Model detection for functional polynomial regression

Qihua Wang
Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences,
Joint work with Tao Zhang and Qingzhao Zhang.

2013-6

OUTLINES

Functional data: Background

- Background: Recent technological advances on collecting and storing data have put statisticians in front of situations where the datasets are of functional nature.
  - Food industry
  - Climate study

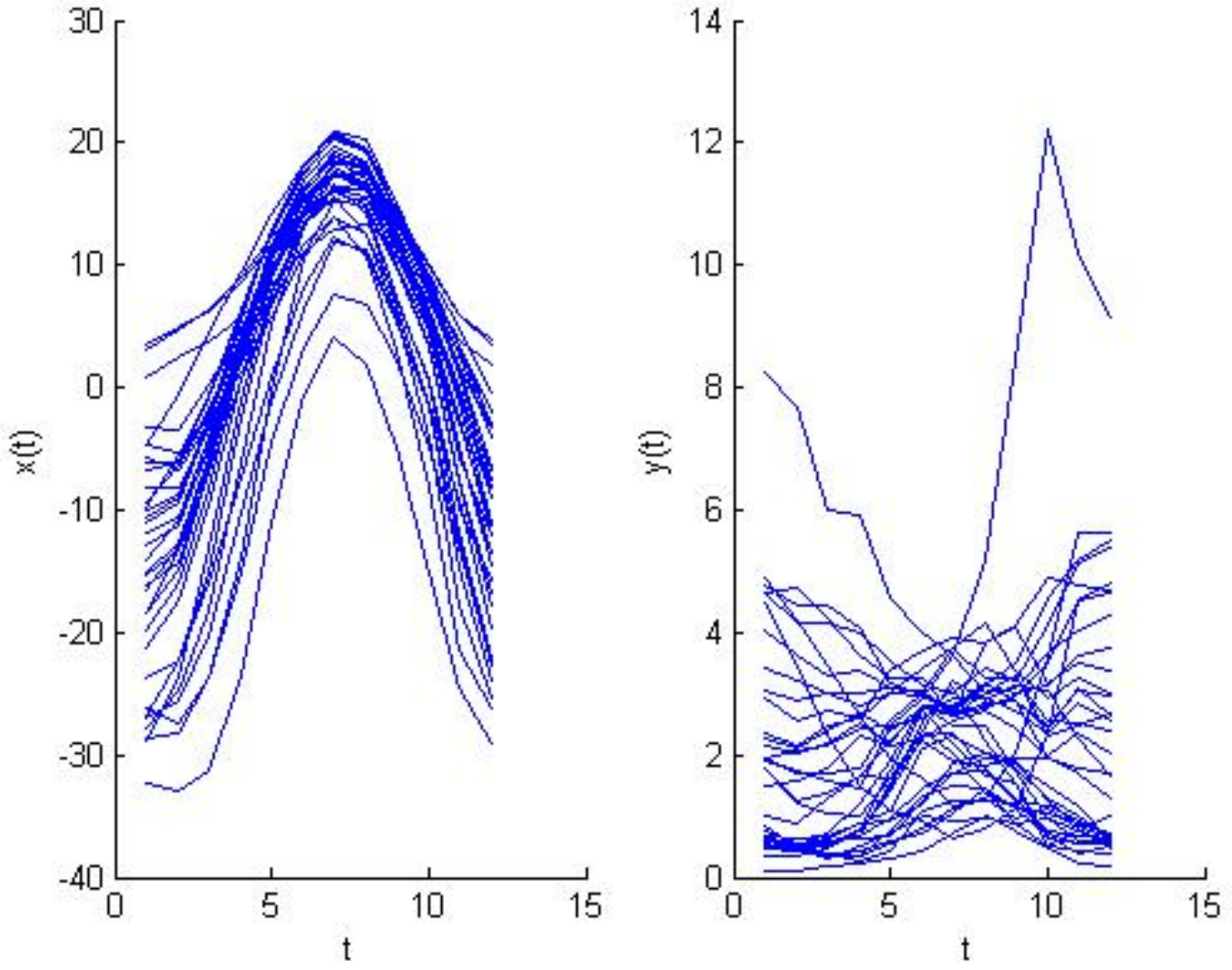

Figure: Canadian climate data, Left: averaged temperature curve each month; Right: the averaged precipitation curve of every month

Functional regression models: Background

Some functional regression models have been proposed, We now list mainly models as following:

- Functional linear models: See, e.g., Cardot et al., 2003; Shen and Faraway, 2004; Cai and Hall, 2006; Hall and Horowitz, 2007
- Functional nonparametric regression model: See, e.g., Ferraty and Vieu (2003), Ferraty and Vieu (2006) and Müller and Stadtmüller (2005)
- Single-index functional regression model: See, e.g., Ait-Saïdi et al. (2008) and Chen et al. (2011)
- Functional polynomial model: Yao and Müller (2010)

