Spline-Backfitted Kernel Smoothing Additive Models in Nonlinear Time Series

Lily Wang[1] Lijian Yang[2]

[1]Department of Statistics
University of Georgia
[2]Department of Statistics& Probability
Michigan State University

NRC, 2009

Outline

1. Motivation
    - Curse of Dimensionality
    - The Model
2. Methodology
    - Estimation Method
    - Asymptotic Results
3. Implementation and Simulation
    - Implementation
    - Simulation
4. Application
    - Data
    - Estimation and Forecast
5. Conclusions

Nonparametric stochastic model

- Nonparametric stochastic model for trend $m(\cdot)$ with time series $\{Y_t, \mathbf{X}_t\}_{t=1}^{n}$

$$Y_t = m(\mathbf{X}_t) + \sigma(\mathbf{X}_t)\varepsilon_t, \quad t = 1, ..., n$$

  - Predictor vector: $\mathbf{X}_t = \left(X_{t,1}, ..., X_{t,d}\right)$
  - Conditional mean function: $m(\mathbf{X}_t) = E(Y_t|\mathbf{X}_t)$
  - Conditional variance function: $\sigma^2(\mathbf{X}_t) = \mathrm{Var}(Y_t|\mathbf{X}_t)$
  - Random error: $E(\varepsilon_t|\mathbf{X}_t) = 0, \mathrm{Var}(\varepsilon_t|\mathbf{X}_t) = 1$

- Optimal convergence rate: $n^{-p/(2p+d)}$ if the pth order derivative of m is continuous ($m \in C^{(p)}[0,1]^d$), Stone (1982)

Curse of Dimensionality

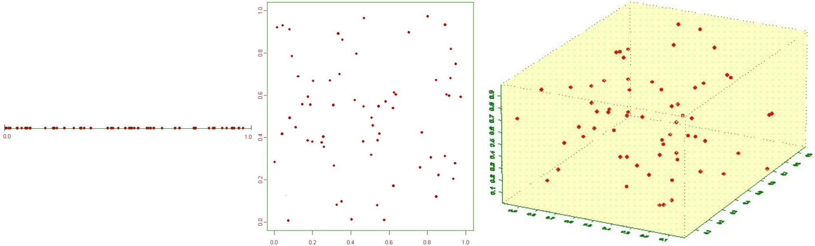

Figure: Curse of dimensionality ($\{\mathbf{X}_i\}_{i=1}^d$ generated from Uniform $[0,1]^d$)

Approaches for High Dimensional Data

- Dimension reduction
  - Sliced Inverse Regression: Li (1991)
  - Single-Index Model: Härdle *et al.* (1989, 1993)
  - Principal Hessian Direction: Cook (1998)
  - Central Mean Subspace: Cook and Li (2002)
- Function approximation
  - Projection Pursuit Regression: Friedman and Stuetzle (1981)
  - **Additive Model**: Hastie and Tibshirani (1990)

The Model

Additive model

$$Y_i = m(\mathbf{X}_i) + \sigma(\mathbf{X}_i)\varepsilon_i, \quad i = 1, \ldots, n$$

$$m(\mathbf{x}) = c + \sum_{\alpha=1}^{d} m_\alpha(x_\alpha), \; Em_\alpha(X_\alpha) \equiv 0, \; \alpha = 1, \ldots, d$$

- Optimal rate: $n^{-p/(2p+1)}$ for $m_\alpha \in C^{(p)}[0,1]$, Stone (1985).
- The “curse of dimensionality” is “avoided” by using additive models.
- In **nonlinear autoregression** data-analytical context, each predictor $X_{i\alpha}$ could be observed lagged values of Y_i, such as $X_{i\alpha} = Y_{i-\alpha}$.
- $\{X_\alpha\}_{\alpha=1}^{d}$ has density function $f_\alpha(\cdot)$ supported on $[0,1]$.

The Model

Additive model

$$Y_i = m\left(\mathbf{X}_i\right) + \sigma\left(\mathbf{X}_i\right)\varepsilon_i, \quad i = 1, ..., n$$

$$m\left(\mathbf{x}\right) = c + \sum_{\alpha=1}^{d} m_\alpha\left(x_\alpha\right), \; Em_\alpha\left(X_\alpha\right) \equiv 0, \; \alpha = 1, ..., d$$

- Optimal rate: $n^{-p/(2p+1)}$ for $m_\alpha \in C^{(p)}[0,1]$, Stone (1985).
- The “curse of dimensionality” is “avoided” by using additive models.
- In **nonlinear autoregression** data-analytical context, each predictor $X_{i\alpha}$ could be observed lagged values of Y_i, such as $X_{i\alpha} = Y_{i-\alpha}$.
- $\{X_\alpha\}_{\alpha=1}^{d}$ has density function $f_\alpha(\cdot)$ supported on $[0,1]$.

Estimation of the Components

- Marginal integration estimation
  Proposed by Linton & Nielsen (1995) for i.i.d data

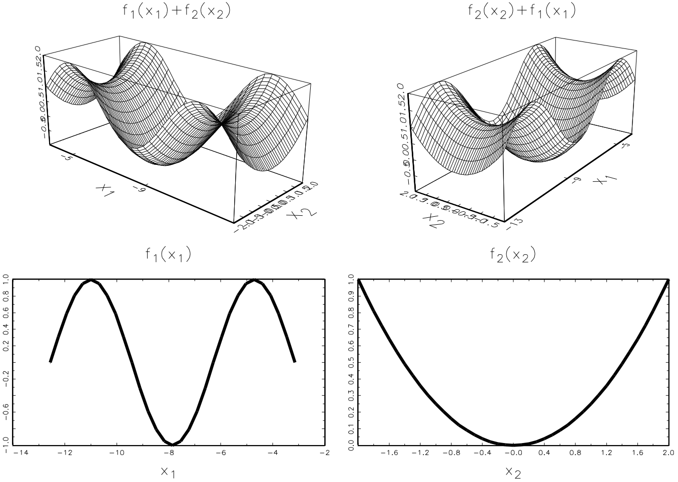

Figure: Illustration of marginal integration

Estimation of the Components

- Marginal integration estimation: Linton & Nielsen (1995)
- Backfitting estimation: Friedman & Stuetzle (1981)
- One step kernel backfitting based on marginal integration: Linton (1997)
- Polynomial spline estimation: Huang & Yang (2004)

