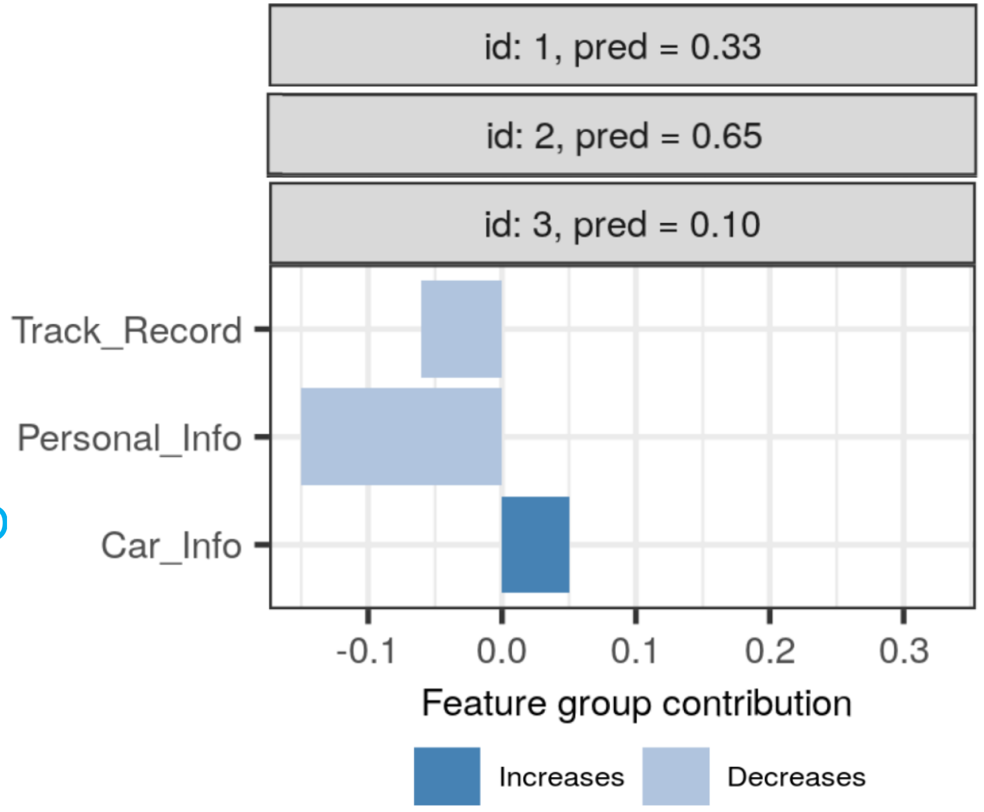
- US Car insurance dataset
  - 10 302 customers with records of crash/no crash + 21 features
  - Fit a random forest model with 500 trees to predict crash based on the 21 features

  - 3 feature groups based on type
    - Track record (4 features): # claims last 5 years, # licence record points, previous licence revokes, time as customer
    - Personal information (13 features): age of driver, education level, # children, job type, # driving children, marital status, gender, distance to work +++
    - Car information (4 features) value of car, age of car, type of car, whether car is red

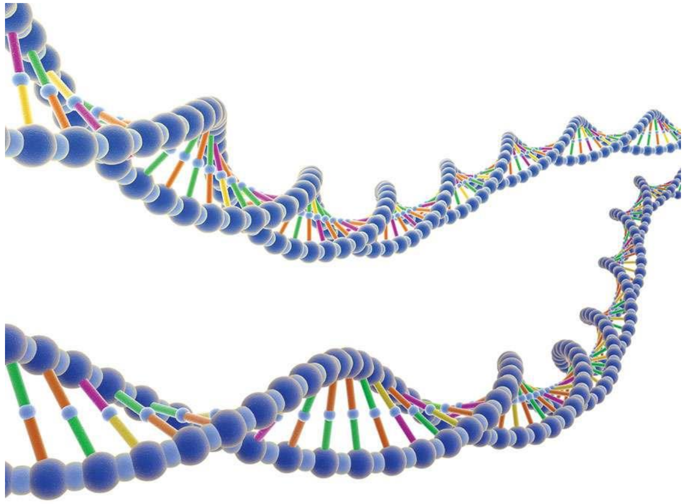
Practical example 1: Car insurance

- Explain predictions for 3 individuals

1. 1 claim last 5 years, 3 licence record points.
   Single mother of 4 (2 driving).
   Driving a SUV, 27 miles to work.
2. Got licence revoked and 10 licence record points.
   37 year old father of 2 (1 driving).
3. 3 claims last 5 years, no licence record points
   60 year old married doctor with no children, with a PhD
   Red sports car.



