Boruta Shap method for associations network inference

Nicolas Captier (PhD student at Institut Curie U900)

We propose here a brief description of our Boruta Shap method for associations network inference. In particular, you will find the pseudo-codes of the BorutaShap algorithm implemented in this repository.

Let us first describe the problem we want to solve. We observe a data set $\mathcal{D} \in \mathbb{R}^{n \times m}$ with m different features (f_i , $i = 1, ..., m$). We assume that these features interact with each other and we would like to infer these interactions using only $\mathcal{D}$. In other words, we would like to infer a network whose nodes would be the different features and whose edges would indicate interactions between features.
For instance, one can think of $\mathcal{D}$ as a transcriptomic data set : the rows represent different patients or biological replicates and the columns (i.e features) represent genes that may be co-regulated.

To solve such a problem, we decompose it into several all-relevant feature selection problems. In other words, we identify several target features and we try to predict each one of them with the rest of the features (or with a subset of features identified as potential predictors) using an ensemble regression algorithm. For each regression, we use different tools to select all the relevant predictors ; we combine a statistical method called Boruta[1] with Shapley values[2]. At the end, each target will be associated with it selected relevant predictors.

Note : *we use ensemble regression algorithms since they do not make any assumption on the nature of the associations and they work reasonably well for small sample sizes. The tuning of the hyperparameters of such algorithms is let to the user.*

Algorithm 1: Boruta - SHAP

Data: $X \in \mathbb{R}^{n \times p}$ (predictors) and $Y \in \mathbb{R}^n$ (response)
Input: number of runs n_{runs}
Output: set of important features, set of unimportant features

for $i = 1, ..., n_{runs}$ **do**
- $\tilde{X} \leftarrow$ concatenate X and X_{shadow} (add shadow features) ;
- Split $\tilde{X}$ into $\tilde{X}_{train}$ and $\tilde{X}_{test}$;
- Run an ensemble algorithm to solve the regression problem $Y \sim \tilde{X}$ with $\tilde{X}_{train}$;
- Compute importance scores with mean absolute Shapley values on $\tilde{X}_{test}$;
- Assign a hit to every feature with higher score than max shadow ;
- Use accumulated hits to identify important and unimportant features ;
- Update X removing unimportant features ;

Algorithm 2: Associations network inference

Data: features $\mathcal{D} = \{f_1, ..., f_m\}$ ($f_i \in \mathbb{R}^n$, $i = 1, ..., m$)
Input: number of runs for Boruta method n_{runs}
Output: graph $G = (N, E)$ with nodes $N = \{f_i , i = 1, ..., m\}$

Identify predictors $P = \{X_1, ..., X_p\} \subset \mathcal{D}$ and responses $R = \{Y_1, ..., Y_r\} \subset \mathcal{D}$;
for $j = 1, ..., r$ **do**
- Build a regression model $\mathcal{R} : Y_j \sim (P\,;\, \text{Confounders})$;
- Important features $\leftarrow$ **Algorithm 1**(X = P + Confounders , Y = Y$_j$, n_{runs}) ;
- For each important features, create a new edge in G which links the response Y_j and this feature ;

---

[1] "Boruta explained exactly how you wished someone explained to you" Mazzanti 2017
[2] "A Unified Approach to Interpreting Model Predictions" Lundberg et al. 2017