Estimation of the Components

- Marginal integration estimation: Linton & Nielsen (1995)
- Backfitting estimation: Friedman & Stuetzle (1981)
- One step kernel backfitting based on marginal integration: Linton (1997)
- Polynomial spline estimation: Huang & Yang (2004)

Estimation of the Components

- Marginal integration estimation: Linton & Nielsen (1995)
- Backfitting estimation: Friedman & Stuetzle (1981)
- One step kernel backfitting based on marginal integration: Linton (1997)
- Polynomial spline estimation: Huang & Yang (2004)

Kernel Smoothing

- A simple nonparametric model

$$Y_i = m(X_i) + \sigma(X_i)\varepsilon_i$$

- Kernel estimator for $m(\cdot)$ with given data $\{X_i, Y_i\}_{i=1}^{n}$

$$\frac{\sum_{i=1}^{n} K_h\left(X_i - x\right) Y_i}{\sum_{i=1}^{n} K_h\left(X_i - x\right)}$$

- Kernel smoothing: a locally weighted average

$$K_h\left(u\right) = K\left(u/h\right)/h$$

where $h \sim n^{-1/5}$ is the bandwidth and K is the kernel function to determine how to assign the weights

Kernel Oracle Smoother

- If $\{m_\beta\}_{\beta\neq\alpha}$ is known by “oracle”, define kernel “oracle smoother”
$$\tilde{m}^*_\alpha(x_\alpha) = \frac{\sum_{i=1}^n K_h(X_{i,\alpha} - x_\alpha)\, Y_{i,\alpha}}{\sum_{i=1}^n K_h(X_{i,\alpha} - x_\alpha)},$$
where the “oracle response”
$Y_{i,\alpha} = Y_i - c - \sum_{\beta\neq\alpha} m_\beta(X_{i,\beta})$
- Asymptotic properties of the oracle smoother (Bosq, 1998)
  - $\sup_{x\in[h,1-h]} \left|\tilde{m}^*_\alpha(x) - m_\alpha(x)\right| = o_p\left(n^{-2/5}\log n\right)$
  - $\sqrt{nh}\left\{\tilde{m}^*_\alpha(x) - m_\alpha(x) - b_\alpha(x)h^2\right\} \xrightarrow{D} N\left\{0, v^2_\alpha(x)\right\}$

$$\begin{aligned} b_\alpha(x) &= \int u^2K(u)\,du\,\{m''_\alpha(x)f_\alpha(x)/2 + m'_\alpha(x)f'_\alpha(x)\}f^{-1}_\alpha(x) \\ v^2_\alpha(x) &= \int K^2(u)\,du E\left[\sigma^2(X_1,\ldots,X_d)\,|X_\alpha = x\right]f^{-1}_\alpha(x) \end{aligned}$$

Kernel Oracle Smoother

- If $\{m_\beta\}_{\beta \neq \alpha}$ is known by “oracle”, define kernel “oracle smoother”
$$\tilde{m}^*_\alpha(x_\alpha) = \frac{\sum_{i=1}^n K_h(X_{i,\alpha} - x_\alpha) Y_{i,\alpha}}{\sum_{i=1}^n K_h(X_{i,\alpha} - x_\alpha)},$$
where the “oracle response”
$Y_{i,\alpha} = Y_i - c - \sum_{\beta \neq \alpha} m_\beta(X_{i,\beta})$

- Asymptotic properties of the oracle smoother (Bosq, 1998)
  - $\sup_{x \in [h,1-h]} |\tilde{m}^*_\alpha(x) - m_\alpha(x)| = o_p(n^{-2/5} \log n)$
  - $\sqrt{nh}\{\tilde{m}^*_\alpha(x) - m_\alpha(x) - b_\alpha(x) h^2\} \xrightarrow{D} N\{0, v^2_\alpha(x)\}$

$$\begin{aligned} b_\alpha(x) &= \int u^2 K(u)\,du \{m''_\alpha(x) f_\alpha(x)/2 + m'_\alpha(x) f'_\alpha(x)\} f^{-1}_\alpha(x) \\ v^2_\alpha(x) &= \int K^2(u)\,du E\left[\sigma^2(X_1, \dots, X_d) \middle| X_\alpha = x\right] f^{-1}_\alpha(x) \end{aligned}$$

Kernel Oracle Estimation

- One step kernel backfitting based on marginal integration, Linton (1997)

  $$\hat{m}_{L,\alpha}(x_\alpha) = \frac{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right) \hat{Y}_{L,i,\alpha}}{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right)},$$

  where the pseudo response $\hat{Y}_{L,i,\alpha} = Y_i - \hat{c} - \sum_{\beta \neq \alpha} \hat{m}_{L,\beta}\left(X_{i,\beta}\right)$ and $\hat{m}_{L,\beta}(\cdot)$ is the marginal integration estimator, $\beta \neq \alpha$, $\hat{c} = \overline{Y}_n$

- $\left|\hat{m}_{L,\alpha}(x) - \tilde{m}_\alpha^*(x)\right| = o_p\left(n^{-2/5}\right)$, $x \in [h, 1-h]$ (pointwise oracle)
- It is computationally intensive: $O\left(n^3\right)$

