

A Hessian Regularized Nonlinear Time Series Model (HRM)

Jie Chen and Xiaoming Huo ^a

December 17, 2007

Abstract

We introduce a nonparametric nonlinear time series model. The key idea is to fit a model via penalization, where the penalty term is an unbiased estimator of the integrated hessian of the underlying function. The underlying model assumption is very general: it has hessian everywhere (or almost everywhere) in its domain. Numerical experiments demonstrate that our model has better predictive power: if the underlying model complies with an existing parametric/semiparametric form (e.g., a threshold autoregressive model (TAR), an additive autoregressive model (AAR), or a functional coefficient autoregressive model (FAR)), our model performs comparably; if the underlying model does *not* comply with any pre-existing form, our model performs better in near all simulations. We name our model a hessian regularized nonlinear model for time series (HRM). We conjecture on theoretical properties and use simulations to back up. We discuss the feasibility of fast computing: our method can be as fast as fitting a smoothing spline in one dimension. Our approach can be viewed as a way to generalize splines to high dimensions (when the number of variates is more than three), under which an analogous analytical derivation can not work due to the effect of dimensionality.

Key Words. penalization estimator, splines, threshold autoregressive (TAR), additive autoregressive (AAR), functional coefficient autoregressive (FAR), hessian, regularization,

nearest neighbors.

^a: Dr. Jie Chen is with Georgia Institute of Technology, School of Industrial and Systems Engineering, jchen@isye.gatech.edu. Professor Xiaoming Huo is with Georgia Institute of Technology, School of Industrial and Systems Engineering, xiaoming@isye.gatech.edu, www.isye.gatech.edu/~xiaoming. This work has been partially supported by National Science Foundation grant DMS 0604736 and 0700152.

1 INTRODUCTION

In time series analysis, the concept of autoregressive (AR) and moving average (MA) has been regularly taught in graduate courses. An example is the ARIMA model. These models are linear, or linear after a pre-specified transform (like exponential). In the past few decades, nonlinear models have been successfully developed and tested in time series. Some works that have been influential to us include the threshold autoregressive model (TAR) (Tong 1990), additive autoregressive model (AAR) (Hastie and Tibshirani 1990), multivariate local polynomial regression model (Cleveland 1979, 1988; Fan and Gijbels 1996), functional coefficient autoregressive model (Chen and Tsay 1993) and its adaptive version (FAR) (Cai et al. 2000; Fan et al. 2003), among many others. Book Fan and Yao (2003) provides an excellent contemporary overview. These developments have significantly broadened our modeling capability in time series. However, they still rely on some particular assumptions on the functional form. For example, in TAR, the model is assumed to be piecewise linear. In AAR, the underlying model is an additive function. In FAR, the linear combination formulation under an autoregressive framework is preserved, while the coefficients are relaxed to be functions of specific types—e.g., in Chen and Tsay (1993), they are functions of one covariate, while in Fan et al. (2003), they are functions of a unique linear transform of all covariates. These preassumptions potentially raise confusion in practice, users pondering which model should be used. The number of the proposed models and the foreseeable possibility of many other alternative models also raise the question: is there a approach that unify them? It is desirable to have an approach that is relatively free from

formulational assumptions.

This paper introduces a numerical method, which only requires the underlying model function to have the hessian almost everywhere. The main idea is to consider a penalized model fitting. The objective function is a sum of a goodness-of-fit measure (which is the residuals sum of squares) and a penalty term multiplied by an algorithmic parameter (λ). The penalty function is an unbiased estimator of the integrated hessian of the underlying function. Numerical solution is developed. It is shown that the solution can be given in a close form; hence implementation is convenient. The hessian estimator utilizes the nearest neighbors of each observation. To our knowledge, the algorithm is at its first appearance. The above approach is attractive because no additional assumption on the model formulation is imposed in advance.

In simulations, we compare our approach with other models. The result is very appealing. When the underlying model complies with one existing form (e.g., it is a TAR model), our method performs comparably with the existing method. When the underlying model deviates from existing forms, our methods consistently outperform existing methods.

Our philosophical starting point is from the literature on smoothing splines. Our current approach can be considered as a way to generalize smoothing splines to high dimensions. To be more specific, we need to bring in the formulation. Consider a univariate time series $X_1, X_2, \dots, X_n, n \geq 1$. A nonlinear data generation mechanism can be written as

$$X_t = f(X_{t-1}, \dots, X_{t-p}) + \varepsilon_t, \quad (1.1)$$

for $p + 1 \leq t \leq n$, and i.i.d. ε_t 's. Function f has p variables. We assume that f has square integrable second partial derivatives. To simplify the notation, denote $\mathbf{Z}_{t-1} = (X_{t-1}, \dots, X_{t-p})^T \in \mathbb{R}^p$; vector $\mathbf{z} = (z_1, \dots, z_p)^T \in \mathbb{R}^p$ is a generic p -dimensional vector. Let $f_{ij}(z) = \frac{\partial^2 f(z)}{\partial z_i \partial z_j}$. Recall the integrated hessian of function f is defined as

$$\mathcal{H}f = \int_{\Omega} \sum_{i,j} |f_{ij}(z)|^2 dz,$$

where subset $\Omega \subset \mathbb{R}$ is the support of function f . Because f has square integrable second derivatives, we must have $\mathcal{H}f < \infty$. It is evident that formula (1.1) encompasses many

nonlinear time series models. For technical reason, we assume that $\Omega(\subset \mathbb{R}^p)$ is compact. Define a functional class: $\mathcal{F} = \{f : \mathcal{H}f < \infty, \text{ and support } f \subset \Omega\}$. A large literature in nonparametric estimation is to consider \hat{f} which is the minimizer in the following problem:

$$\min_{f \in \mathcal{F}} \sum_{t=p+1}^N [X_t - f(\mathbf{Z}_{t-1})]^2 + \lambda \mathcal{H}f. \quad (1.2)$$

(Adding a penalty term $\mathcal{H}f$ is a standard approach in regularization.) Function $\mathcal{H}f$ is called the “bending energy” in deriving the thin-plate spline (Wahba 1990). Note that if $p = 1$, then the minimizer in (1.2) is the natural cubic spline, whose knots are at \mathbf{Z}_t 's, $p \leq t \leq n-1$. If $p = 2$ or 3 , then the solution to (1.2) is the thin plate spline (Wahba 1990). However when $p \geq 4$, the above problem is not well defined (Green and Silverman 1994, Section 7.9). Hence for $p > 3$, high-dimensional splines can not be derived by following the formulation (1.2).

Our proposed model—hessian regularized nonlinear time series model (HRM)—uses a surrogate penalty function in (1.2). In a nutshell, the new penalty function is a sum of estimated hessian at observed points. At each point, the nearest neighbors are utilized, and the pointwise estimate is the Frobenius norm of the least squares estimate of the hessian (which ought to be a matrix). At this point, theoretical consistency of the numerical approach remains a conjecture. However, simulations provide strong evidence that the consistency holds with significant generality. By considering special cases, some speculative analysis is provided, in order to demonstrate the theoretical derivation that may take place. A full scale proof seems to be too complex to be accommodated here, and is relegated to a future publication. Most importantly, simulations demonstrate the advantage of our method to existing state-of-the-art nonlinear models.

The choice of the algorithmic parameter λ is studied. We found that the generalized cross validation principle can be adopted successfully. Fast computational algorithm is known for smoothing splines. We discuss how this technique can be integrated into fitting our model.

The rest of this paper is organized as follows. Section 2 derives a numerical approximation to the functional, given an analytical solution is not available. Section 3 derives a

theorem that reveals some conditions under which our proposed method should work. Section 4 discusses issues related to how to choose an optimal value of λ . Section 5 introduces a fast computing approach. Section 6 presents numerical results for some simulations and real data analysis. Relevant discussion is in Section 7. Section 8 presents conclusion.

2 NUMERICAL APPROXIMATION TO HESSIAN

We propose a numerical approach that emulates problem (1.2). The key idea is to introduce a least squares estimator of the hessian matrix $\mathcal{H}f(\mathbf{Z}_{t-1})$ at locations $\mathbf{Z}_{t-1}, t = p+1, p+2, \dots, n$.

