This is a 1-slide summary. 5 min/5 slides lightning talks are given on request

groupSHAP

Efficient Shapley value explanation through feature groups

Explain prediction $f(x)$ for individual with features $x = x^*$

$S =$ a subset of the M features

$v(S) = E[f(x) | x_S = x_S^*]$

$w(S) =$ weight function

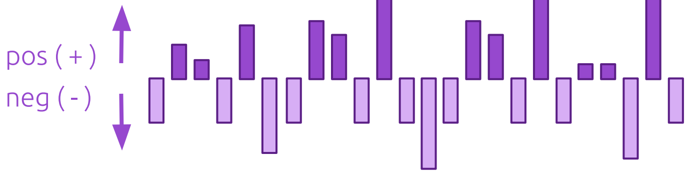
regular SHAP

SHAP formula for feature j

$$\phi_j = \underbrace{\sum_{\text{all } S \text{ without } j}}_{2^M \text{ terms}} w(S) \underbrace{(v(S \cup \{j\}) - v(S))}_{\text{marginal contribution}}$$

computationally <u>intractable</u> for many features

Shapley values contribution ϕ_j per feature



looong list for many features hard to interpret

Divide the M features into G feature groups

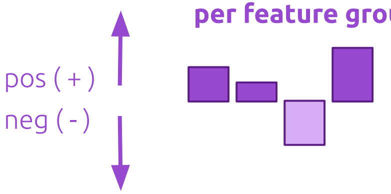
groupSHAP

SHAP formula for feature group g

$$\psi_g = \underbrace{\sum_{\text{all } T \text{ without } g}}_{2^G \text{ terms} << 2^M} w(T)(v(T \cup \{g\}) - v(T))$$

computationally <u>tractable</u> unless G too large

Shapley value contribution ψ_g per feature group



short list easy to interpret

Martin Jullum
jullum@nr.no

groupSHAP

Efficient Shapley value explanation through feature groups

Martin Jullum
Norwegian Computing Center

SHAP (SHapley Additive exPlanations)
The practical issue

Notation

S = a subset of the M features
$v(S) = E[f(x)|x_S = x^*_S]$
$w(S)$ = weight function

- Problem: Want to explain predictions from a model $f(x)$ with *M* features
- Shapley value formula for feature *j*

$$\phi_j = \sum_{\underbrace{\text{all } S \text{ without } j}_{2^M \text{ terms}}} w(S)\underbrace{(v(\mathcal{S} \cup \{j\}) - v(\mathcal{S}))}_{\text{marginal contribution}}$$

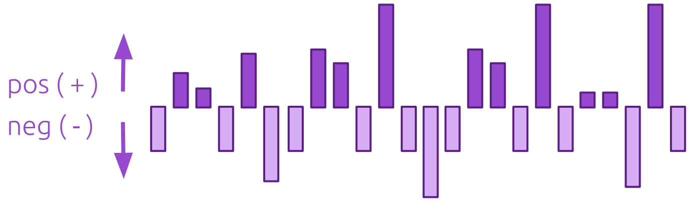
- Computational complexity

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

Generally nonsatisfactory approximation methods exists

- KernelSHAP*
  - Inaccurate for large *M*
- TreeSHAP/TreeExplainer*
  - Limited to tree-based models
- DASP (Deep Approximate Shapley Propagation)
  - Limited to neural Networks

Shapley values contribution ϕ_j per feature



- In any case: Potentially too long list of contributions from <u>dependent</u> features

*The popular python library *shap* uses KernelSHAP/TreeSHAP

groupSHAP
The idea

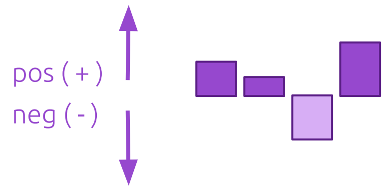
- A fundamentally simple solution:
  - Divide the M features into G feature groups
  - Replace the feature subsets S in the Shapley formula by feature group subsets T
- Shapley formula for feature group g

$$\psi_g = \sum_{\text{all } T \text{ without } g} w(T)(v(T \cup \{g\}) - v(T))$$

- $2^G \text{ terms} << 2^M \Rightarrow \text{Computationally tractable}$

- Still a Shapley value, so all properties remains
- Potential grouping criteria
  - Type of feature
  - Origin/source of feature
  - Feature dependence (high-dependence features in same group)

Shapley value contribution ψ_j per feature group



Example groups, car insurance:

- Car info
- main driver info
- other driver info
- previous incident info

groupSHAP
Theoretical result

Is group-wise Shapley values (using groupSHAP) ever the same as summing feature-wise Shapley values?

I.e. do we ever have? $\psi_g = \sum_{j \in g} \phi_j$

YES!

Any partially additively separable function $(f(\boldsymbol{x}) = \sum_{g=1}^{G} f_g(\boldsymbol{x}_g))$

with between-group feature independence $(\boldsymbol{x}_g \perp\!\!\!\perp \boldsymbol{x}_{g'})$

has $\psi_g = \sum_{j \in g} \phi_j$

groupSHAP

Practical use (through R-package *shapr*)

groupSHAP branch on GitHub (under development)

Code example

```r
remotes::install_github("NorskRegnesentral/shapr",ref = "groupSHAP")

# Loading the Boston housing data set
data("Boston", package = "MASS")
x_var <- c("lstat", "rm","dis","indus" ,"nox","tax")
y_var <- "medv"

x_train <- as.matrix(Boston[1:50, x_var])
y_train <- Boston[1:500, y_var]
x_test <- as.matrix(Boston[501:504, x_var])

# Fitting a basic xgboost model
model <- xgboost::xgboost(
  data = x_train,
  label = y_train,
  nround = 20,
  verbose = FALSE
)

# Define the feature groups
group <- list(c("lstat","rm","dis"),
              c("indus","nox"),
              "tax")

# Prepare the data for explanation
explainer <- shapr::shapr(x_train, model, group = group)

# Run the explainer
explanation <- shapr::explain(x_test,
                              explainer,
                              approach = "empirical",
                              prediction_zero = mean(y_train))

# Plot the group-wise shapley values
plot(explanation,plot_phi0 = F)
```

Resulting explanation figure