Parallel Programming in R

Fabrice Rossi

December 18, 2017

1 Foreach

The following exercises illustrate the use of the `foreach` construction for the `foreach` package.

1.1 Bootstrap

The goal of the exercise is to implement a bootstrap estimator for the empirical risk of a k nearest neighbor (knn) classifier.

1. Generate an artificial data set of size n in $\mathbb{R}^p$ as follows:
   - generate a vector Y of labels A and B of size n approximately balanced
   - conditionally on Y generate X with a isotropic Gaussian distribution with a fixed variance σ^2 and a mean that depends on the value of Y
   - gather the data into a dataframe
2. Write a `boot` function that generate a bootstrap sample stratified according to a vector of factors. The sample should be returned as a vector selected indices among the indices needed to map the full vector of factors.
3. Write a `knn.boot.seq` function that receives as inputs a data frame, the index of the target variable in the data frame, k and a number of bootstrap samples. The function should return a list containing a bootstrap based estimation of the error rate of the knn classifier as well as its estimated variance, for the given value of k. For this function, the `knn` function provided by the `class` package can be used.
4. Implement a parallel version of `knn.boot.seq`, `knn.boot.foreach`, with the same semantic but based on the `foreach` loop.
5. Using the `proxy` package, implement a `mknn` function with a similar interface as the one of `knn` from the `class` package but that can output prediction of the knn algorithm for multiple values of `k` at once.

6. Use `mknn` to implement a parallel version of `knn.boot.seq`, `mknn.boot.foreach` which similar results but for multiple values of k, using the **same** bootstrap samples for different k.

7. Use `foreach` nested loops to achieve the same functionality but using the base `knn` function.

8. Compare the performances of the solutions.

1.2 Naive Bayes Classifier

The goal of the exercise is to implement a naive bayes classifier (NBC) efficiently.

1. Write a `nbc` function that computes all the probabilities needed for a NBC from a dataset with factor valued variables only.

2. Write a `predict.nbc` function that handles the prediction on new data for the NBC model.

3. Write a parallel version of `nbc`, `nbc.foreach` based on the `foreach` construction. It is recommend to handle variables in a parallel way, not objects.

4. Implement a forward feature selection model for `nbc` using a parallel evaluation of the features.