Kernel Oracle Estimation

- One step kernel backfitting based on marginal integration, Linton (1997)

  $$\hat{m}_{L,\alpha}\left(x_\alpha\right) = \frac{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right) \hat{Y}_{L,i,\alpha}}{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right)},$$

  where the pseudo response $\hat{Y}_{L,i,\alpha} = Y_i - \hat{c} - \sum_{\beta\neq\alpha} \hat{m}_{L,\beta}\left(X_{i,\beta}\right)$ and $\hat{m}_{L,\beta}\left(\cdot\right)$ is the marginal integration estimator, $\beta \neq \alpha$, $\hat{c} = \overline{Y}_n$

- $\left|\hat{m}_{L,\alpha}\left(x\right) - \tilde{m}_{\alpha}^{*}\left(x\right)\right| = o_p\left(n^{-2/5}\right)$, $x \in [h, 1-h]$ (pointwise oracle)
- It is computationally intensive: $O\left(n^3\right)$

Kernel Oracle Estimation

☛ One step kernel backfitting based on marginal integration, Linton (1997)

$$\hat{m}_{L,\alpha}(x_\alpha) = \frac{\sum_{i=1}^{n} K_h(X_{i,\alpha} - x_\alpha)\, \hat{Y}_{L,i,\alpha}}{\sum_{i=1}^{n} K_h(X_{i,\alpha} - x_\alpha)},$$

where the pseudo response $\hat{Y}_{L,i,\alpha} = Y_i - \hat{c} - \sum_{\beta \neq \alpha} \hat{m}_{L,\beta}(X_{i,\beta})$ and $\hat{m}_{L,\beta}(\cdot)$ is the marginal integration estimator, $\beta \neq \alpha$, $\hat{c} = \overline{Y}_n$

- $\left|\hat{m}_{L,\alpha}(x) - \tilde{m}_\alpha^*(x)\right| = o_p\left(n^{-2/5}\right)$, $x \in [h, 1-h]$ (pointwise oracle)
- It is computationally intensive: $O\left(n^3\right)$

Constant Spline Smoothing

- Equally spaced interior knots $\{t_J\}_{J=1}^{N}$ $(N \sim n^{2/5} \log n)$
  $$t_0 = 0 < t_1 < \cdots < t_N < 1 = t_{N+1}$$
  where the subinterval width $H = 1/\left(N+1\right)$
- Constant spline space G with basis
  $$\left\{1, I_{J,\alpha}\left(x_\alpha\right) = I_{[t_J, t_{J+1})}\left(x_\alpha\right), 1 \le \alpha \le d, 1 \le J \le N\right\}$$
- Constant spline estimator of m is
  $$\hat{m}(x_1, ..., x_d) = \hat{\lambda}_0 + \sum_{\alpha=1}^{d} \sum_{J=1}^{N} \hat{\lambda}_{J,\alpha} I_{J,\alpha}\left(x_\alpha\right)$$

$$\left\{\hat{\lambda}_0, \hat{\lambda}_{1,1}, ..., \hat{\lambda}_{N,d}\right\}^T = \arg\min_{R^{dN+1}} \sum_{i=1}^{n} \left\{Y_i - \lambda_0 - \sum_{\alpha=1}^{d} \sum_{J=1}^{N} \lambda_{J,\alpha} I_{J,\alpha}\left(X_{i,\alpha}\right)\right\}^2$$

Pros and Cons of Kernel/Spline Smoothing

- Kernel smoothing
  1. Good asymptotic properties
  2. Computationally intensive
- Polynomial spline smoothing
  1. Computationally expedient
  2. Simple implementation
  3. Explicit formula
  4. Asymptotic distribution unavailable

Pros and Cons of Kernel/Spline Smoothing

- Kernel smoothing
  1. Good asymptotic properties
  2. Computationally intensive

- Polynomial spline smoothing
  1. Computationally expedient
  2. Simple implementation
  3. Explicit formula
  4. Asymptotic distribution unavailable

Figure: Spline and kernel smoothing

Spline-Backfitted Kernel (SPBK) Estimation

— Wang and Yang (2007)

Stage 1: Spline Smoothing

Pre-estimate $\{m_\alpha(x_\alpha)\}_{\alpha=1}^{d}$ by its pilot estimator $\{\hat{m}_\alpha(x_\alpha)\}_{\alpha=1}^{d}$ through an under-smoothed constant spline procedure. The empirically centered pilot estimator is

$$\hat{m}_\alpha\left(x_\alpha\right) \hat{=} \sum_{J=1}^{N} \hat{\lambda}_{J,\alpha} I_{J,\alpha}\left(x_\alpha\right) - n^{-1} \sum_{i=1}^{n} \sum_{J=1}^{N} \hat{\lambda}_{J,\alpha} I_{J,\alpha}\left(X_{i,\alpha}\right)$$ (Pilot estimator)

Stage 2: Kernel Smoothing

Construct the pseudo-response $\hat{Y}_{i,\alpha} = Y_i - \hat{c} - \sum_{\beta \neq \alpha} \hat{m}_\beta\left(X_{i,\beta}\right)$ and approximate $m_\alpha(x_\alpha)$ by the kernel estimator based on $\left\{\hat{Y}_{i,\alpha}, X_{i,\alpha}\right\}_{i=1}^{n}$,

$$\hat{m}_\alpha^*\left(x_\alpha\right) = \frac{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right) \hat{Y}_{i,\alpha}}{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right)}$$ (SPBK estimator)

Spline-Backfitted Kernel (SPBK) Estimation

— Wang and Yang (2007)

Stage 1: Spline Smoothing

Pre-estimate $\{m_\alpha(x_\alpha)\}_{\alpha=1}^d$ by its pilot estimator $\{\hat{m}_\alpha(x_\alpha)\}_{\alpha=1}^d$ through an under-smoothed constant spline procedure. The empirically centered pilot estimator is

$$\hat{m}_\alpha\left(x_\alpha\right) \hat{=} \sum_{J=1}^{N} \hat{\lambda}_{J,\alpha} I_{J,\alpha}\left(x_\alpha\right) - n^{-1} \sum_{i=1}^{n} \sum_{J=1}^{N} \hat{\lambda}_{J,\alpha} I_{J,\alpha}\left(X_{i,\alpha}\right) \quad \text{(Pilot estimator)}$$