Functional polynomial model

- Model setups:
  1. Functional polynomial regression model with scale response Y and functional predictor X:

$$\begin{aligned} Y = \quad & \alpha + \int_T \gamma_1(s) X^c(s) ds + \cdots \\ & + \int_{T^p} \gamma_p(t_1, \cdots, t_p) X^c(t_1) \cdots X^c(t_p) dt_1 \cdots dt_p + \varepsilon, \end{aligned} \tag{2.1}$$

  where $E(\varepsilon|X) = 0$ and α is an intercept, $\gamma_j, 1 \leq j \leq p$, are the jth order regression parameter functions, respectively.
  $X^c(s) = X(s) - \mu_X(s)$ denotes the centered predictor process.

Existing work

- Yao and Müller (2010) mainly studied the estimating problem for the functional quadratic regression model which is a special case of model (2.1) with $p = 2$.

Our work

- Develop a method to identify which orders in functional polynomial are significant. i.e: detect the sets

$$A = \{j : \|\gamma_j\|_1 \neq 0, j \in \Omega\},$$

where $\Omega = \{0, 1, 2, \cdots, p\}$ and
$\|\gamma_j\|_1 = \{\int_T \gamma_j^2(t_1, \cdots, t_j)dt_1 \cdots dt_j\}^{1/2}, j = 1, \cdots, p.$。

- Estimate the function γ_j.

Parsimonious representations

- Pocesses X can be represented as:

$$X(s) = \mu_X(s) + \sum_{j=1}^{K} \eta_j \phi_j(s), \tag{2.2}$$

- The regression parameter functions in (2.1) can be represented as

$$\gamma_j(t_1, \cdots, t_j) = \sum_{k_1, \cdots, k_j=1}^{K} \gamma_{k_1, \cdots, k_j} \phi_{k_1}(t_1) \cdots \phi_{k_j}(t_j) \tag{2.3}$$

Parsimonious representations

- Applying the orthonormality property of the eigenfunctions, we have

$$\begin{aligned} Y = \quad & \alpha + \sum_{j=1}^{K} \gamma_j \eta_j + \sum_{k=1}^{K} \sum_{j=1}^{k} \gamma_{j,k} \eta_j \eta_k \\ & + \cdots + \sum_{j_1 \le \cdots \le j_p}^{K} \gamma_{j_1,\cdots,j_p} \eta_{j_1} \cdots \eta_{j_p} + \tilde{\varepsilon}. \end{aligned}$$

- Let

$$\begin{aligned} &\beta_0 = \alpha, \beta_1 = (\gamma_1, \cdots, \gamma_K)^\top, \\ &\beta_2 = (\gamma_{1,1}, \gamma_{2,1}, \gamma_{2,2} \cdots, \gamma_{K,K})^\top, \cdots, \\ &\beta_p = (\gamma_{1,1,\cdots,1}, \cdots, \gamma_{K,K,\cdots,K})^\top, \\ &U_0 = 1, U_1 = (\eta_1, \cdots, \eta_K)^\top, \\ &U_2 = (\eta_1\eta_1, \eta_2\eta_1, \eta_2\eta_2, \cdots, \eta_K\eta_K)^\top, \cdots, \\ &U_p = (\eta_1\eta_1 \cdots \eta_1, \cdots, \eta_K\eta_K \cdots \eta_K)^\top. \end{aligned}$$

Main idea

- Model (2.1) becomes

$$Y = \sum_{j=0}^{p} U_j^{\top} \beta_j + \varepsilon.$$

- $$\|\gamma_j\|_1 \neq 0 \Leftrightarrow \|\beta_j\| \neq 0$$
$$\|\gamma_j\|_1 = 0 \Leftrightarrow \|\beta_j\| = 0$$

  where $\|\cdot\|$ is the L_2 norm of a vector,

Method of detection

1. Estimate η_{ij}
   - Estimate the covariance function

$$\hat{G}_X(s,t) = \frac{1}{n}\sum_{i=1}^{n}\{X_i(s) - \bar{X}(s)\}\{X_i(t) - \bar{X}(t)\},$$

     where $\bar{X}(s) = \frac{1}{n}\sum_{i=1}^{n} X_i(s)$。
   - Estimate the eigenvalues and eigenfunctions

$$\hat{G}_X(s_1,s_2) = \sum_{k=1}^{\infty}\hat{\nu}_k\hat{\phi}_k(s_1)\hat{\phi}_k(s_2),$$

