Automatic Piecewise Linear Regression version 9.3.0

Von Ottenbreit Data Science

Automatic Piecewise Linear Regression (APLR)

- Automatically handles variable selection, non-linear relationships and interactions
- Empirical tests show that APLR is often able to compete with tree-based methods on predictiveness
- APLR produces interpretable models

Some useful functionality

- Allows to specify the loss function among several built-in options such as *mse* (default), *poisson*, *gamma*, etc
- Allows to specify the link function among built in variants: *identity* (default), *logit* and *log*
- Allows to specify the function for calculating the validation set tuning metric among built-in variants such as *default* (same as loss function), *mse*, *negative_gini* etc
- Alternatively allows to pass custom Python functions for each of the above
- The user can specify a list of predictors to be prioritized. Terms based on these will be added or updated in each boosting step as long as training error is reduced. This can be particularly useful to include the full effect of a predictor that is weak but important
- The user can specify a list of predictors with monotonic constraints. This can improve model realism, usually with only a minimal loss increase
- The user can provide constraints on which predictors are allowed in interaction terms
- In addition to solving regression problems APLR can also be used for classification. The object APLRClassifier fits a logit APLR model for each response category. When predicting, each observation is predicted to the class with the highest predicted class probability

APLR fitting procedure: Important hyperparameters

- *m* (default is 3000) is the maximum boosting steps to try. Should be large enough to minimize the validation error
- *v* (default is 0.1) is the learning rate. Should ideally be <= 0.1
- *max_interaction_level* (default is 1) specifies the maximum allowed interaction depth. Should be tuned by for example doing a grid search
- *min_observations_in_split* (default is 20) specifies the minimum number of training observations where a term must not have a value of zero due to the max, min or indicator functions (number of effective observations). The higher value the lower variance (the term relies on more observations) but higher bias (less fine grained splits). Should be tuned by for example doing a grid search

References

- Link to published article: https://link.springer.com/article/10.1007/s00180-024-01475-4