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

Ville Turunen

ville.t.turunen@tkk.fi

Outline

- 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

$$\lim_{n\to\infty} X_n = X, a.co.$$

- *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 $\widehat{r}(\mathcal{X})$
  - $\lim_{n\to\infty} \widehat{r}(\mathcal{X}) = r(\mathcal{X}), a.co.$ (Theorem 6.1)
- similar results for $\widehat{m}(\mathcal{X})$, $\widehat{\theta}(\mathcal{X})$, $\widehat{t_\alpha}(\mathcal{X})$, $\widehat{F}_Y^{\mathcal{X}}(y)$
- i.e. all the estimators are consistent

Rates of convergence

- results for rates of convergence in Section 6.3
- e.g for regression estimator $\widehat{r}(\mathcal{X})$
  - $\widehat{r}(\mathcal{X}) - r(\mathcal{X}) = O(h^{\beta}) + O_{a.co}(\sqrt{\frac{\log n}{n\varphi_{\mathcal{X}}(h)}})$ (Theorem 6.11)
- bias + dispersion
- bias increases and dispersion decreases with bandwidth h
- how to automatically choose h?

- also depends on unknown constant β
- $\rightarrow$ 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 $(\mathbf{x_i}, y_i)_{i=1,\ldots,n}$
  - $\mathbf{x_i} = \{\mathcal{X}_i(t_1), \ldots, \mathcal{X}_i(t_J)\}$ (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:`
  - $R^{kernel}(\mathbf{x}) = \frac{\sum_{i=1}^{n} y_i K(\mathbf{d_q}(\mathbf{x}_i,\mathbf{x})/h)}{K(\mathbf{d_q}(\mathbf{x_i},\mathbf{x})/h)}$ (5.23)
  - user selects bandwidth h

- `funopare.kernel.cv:`
  - bandwidth h_{opt} found automatically by cross-validation
  - $h_{opt} = \arg\min_h CV(h)$
  - $CV(h) = \sum_{i=1}^{n} (y_i - R_{(-i)}^{kernel}(\mathbf{x_i}))^2$
  - $R_{(-i)}^{kernel}(\mathbf{x}) = \frac{\sum_{j=1, j\neq i}^{n} y_j K(\mathbf{d_q}(\mathbf{x_j},\mathbf{x})/h))}{\sum_{j=1, j\neq i}^{n} K(\mathbf{d_q}(\mathbf{x_j},\mathbf{x})/h))}$

k-Nearest Neighbours estimators

- use bandwidth $h_k(\mathbf{x})$ for which there are exactly k curves among the $\mathbf{x}_i$s such that $d_q(\mathbf{x}_i, \mathbf{x}) < h_k(\mathbf{x})$

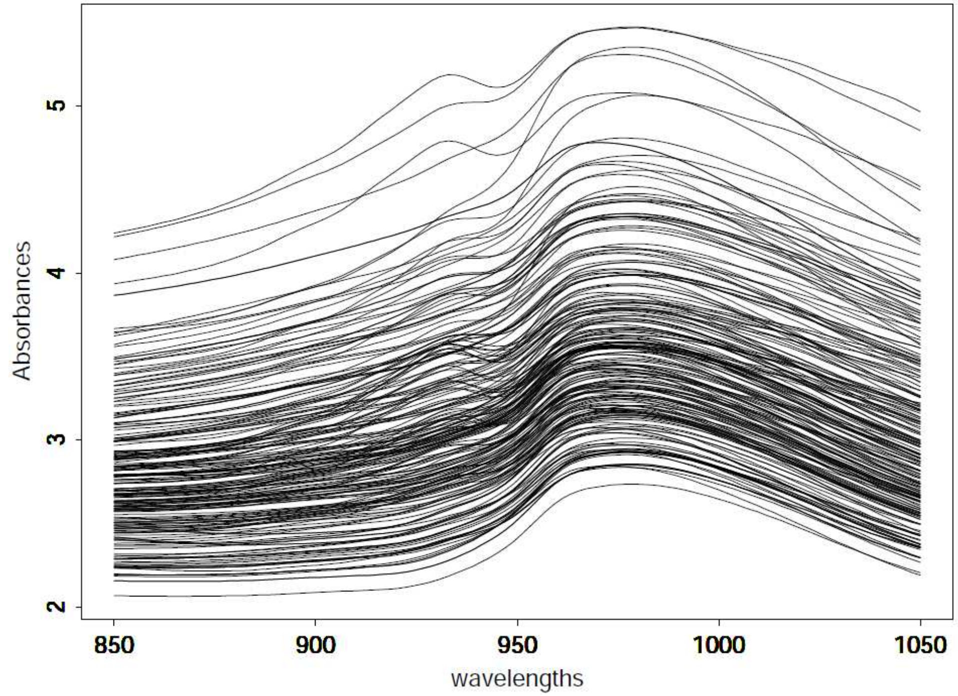
- `funopare.knn`: user selects k
- `funopare.knn.gcv`: global k using CV
- `funopare.knn.lcv`: local k using CV
  - the optimal number of neighbours can change from one curve to another