   - Estimate functional principal components scores

$$\hat{\eta}_{ij} = \sum_{m=2}^{N}[X_i(s_{im}) - \bar{X}(s_{im})]\hat{\phi}_j(s_{im})(s_{im} - s_{i,m-1}), \quad j = 1,2,\cdots,$$

Method of detection: cont's

2. Adaptive group lasso:

$$Q(\beta) = \frac{1}{2}\sum_{i=1}^{n}(Y_i - \sum_{j=0}^{p}\hat{U}_{ij}^{\top}\beta_j)^2 + n\sum_{j=0}^{p}c_j\lambda_j\|\beta_j\|, \qquad (2.6)$$

where $\|\cdot\|$ is the L_2 norm of a vector, λ_j are the tuning parameters and c_j $(j = 0, 1, \cdots, p)$ are constants for adjustment of the size of group j.

Estimate the function γ_l

- let

$$\begin{aligned}\hat{\beta} = \quad & (\hat{\alpha}, \hat{b}_1, \cdots, \hat{b}_K, \hat{b}_{1,1}, \hat{b}_{2,1}, \hat{b}_{2,2} \\ & \cdots, \hat{b}_{K,K}, \cdots, \hat{b}_{1,1,\cdots,1}, \cdots, \hat{b}_{K,K,\cdots,K}).\end{aligned}$$

-

$$\hat{\gamma}_l(t_1, \cdots, t_l) = \sum_{j_1 \leq \cdots \leq j_l}^{K} \hat{b}_{j_1,\cdots,j_l} \hat{\phi}_{j_1}(t_1) \cdots \hat{\phi}_{j_l}(t_l), 1 \leq l \leq p.$$

Tuning parameter selection

- Tuning parameter selection:

$$\lambda_j = \frac{\lambda}{||\tilde{\beta}_j||},$$

  where λ is a positive constant and $\tilde{\beta} = (\tilde{\beta}_0^\top, \cdots, \tilde{\beta}_p^\top)^\top$ is the unpenalized least squares estimator.
- λ can be selected by BIC, AIC and CV

Asyptotic Properties

- **Theorem 1**
  Under assumptions (a)-(e), we have

$$\|\hat{\gamma}_j - \gamma_j\| = o_p(1), j \in \Omega.$$

Let $\hat{A}_n = \{j : \|\hat{\gamma}_j\| \neq 0\}$

Theorem 2
Under assumptions (a)-(e), we have

$$P(\hat{A}_n = A) \to 1.$$

Simulation study: basics

1. 200 simulations
2. training sample: $n = 100$, $n = 200$ and $n = 400$;
   test sample: $m = 100$
3. $$X_i(s) = s + \sin(s) + \sum_{j=1}^{2} \eta_{ij}\phi_j(s),$$

   where $\phi_1(s) = -\sqrt{2}\cos(2\pi s), s \in [0, 1]$,
   $\phi_2(s) = \sqrt{2}\sin(2\pi s), s \in [0, 1]$, the corresponding eigenvalues were chosen as $\lambda_1 = 2$ and $\lambda_2 = 1$ and η_{ij} were generated from $N(0, \lambda_j)$.

Example 1: Functional quadratic models

- $$\begin{aligned} \textit{Model I} &: Y_i = 2 + 2\eta_{i1} + \eta_{i2} + e_i, \\ \textit{Model II} &: Y_i = 2 + \tfrac{1}{3}\eta_{i1}^2 + \eta_{i2}^2 + e_i, \\ \textit{Model III} &: Y_i = \tfrac{1}{2}\eta_{i1} + \tfrac{1}{4}\eta_{i1}\eta_{i2} + e_i, \end{aligned}$$

  where e_i were generated from $N(0, \sigma)$.
- Model *I* can be seen as a functional quadratic model without quadratic term; Model *II* a functional quadratic model without linear term; Model *III* a functional quadratic model without constant term.

Example 2: Functional cubic model

- Responses Y_i were generated from:

$$\begin{aligned} \textit{Model M} &: Y_i = 2 + \eta_{i1} + \eta_{i2}^2 + e_i, \\ \textit{Model V} &: Y_i = 2 + \eta_{i2}^2 + e_i, \\ \textit{Model VI} &: Y_i = 2 + 2\eta_{i1} + \eta_{i2} + e_i, \end{aligned}$$

  where e_i were simulated as $N(0, \sigma)$.

- Model *M* can be seen a functional cubic model without cubic term ; Model *V* a functional cubic model without linear term and cubic term; Model *VI* a functional cubic model with only constant and linear term.

Study of model detection

- We use the percentage of
  - ”C”:the true polynomial model is correctly identified
  - ”U”:at least an important order is ignored
  - ”O”: all the significant orders are detected while at least one spurious order is included in the selected model;

Table: Summary of model selection (MS) in Example 1.