Because \mathbf{Z}_t 's are random variables, we slightly modify the original hessian functional $\mathcal{H}f$, so that the density function of \mathbf{Z}_t 's can be involved. The new hessian functional is:

$$\int_{\Omega} \sum_{i,j} |f_{ij}(z)|^2 g(z) dz, \quad (2.3)$$

where $g(z)$ is the density function of z . Actually, $\mathcal{H}f$ is a special case of (2.3), if z is uniformly distributed. Thus, objective function (1.2) becomes

$$\min_{f \in \mathcal{F}} \sum_{t=p+1}^N [X_t - f(\mathbf{Z}_{t-1})]^2 + \lambda \int_{\Omega} \sum_{i,j} |f_{ij}(z)|^2 g(z) dz \quad (2.4)$$

An unbiased estimator of functional (2.3) is $\frac{1}{n-p} \sum_{t=p+1}^n \|\mathcal{H}f(\mathbf{Z}_{t-1})\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm (or Euclidean norm) of a matrix. Therefore, a numerical approximation of problem (2.4) is

$$\min_{f \in \mathcal{F}} \sum_{t=p+1}^N [X_t - f(\mathbf{Z}_{t-1})]^2 + \lambda \sum_{t=p+1}^n \|\mathcal{H}f(\mathbf{Z}_{t-1})\|_F^2. \quad (2.5)$$

2.1 Local Least Square Estimate

We now consider an estimate of functional $\|\mathcal{H}f(\mathbf{Z}_{t-1})\|_F^2$. Recall we have

$$\mathbf{Z}_{t-1} = (X_{t-1}, \dots, X_{t-p})^T \in \mathbb{R}^p.$$

Consider a set $\mathcal{V} = \{\mathbf{Z}_{t-1}, p+1 \leq t \leq n\}$ is a collection of $(n-p)$ p -dimensional vectors. Assume $\mathbf{V}_0 = \mathbf{Z}_{t-1}$, for $p+1 \leq t \leq n$. Let $\mathbf{V}_i, i = 1, 2, \dots, k$, denote the k ($k \geq 1$) nearest

neighbors of \mathbf{V}_0 , while $\mathbf{V}_i \in \mathcal{V}$. Let $\bar{\mathbf{V}} = \frac{1}{k+1} \sum_{i=0}^k \mathbf{V}_i$, i.e., $\bar{\mathbf{V}}$ is the average of the $k+1$ vectors. A Taylor expansion at point $\bar{\mathbf{V}}$ generates the following approximation

$$f(\mathbf{V}_i) \approx f(\bar{\mathbf{V}}) + (\mathbf{V}_i - \bar{\mathbf{V}})^T \mathcal{J}f(\bar{\mathbf{V}}) + \frac{1}{2}(\mathbf{V}_i - \bar{\mathbf{V}})^T \mathcal{H}f(\bar{\mathbf{V}})(\mathbf{V}_i - \bar{\mathbf{V}}),$$

$$i = 0, 1, \dots, k,$$

where $f(\bar{\mathbf{V}})$ is the value of function f at location $\bar{\mathbf{V}}$, $\mathcal{J}f(\bar{\mathbf{V}})$ is the Jacobian at $\bar{\mathbf{V}}$, and $\mathcal{H}f(\bar{\mathbf{V}})$ is the hessian matrix at $\bar{\mathbf{V}}$. Note we have $\mathcal{J}f(\bar{\mathbf{V}}) \in \mathbb{R}^p$ and $\mathcal{H}f(\bar{\mathbf{V}}) \in \mathbb{R}^{p \times p}$.

If f is analytical, then the above approximation is close. A matrix version of the above approximation is

$$\mathbf{f}^* \approx \mathbf{1}_{k+1} \cdot c + \mathbf{V} \cdot \mathbf{J} + \frac{1}{2} \mathbf{C} \cdot \mathbf{H}, \quad (2.6)$$

where

$$\mathbf{f}^* = (f(\mathbf{V}_0), f(\mathbf{V}_1), \dots, f(\mathbf{V}_k))^T \in \mathbb{R}^{k+1},$$

$$\mathbf{1}_{k+1} = (1, \dots, 1)^T \in \mathbb{R}^{k+1},$$

c is a constant; the i th ($1 \leq i \leq k+1$) row of matrix \mathbf{V} , $\mathbf{V} \in \mathbb{R}^{(k+1) \times p}$, is $(\mathbf{V}_{i-1} - \bar{\mathbf{V}})^T$; vector $\mathbf{J} \in \mathbb{R}^p$ is the Jacobian ($J_i = f_i(\bar{\mathbf{V}})$) at $\bar{\mathbf{V}}$. The i th row of matrix \mathbf{C} , $\mathbf{C} \in \mathbb{R}^{(k+1) \times \frac{p^2+p}{2}}$, is $\mathcal{V}_1[(\mathbf{V}_i - \bar{\mathbf{V}})(\mathbf{V}_i - \bar{\mathbf{V}})^T]$, where for an arbitrary column vector $\mathbf{x} = (x_1, x_2, \dots, x_p)^T$, we define $\mathcal{V}_1[\mathbf{x} \cdot \mathbf{x}^T] = (x_1^2, x_2^2, \dots, x_p^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_1x_p, \sqrt{2}x_2x_3, \dots, \sqrt{2}x_2x_p, \dots, \sqrt{2}x_{p-1}x_p) \in \mathbb{R}^{\frac{p^2+p}{2}}$. Vector \mathbf{H} is a vectorization with respect to the hessian matrix at $\bar{\mathbf{V}}$, after eliminating identical entries: $\mathbf{H} = \mathcal{V}_2[\mathcal{H}f(\bar{\mathbf{V}})]$, where for a symmetric matrix $\mathbf{S} = (S_{ij}) \in \mathbb{R}^{p \times p}$, we have $\mathcal{V}_2[\mathbf{S}] = (S_{11}, S_{22}, \dots, S_{pp}, \sqrt{2}S_{12}, \sqrt{2}S_{13}, \dots, \sqrt{2}S_{1p}, \sqrt{2}S_{23}, \dots, \sqrt{2}S_{2p}, \dots, \sqrt{2}S_{p-1,p})^T$. It is a standard exercise to verify that $\mathbf{1}^T \mathbf{V} = \mathbf{0}$.

A partial implementation of QR-decomposition (via, e.g., a modified Gram-Schmidt algorithm) can produce

$$\begin{bmatrix} \mathbf{1}_{k+1} & \mathbf{V} & \frac{1}{2} \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{I}_{(p^2+p)/2} \end{bmatrix}, \quad (2.7)$$

where columns of $\mathbf{Q}_1 \in \mathbb{R}^{(k+1) \times (p+1)}$ are orthonormal ($\mathbf{Q}_1^T \mathbf{Q}_1 = \mathbf{I}_{p+1}$), and columns of $\mathbf{Q}_2 \in \mathbb{R}^{(k+1) \times \frac{p^2+p}{2}}$ are orthogonal to the columns of \mathbf{Q}_1 (i.e., $\mathbf{Q}_2^T \mathbf{Q}_1 = \mathbf{0}$).

From (2.6), we have

$$\begin{aligned}\mathbf{Q}_2^T \mathbf{f}^* &= \begin{pmatrix} \mathbf{0} & \mathbf{Q}_2^T \mathbf{Q}_2 \end{pmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{I}_{(p^2+p)/2} \end{bmatrix} \begin{bmatrix} c \\ \mathbf{J} \\ \mathbf{H} \end{bmatrix} \\ &= \mathbf{Q}_2^T \mathbf{Q}_2 \mathbf{H}.\end{aligned}$$

Hence, a least-squares estimator of \mathbf{H} is

$$\hat{\mathbf{H}} = (\mathbf{Q}_2^T \mathbf{Q}_2)^+ \mathbf{Q}_2^T \mathbf{f}^*,$$

where $(\cdot)^+$ denotes a pseudo-inverse of a matrix.

For the local hessian matrix, we have

$$\begin{aligned}\|\hat{\mathcal{H}}f(\mathbf{Z}_{t-1})\|_F^2 &= \|\hat{\mathbf{H}}\|_2^2 = \hat{\mathbf{H}}^T \hat{\mathbf{H}} \\ &= (\mathbf{f}^*)^T \mathbf{Q}_2 (\mathbf{Q}_2^T \mathbf{Q}_2)^+ (\mathbf{Q}_2^T \mathbf{Q}_2)^+ \mathbf{Q}_2^T \mathbf{f}^*.\end{aligned}$$

2.2 Global Estimation and a Close Form Solution

Now we consider the objective function in (2.5). We will show that it is a quadratic form.

To construct the matrix form of (2.5), we introduce the following notation:

$$\mathbf{K}_{t-1} = \mathbf{Q}_2 (\mathbf{Q}_2^T \mathbf{Q}_2)^+ (\mathbf{Q}_2^T \mathbf{Q}_2)^+ \mathbf{Q}_2^T.$$

We also bring in a selection matrix \mathbf{S}_{t-1} , $p+1 \leq t \leq n$. Matrix \mathbf{S}_{t-1} , $\mathbf{S}_{t-1} \in \mathbb{R}^{(k+1) \times (n-p)}$, is made by two possible components: 0 and 1. For $\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_k$ and $\mathbf{Z}_p, \dots, \mathbf{Z}_{n-1}$ that are defined before, \mathbf{S}_{t-1} satisfies

$$(\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_k) = (\mathbf{Z}_p, \mathbf{Z}_{p+1}, \dots, \mathbf{Z}_{n-1}) \mathbf{S}_{t-1}^T, \forall t.$$

Apparently we have $\mathbf{f}^* = \mathbf{S}_{t-1} \mathbf{f}$, where $\mathbf{f} = (f(\mathbf{Z}_p), f(\mathbf{Z}_{p+1}), \dots, f(\mathbf{Z}_{n-1}))^T$. We have

$$\sum_{t=p+1}^n \|\hat{\mathcal{H}}f(\mathbf{Z}_{t-1})\|_F^2 = \sum_{t=p}^{n-1} (\mathbf{f}^T \mathbf{S}_t^T \mathbf{K}_t \mathbf{S}_t \mathbf{f}).$$

Let $\mathbf{M} = (\mathbf{S}_p^T, \dots, \mathbf{S}_{n-1}^T) \text{diag}\{\mathbf{K}_p, \mathbf{K}_{p+1}, \dots, \mathbf{K}_{n-1}\} \begin{pmatrix} \mathbf{S}_p \\ \vdots \\ \mathbf{S}_{n-1} \end{pmatrix}$, we have

$$\sum_{t=p+1}^n \|\hat{\mathcal{H}}f(\mathbf{z}_{t-1})\|_F^2 = \mathbf{f}^T \mathbf{M} \mathbf{f},$$

which is a quadratic function of \mathbf{f} .