Stage 2: Kernel Smoothing

Construct the pseudo-response $\hat{Y}_{i,\alpha} = Y_i - \hat{c} - \sum_{\beta \neq \alpha} \hat{m}_\beta\left(X_{i,\beta}\right)$ and approximate $m_\alpha(x_\alpha)$ by the kernel estimator based on $\left\{\hat{Y}_{i,\alpha}, X_{i,\alpha}\right\}_{i=1}^{n}$,

$$\hat{m}_\alpha^*\left(x_\alpha\right) = \frac{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right) \hat{Y}_{i,\alpha}}{\sum_{i=1}^{n} K_h\left(X_{i,\alpha} - x_\alpha\right)} \quad \text{(SPBK estimator)}$$

Assumptions

- (A1) $m_\alpha(\cdot) \in C^{(2)}[0,1]$, $\{m_\beta\}_{\beta \neq \alpha}$ is Lipschitz continuous on $[0,1]$
- (A2) The strong mixing coefficient of order k, $\alpha(k) \leq K_0 e^{-\lambda_0 k}$

$$\alpha(k) = \sup_{B \in \sigma\{X_s, Y_s, s \leq t\}, C \in \sigma\{X_s, Y_s, s \geq t+k\}} |P(B \cap C) - P(B)P(C)|$$

- (A3) Standard deviation $\sigma(\mathbf{x})$ is continuous and positive on $[0,1]^d$
- (A4) Density $f(\mathbf{x})$ *of* $\mathbf{X}$ is continuous and positive on $[0,1]^d$
- (A5) Number of knots for pilot **spline** estimation $N = N_n \sim n^{2/5} \log(n)$
- (A6) **Kernel** function K is a pdf, symmetric, Lipschitz continuous and supported on $[-1,1]$ with bandwidth $h = h_n \sim n^{-1/5}$

Main Results for SPBK Estimator

For stationary sequence $\{Y_i, \mathbf{X}_i\}_{i=1}^n$, under (A1)-(A6),

Theorem (Uniformly Consistency)

$$\sup_{x\in[0,1]} \left|\hat{m}^*_\alpha(x) - \tilde{m}^*_\alpha(x)\right| = o_p\left(n^{-2/5}\right)$$

Theorem (Normality)

- $\sqrt{nh}\left\{\hat{m}^*_\alpha(x) - m_\alpha(x) - b_\alpha(x)h^2\right\} \xrightarrow{D} N\left\{0, v^2_\alpha(x)\right\}, x \in [0,1]$

$$\begin{aligned} b_\alpha(x) &= \int u^2K(u)du\left\{m''_\alpha(x)f_\alpha(x)/2 + m'_\alpha(x)f'_\alpha(x)\right\}f^{-1}_\alpha(x) \\ v^2_\alpha(x) &= \int K^2(u)duE\left\{\sigma^2(X_1, ..., X_d)\,|X_\alpha = x\right\}f^{-1}_\alpha(x) \end{aligned}$$

- $\sqrt{nh}\left\{\hat{m}^*(\mathbf{x}) - m(\mathbf{x}) - b(\mathbf{x})h^2\right\} \xrightarrow{D} N\left\{0, v^2(\mathbf{x})\right\}$

$$\hat{m}^*(\mathbf{x}) = \hat{c} + \sum_{\alpha=1}^{d}\hat{m}^*_\alpha(x_\alpha), b(\mathbf{x}) = \sum_{\alpha=1}^{d} b_\alpha(x_\alpha), v^2(\mathbf{x}) = \sum_{\alpha=1}^{d} v^2_\alpha(x_\alpha)$$

Main Results for SPBK Estimator

For stationary sequence $\{Y_i, \mathbf{X}_i\}_{i=1}^n$, under (A1)-(A6),

Theorem (Uniformly Consistency)

$$\sup_{x\in[0,1]} \left|\hat{m}_\alpha^*(x) - \tilde{m}_\alpha^*(x)\right| = o_p\left(n^{-2/5}\right)$$

Theorem (Normality)

- $\sqrt{nh}\left\{\hat{m}_\alpha^*(x) - m_\alpha(x) - b_\alpha(x)h^2\right\} \xrightarrow{D} N\left\{0, v_\alpha^2(x)\right\}, x \in [0,1]$

$$\begin{aligned} b_\alpha(x) &= \int u^2 K(u)\,du \left\{m_\alpha''(x) f_\alpha(x)/2 + m_\alpha'(x) f_\alpha'(x)\right\} f_\alpha^{-1}(x) \\ v_\alpha^2(x) &= \int K^2(u)\,du E\left\{\sigma^2(X_1, ..., X_d) \,|X_\alpha = x\right\} f_\alpha^{-1}(x) \end{aligned}$$

- $\sqrt{nh}\left\{\hat{m}^*(\mathbf{x}) - m(\mathbf{x}) - b(\mathbf{x})h^2\right\} \xrightarrow{D} N\left\{0, v^2(\mathbf{x})\right\}$

$$\hat{m}^*(\mathbf{x}) = \hat{c} + \sum_{\alpha=1}^{d} \hat{m}_\alpha^*(x_\alpha), b(\mathbf{x}) = \sum_{\alpha=1}^{d} b_\alpha(x_\alpha), v^2(\mathbf{x}) = \sum_{\alpha=1}^{d} v_\alpha^2(x_\alpha)$$

Implementation

Stage 1

- Number of interior knots of the spline smoother

$$N_n = \min\left(c\left[n^{2/5}\log n\right]+1, \left[(n/2-1)\,d^{-1}\right]\right),$$

where c is the tuning parameter

Stage 2

- Quartic kernel

$$K(u)=\frac{15}{16}\left(1-u^2\right)^2 I_{[-1,1]}(u)$$

- Rule-of-thumb bandwidth

$$h_{\text{rot}} = \left[\frac{\int K^2(u)\,du\cdot\hat{\sigma}^2}{\left\{\int u^2K(u)\,du\right\}^2\cdot\sum_{i=1}^n\left\{\hat{m}''_\alpha(X_{i,\alpha})\right\}^2}\right]^{1/5}$$