Model	n	MS by BIC			MS by AIC			MS by CV		
		O	C	U	O	C	U	O	C	U
					$\sigma = 0.5$					
	100	0.010	0.970	0.020	0.070	0.930	0.000	0.195	0.805	0.000
I	200	0.000	1.000	0.000	0.040	0.960	0.000	0.140	0.860	0.000
	400	0.000	1.000	0.000	0.000	0.100	0.000	0.020	0.980	0.000
	100	0.145	0.855	0.000	0.185	0.815	0.000	0.175	0.825	0.000
II	200	0.075	0.925	0.000	0.100	0.900	0.000	0.100	0.900	0.000
	400	0.025	0.975	0.000	0.025	0.975	0.000	0.025	0.975	0.000
	100	0.290	0.655	0.055	0.305	0.655	0.040	0.290	0.665	0.040
III	200	0.135	0.860	0.005	0.135	0.860	0.005	0.135	0.860	0.005
	400	0.055	0.940	0.005	0.055	0.940	0.005	0.055	0.940	0.005
					$\sigma = 4$					
	100	0.030	0.970	0.000	0.175	0.825	0.000	0.300	0.700	0.000
I	200	0.015	0.985	0.000	0.165	0.835	0.000	0.210	0.790	0.000
	400	0.000	1.000	0.000	0.080	0.920	0.000	0.195	0.805	0.000
	100	0.140	0.860	0.000	0.330	0.670	0.000	0.375	0.625	0.000
II	200	0.120	0.880	0.000	0.375	0.625	0.000	0.400	0.600	0.000
	400	0.075	0.920	0.000	0.335	0.665	0.000	0.335	0.665	0.000
	100	0.110	0.255	0.635	0.400	0.375	0.225	0.255	0.535	0.210
III	200	0.070	0.565	0.365	0.475	0.470	0.055	0.270	0.650	0.080
	400	0.245	0.590	0.165	0.375	0.610	0.015	0.320	0.660	0.020

Table: Summary of model selection (MS) in Example 2.

Model	n	MS by BIC			MS by AIC			MS by CV		
		O	C	U	O	C	U	O	C	U
					$\sigma = 0.5$					
	100	0.040	0.910	0.050	0.185	0.765	0.050	0.225	0.735	0.040
IV	200	0.020	0.970	0.001	0.185	0.805	0.010	0.185	0.805	0.010
	400	0.000	0.995	0.005	0.145	0.850	0.005	0.315	0.680	0.005
	100	0.100	0.900	0.000	0.450	0.550	0.000	0.500	0.500	0.000
V	200	0.110	0.890	0.000	0.375	0.620	0.005	0.425	0.565	0.010
	400	0.040	0.955	0.005	0.390	0.605	0.005	0.430	0.565	0.005
	100	0.015	0.985	0.000	0.080	0.920	0.000	0.265	0.735	0.000
VI	200	0.000	1.000	0.000	0.100	0.900	0.000	0.270	0.730	0.000
	400	0.000	1.000	0.000	0.105	0.895	0.000	0.130	0.870	0.000
					$\sigma = 4$					
	100	0.365	0.575	0.000	0.475	0.475	0.050	0.580	0.370	0.050
IV	200	0.220	0.775	0.005	0.325	0.670	0.005	0.470	0.525	0.005
	400	0.115	0.880	0.005	0.260	0.735	0.005	0.430	0.565	0.005
	100	0.200	0.800	0.000	0.400	0.600	0.000	0.600	0.400	0.000
V	200	0.100	0.895	0.005	0.380	0.615	0.005	0.645	0.350	0.005
	400	0.035	0.950	0.015	0.280	0.715	0.005	0.505	0.490	0.005
	100	0.115	0.885	0.000	0.245	0.755	0.000	0.490	0.510	0.000
VI	200	0.030	0.970	0.000	0.225	0.775	0.000	0.410	0.590	0.000
	400	0.005	0.995	0.000	0.120	0.880	0.000	0.215	0.785	0.000

The result of model detection

- BIC tuning selection method does better than AIC and CV
- AIC and CV selectors tend to select more orders of polynomial.
- The percentage of correct detection under BIC tuning selector increases as the sample size increases.

