T-61.6030 Introductory Elements of Functional Data Analysis: Ferraty: Chapters 6,7

Ville Turunen

ville.t.turunen@tkk.fi
Chapter 6: Some selected asymptotics
Chapter 7: Computational issues
Almost complete convergence

- asymptotic results for the functional estimators from Ch. 5
- presented in terms of *almost complete convergence*

**Definition A.1.** One says that \((X_n)_{n \in \mathbb{N}}\) converges almost completely to some r.r.v. \(X\), if and only if

\[
\forall \epsilon > 0, \quad \sum_{n \in \mathbb{N}} P(|X_n - X| > \epsilon) < \infty
\]

It is denoted by

\[
\lim_{n \to \infty} X_n = X, \text{a.co.}
\]

- convergence in probability and *almost sure convergence* follow
Some selected asymptotics

- main assumptions
  - small ball probability around $\mathcal{X}$ is nonnull
  - bandwidth $h$ is a positive sequence that limits to zero
  - $K$ is a kernel of type I or type II
  - (plus couple more)

- e.g. for regression estimator $\hat{r}(\mathcal{X})$
  - $\lim_{n \to \infty} \hat{r}(\mathcal{X}) = r(\mathcal{X}), a.co.$ (Theorem 6.1)

- similar results for $\hat{m}(\mathcal{X}), \hat{\theta}(\mathcal{X}), \hat{t}_\alpha(\mathcal{X}), \hat{F}_Y^{\mathcal{X}}(y)$

- i.e. all the estimators are consistent
Rates of convergence

- results for rates of convergence in Section 6.3
- e.g., for regression estimator \( \hat{r}(\mathcal{X}) \)
  \[ \hat{r}(\mathcal{X}) - r(\mathcal{X}) = O(h^\beta) + O_{a.co}\left(\sqrt{\frac{\log n}{n\varphi_X(h)}}\right) \]  (Theorem 6.11)
- bias + dispersion
- bias increases and dispersion decreases with bandwidth \( h \)
- how to automatically choose \( h \)?
- also depends on unknown constant \( \beta \)
- \( \rightarrow \) how can MSE-loss function be obtained?
Computational issues

- Ch.7 presents functional nonparametric routines written in R
- downloadable from the book website
- example: predicting fat content
Computing estimators

- prediction problem
  - observed \((x_i, y_i)_{i=1,...,n}\)
  - \(x_i = \{\mathcal{X}_i(t_1), \ldots, \mathcal{X}_i(t_J)\}\) (discretized)
  - \(\mathcal{X}_i = \{\mathcal{X}_i(t); t \in T\}\) (curve)

- kernel estimators with automatic selections for the smoothing parameter

- semi-metrics: `semimetric.pca`, `semimetric.deriv`

- kernel functions: `triangle`, `quadratic`,
  `triangle.integrated`, `quadratic.integrated`
Functional kernel estimators

- **funopare.kernel:**
  \[ R_{\text{kernel}}(x) = \frac{\sum_{i=1}^{n} y_i K(d_q(x_i, x)/h)}{K(d_q(x_i, x)/h)} \] (5.23)
  
  - user selects bandwidth \( h \)

- **funopare.kernel.cv:**
  
  - bandwidth \( h_{opt} \) found automatically by cross-validation
  
  \[ h_{opt} = \arg \min_h CV(h) \]
  
  \[ CV(h) = \sum_{i=1}^{n} (y_i - R_{\text{kernel}}(x_i)^{-i})(x_i))^2 \]
  
  \[ R_{\text{kernel}}(x)^{-i} = \frac{\sum_{j=1, j \neq i}^{n} y_j K(d_q(x_j, x)/h))}{\sum_{j=1, j \neq i}^{n} K(d_q(x_j, x)/h))} \]
k-Nearest Neighbours estimators

- use bandwidth $h_k(x)$ for which there are exactly $k$ curves among the $x_i$s such that $d_q(x_i, x) < h_k(x)$

- `funopare.knn`: user selects $k$
- `funopare.knn.gcv`: global $k$ using CV
- `funopare.knn.lcv`: local $k$ using CV
  - the optimal number of neighbours can change from one curve to another
Functional conditional quantiles and modes

- local $k$-NN approach
- $h_k$ bandwidth for the curves as before
- $g_\kappa$ bandwidth for the responses
- optimal values $k_{opt}(x), \kappa_{opt}(y)$ obtained by cross-validation
  where learning set is split to two

- `funopare.quantile.lcv`: $t_{kN N}(x)$
- `funopare.mode.lcv`: $\theta_{kN N}(x)$
**Spectrometric data**

![Spectrometric data graph and table]

**Graph Description:**
- The graph shows absorbance values over a range of wavelengths, from 850 to 1050.
- There are 215 rows of data, each representing a different sample or observation.

**Table Description:**
- The table below the graph provides a grid layout for the data:
  - **Row 1:** Sample 1 at λ₁, ..., λ₁₀₀, y₁
  - **Row i:** Sample i at λ₁, ..., λ₁₀₀, yᵢ
  - **Row 215:** Sample 215 at λ₁, ..., λ₁₀₀, y₂₁₅

**Table Data:**
- The table columns are labeled as follows:
  - **Col 1:** \( x₁(λ₁) \) to \( x₁(λ₁₀₀) \)
  - **Col j:** \( x₁(λⱼ) \) to \( x₁(λ₁₀₀) \)
  - **Col 100:** \( x₁(λ₁₀₀) \)
  - **Col 101:** \( y₁ \) to \( y₂₁₅ \)
Functional prediction of fat content (1/3)

- learning set: first 160 units
- test set: last 55 units

- performance measures:
  - distribution of square errors:
    \[ se_i = (y_i - \hat{y}_i)^2, \ i = 161, \cdots, 215 \]
  - empirical mean square errors:
    \[ MSE = \frac{1}{55} \sum_{i=161}^{215} se_i \]

- three routines:
  - functional conditional expectation (regression)
  - functional conditional mode
  - functional conditional median

- all with \( k \)-NN bandwidth and locally optimized \( k \)
Functional prediction of fat content (2/3)

Cond. Expect.: \( \text{MSE} = 3.5 \)

Cond. Mode: \( \text{MSE} = 3.61 \)

Cond. Median: \( \text{MSE} = 3.44 \)
Square Error

Cond. Expectation  Cond. Mode  Cond. median  Multimethods
MSE=3.5  MSE=3.61  MSE=3.44  MSE=2.21

*multimethods*: average of the three methods