Implementation

Stage 1

- Number of interior knots of the spline smoother

$$N_n = \min\left(c\left[n^{2/5}\log n\right]+1, \left[(n/2-1)\,d^{-1}\right]\right),$$

where c is the tuning parameter

Stage 2

- Quartic kernel

$$K(u) = \frac{15}{16}\left(1-u^2\right)^2 I_{[-1,1]}(u)$$

- Rule-of-thumb bandwidth

$$h_{\text{rot}} = \left[\frac{\int K^2(u)\,du \cdot \hat{\sigma}^2}{\left\{\int u^2 K(u)\,du\right\}^2 \cdot \sum_{i=1}^{n}\left\{\hat{m}''_{\alpha}\left(X_{i,\alpha}\right)\right\}^2}\right]^{1/5}$$

Simulation

NAAR(3) Model

Time series $\{Y_t\}_{t=-1999}^{n+3}$ is generated from a nonlinear additive AR(3) model

$$\begin{aligned} Y_t &= 1.5\sin\left(\frac{\pi}{2}Y_{t-2}\right) - 1.0\sin\left(\frac{\pi}{2}Y_{t-3}\right) + \sigma_0\varepsilon_t \\ &\equiv m\left(\mathbf{X}_t\right) + \sigma_0\varepsilon_t \equiv c + \sum_{\alpha=1}^{3} m_\alpha\left(X_{t,\alpha}\right) + \sigma_0\varepsilon_t \end{aligned}$$

where the predictor vector $\mathbf{X}_t = \{Y_{t-1}, Y_{t-2}, Y_{t-3}\}$

- $\{\varepsilon_t\}_{t=-1996}^{n+3}$: i.i.d. standard normal errors, $\sigma_0 = 0.5, 1.0$
- The first 2000 obs are discarded to make $\{Y_t, \mathbf{X}_t\}_{t=4}^{n+3}$ stationary

Simulation

NAAR(3) Model

Time series $\{Y_t\}_{t=-1999}^{n+3}$ is generated from a nonlinear additive AR(3) model

$$\begin{aligned} Y_t &= 1.5\sin\left(\frac{\pi}{2}Y_{t-2}\right) - 1.0\sin\left(\frac{\pi}{2}Y_{t-3}\right) + \sigma_0\varepsilon_t \\ &\equiv m\left(\mathbf{X}_t\right) + \sigma_0\varepsilon_t \equiv c + \sum_{\alpha=1}^{3} m_\alpha\left(X_{t,\alpha}\right) + \sigma_0\varepsilon_t \end{aligned}$$

where the predictor vector $\mathbf{X}_t = \{Y_{t-1}, Y_{t-2}, Y_{t-3}\}$

- $\{\varepsilon_t\}_{t=-1996}^{n+3}$: i.i.d. standard normal errors, $\sigma_0 = 0.5, 1.0$
- The first 2000 obs are discarded to make $\{Y_t, \mathbf{X}_t\}_{t=4}^{n+3}$ stationary

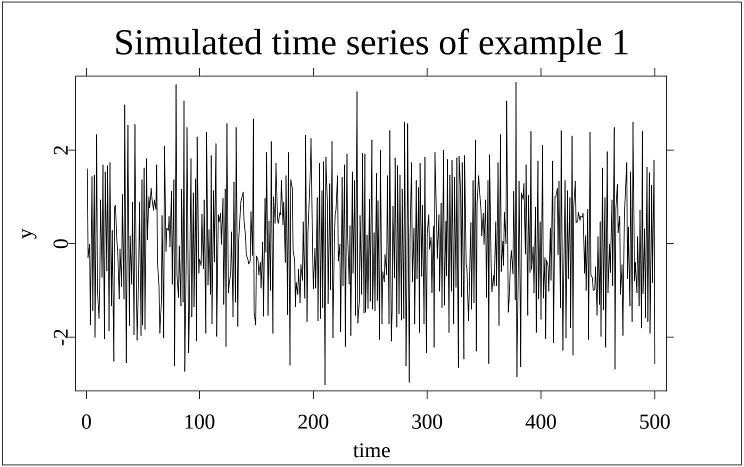

Figure: Plot of the simulated time series ($n = 500$)

Component Functions to be Estimated

1. $m_1\left(x_1\right) \equiv 0$
2. $m_2\left(x_2\right) \equiv 1.5 \sin\left(\frac{\pi}{2} x_2\right) - E\left[1.5 \sin\left(\frac{\pi}{2} Y_t\right)\right]$
3. $m_3\left(x_3\right) \equiv -1.0 \sin\left(\frac{\pi}{2} x_3\right) - E\left[-1.0 \sin\left(\frac{\pi}{2} Y_t\right)\right]$