Prediction

- For prediction, we compare the performance of
  - the proposed estimators based on three tuning selectors (denote as 'AIC', 'CV' and 'BIC', respectively)
  - the oracle estimators (based on the true model, denote as 'OE')
  - unpenalized estimators (based on the full model, denote as 'UE').
- the 25th, 50th and 75th percentiles of relative prediction error (RPE)

$$RPE_i = (Y_i - \hat{Y}_i)^2 / Y_i^2,$$

Table: The prediction results for model *I*

model	n	method	25		50		75	
$\sigma = 0.5$	100	*BIC*	0.1251	(0.3092)	0.3463	(0.7914)	1.5541	(3.4710)
		AIC	0.1256	(0.3118)	0.3465	(0.7925)	1.5582	(3.4821)
		CV	0.1260	(0.3124)	0.3465	(0.7933)	1.5512	(3.4660)
		OE	0.1248	(0.3068)	0.3476	(0.7864)	1.5490	(3.5500)
		UE	0.1256	(0.3115)	0.3516	(0.8074)	1.5747	(3.5317)
	200	*BIC*	0.0927	(0.2666)	0.2689	(0.7101)	1.1690	(3.0608)
		AIC	0.0927	(0.2667)	0.2689	(0.7101)	1.1692	(3.0607)
		CV	0.0928	(0.2667)	0.2680	(0.7074)	1.1622	(3.0340)
		OE	0.0925	(0.2644)	0.2694	(0.7089)	1.1878	(3.0919)
		UE	0.0915	(0.2630)	0.2697	(0.7105)	1.1753	(3.0753)
	400	*BIC*	0.0968	(0.2853)	0.2843	(0.7704)	1.1143	(2.8293)
		AIC	0.0968	(0.2853)	0.2843	(0.7704)	1.1143	(2.8293)
		CV	0.0972	(0.2864)	0.2843	(0.7705)	1.1105	(2.8205)
		OE	0.0971	(0.2863)	0.2857	(0.7706)	1.1137	(2.8224)
		UE	0.0973	(0.2858)	0.2836	(0.7665)	1.1177	(2.8304)
$\sigma = 4$	100	*BIC*	0.1434	(0.2928)	0.4507	(0.7206)	1.8187	(2.4220)
		AIC	0.1433	(0.2921)	0.4588	(0.7431)	1.8284	(2.4355)
		CV	0.1446	(0.2955)	0.4558	(0.7335)	1.7951	(2.3829)
		OE	0.1390	(0.2849)	0.4571	(0.7343)	1.8131	(2.4238)
		UE	0.1420	(0.2879)	0.4652	(0.7557)	1.8568	(2.4449)
	200	*BIC*	0.1157	(0.2509)	0.3869	(0.6741)	1.5314	(2.1307)
		AIC	0.1159	(0.2510)	0.3894	(0.6786)	1.5507	(2.1713)
		CV	0.1171	(0.2542)	0.3870	(0.6716)	1.5368	(2.1537)
		OE	0.1133	(0.2454)	0.3849	(0.6709)	1.5543	(2.1621)
		UE	0.1158	(0.2515)	0.3895	(0.6729)	1.5818	(2.1277)
	400	*BIC*	0.1181	(0.2679)	0.3945	(0.6862)	1.5096	(1.9878)
		AIC	0.1184	(0.2681)	0.3944	(0.6850)	1.5142	(1.9822)
		CV	0.1191	(0.2706)	0.3920	(0.6795)	1.4974	(1.9599)
		OE	0.1171	(0.2679)	0.3945	(0.6871)	1.5369	(2.0268)
		UE	0.1157	(0.2594)	0.3949	(0.6886)	1.5400	(2.0373)

Table: The prediction results for model V.