Problem (2.5) becomes

$$\min_{\mathbf{f}} \|\mathbf{Y} - \mathbf{f}\|_2^2 + \lambda \mathbf{f}^T \mathbf{M} \mathbf{f},$$

where we have variable $\mathbf{f} \in \mathbb{R}^{n-p}$, vector $\mathbf{Y} = (X_{p+1}, \dots, X_n)^T \in \mathbb{R}^{n-p}$, and matrix \mathbf{M} is defined before. The estimator of \mathbf{f} becomes

$$\hat{\mathbf{f}} = (\mathbf{I}_{n-p} + \lambda \cdot \mathbf{M})^{-1} \cdot \mathbf{Y}. \quad (2.8)$$

In numerical implementation, it may be faster to treat $\hat{\mathbf{f}}$ as a solution to the following system of linear equations: $(\mathbf{I}_{n-p} + \lambda \cdot \mathbf{M}) \cdot \hat{\mathbf{f}} = \mathbf{Y}$.

2.3 Null Space of Matrix \mathbf{M}

The estimator in (2.8) requires inverting an $(n-p) \times (n-p)$ matrix, which can be challenging. It is easy to verify that matrix \mathbf{M} is positive-semidefinite. Moreover, matrix \mathbf{M} has eigenvalue 0, whose multiplicity is at least $p+1$, conditioning that the following matrix is of full column rank:

$$\begin{pmatrix} 1 & \mathbf{z}_p^T \\ \vdots & \vdots \\ 1 & \mathbf{z}_{n-1}^T \end{pmatrix}.$$

As a matter of fact, every column of the above matrix solves the equation $\mathbf{M} \cdot \mathbf{x} = 0$.

2.4 Prediction

We would like to estimate $f(\mathbf{Z})$ at a new point $\mathbf{Z} \in \mathbb{R}^p$. First we identify the $k+1$ ($k \geq 1$) nearest neighbors of \mathbf{Z} for the vectors in the set \mathcal{V} . The reason for choosing the $k+1$ instead

of k nearest neighbors is that we want a similar expression in the prediction step as in the estimation step. Recall \mathcal{V} contains all the p -dimensional vectors generated by a scanning window going through the time series. Without loss of generality, let $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_{k+1}$ denote the $k + 1$ nearest neighbors. Let $\bar{\mathbf{V}}$ denote the average: $\bar{\mathbf{V}} = \frac{1}{k+1} \sum_{i=1}^{k+1} \mathbf{V}_i$. Recall $\mathcal{J}f(\bar{\mathbf{V}})$ denotes the Jacobian at $\bar{\mathbf{V}}$, and $\mathcal{H}f(\bar{\mathbf{V}})$ denotes the hessian matrix at $\bar{\mathbf{V}}$. A second order approximation via Taylor expansion at point $\bar{\mathbf{V}}$ yields

$$f(\mathbf{V}_i) - f(\bar{\mathbf{V}}) = (\mathbf{V}_i - \bar{\mathbf{V}})^T \mathcal{J}f(\bar{\mathbf{V}}) + \frac{1}{2}(\mathbf{V}_i - \bar{\mathbf{V}})^T \mathcal{H}f(\bar{\mathbf{V}})(\mathbf{V}_i - \bar{\mathbf{V}}).$$

Recall $\hat{f}(\mathbf{V}_1), \dots, \hat{f}(\mathbf{V}_{k+1})$ are the fitted values at $\mathbf{V}_1, \dots, \mathbf{V}_{k+1}$. Similar to the analysis in establishing the least squares estimators for hessian, we have the following equation:

$$\begin{aligned} \begin{pmatrix} \hat{f}(\mathbf{V}_1) \\ \vdots \\ \hat{f}(\mathbf{V}_{k+1}) \end{pmatrix} &= \mathbf{1}_{k+1} f(\bar{\mathbf{V}}) + \begin{pmatrix} (\mathbf{V}_1 - \bar{\mathbf{V}})^T \\ \vdots \\ (\mathbf{V}_{k+1} - \bar{\mathbf{V}})^T \end{pmatrix} \mathcal{J}f(\bar{\mathbf{V}}) \\ &+ \frac{1}{2} \begin{pmatrix} \mathcal{V}_1[(\mathbf{V}_1 - \bar{\mathbf{V}})(\mathbf{V}_1 - \bar{\mathbf{V}})^T] \\ \vdots \\ \mathcal{V}_1[(\mathbf{V}_{k+1} - \bar{\mathbf{V}})(\mathbf{V}_{k+1} - \bar{\mathbf{V}})^T] \end{pmatrix} \mathcal{V}_2[\mathcal{H}f(\bar{\mathbf{V}})], \end{aligned} \quad (2.9)$$

where vectorization operators $\mathcal{V}_1(\cdot)$ and $\mathcal{V}_2(\cdot)$ have been defined earlier. Note $f(\bar{\mathbf{V}})$ is a scalar, vector $\mathcal{J}f(\bar{\mathbf{V}})$ is p -dimensional, and matrix $\mathcal{H}f(\bar{\mathbf{V}})$ contains $p(p + 1)/2$ unknown variables. If we have

$$k \geq p + \frac{p(p+1)}{2} = \frac{1}{2}(p+1)(p+2),$$

then least squares estimators can be established for $f(\bar{\mathbf{V}})$, $\mathcal{J}f(\bar{\mathbf{V}})$ and $\mathcal{H}f(\bar{\mathbf{V}})$ on the right hand side of (2.9). Hence, an estimated value of $f(\cdot)$ at \mathbf{Z} is

$$\hat{f}(\mathbf{Z}) = \hat{f}(\bar{\mathbf{V}}) + (\mathbf{Z} - \bar{\mathbf{V}})^T \hat{\mathcal{J}}f(\bar{\mathbf{V}}) + \frac{1}{2}(\mathbf{Z} - \bar{\mathbf{V}})^T \hat{\mathcal{H}}f(\bar{\mathbf{V}})(\mathbf{Z} - \bar{\mathbf{V}}). \quad (2.10)$$

A variation of the above is to ignore the quadratic terms in the right hand sides of (2.9) and (2.10). Hence instead of a quadratic prediction, we adopt a linear prediction, where the first order approximation via Taylor expansion is applied, and the least squares estimators are established for $f(\bar{\mathbf{V}})$ and $\mathcal{J}f(\bar{\mathbf{V}})$. Thus, the prediction of $f(\cdot)$ at \mathbf{Z} is

$$\hat{f}(\mathbf{Z}) = \hat{f}(\bar{\mathbf{V}}) + (\mathbf{Z} - \bar{\mathbf{V}})^T \hat{\mathcal{J}}f(\bar{\mathbf{V}}).$$

The simulation results show that the linear prediction is more robust than the quadratic prediction. Thus, in our numerical experiments, only the linear prediction is utilized.

3 WHEN DOES THE NUMERICAL APPROACH WORK?

We have introduced a numerical approximation to (2.4). Given an underlying function f that is smooth enough, when the estimator $\hat{\mathbf{f}}$ will converge to the underlying function? In this section, we study quantity $\|\hat{\mathbf{f}}_n - \mathbf{f}\|_2^2/n$, where $\hat{\mathbf{f}}_n$ is the same as $\hat{\mathbf{f}}$ but with a subscript to indicate the length of the time series n , and vector \mathbf{f} is the true value of the function at \mathbf{Z}_t 's. We conjecture that $\|\hat{\mathbf{f}}_n - \mathbf{f}\|_2^2/n \rightarrow 0$ under certain conditions that only depend on matrix \mathbf{M} and underlying function $f(\cdot)$. These conditions can be verified numerically, hence can be checked in simulations. Our conditions are analogous to the conditions for Sobolev space, which has played an important role in determining the optimal rate of estimation within a certain functional class (Johnstone 2004). In the following, we first establish an upper bound. Then we present a special case under which the derived upper bound converges to zero when n goes to infinity. The special case motivates us to believe that the convergence is true in more general case, however, a rigorous proof will not be in the present paper.

3.1 An Upper Bound

We establish an upper bound for the error $(\|\hat{\mathbf{f}}_n - \mathbf{f}\|_2^2)$. Recall

$$\hat{\mathbf{f}}_n = (\mathbf{I}_{n-p} + \lambda_n \mathbf{M})^{-1} \mathbf{Y},$$

Consequently, we have

$$\hat{\mathbf{f}}_n - \mathbf{f} = (\mathbf{I} + \lambda_n \mathbf{M})^{-1} \mathbf{f} - \mathbf{f} + (\mathbf{I} + \lambda_n \mathbf{M})^{-1} \boldsymbol{\varepsilon}.$$

where $\boldsymbol{\varepsilon} = (\varepsilon_{p+1}, \dots, \varepsilon_n)^T$.

Let $\mathbf{M} = \mathbf{U}^T \mathbf{D} \mathbf{U}$ be the eigenvalue decomposition of matrix \mathbf{M} . Denote $\mathbf{D} = \text{diag}\{d_1, \dots, d_{n-p}\}$. Let $f'_i = (\mathbf{U}\mathbf{f})_i$ and $\varepsilon'_i = (\mathbf{U}\boldsymbol{\varepsilon})_i$.

Lemma 3.1 *There exists a constant c_n (e.g. $c_n = 2 \log n$), such that for $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$,*

$$Pr\{|\varepsilon'_i|^2 < c_n \sigma^2, \forall i\} \rightarrow 1, \text{ as } n \rightarrow \infty.$$

The above is a well-known property of normally distributed random variables.