Functional conditional quantiles and modes

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

- `funopare.quantile.lcv`: $t_\alpha^{kNN}(\mathbf{x})$
- `funopare.mode.lcv`: $\theta^{kNN}(\mathbf{x})$

Spectrometric data



	Col 1	$\cdots$	Col j	$\cdots$	Col 100	Col 101
Row 1	$\chi_1(\lambda_1)$	$\cdots$	$\chi_1(\lambda_j)$	$\cdots$	$\chi_1(\lambda_{100})$	y_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Row i	$\chi_i(\lambda_1)$	$\cdots$	$\chi_i(\lambda_j)$	$\cdots$	$\chi_i(\lambda_{100})$	y_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Row 215	$\chi_{215}(\lambda_1)$	$\cdots$	$\chi_{215}(\lambda_j)$	$\cdots$	$\chi_{215}(\lambda_{100})$	y_{215}

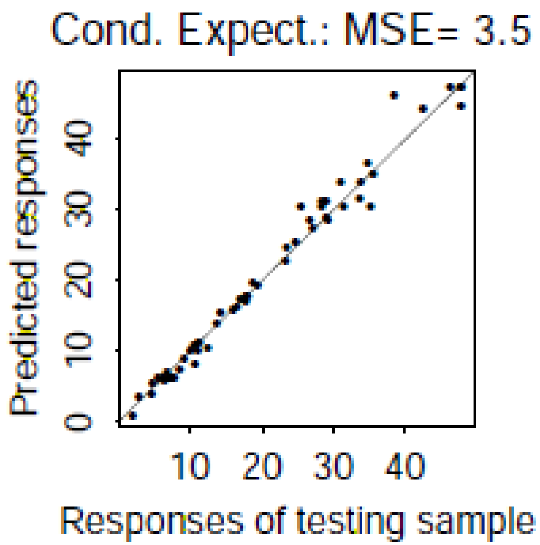
Functional prediction of fat content (1/3)

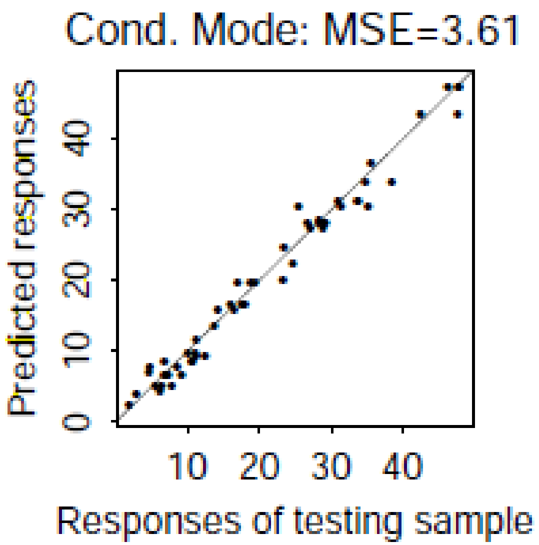
- learning set: first 160 units
- test set: last 55 units

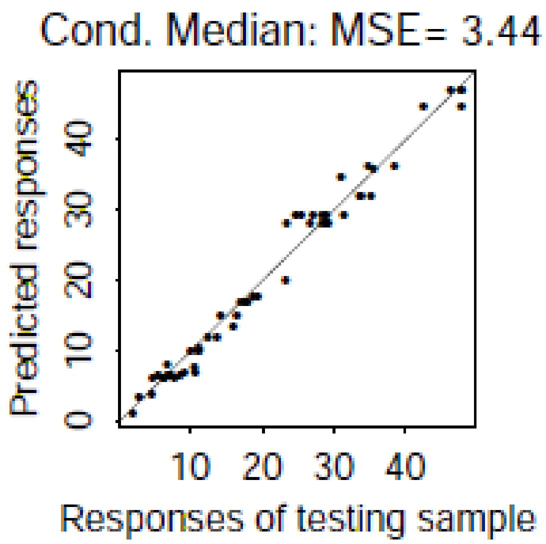
- performance measures:
  - distribution of square errors:
    $se_i = (y_i - \widehat{y_i})^2, i = 161, \cdots, 215$
  - empirical mean square errors: $MSE = \frac{1}{55} \sum_{i=161}^{215} se_i$