model	n	method	25		50		75	
$\sigma = 0.5$	100	*BIC*	0.0083	(0.0027)	0.0375	(0.0072)	0.1144	(0.0207)
		AIC	0.0084	(0.0023)	0.0384	(0.0064)	0.1138	(0.0210)
		CV	0.0086	(0.0030)	0.0388	(0.0071)	0.1156	(0.0216)
		OE	0.0082	(0.0034)	0.0359	(0.0083)	0.1136	(0.0269)
		UE	0.0091	(0.0027)	0.0397	(0.0069)	0.1212	(0.0250)
	200	*BIC*	0.0079	(0.0065)	0.0342	(0.0114)	0.1040	(0.0310)
		AIC	0.0080	(0.0065)	0.0346	(0.0113)	0.1041	(0.0298)
		CV	0.0080	(0.0067)	0.0347	(0.0113)	0.1045	(0.0306)
		OE	0.0078	(0.0071)	0.0342	(0.0115)	0.1038	(0.0298)
		UE	0.0083	(0.0071)	0.0355	(0.0114)	0.1064	(0.0296)
	400	*BIC*	0.0074	(0.0081)	0.0323	(0.0100)	0.0976	(0.0237)
		AIC	0.0074	(0.0081)	0.0327	(0.0105)	0.0987	(0.0240)
		CV	0.0073	(0.0077)	0.0327	(0.0104)	0.0982	(0.0239)
		OE	0.0073	(0.0087)	0.0325	(0.0098)	0.0983	(0.0244)
		UE	0.0074	(0.0074)	0.0329	(0.0100)	0.0996	(0.0244)
$\sigma = 4$	100	*BIC*	0.0520	(0.0231)	0.1917	(0.0320)	1.0023	(0.6024)
		AIC	0.0493	(0.0193)	0.1917	(0.0293)	1.0868	(0.7110)
		CV	0.0510	(0.0215)	0.1852	(0.0352)	1.0381	(0.7024)
		OE	0.0478	(0.0224)	0.1816	(0.0332)	1.1415	(0.7739)
		UE	0.0517	(0.0233)	0.1976	(0.0367)	1.2390	(0.9016)
	200	*BIC*	0.0487	(0.0180)	0.1820	(0.0416)	1.0081	(0.7938)
		AIC	0.0474	(0.0177)	0.1836	(0.0422)	1.0337	(0.8077)
		CV	0.0474	(0.0180)	0.1811	(0.0405)	1.0271	(0.8130)
		OE	0.0456	(0.0178)	0.1803	(0.0416)	1.0310	(0.8037)
		UE	0.0476	(0.0177)	0.1891	(0.0427)	1.0716	(0.8322)
	400	*BIC*	0.0465	(0.0238)	0.1771	(0.0354)	0.9895	(0.8334)
		AIC	0.0459	(0.0239)	0.1778	(0.0369)	0.9990	(0.8336)
		CV	0.0462	(0.0256)	0.1774	(0.0357)	0.9873	(0.8313)
		OE	0.0453	(0.0251)	0.1767	(0.0353)	1.0128	(0.8509)
		UE	0.0461	(0.0251)	0.1803	(0.0392)	1.0158	(0.8544)

The result of prediction

- our penalized method with BIC, AIC and CV tuning selection tools have better performance than unpenalized estimators
- the BIC and CV are comparable for prediction.
- the oracle estimator has the best overall prediction performance,

Application: Tecator Data

- Y: the percentage of fat.
  X(t): the second derivatives of the curves
- The aim is to predict the percentage of fat Y given the corresponding curve X(t).
- The total number of samples is 215; the first 160 spectra have been used to train the methods and sample 161-215 for testing.

