ExactSampling: risk evaluation using exact resampling methods for the $k$ Nearest Neighbor algorithm

Kai Li$^1$, Alain Céllisse$^{2,4}$, Michel Koskas$^1$, Tristan Mary-Huard$^{1,3,*}$

2. Laboratoire de Mathématique Paul Painlevé, UMR 8524 CNRS - Université Lille 1, France.
3. UMR de Génétique Végétale, INRA, Université Paris-Sud, CNRS, Gif-sur-Yvette, France
4. Equipe-projet MODAL, INRIA Lille, France

*Contact author: maryhuar@agroparistech.fr

Keywords: Machine learning, cross-validation, bootstrap

The $k$-Nearest Neighbor algorithm ($k$NN) is a popular method to perform either regression or classification. It consists in predicting the response of a new observation according to the $k$ closest observations in the training sample. While simple, the $k$NN algorithm demonstrated good performances on many applications (Hastie et al., 2001), and has already been implemented in several R packages (class, FNN).

The performance of the $k$NN algorithm highly depends on the tuning of parameter $k$, that should be performed adaptively to the data at hand. To do so, resampling strategies such as Bootstrap or Leave-$p$-out ($L_p$O) cross-validation can be used to estimate the prediction performance obtained with different values of $k$, and select the optimal value $k^*$ that minimizes the prediction error rate. However, the computational cost of such strategies is prohibitive. In practice one often needs to limit the number of resamplings as the training sample size gets large, yielding poor approximation of the actual risk. Recently computational shortcuts have been derived to compute the exact $L_p$O or Bootstrap risk estimators in a short computational time, for instance linear with respect to the number of observations in the case of $L_p$O, whatever $p$ (Celisse, 2011). These efficient strategies are now implemented in a package called ExactSampling. In this package additional computational shortcuts are also provided in settings where exacts formulas are not available for the bootstrap estimator. In particular the approximation level can be specified beforehand by the practitioner.

From an algorithmic point of view, we improve the computational time of the nearest neighbor search step, which is the most consuming one. This is done thanks to a new combination of classical algorithms. All functions have been developed using the C programming language. From a theoretical point of view, the package may be used to investigate the properties of resampling methods that are still poorly understood. In practice ExactSampling can be of great help to tune the $k$ parameter of the kNN when dealing with large dataset: while a naive implementation of the leave-10-out cross validation procedure would be totally inefficient to deal with a dataset of size 100 (hundreds of hours of computational time), the knn.cv function of the ExactSampling package computes this estimator within a few minutes for datasets of size 100,000.

References

Celisse, A. Mary-Huard, T. (2011). Exact cross-validation for knn and applications to passive and active learning in classification. JSFdS 152(3).