We have the following inequality, with high probability,

$$\begin{aligned} \frac{1}{2} \|\hat{\mathbf{f}}_n - \mathbf{f}\|_2^2 &\leq \sum_{i=1}^{n-p} \frac{(\lambda_n d_i)^2 (f'_i)^2 + (\varepsilon'_i)^2}{(1 + \lambda_n d_i)^2} \\ &\leq \sum_{i=1}^{n-p} \frac{(\lambda_n d_i)^2 (f'_i)^2 + c_n \sigma^2}{(1 + \lambda_n d_i)^2}. \end{aligned}$$

The last inequality utilizes the preceding lemma. We consider a function: for $\alpha \geq 0$.

$$g(\alpha) = \frac{\alpha^2 (f'_i)^2 + c_n \sigma^2}{(1 + \alpha)^2}.$$

The following can be verified through elementary calculation.

1. $g(0) = c_n \sigma^2, g(\infty) = (f'_i)^2$.
2. When $0 < \alpha < \frac{c_n \sigma^2}{(f'_i)^2}$, we have $g'(\alpha) < 0$; when $\alpha > \frac{c_n \sigma^2}{(f'_i)^2}$, we have $g'(\alpha) > 0$.
3. The minimum point is $g\left(\frac{c_n \sigma}{(f'_i)^2}\right) = \frac{(c_n \sigma^2)(f'_i)^2}{(f'_i)^2 + c_n \sigma^2}$.
4. If $(f'_i)^2 < c_n \sigma^2$ and $\alpha > \frac{1}{\gamma} \frac{c_n \sigma^2}{(f'_i)^2}$, where $\gamma \geq 1$, we have $g(\alpha) < \gamma (f'_i)^2$.
5. If $c_n \sigma^2 < (f'_i)^2$ and $\alpha < \gamma \frac{c_n \sigma^2}{(f'_i)^2}$, where $\gamma \geq 1$, we have $g(\alpha) < \gamma \cdot c_n \sigma^2$.

Consider two quantities:

$$\begin{aligned} a_n &= \max_i \left\{ \frac{c_n \sigma^2}{(f'_i)^2} \cdot \frac{1}{d_i} : \text{for } i \text{ such that } (f'_i)^2 < c_n \sigma^2 \right\}, \\ b_n &= \min_i \left\{ \frac{c_n \sigma^2}{(f'_i)^2} \cdot \frac{1}{d_i} : \text{for } i \text{ such that } (f'_i)^2 > c_n \sigma^2 \right\}. \end{aligned}$$

It won't be interesting if $a_n \leq b_n$. We suppose that

$$\gamma_n = \sqrt{\frac{a_n}{b_n}} \geq 1.$$

We pick $\lambda_n = \frac{a_n}{\gamma_n} = b_n \cdot \gamma_n = \sqrt{a_n b_n}$. We have the following main result.

Theorem 3.2 *For the aforementioned λ_n , we have, with high probability,*

$$\frac{1}{2} \|\hat{\mathbf{f}}_n - \mathbf{f}\|_2^2 \leq \gamma_n \cdot \sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2].$$

The proof is an application of the preceding analysis. If the right hand side converges to a finite number, then convergence is achieved. Section 6.1 provides some examples where $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ does decay with the increasing n . It is still an open question whether γ_n remains a constant.

3.2 Speculation on a Formal Proof of Convergence

It is known that $\sum_{i=1}^{n-p} (f'_i)^2/n = \|\mathbf{f}\|_2^2/n = O(1)$, i.e., such a quantity tends to be a constant. If sequence (f'_i) has the same property as the Fourier coefficients of a function in a Sobolev's ellipsoid, then $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]$ could have lower order than $O(n)$. Then it will be possible that $n^{-1} \gamma_n \sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2] \rightarrow 0$.

We can imagine for at least one special case that the above is true. Assume that $p = 1$. (Note it is against some previous argument; so it is for illustration only.) Suppose the variables z_{t-1} (now they are scalars) are equally spaced. It is easy to verify that the first selection matrix \mathbf{S}_1 selects the k leftmost points. The second selection matrix \mathbf{S}_2 selects the points having from the second smallest to the $(k+2)$ nd smallest coordinates... In general, the selection matrix \mathbf{S}_i selects the $k+1$ consecutive points starting from the i th smallest coordinate (see Fig. 1). On the other hand, all matrices \mathbf{K}_{t-1} are identical. Hence it is not hard to see that matrix \mathbf{M} approximate a Toeplitz matrix with identical diagonal and off-diagonal entries. Recall matrix \mathbf{M} is also symmetric. Hence the eigenvectors of matrix \mathbf{M} should be Fourier sequences. Hence if the underlying model corresponds to a function in a Sobolev's ellipsoid, then sequence (f'_i) has the property to ensure that the quantity $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]$ converges.

A rigorous proof will be a delicate problem. In simulated examples, in almost all the cases, we observe that the sequence $|f'_i|$ (after being sorted at a decreasing order) decays like an inverse polynomial (i.e., $x^{-\beta}$ for $\beta > 0$; in fact, in most cases, we observe that $\beta > 1/2$).

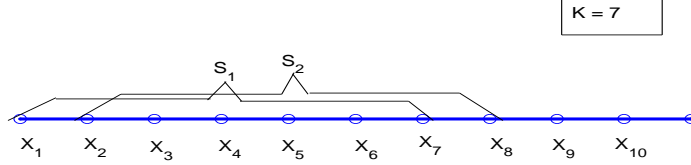


Figure 1: Illustration of the special case for the proof of convergence

From the above theorem, our estimator will converge to the truth in those situations.

4 CHOICE OF PENALTY PARAMETER λ

In Section 3, a theoretical appropriation λ_n is provided. However, in practice, the underlying function $f(\cdot)$ is not available; hence one can not utilize the theoretical formula. Generalized cross validation can be adopted. Consider the generalized cross validation function $\text{GCV}(\lambda)$:

$$\text{GCV}(\lambda) = \frac{1}{n-p} \sum_{k=p+1}^n \left(\frac{X_k - \hat{f}_\lambda(\mathbf{Z}_{k-1})}{1 - \frac{1}{n-p} \text{Tr}(\mathbf{A}(\lambda))} \right)^2, \quad (4.11)$$

where $\mathbf{A}(\lambda) = (\mathbf{I}_{n-p} + \lambda \mathbf{M})^{-1}$. The optimal value of the penalty parameter λ can be estimated by minimizing the above GCV function. The justification is relatively straightforward and can be found in thesis Chen (2007).

The derivation of the GCV function uses an approximation, because the numerical approximation of hessian functional is applied. The following provides more justification on applying generalized cross validation here. To facilitate the following analysis, let us recall some notations. Recall that the eigenvalue decomposition of matrix \mathbf{M} is $\mathbf{M} = \mathbf{U}^T \mathbf{D} \mathbf{U}$, where $\mathbf{D} = \text{diag}\{d_1, \dots, d_{n-p}\}$, and $0 \leq d_1 \leq d_2 \leq \dots \leq d_{n-p}$. Recall that $\varepsilon'_i = (\mathbf{U}\varepsilon)_i$.

We define $y'_i = (\mathbf{UY})_i$. Note vectors $\mathbf{f}, \mathbf{Y}, \varepsilon$ have been used in Section 3. If we know the true value of the function at every point (i.e., \mathbf{f} is known), then the mean square error as a function of λ is

$$\begin{aligned}
\text{MSE}(\lambda) &= \frac{1}{n-p} \|\mathbf{f} - \mathbf{A}(\lambda)\mathbf{Y}\|_2^2 \\
&= \frac{1}{n-p} \|[\mathbf{I} - \mathbf{A}(\lambda)]\mathbf{Y} - \varepsilon\|_2^2 \\
&= \frac{1}{n-p} \{ \|[\mathbf{I} - \mathbf{A}(\lambda)]\mathbf{Y}\|_2^2 + \|\varepsilon\|_2^2 - 2\varepsilon^T[\mathbf{I} - \mathbf{A}(\lambda)]\mathbf{Y} \} \\
&= \frac{1}{n-p} [\lambda^2 h_1(\lambda) + \|\varepsilon\|_2^2 - 2\lambda h_2(\lambda)],
\end{aligned} \tag{4.12}$$

where

$$h_1(\lambda) = \sum_i \frac{d_i^2 (y'_i)^2}{(1 + \lambda d_i)^2},$$

and

$$h_2(\lambda) = \sum_i \frac{d_i \cdot \varepsilon'_i \cdot y'_i}{1 + \lambda d_i}.$$

On the other hand, for GCV, we have

$$\text{GCV}(\lambda) = (n-p) \frac{h_1(\lambda)}{[h_3(\lambda)]^2}, \tag{4.13}$$

where

$$h_3(\lambda) = \sum_i \frac{d_i}{1 + \lambda d_i}.$$

Hopefully, one can establish a quantitative connection between the minimizer of (4.12) and the minimizer of (4.13). This chapter has not pursued further. Note that if factor $\varepsilon'_i \cdot y'_i$ can be treated as a constant, there is a strong similarity between $h_2(\lambda)$ and $h_3(\lambda)$. In simulations, we plot $\text{GCV}(\lambda)$ and $\text{MSE}(\lambda)$ for multiple examples. In almost all the cases, we observe that the minimum of the two function are close. This in some sense validates the use of GCV to choose λ . More details are given in Section 6.2.