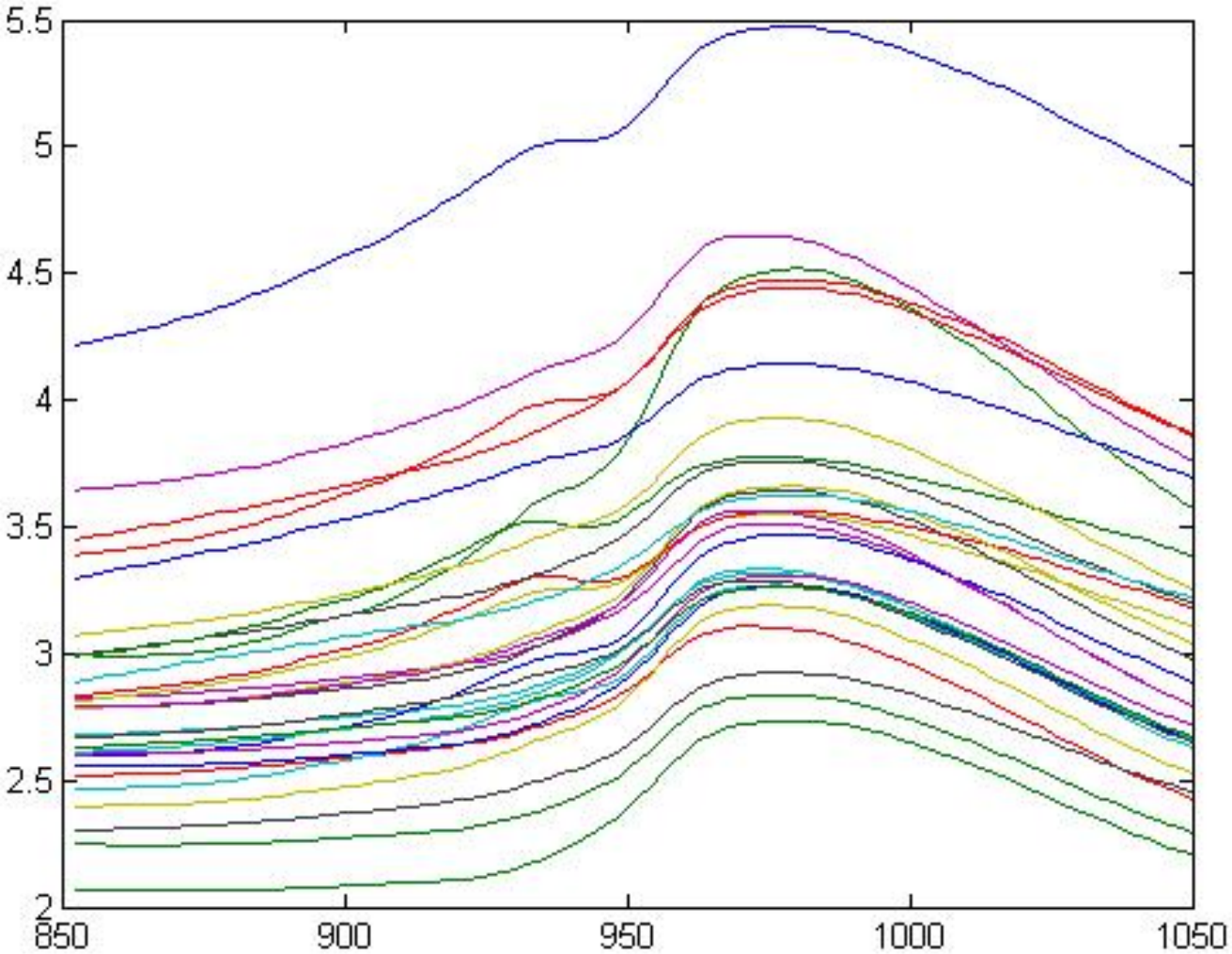

Figure: 普数据

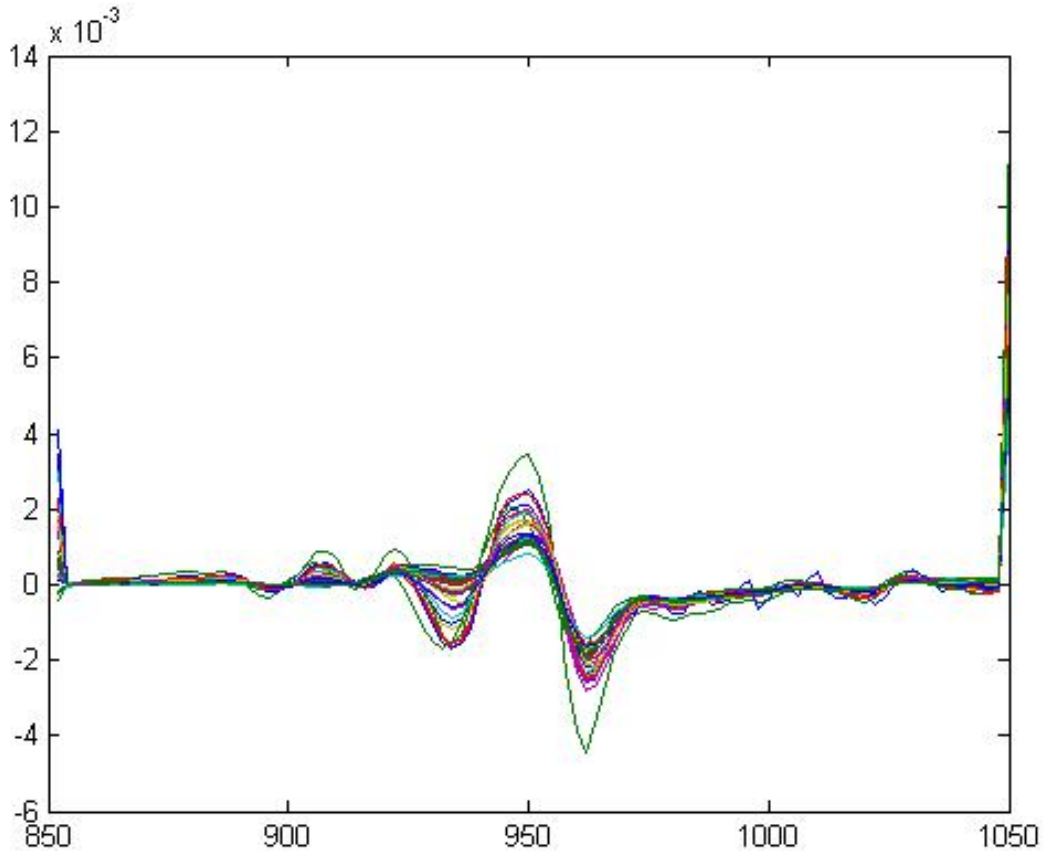

Figure: 相应的二阶导数.

Summary of model detection

- Penalized method with BIC, AIC and CV tuning selection tools select the functional quadratic models.
- This conclusion is consistency with that of Ferraty et al. (2012): the functional linear model can not reflect the relationship between response and covariates correctly.