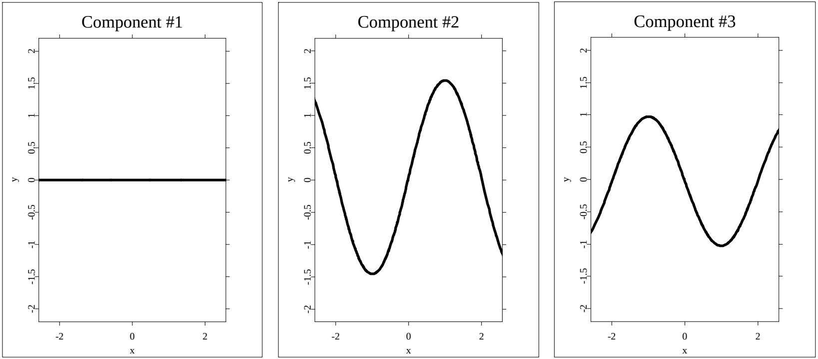

Figure: Plots of the true components $m_\alpha(x_\alpha)$

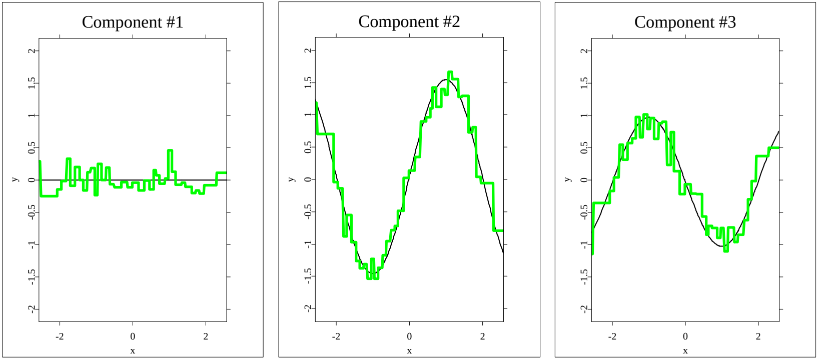

Figure: The first stage pilot estimates (green) of $m_\alpha(x_\alpha)$ (black)

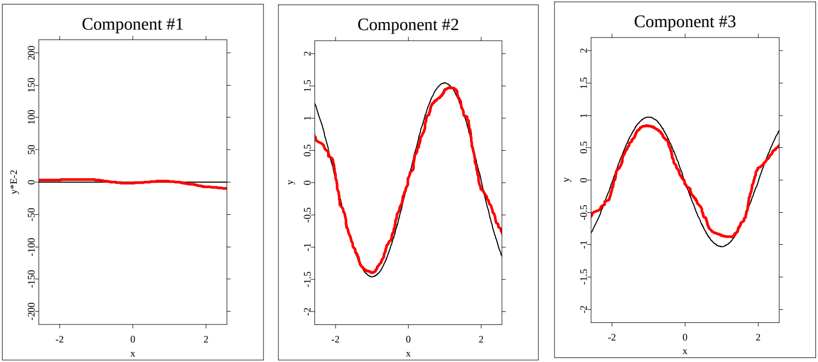

Figure: The SPBK estimates (red) of $m_\alpha(x_\alpha)$ (black)

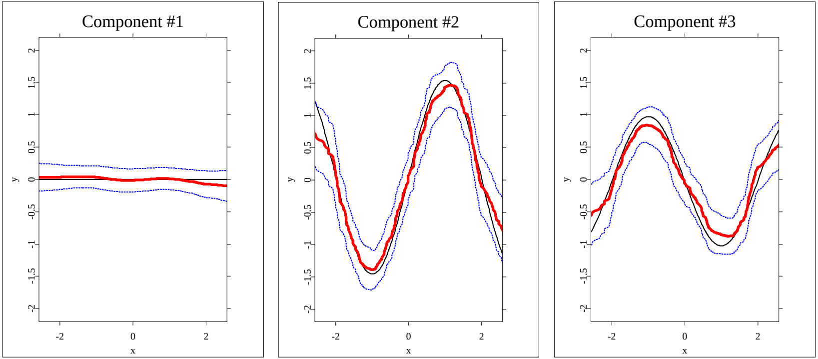

Figure: Plots of the SPBK (red) and 95% pointwise confidence intervals (upper and lower blue) of $m_\alpha(x_\alpha)$ (black)

Table 1: Average squared errors based on 100 replications

σ_0	n	c	component #1		component #2		component #3	
			1st step	2nd step	1st step	2nd step	1st step	2nd step
0.5	100	0.5	0.1231	0.0461	0.1476	0.0645	0.1254	0.0681
		1.0	0.1278	0.0520	0.1404	0.0690	0.1318	0.0726
	200	0.5	0.0539	0.0125	0.0616	0.0275	0.0577	0.0252
		1.0	0.0841	0.0144	0.0839	0.0290	0.0848	0.0285
	500	0.5	0.0263	0.0031	0.0306	0.0107	0.0278	0.0102
		1.0	0.0595	0.0044	0.0578	0.0115	0.0605	0.0119
	1000	0.5	0.0169	0.0015	0.0210	0.0053	0.0178	0.0054
		1.0	0.0364	0.0018	0.0367	0.0054	0.0375	0.0059
1.0	100	0.5	0.3008	0.0587	0.3298	0.1427	0.3236	0.1393
		1.0	0.3088	0.0586	0.3369	0.1364	0.3062	0.1316
	200	0.5	0.1742	0.0256	0.1783	0.0802	0.1892	0.0701
		1.0	0.2899	0.0328	0.2830	0.0824	0.3043	0.0721
	500	0.5	0.0924	0.0065	0.1124	0.0421	0.1004	0.0345
		1.0	0.2299	0.0078	0.2305	0.0458	0.2314	0.0362
	1000	0.5	0.0616	0.0033	0.0637	0.0270	0.0646	0.0224
		1.0	0.1460	0.0034	0.1433	0.0275	0.1429	0.0219

Table 2: Computing time

Sample Size	**n** = **100**	**n** = **200**	**n** = **400**	**n** = **1000**
Marginal Integration	10	76	628	10728
SPBK	.7	.9	1.2	4.5

- Time is measured by seconds per replication
- PC Configuration: CPU (Pentium IV 1.86GHz), memory (1 GB)
- Software: XploRe

US Unemployment Rate

Data from US Bureau of Labor Statistics

Non-seasonally adjusted quarterly series of US unemployment rate from the first quarter of 1948 to the first quarter of 2003, which covers unemployed people of 16 years old and older of all ethnic origins, races and sexes.