5 ABOUT FAST COMPUTING

Recall that our estimator has the form $\hat{\mathbf{f}} = (\mathbf{I} + \lambda \mathbf{M})^{-1} \mathbf{Y}$. Such a solution bears strong

similarity with the solution to smoothing splines (Green and Silverman 1994, Section 2.3). It is well known that for smoothing splines, by taking advantage of a band matrix, fast computing is feasible (Reinsch 1967; Green and Silverman 1994, Section 2.3.3). A similar analysis can be developed for our method. It is not hard to observe that

$$(\mathbf{I} + \lambda\mathbf{M})\hat{\mathbf{f}} = \mathbf{Y}.$$

Denoting

$$\mathbf{B} = \text{diag}\{\mathbf{K}_p, \dots, \mathbf{K}_{n-1}\},$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_p \\ \vdots \\ \mathbf{S}_{n-1} \end{pmatrix},$$

and $\mathbf{r} = \mathbf{B}\hat{\mathbf{f}}$, we have

$$\begin{aligned} \hat{\mathbf{f}} &= \mathbf{Y} - \lambda\mathbf{M}\hat{\mathbf{f}} \\ &= \mathbf{Y} - \lambda\mathbf{S}^T\mathbf{B}\hat{\mathbf{f}} \\ &= \mathbf{Y} - \lambda\mathbf{S}^T\mathbf{r}. \end{aligned} \tag{5.14}$$

The above leads to

$$\mathbf{B}^{-1}\mathbf{r} = \hat{\mathbf{f}} = \mathbf{S}\mathbf{Y} - \lambda\mathbf{S}\mathbf{S}^T\mathbf{r},$$

which is equivalent to

$$(\mathbf{B}^{-1} + \lambda\mathbf{S}\mathbf{S}^T)\mathbf{r} = \mathbf{S}\mathbf{Y}. \tag{5.15}$$

Matrix \mathbf{B}^{-1} is blocky diagonal with $(k+1) \times (k+1)$ diagonal submatrices. If matrix $\mathbf{S}\mathbf{S}^T$ is a blocky band matrix, then the standard technique in manipulating band matrices (Green and Silverman 1994, Section 2.6) can be adopted. Hence an $O(nk^2)$ algorithm is available to find \mathbf{r} via (5.15). Our estimate $\hat{\mathbf{f}}$ then can be obtained via (5.14). The overall complexity is still $O(nk^2)$.

Unfortunately, in the previous algorithm, matrix $\mathbf{S}\mathbf{S}^T$ is not guaranteed to be a blocky band matrix. One can propose a modified version of our algorithm to facilitate fast computation. We discuss two possible approaches. Recall we are considering modeling $X_t =$

$f(\mathbf{Z}_{t-1}) + \varepsilon_t$, $p+1 \leq t \leq n$. Let $\{P(p+1), \dots, P(n)\}$ be a permutation of $p+1, \dots, n-1, n$. It is equivalent to consider modeling with dataset $X_{P(t)} = f(\mathbf{Z}_{P(t)-1}) + \varepsilon_{P(t)}$, $p+1 \leq t \leq n$. Now we consider the distance matrix $\{D_{ij}\}_{(n-p) \times (n-p)}$, where $D_{ij} = \|\mathbf{Z}_{i+p-1} - \mathbf{Z}_{j+p-1}\|_2$, i.e., the ℓ_2 -distance (Euclidean distance). It is evident that $D_{ii} = 0$. For a fixed permutation P , define $\tilde{D}_{ij}(P) = \|\mathbf{Z}_{P(i+p)-1} - \mathbf{Z}_{P(j+p)-1}\|_2$.

Approach 1: Find permutation P such that the quantity

$$\max\{\tilde{D}_{ij}(P) : |i - j| \leq k\} \quad (5.16)$$

is minimized. Then “the k nearest neighbors” of \mathbf{Z}_t are specified to be the k closest elements of $P(t)$ in the sequence $\{P(p+1), \dots, P(n)\}$. One can verify that the matrix $\mathbf{S}\mathbf{S}^T$ is a blocky band matrix.

Approach 2: A drawback of Approach One is that the permutation P may not be found easily. A compromise is to use some heuristic iterative algorithm to minimize the quantity in (5.16), and then still use the k nearest neighbors in the Euclidean distance. Hopefully, the off-diagonal submatrix of $\mathbf{S}\mathbf{S}^T$ becomes a zero matrix when it is far away from the diagonal. An example of such an algorithm is the Jacobi’s method to find eigenvalues and eigenvectors (Lange 1999). The simulation study and further research is left as future work.

6 NUMERICAL EXPERIMENTS

6.1 Simulations Regarding the Convergence Theorem

The following model is utilized to generate four times series:

$$X_t = f(X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}) + \varepsilon_t, \quad \text{for } t \geq 5,$$

where $\varepsilon_t \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2)$, where $\sigma = 1$ for the first two time series, $\sigma = 0.2$ for the third time series, and $\sigma = 0.8$ for the last time series. Under the previous formulation, these are the cases when $p = 4$. The function $f(X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4})$ is defined as:

- in the first time series, we have

$$f(x_1, x_2, x_3, x_4) = a_1 + a_2 + a_3 + a_4,$$

where

$$\begin{aligned} a_1 &= -x_2 \exp(-x_2^2/2), \\ a_2 &= \frac{x_1}{1+x_2^2} \cos(1.5x_2), \\ a_3 &= \frac{4x_3}{1+0.8x_3^2}, \\ a_4 &= \frac{\exp(3(x_4-2))}{1+\exp(3(x_4-2))}; \end{aligned}$$

- in the second time series, we have

$$f(x_1, x_2, x_3, x_4) = \frac{2x_1}{1+0.8x_1^2} - \frac{2x_2}{1+0.8x_2^2} + \frac{2x_3}{1+0.8x_3^2} - \frac{2x_4}{1+0.8x_4^2};$$

- in the third time series, we have

$$f(x_1, x_2, x_3, x_4) = a_1x_1 + a_2x_2 + a_1x_3 + a_2x_4, \quad (6.17)$$

where

$$\begin{aligned} a_1 &= 0.2 + (0.3 + x_1) \exp(-4x_1^2), \\ a_2 &= -0.4 - (0.7 + 1.3x_1) \exp(-4x_1^2); \end{aligned}$$

- in the last time series, we have

$$f(x_1, x_2, x_3, x_4) = \frac{0.25x_4}{1+1.2x_1^2} - \frac{0.4x_1}{1+0.6x_2^2} + \frac{0.5x_2}{1+0.8x_3^2} - \frac{0.75x_3}{1+x_4^2} + \frac{\exp(1.5(x_4-2))}{1+\exp(3(x_4-2))}.$$

The simulated time series can be seen in Fig. 2.

For each time series model, time series with the different length ranging from 200 to 3000 are generated. The increment of the length of time series is 200 data points each time. Two quantities $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ and $\sum_{i=1}^{n-p} (f'_i)^2/n$ are calculated, and illustrated

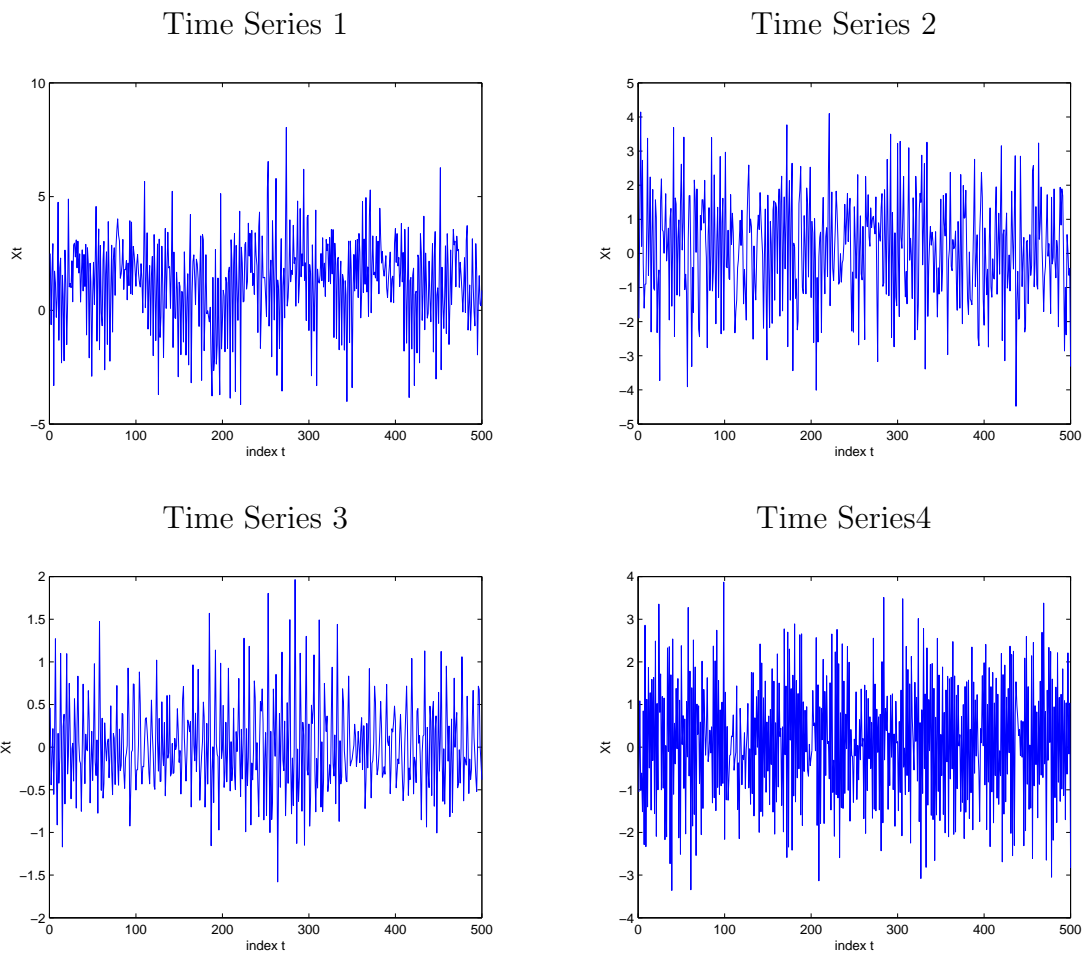


Figure 2: The simulated four time series with 500 data points. The data generation mechanism is described in Section 6.1.