Practical example 2: Gene data

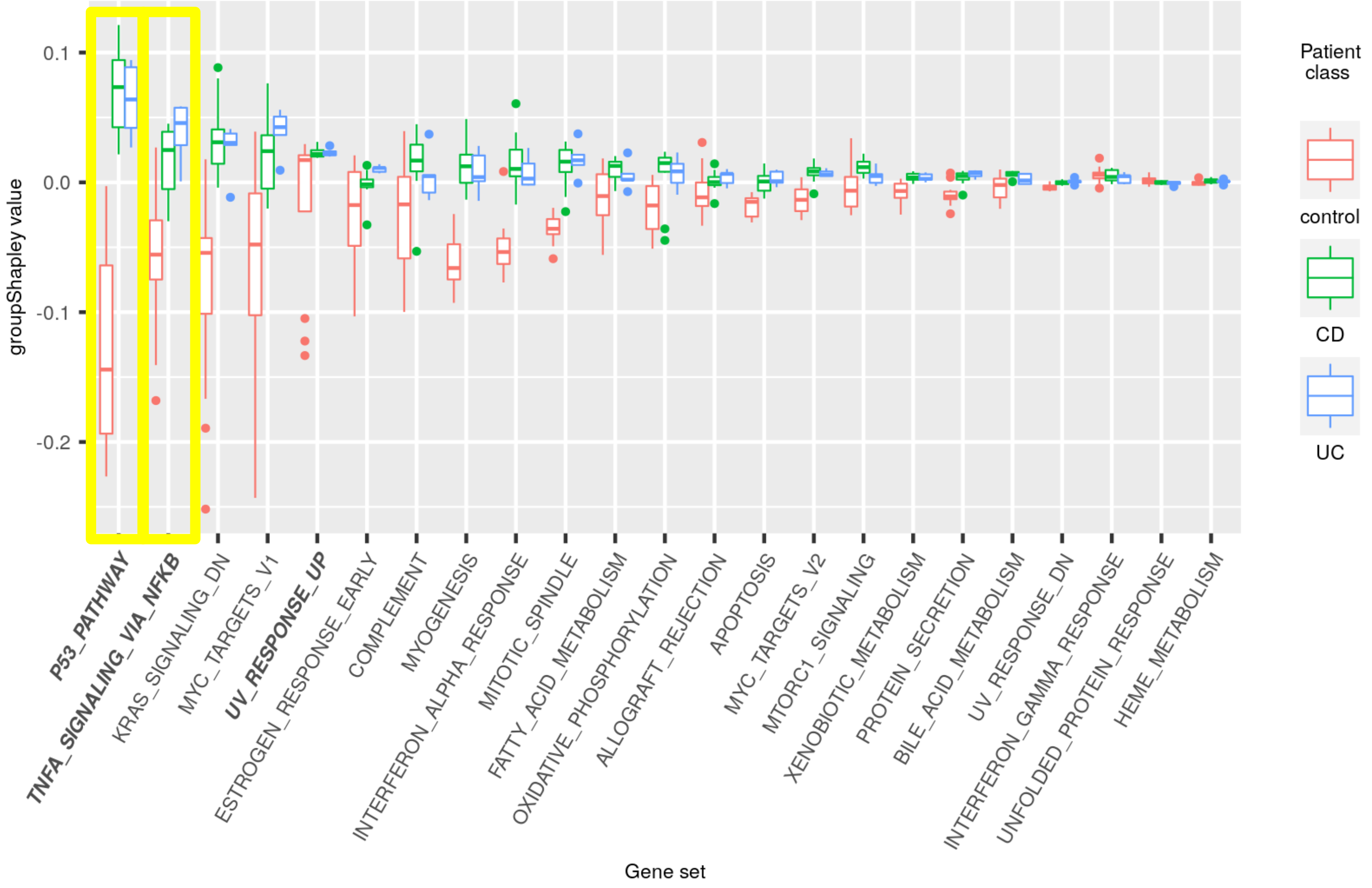


- Disease classification with high dimensional gene data
  - 127 patients where 85 are diseased with either Crohn's disease (CD) or Ulcerative colitis (UC) + 42 healthy controls.
  - 4 834 genes (after pre-processing)
  - Using 100 random individuals, we fit a Lasso penalized linear regression model to predict P(diseased with either CD or UC) based on the patient's genes

- Feature groups
  - Use the so-called Hallmark gene set to group the features (genes) into 23 different groups commonly used in gene set enrichment analysis
  - The Hallmark gene set "conveys a specific biological state or process" (Liberzon et al., 2015)

Practical example 2: Gene data

- Compute groupShapley values for the remaining 27 patients
- Make separate groupShapley boxplots for UC, CD and controls
- Can we identify genetical similarities and differences for UC and CD?
- Note: Model not trained to separate UC and CD



Software

The Journal of Open Source Software

shapr: An R-package for explaining machine learning models with dependence-aware Shapley values

Nikolai Sellereite[1] and Martin Jullum[1]

- R-package shapr
  - Computes Shapley values for any model $f(\boldsymbol{x})$ with different dependence-aware methods for estimating $v(S)$
  - All functionality works for both feature-wise and group-wise Shapley values
  - Currently undergoing heavy restructuring to allow
    - Parallellization
    - Reduce memory usage
    - Causal Shapley values
    - Improved user experience +++
    - Python wrapper

Take home points

- The Shapley value framework from game theory can be used to explain predictions from any ML model
- Shapley value measures the value of observing each feature
- There are two main challenges with such explanations
  - Computational complexity -> Approximate by sample of subsets S, or explain feature groups instead
  - Estimating contribution function v(S) -> Several differnet methods for different settings
- There exists other prediction explanation games:
  - Interventional Shapley values can be used for “debugging”
  - Causal Shapley values is promising if you have prior causal knowledge