Prediction

- For prediction, We compare the prediction performance of
  - the proposed estimator (including BIC,AIC and CV).
  - functional linear regression (denoted by ‘linear’)
  - unpenalized estimating method based on the full model (denoted by ‘UE’)
- the 25th, 50th and 75th percentiles of relative prediction error (RPE)

Table: Summary of *RPE* for different method

Method	25	50	75
BIC	0.0002	0.0026	0.0055
AIC	0.0002	0.0026	0.0055
CV	0.0003	0.0027	0.0055
UE	0.0003	0.0026	0.0059
Linear	0.0037	0.0165	0.0326

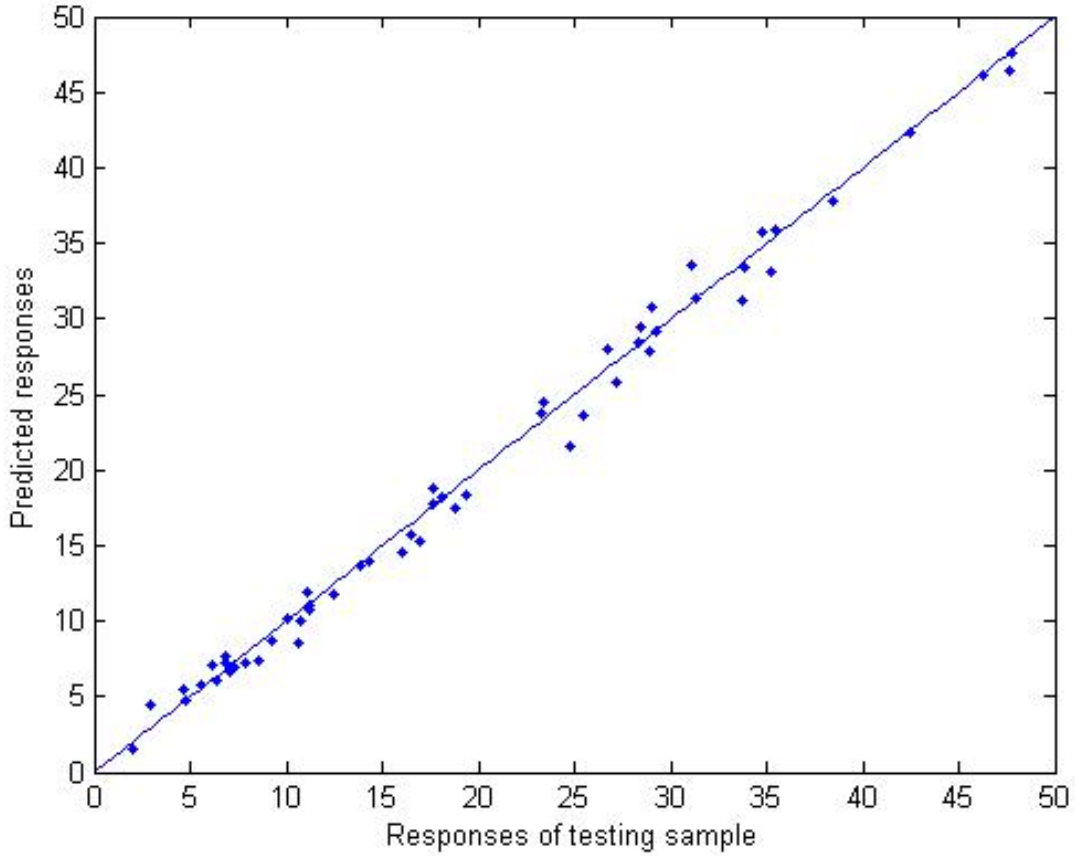

Figure: the scatter plot of the true value and predicted values in the test set

Prediction

- We also compare the prediction performance of
  - the proposed estimator (including BIC,AIC and CV).
  - different functional projection pursuit regression methods (denoted by FPPR- Step1(8 knots), FPPR - Step 1 and Step 2(15 knots), FPPR and NPM, respectively) due to Ferraty et al. (2012).
  - the stepwise algorithm method (denote *FH*) due to Ferraty and Hall(2010)
  - functional nonparametric regression method.
- Mean Square Error of Prediction (*MSEP*)

$$MSEP = \frac{1}{55} \sum_{i=161}^{215} (Y_i - \hat{Y}_i)^2,$$

Table: Summary of *MSEP* for different method

Method	*MSEP*
BIC	1.2442
AIC	1.2442
CV	1.2770
*FPPR – Step*1(8 *knots*)	3.2893
FPPR – Step 1 *and Step* 2(15 *knots*)	2.0368
FPPR and NPM	1.6473
FH	1.2000
Functional nonparametric regression	1.9000

Summary

- We develop a detection method to identify which orders in functional polynomial are significant.
- We proved the consistency of the estimator and the consistency of model selection.
- It is meaningful to extend current work to other data structures such as missing or censored data problem in the future.

Thanks very much!