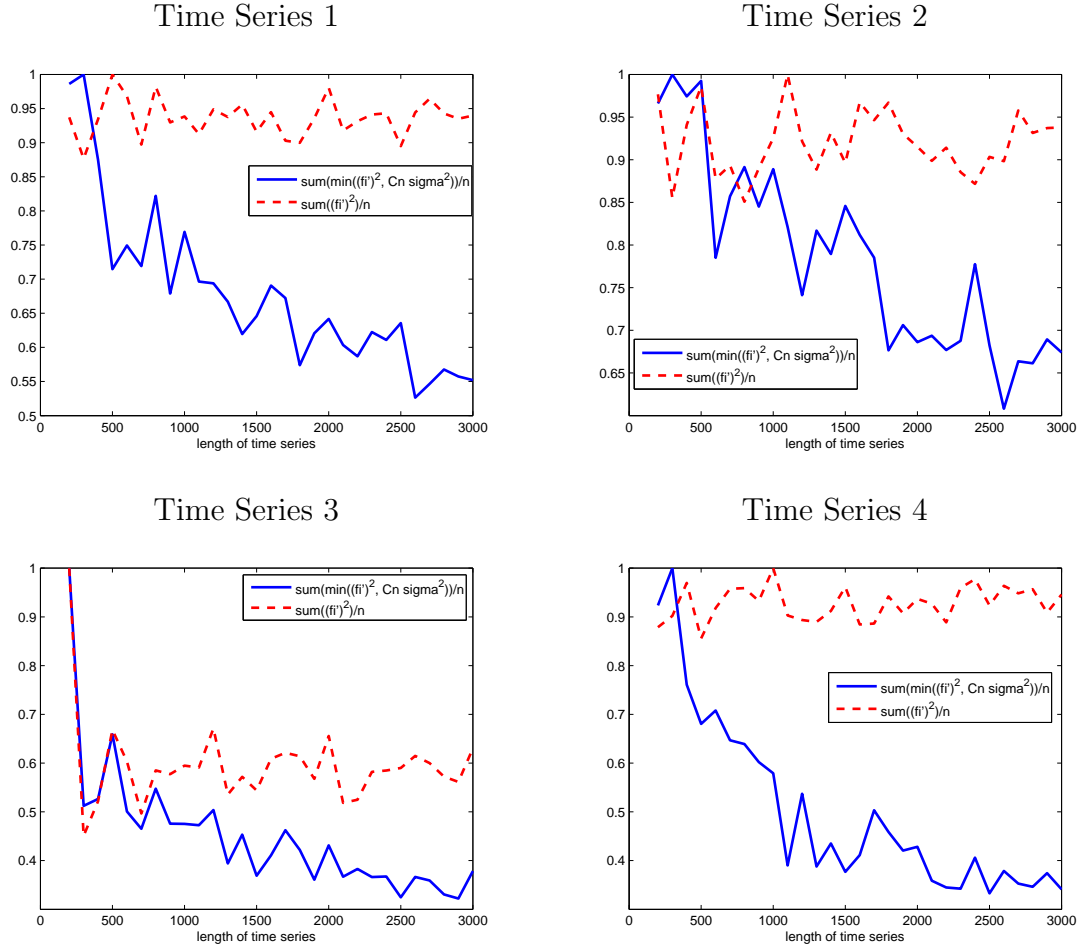


Figure 3: The trend of $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ and $\sum_{i=1}^{n-p} (f'_i)^2/n$ as n is increasing. The length of time series n ranges from 200 to 3000. In order to compare two quantities more clearly, we normalize the two quantity sequences by deviding their maximal values, respectively. Without normalization, $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ is always smaller than $\sum_{i=1}^{n-p} (f'_i)^2/n$.

for each time series in Fig.3. Quantity $\sum_{i=1}^{n-p} (f'_i)^2/n$ fluctuates around a constant with the different length of time series, while quantity $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ appears to decay as the length of the time series is increasing. The above observation verifies our conjecture to some degree: although $\sum_{i=1}^{n-p} (f'_i)^2/n$ tends to be a constant, $\sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2]/n$ could have lower order, i.e., it is possible that $n^{-1} \gamma_n \sum_{i=1}^{n-p} \min[(f'_i)^2, c_n \sigma^2] \rightarrow 0$.

6.2 Adoption of the Generalized Cross Validation Principle

Using simulations of the four time series in the last section, we study the relation between the minimizers of $\text{GCV}(\cdot)$ and $\text{MSE}(\cdot)$. Since the data generation mechanism is known, we can plot $\text{MSE}(\lambda)$ for a range of values for λ . Function $\text{MSE}(\cdot)$ is plotted against $\text{GCV}(\cdot)$ in Fig. 4. It is evident that the minimizer of GCV also renders a small value of MSE , which can be considered as a validation of using GCV function to choose for optimal λ .

6.3 Synthetic Examples

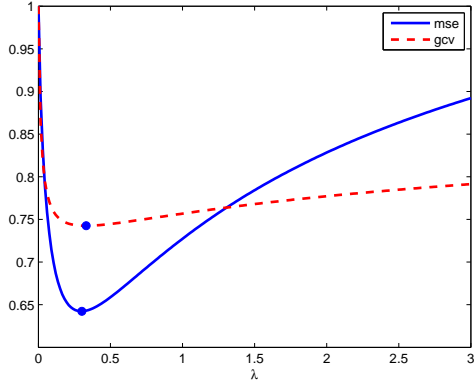
This section contains three parts. The first two parts consist of simulation models, which are chosen from functional coefficient autoregressive model (FAR) and threshold autoregressive model (TAR), respectively. The last part consists of two simulation models, which are nonlinear models and belong to none of FAR, TAR or additive autoregressive model (AAR). For each of the model, three types of prediction errors—one-step prediction errors, iterative two-step prediction errors, and direct two-step prediction errors—are computed for AAR, FAR, TAR, AR, Loess, Locpoly and our method. The difference between *iterative* two-step prediction and *direct* two-step prediction can be found in Fan and Yao (2003, Section 8.3.6). The following is a brief introduction of the models we applied for comparison.

- AAR(p): additive nonlinear autoregressive model with the embedding dimension p .

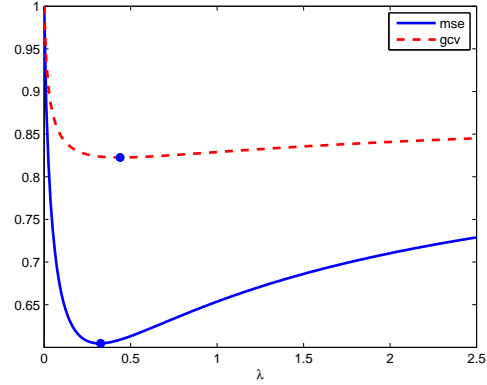
The formula is

$$X_t = f_1(X_{t-1}) + \dots + f_p(X_{t-p}) + \varepsilon_t.$$

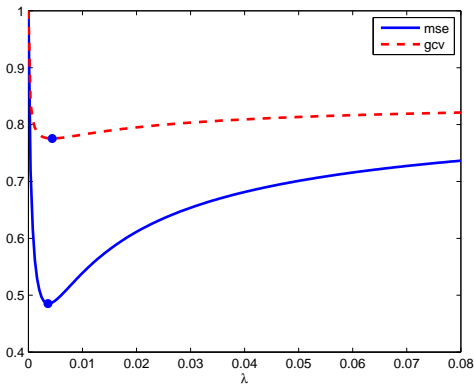
Time Series 1 : MSE(λ) vs. GCV(λ)



Time Series 2 : MSE(λ) vs. GCV(λ)



Time Series 3 : MSE(λ) vs. GCV(λ)



Time Series 4 : MSE(λ) vs. GCV(λ)

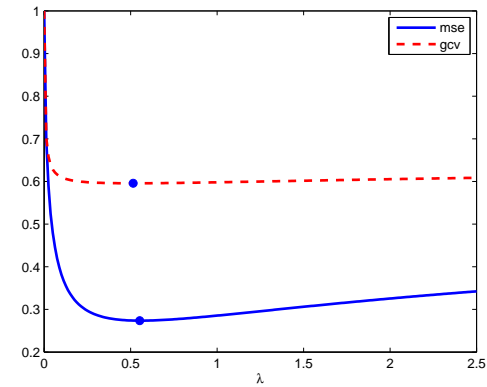


Figure 4: The functions GCV(\cdot) and MSE(\cdot) of the four time series. The GCV and MSE achieve minima at $(0.3317, 0.3015)$, $(0.4397, 0.3266)$, $(0.0044, 0.0036)$ and $(0.5151, 0.5528)$ respectively in the above four cases. The minima are marked with circles. For comparison, the maximal values of functions GCV and MSE are normalized to 1.