- You can do most (hopefully all quite soon) types of prediction explanations efficiently with the shapr R-package
  - See package vignette at https://norskregnesentral.github.io/shapr/ for an intro

References

1. Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: More accurate approximations to shapley values. Artificial Intelligence **298**, 103502 (2021)
2. Aas, K., Nagler, T., Jullum, M., Løland, A.: Explaining predictive models using shapley values and non-parametric vine copulas. Dependence Modeling **9**, 62–81 (2021)
3. Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? arXiv preprint arXiv:2006.16234 (2020)
4. Heskes, T., Sijben, E., Bucur, I.G., Claassen, T.: Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. Advances in Neural Information Processing Systems **33** (2020)
5. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable ai: A causal problem. In: International Conference on Artificial Intelligence and Statistics. pp. 2907–2916. PMLR (2020)
6. Jullum, M., Redelmeier, A., Aas, K.: Efficient and simple prediction explanations with groupshapley: A practical perspective. In: Proceedings of the 2nd Italian Workshop on Explainable Artificial Intelligence. pp. 28–43. CEUR Workshop Proceedings (2021)
7. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. pp. 4768–4777 (2017)
8. Olsen, L.H.B., Glad, I.K., Jullum, M., Aas, K.: Using shapley values and variational autoencoders to explain predictive models with dependent mixed features. arXiv preprint arXiv:2111.13507 (2021)
9. Redelmeier, A., Jullum, M., Aas, K.: Explaining predictive models with mixed features using shapley values and conditional inference trees. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 117–137. Springer (2020)
10. Sellereite, N., Jullum, M.: shapr: An r-package for explaining machine learning models with dependence-aware shapley values. Journal of Open Source Software **5**(46), 2027 (2020)
11. Shapley, L.: A value fo n-person games. Ann. Math. Study28, Contributions to the Theory of Games, ed. by HW Kuhn, and AW Tucker pp. 307–317 (1953)

Contact

jullum@nr.no

https://martinjullum.netlify.app