7

Computational Issues

This chapter is devoted to the implementation of the functional nonparametric prediction methods based on regression, conditional quantiles and conditional mode with special attention to the regression ones. It concerns mainly users/practitioners wishing to test such functional statistical techniques on their datasets. The main goal consists in presenting several routines written in S+ or R in order to make any user familiar with such statistical methods. In particular, we build procedures with automatic choice of the smoothing parameters (bandwidths), which is especially interesting for practitioners. This chapter is written to be self-contained. However, in order to make this chapter easier to understand, we recommend the reading of the "nontheoretical" Chapters 2, 3, 4 and 5. After the description of the various routines, we propose a short case study which allows one to understand how work such procedures and how they can be easily implemented. Finally, the source codes, functional datasets, descriptions of the $R/S+$ routines and guidelines for use are given with much more detail in the companion website *http://www.lsp.ups-tlse.fr/staph/npfda*.

7.1 Computing Estimators

We focus on the various functional nonparametric prediction methods. For each of them, we present the kernel estimator and its corresponding implementations through $R/S+$ subroutine. Most of the programs propose an automatic method for selecting the smoothing parameter (i.e., for choosing the bandwidths), which makes these procedures particularly attractive for practitioners. Concerning the functional nonparametric regression method, special attention is paid because it was the first one developed from an historical point of view and because it is the most popular from a general statistical point of view. Therefore, in this regression context we propose several kernel estimators with various automatic selections of the smoothing parameter.

Note that we consider only two families of semi-metrics (computed via `semimetric.pca` or `semimetric.deriv`) described in Chapter 3, two basic kernel functions (see routines `triangle` or `quadratic` in the companion website[1]) described in Chapter 4 and the corresponding integrated ones (see routines `integrated.quadratic` and `integrated.triangle` in the website[1]) described in Chapter 5. However, the following package of subroutines can be viewed as a basic library; any user, according to his statistical and programmer's level, can increase this library by adding his own kernels, integrated kernels or semi-metrics routines.

We recall that we focus on the prediction problem which corresponds to the situation when we observe n pairs $(\boldsymbol{x}_i, y_i)_{i=1,\ldots,n}$ independently and identically distributed: $\boldsymbol{x}_i = \{\chi_i(t_1), \ldots, \chi_i(t_J)\}$ is the discretized version of the curve $\chi_i = \{\chi_i(t);\ t \in T\}$ measured at J points $t_1, \ldots, t_J$ whereas the y_i's are scalar responses. In addition, $\boldsymbol{d}_q(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ denotes any semi-metric (index of proximity) between the observed curves $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$. So, the statistical problem consists in predicting the responses from the curves.

7.1.1 Prediction via Regression

First, we consider the kernel estimators defined previously in (5.23). It achieves the prediction at an observed curve $\boldsymbol{x}_{i'}$ by building a weighted average of the y_i's for which the corresponding $\boldsymbol{x}_i$ is such that the quantity $\boldsymbol{d}_d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ is smaller than a positive real parameter h called bandwidth. In a second attempt, we will consider a slightly modified version in which we replace the bandwidth h by the number k of $\boldsymbol{x}_i$'s that are taken into account to compute the weighted average; such methods use the terminology k-Nearest Neighbours . For both kernel and k-NN estimators, we propose various procedures, the most basic one being the case when the user fixes himself the smoothing parameter h or k. Any other routine achieves an automatic selection of the smoothing parameter. So, if the practitioner wishes to test several different bandwidths, the basic routines can be used, or, in the opposite case, let the other routines automatically choose them.

- **Functional kernel estimator without bandwidth selection**

  The main goal is to compute the quantity:

$$R^{kernel}(\boldsymbol{x}) \;=\; \frac{\sum_{i=1}^{n} y_i K\left(\boldsymbol{d}_q(\boldsymbol{x}_i, \boldsymbol{x})/h\right)}{\sum_{i=1}^{n} K\left(\boldsymbol{d}_q(\boldsymbol{x}_i, \boldsymbol{x})/h\right)},$$

  where $(\boldsymbol{x}_i, y_i)_{i=1,\ldots,n}$ are the observed pairs and $\boldsymbol{x}$ is an observed curve at which the regression is estimated. The user has to fix the bandwidth h, the semi-metric $\boldsymbol{d}_q(.,.)$ and the kernel function $K(.)$. The routine `funopare.kernel` computes the quantities

[1] http://www.lsp.ups-tlse.fr/staph/npfda