- FAR(p, d): functional coefficient autoregressive model (Chen and Tsay 1993) with p lags and X_{t-d} being the model dependent variable (see Fan and Yao (2003, pp. 318) for additional details). The formula is

$$X_t = f_1(X_{t-d})X_{t-1} + \cdots + f_p(X_{t-d})X_{t-p} + \varepsilon_t.$$

- AR(p): autoregressive model with p lags.
- TAR($p_1, p_1; d$): threshold autoregressive model (Tong 1990, Section 3.3), where p_1 and p_2 are autoregressive orders for low and high regime respectively, and d is the time delay or time lag for the threshold variable. The formula is

$$X_t = \begin{cases} b_{10} + b_{11}X_{t-1} + \cdots + b_{1p_1}X_{t-p_1} + \varepsilon_t, & \text{if } X_{t-d} \leq c, \\ b_{20} + b_{21}X_{t-1} + \cdots + b_{2p_2}X_{t-p_2} + \varepsilon_t, & \text{if } X_{t-d} > c. \end{cases}$$

- Loess(p)(or Lowess(p)): locally weighted scatterplot smoothing with p covariates (Cleveland 1979, 1988). It is a local polynomial regression with tricubic weighting.
- Locpoly(p): multivariate local polynomial regression with Epanechnikov kernel. p is the number of covariates (Fan and Gijbels 1996; Yatchew 2003; Pagan and Ullah 1999).

All the models listed above, other than AR model, are nonlinear models. For Loess, Locpoly and our model, the first order (linear) prediction is utilized for all the numerical experiments.

In each experiment for each model, we produce 300 simulations. In each simulation, a time series with length 602 is generated from the simulation model. We make the one-step and two-step predictions based on the first 600 data points, and then calculate the three types of prediction errors by comparing with the observed values, i.e., the last two generated data points. The mean, median, and standard deviation of the absolute prediction errors are computed over the 300 simulations for the each type of prediction errors. [The statistics are denoted by mean, median, std respectively in the tables of this section.] The mean

square prediction error (MSPE) for each type of prediction errors is also calculated. In all the simulations, we fix the number of the nearest neighbors in our method as $k = 20$.

The software for FAR was downloaded from

<http://orfe.princeton.edu/~jqfan/fan/nls.html>. (A supplement of Fan and Yao (2003).)

Implementation of TAR and AAR is based on an online software package that is downloadable at

<http://cran.r-project.org/src/contrib/Descriptions/tsDyn.html>. (Maintainer: Antonio, Fabio Di Narzo.)

Implementation of Locpoly is based on an online software package that is downloadable at

<http://cran.r-project.org/src/contrib/Descriptions/JLLprod.html>. (Maintainer: David Tomás, Jacho-Chávez.)

Implementation of AR can be found in the Matlab system identification toolbox , and Loess is implemented based on the function “loess” in standard R package “stats.”

6.3.1 A Functional Coefficient Autoregressive Model

The first model is model (6.17), an FAR model. The model is initialized with $X_0 = X_1 = X_2 = X_3 = 2$, and the first 100 data points are warm-up period.

Table 1 shows that there is no significant difference between our method (HRM) and the method specific for FAR model. “1-s” stands for one-step prediction; “Ite” stands for iterative two-step prediction; “Dir” stands for direct two-step prediction. Among nonlinear ones, FAR and our model give the most accurate performance for this example. Linear AR model does not fit the nonlinear situation very well.

Table 1: Prediction error under an FAR model.

| | <u>HRM</u> | | | <u>AAR(4)</u> | | <u>FAR(4,1)</u> | | |
|--------|------------|------|------|---------------|------|-----------------|------|------|
| | 1-s | Ite | Dir | 1-s | Ite | 1-s | Ite | Dir |
| mean | 0.16 | 0.18 | 0.18 | 0.19 | 0.20 | 0.16 | 0.17 | 0.20 |
| median | 0.13 | 0.15 | 0.15 | 0.16 | 0.17 | 0.14 | 0.14 | 0.17 |
| std | 0.12 | 0.13 | 0.14 | 0.15 | 0.15 | 0.12 | 0.13 | 0.15 |
| MSPE | 0.04 | 0.05 | 0.05 | 0.06 | 0.06 | 0.04 | 0.05 | 0.06 |

| | <u>AR(4)</u> | | <u>Loess(4)</u> | | <u>Locpoly(4)</u> | |
|--------|--------------|------|-----------------|------|-------------------|------|
| | 1-s | Dir | 1-s | Ite | 1-s | Ite |
| mean | 0.60 | 0.57 | 0.19 | 0.20 | 0.18 | 0.19 |
| median | 0.53 | 0.45 | 0.16 | 0.17 | 0.14 | 0.15 |
| std | 0.43 | 0.43 | 0.15 | 0.15 | 0.15 | 0.14 |
| MSPE | 0.54 | 0.51 | 0.06 | 0.06 | 0.05 | 0.06 |

6.3.2 A Threshold Autoregressive Model

The second model is a TAR model,

$$X_t = \begin{cases} 0.62 + 1.25X_{t-1} - 0.43X_{t-2} + 0.3X_{t-3} - 0.2X_{t-4} + \varepsilon_t, & \text{if } X_{t-2} \leq 2.25, \\ 2.25 + 1.52X_{t-1} - 1.24X_{t-2} - 1.25X_{t-3} + 0.4X_{t-4} + \varepsilon_t, & \text{if } X_{t-2} > 2.25, \end{cases}$$

where $\{\varepsilon_t\}$ are i.i.d. from $N(0, 1.5^2)$. The model is initialized with $X_0 = X_1 = X_2 = X_3 = 0$, and the first 100 data points are warm-up period.

From Table 2, we can see that TAR outperforms all the other methods. This is not surprising given the data generation mechanism. Because TAR may be considered as a special case of FAR, FAR(4,2) is applied for the example. Our method performs similarly to AAR and FAR. Loess and Locpoly have the worst performance among the nonlinear methods. Once again linear AR model does not fit the nonlinear situation well.

One reason for the underperformance of our method possibly is the discontinuity of the underlying model on boundaries. Recall our method assumes that the underlying function f is differentiable, and we penalize on its hessian. It will be interesting to further study

Table 2: Prediction error for a TAR model.

| | <u>HRM</u> | | | <u>AAR(4)</u> | | <u>TAR(4,4;2)</u> | |
|--------|------------|------|------|---------------|------|-------------------|------|
| | 1-s | Ite | Dir | 1-s | Ite | 1-s | Ite |
| mean | 1.42 | 2.30 | 2.31 | 1.39 | 2.30 | 1.25 | 2.14 |
| median | 1.10 | 1.87 | 1.94 | 1.10 | 1.85 | 1.03 | 1.72 |
| std | 1.32 | 1.90 | 1.95 | 1.27 | 1.84 | 1.00 | 1.55 |
| MSPE | 3.75 | 8.89 | 9.15 | 3.55 | 8.63 | 2.59 | 6.98 |

| | <u>FAR(4,2)</u> | | | <u>AR(4)</u> | | <u>Loess(4)</u> | | <u>Locpoly(4)</u> | |
|--------|-----------------|------|-------|--------------|-------|-----------------|-------|-------------------|-------|
| | 1-s | Ite | Dir | 1-s | Ite | 1-s | Ite | 1-s | Ite |
| mean | 1.45 | 2.43 | 2.48 | 3.76 | 6.08 | 2.16 | 4.45 | 1.69 | 2.49 |
| median | 1.15 | 1.81 | 1.82 | 2.62 | 4.14 | 1.79 | 3.57 | 1.30 | 2.00 |
| std | 1.23 | 2.01 | 2.38 | 4.07 | 6.79 | 1.72 | 3.36 | 1.53 | 2.07 |
| MSPE | 3.60 | 9.95 | 11.84 | 30.66 | 82.94 | 7.61 | 31.10 | 5.20 | 10.46 |

the dependence of our method on the regularity of the underlying model.

6.3.3 Other More Generic Models

The above two examples show that even when the data are generated from a perfect model, e.g. AAR, TAR, or FAR, our method still produce comparable prediction results with the original generating model.

The third model is

$$\begin{aligned}
X_t = & -X_{t-4} \exp(-2X_{t-3}^2) + \frac{1}{1 + 4X_{t-2}^2} \cos(1.5X_{t-1})X_{t-1} \\
& + \frac{X_{t-3}}{1 + 4X_{t-1}^2} + \frac{\exp(1.5(X_{t-4} - 1))}{1 + \exp(1.5(X_{t-4} - 1))} + \varepsilon_t,
\end{aligned} \tag{6.18}$$

where $\{\varepsilon_t\}$ are i.i.d. from $N(0, 0.5^2)$. The model is initialized with $X_j \sim N(0, 0.2^2)$, where $j = 0, \dots, 3$, and the first 100 data points are warm-up period. The fourth model is

$$\begin{aligned}
X_t = & [0.2 + (0.3 + X_{t-3}) \exp(-4X_{t-4}^2)]X_{t-1} \\
& + [-0.4 - (0.7 + 1.3X_{t-3}) \exp(-4X_{t-4}^2)]X_{t-2} + \varepsilon_t,
\end{aligned} \tag{6.19}$$

where $\{\varepsilon_t\}$ are i.i.d. from $N(0, 0.5^2)$. The model is initialized with $X_j \sim N(0, 1)$, where $j = 0, \dots, 3$, and the first 100 data points are warm-up period.