- three routines:
  - functional conditional expectation (regression)
  - functional conditional mode
  - functional conditional median
- all with k-NN bandwidth and locally optimized k

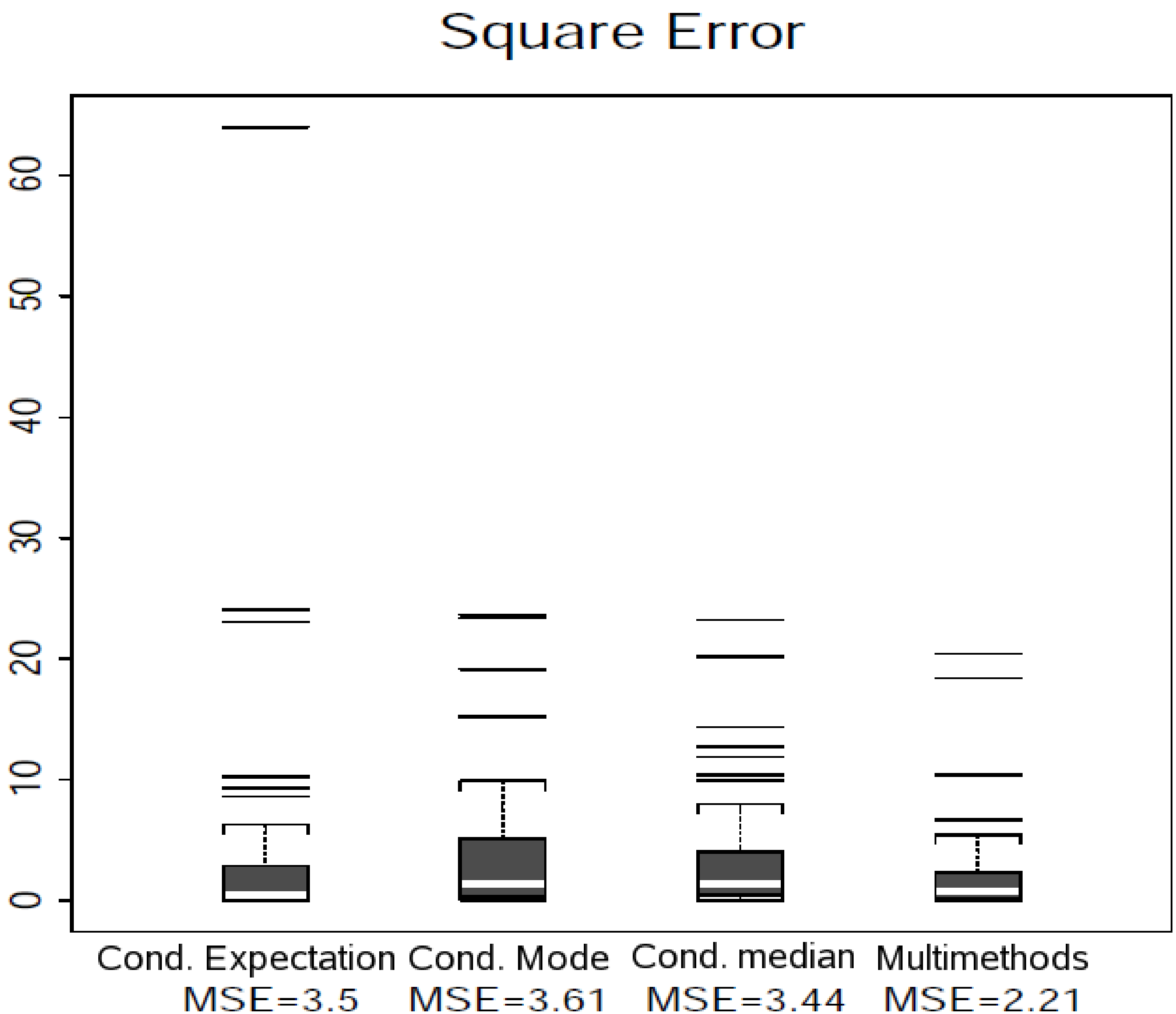
Functional prediction of fat content (2/3)







Functional prediction of fat content (3/3)



- *multimethods*: average of the three methods