Predictive Analytics and Comparison (PAC) framework – Matlab Package

By Ricardo Rendall & Marco Reis

Summary

The Predictive Analytics and Comparison (PAC) framework was devised as an effective solution for “data rich/knowledge poor” scenarios, where a priori knowledge is insufficient to select a suitable prediction method. In these scenarios, practitioners often adopt their favorite method(s) but at the expense of ignoring other approaches that might be better suited for the prediction problem at hand. Thus, this Matlab® package contains methods from different classes and covers a wide spectrum of prior assumptions regarding the predictor variables (e.g. collinearity levels), the response variable and the relationship between them (e.g. non-linearity). To select the best predictive method for a given application, PAC screens different regression methods and robustly assess their prediction performance based on double cross-validation. Afterwards, the best method(s) or class of methods can be investigated in order to identify the most relevant predictors.

What is included?

There are four classes of regression methods considered in the current version of the PAC package: variable selection, penalized regression, latent variable methods and tree-based ensembles. The class of variable selection includes forward stepwise regression. The penalized regression class contains ridge regression, least absolute shrinkage and selector operator (lasso) and elastic nets. The class of latent variable methods contains principal component regression and partial least squares. Lastly, the tree-based ensembles includes bagging of regression trees, random forests and boosting of regression trees. For completeness, multiple linear regression is also included.

Advantages

The PAC package offers the following benefits:

- The package covers several classes of regression methods, which helps practitioners identify the most relevant method(s) or class of methods for their application;
- Model building is easier due to adoption of object-oriented programming: the methods share the same syntax for both model training and predicting samples with unknown response values;
- The settings for model training (e.g. k in k-fold cross validation, the number of repeated iterations of cross-validation) can be easily changed by the user;
- The range of hyper-parameters tested for each method during model training can also be modified by the user;
- It is simple to set a comparison based on double cross-validation;
- Example scripts are provided where the user only needs to change the dataset in order to conduct the comparison.