- ☞ $\{R_t\}_{t=1}^{221}$: Non-seasonally adjusted quarterly unemployment rate series
- ☞ Goal : Forecast the future unemployment rate
- ☞ Method: Use the first $221 - k$ obs for modelling to forecast the last k obs, $k = 10, 20, 30, 40$

US Unemployment Rate

Data from US Bureau of Labor Statistics

Non-seasonally adjusted quarterly series of US unemployment rate from the first quarter of 1948 to the first quarter of 2003, which covers unemployed people of 16 years old and older of all ethnic origins, races and sexes.

- $\{R_t\}_{t=1}^{221}$: Non-seasonally adjusted quarterly unemployment rate series
- Goal : Forecast the future unemployment rate
- Method: Use the first $221 - k$ obs for modelling to forecast the last k obs, $k = 10, 20, 30, 40$

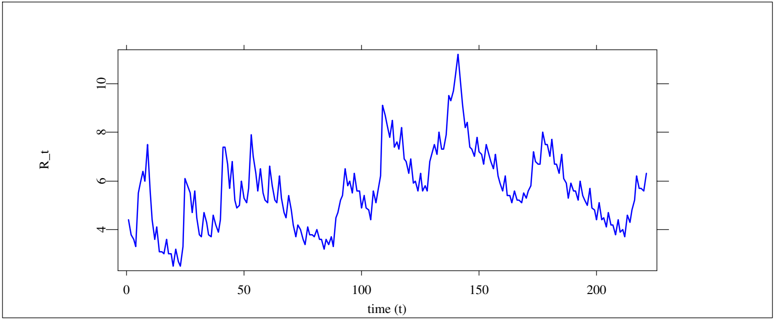

Figure: Non-seasonally adjusted quarterly series of US unemployment rate

US Unemployment Rate (Cont.)

- $\{R_t\}_{t=1}^{221}$ is NOT stationary
- Deseasonalize: Let $Y_t =\triangle_4 R_t = R_{t+4} - R_t$
- $\{Y_t\}_{t=1}^{217}$ reject the unit-root hypothesis

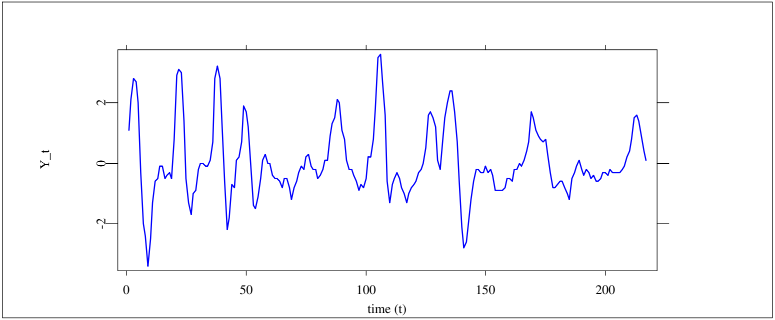

Figure: Seasonally adjusted quarterly series of US unemployment rate

Unemployment Rate: The Model

- The model

$$Y_t = m\left(\mathbf{X}_t\right) + \sigma\left(\mathbf{X}_t\right)\varepsilon_t \equiv c + \sum_{\alpha=1}^{p} m_\alpha\left(X_{t,\alpha}\right) + \sigma\left(\mathbf{X}_t\right)\varepsilon_t$$

- Predictor vector $\left\{X_{t,1}, X_{t,2}\right\} = \left\{Y_{t-1}, Y_{t-2}\right\}$ was selected according to the BIC in Huang and Yang (2004)
- The estimated multivariate regression function:

$$\hat{m}\left(\mathbf{X}_t\right) = \hat{c} + \hat{m}_1\left(Y_{t-1}\right) + \hat{m}_2\left(Y_{t-2}\right)$$

Unemployment Rate: The Model

- The model

$$Y_t = m(\mathbf{X}_t) + \sigma(\mathbf{X}_t)\varepsilon_t \equiv c + \sum_{\alpha=1}^{p} m_\alpha(X_{t,\alpha}) + \sigma(\mathbf{X}_t)\varepsilon_t$$

- Predictor vector $\{X_{t,1}, X_{t,2}\} = \{Y_{t-1}, Y_{t-2}\}$ was selected according to the BIC in Huang and Yang (2004)
- The estimated multivariate regression function:

$$\hat{m}(\mathbf{X}_t) = \hat{c} + \hat{m}_1(Y_{t-1}) + \hat{m}_2(Y_{t-2})$$

Unemployment Rate: The Model

- The model

$$Y_t = m(\mathbf{X}_t) + \sigma(\mathbf{X}_t)\varepsilon_t \equiv c + \sum_{\alpha=1}^{p} m_\alpha(X_{t,\alpha}) + \sigma(\mathbf{X}_t)\varepsilon_t$$

- Predictor vector $\{X_{t,1}, X_{t,2}\} = \{Y_{t-1}, Y_{t-2}\}$ was selected according to the BIC in Huang and Yang (2004)
- The estimated multivariate regression function:

$$\hat{m}(\mathbf{X}_t) = \hat{c} + \hat{m}_1(Y_{t-1}) + \hat{m}_2(Y_{t-2})$$

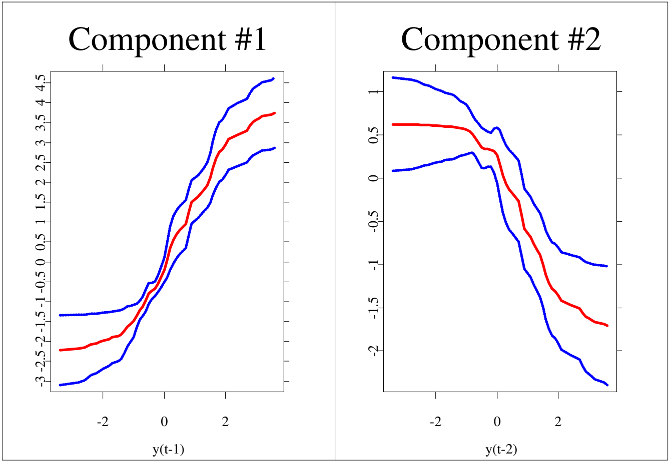

Figure: Estimation of the additive components for $k = 10$

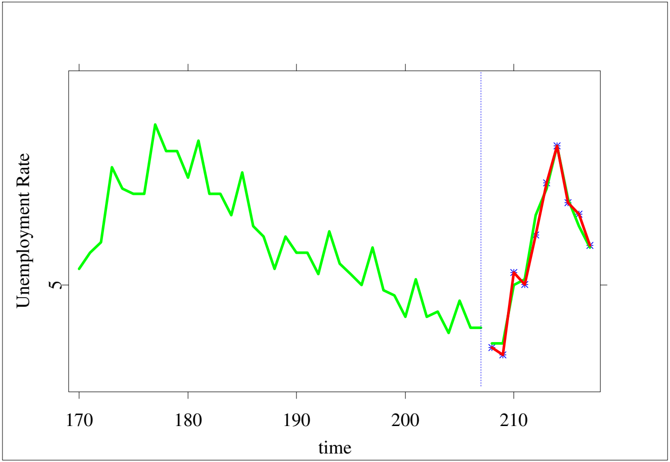

Figure: Out-of-sample forecasts, $k = 10$ obs, red (SPBK), green (real)

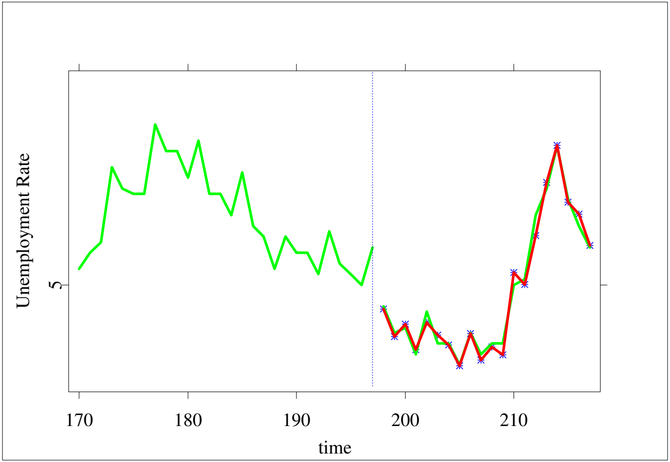

Figure: Out-of-sample forecasts, $k = 20$ obs, red (SPBK), green (real)

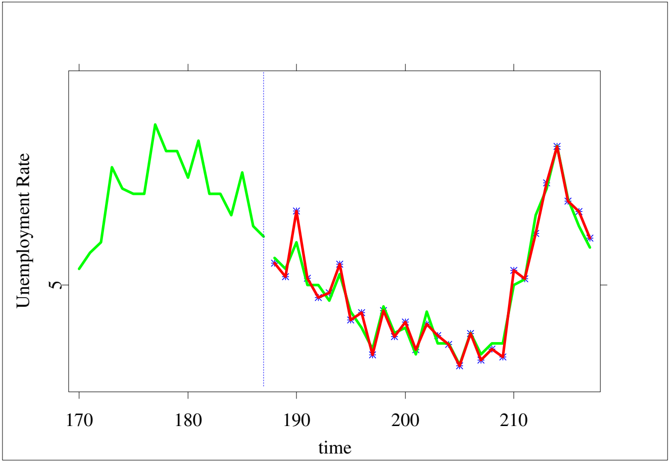

Figure: Out-of-sample forecasts, $k = 30$ obs, red (SPBK), green (real)

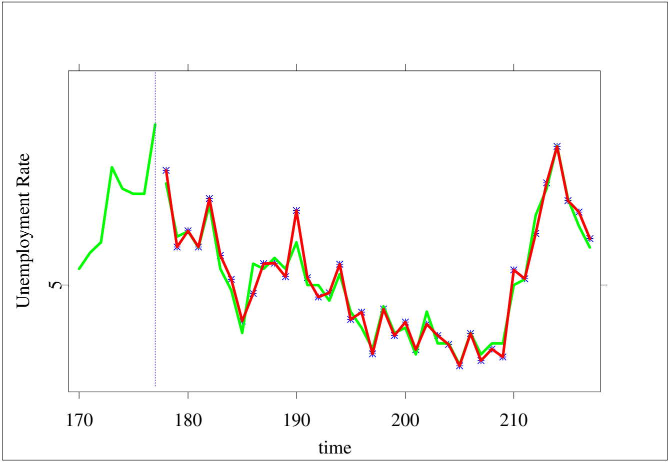

Figure: Out-of-sample forecasts, $k = 40$ obs, red (SPBK), green (real)

Table 3: Mean squared prediction errors (MSPE)

Model	Nonlinear	Linear
Selected lags	1,2	1,2,4,5,8
$k = 10$	0.02211	0.05774
$k = 20$	0.02130	0.04632
$k = 30$	0.03718	0.05423
$k = 40$	0.04556	0.06875

Remark: $\text{MSPE} = \frac{1}{k} \sum_{t=n-k+1}^{n} \left(R_t - \hat{R}_t \right)^2$

Conclusions

Computationally expedient

- It is useful for analyzing time series with many lagged variables

Intuitively appealing

- Approximating a curve by the piecewise polynomial is well known
- Backfitting algorithm is easily understood

Theoretically reliable

- It inherits all the asymptotic properties of the oracle smoother

Conclusions

Computationally expedient

- It is useful for analyzing time series with many lagged variables

Intuitively appealing

- Approximating a curve by the piecewise polynomial is well known
- Backfitting algorithm is easily understood

Theoretically reliable

- It inherits all the asymptotic properties of the oracle smoother

Conclusions

Computationally expedient

- It is useful for analyzing time series with many lagged variables

Intuitively appealing

- Approximating a curve by the piecewise polynomial is well known
- Backfitting algorithm is easily understood

Theoretically reliable

- It inherits all the asymptotic properties of the oracle smoother