The above two examples are nonlinear and can not be included by AAR, FAR, or TAR. Table 3 demonstrates that our method outperforms all these three methods in both one-step and iterative two-step predictions.

All of Loess, Locpoly and our method are nonlinear models, and do not require any specific model structure. Although all these three methods are local approaches, the great advantage of our method is that the penalty term of hessian functional can also take into account of the global properties of the data. For the above four examples, Loess always has the worst performance among the three. Occasionally, our method and Locpoly have the comparable performance, e.g., for the one-step prediction in the third example, but our method often performs better than Locpoly, e.g., in the first two examples.

The third and fourth examples illustrate the flexibility of our method, as our method does not enforce any specific structure on the model. For some cases when the data are not generated from a perfect model, e.g. AAR, TAR, or FAR, our method can generate more accurate prediction results than the other models.

6.4 Real Datasets

We apply our method to some well-studied datasets. Comparison with other reported works is carried out. In many cases, we outperform models that are used by other researchers.

6.4.1 Sunspot Data

Sunspot data is well studied in the literature ((Fan and Yao 2003; Chen and Tsay 1993) and many more). Table 8.5 in Fan and Yao (2003) summarizes results for several previous models, including:

- FAR-1, a functional coefficient autoregressive model fitted via local polynomial methods, as specified by equation (8.19) together with Figure 8.5 in Fan and Yao (2003);

- FAR-2, a functional coefficient autoregressive model fitted by Chen and Tsay (1993), with exact formula given in (8.18) in Fan and Yao (2003);
- TAR, a threshold autoregressive model that is specified in (8.20) in Fan and Yao (2003).

In all the above models, the number of lags is $p = 8$. In our model, we choose $p = 6$. (We choose $p = 6$ because the GCV and prediction errors seem to be stabilized at this point.) The number of the nearest neighbors $k = 29$ is chosen by minimizing the aforementioned generalized cross validation function $\text{GCV}(\cdot)$ as a function of both λ and k .

Table 4 presents prediction errors of our method (column HRM) together with errors of the above three methods (copied from Fan and Yao (2003, Table 8.5)). Our method generates the smallest one-step average absolute prediction error. Our average absolute prediction error of two-step prediction is slightly worse than FAR-1. However, we still outperform FAR-2 and TAR. Note the above is achieved by using six (instead of eight) predictors in our method.

6.4.2 Blowfly Data

We apply our method to the blowfly data. It is known that the first 206 data points are nonlinear, and the remaining data points are almost linear (Tsay 1988). Thus we use the first 195 data points as training data, then make postsample prediction for data points 196 to 210. Log transformation is taken on the blowfly data in accordance with the literature. To get rid of the round-off error of calculating optimal λ for GCV in our method, we multiply the log data by 10. The results are reported in Table 5.

Other four models are compared. A threshold autoregressive model $\text{TAR}(1, 3; 8)$ is suggested in Tong (1990, pp. 337), We apply the model with the same order but refit the model (column TAR), because the original model is applied to a different segment of the time series. To verify the order of the TAR model for our training data, We automatically select the best order of the TAR model with respect to the pooled AIC criteria, using the “selectSETAR” procedure in R package “tsDyn”. By fixing the threshold variable as

the 8th lag, TAR(2, 3; 8) is selected. As the second variable in the lower regime is not significant, it means that TAR(1, 3; 8) is almost same as TAR(2, 3; 8).

The second and third models are functional coefficient autoregressive (FAR) models with different number of dependent variables:

- FAR-1: $X_t = f_0(X_{t-8}) + f_1(X_{t-8})X_{t-1} + f_2(X_{t-8})X_{t-2} + f_3(X_{t-8})X_{t-3} + \varepsilon_t$.
- FAR-2: $X_t = f_0(X_{t-8}) + f_1(X_{t-8})X_{t-1} + f_2(X_{t-8})X_{t-2} + f_3(X_{t-8})X_{t-3} + f_4(X_{t-8})X_{t-4} + \varepsilon_t$.

A similar model with two dependent variables is given in Xia and Li (1999). Note that the above FAR-1 model corresponds to the TAR model given in Tong (1990). Moreover, we observe that more dependent variables can dramatically reduce the prediction errors of the FAR model for our training data. The fourth one is a standard autoregressive model with 8 lags.

For comparison, applying our algorithm, we fit the following generic model,

$$X_t = f(X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}, X_{t-8}) + \varepsilon.$$

The number of the nearest neighbors $k = 21$ is chosen by minimizing the GCV function.

From Table 5, it is observed that when our model is applied, both the average one-step prediction error and the average two-step prediction error are better than those of four competing methods. This example once again demonstrates the powerfulness of our algorithm in time series prediction.

7 DISCUSSION

An alternative penalty is the Laplacian:

$$\mathcal{L}f = \int_{\Omega} \sum_i |f_{ii}(z)|^2 dz.$$

The difference between the hessian ($\mathcal{H}f$) and Laplacian ($\mathcal{L}f$) is that the latter does not consider the cross terms. It is known that $\mathcal{H}f$ is translation and rotation invariant, while

$\mathcal{L}f$ is not. Hence, we prefer the hessian. Another interesting alternative is to consider a modified function:

$$\int_{\Omega} \sum_{i,j} |f_{ij}(z)| dz. \quad (7.20)$$

Note the above penalty function uses sum of absolute values, instead of sum of squares. Penalty function (7.20) may have some nice property. However, it is known that minimizing the sum of absolute values (i.e., the ℓ_1 norm) is much more computational demanding than minimizing the sum of squares (i.e., the ℓ_2 norm). A penalty function like (7.20) has been explored in triograms (Koenker and Mizera 2004).

8 CONCLUSION

We introduce a hessian regularized nonlinear time series model for prediction in time series. The approach is especially powerful when the number of dependent variables is greater than three, which can not be handled by natural cubic splines and thin plate splines. Moreover, our approach is nonlinear and nonparametric, and does not enforce any specific structure on the model. Both the theoretical and simulation results provide a strong verification and support of our model.

References

- Cai, Z., Fan, J., and Yao, Q. (2000), “Functional-coefficient regression models for nonlinear time series models,” *J. Am. Statist. Ass.*, 95, 941–956.
- Chen, J. (2007), “Theoretical Results and Applications Related to Dimension Reduction,” Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, downloadable at etd.gatech.edu.
- Chen, R. and Tsay, R. S. (1993), “Functional-coefficient autoregressive models,” *J. Am. Statist. Ass.*, 88, 298–308.
- Cleveland, W. S. (1979), “Robust locally weighted regression and smoothing scatterplots,” *J. Amer. Statist. Assoc.*, 74, 829–836.

- (1988), “Locally weighted regression: an approach to regression analysis by local fitting,” *J. Amer. Statist. Assoc.*, 83, 596–610.
- Fan, J. and Gijbels, I. (1996), *Local polynomial modelling and its applications*, vol. 66 of *Monographs on statistics & applied probability*, New York, NY: Chapman & Hall.
- Fan, J. and Yao, Q. (2003), *Nonlinear Time Series: Nonparametric and Parametric Methods*, New York: Springer.
- Fan, J., Yao, Q., and Cai, Z. (2003), “Adaptive varying-coefficient linear models,” *J. R. Statist. Soc. B*, 65, 57–80.
- Green, P. J. and Silverman, B. W. (1994), *Nonparametric regression and generalized linear models: a roughness penalty approach*, vol. 58 of *Monographs on statistics and applied probability*, New York, NY: Chapman & Hall.
- Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized additive models*, vol. 43 of *Monographs on statistics and applied probability*, New York, NY: Chapman and Hall.
- Johnstone, I. M. (2004), *Function Estimation and Gaussian Sequence Models*, un published monograph ed., available at www-stat.stanford.edu/~imj/baseb.pdf.
- Koenker, R. and Mizera, I. (2004), “Penalized triograms: total variation regularization for bivariate smoothing,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 66, 145–163.
- Lange, K. (1999), *Numerical analysis for statisticians*, New York, NY: Springer.
- Pagan, A. and Ullah, A. (1999), *Nonparametric Econometrics*, New York: Cambridge Universtiy Press.
- Reinsch, C. H. (1967), “Smoothing By Spline Functions,” *Numerische Mathematik*, 10, 177–183.
- Tong, H. (1990), *Nonlinear Time Series: A Dynamical System Approach*, New York: Oxford University Press.

Tsay, R. S. (1988), “Non-linear time series analysis of blowfly population,” *J. Time Ser. Anal.*, 9, 247–264.

Wahba, G. (1990), *Spline Models for Observational Data (CBMS-NSF Regional Conference Series in Applied Mathematics)*, Philadelphia, PA: SIAM: Society for Industrial and Applied Mathematics.

Xia, Y. and Li, W. K. (1999), “On the estimation and testing of functional-coefficient linear models,” *Statistica Sinica*, 9, 735–757.

Yatchew, A. (2003), *Semiparametric Regression for the Applied Econometrician*, New York: Cambridge University.

Table 3: Prediction error under two nonlinear models.

(a) Under model (6.18)

| | <u>HRM</u> | | | <u>AAR(4)</u> | | <u>AR(4)</u> | | <u>Loess(4)</u> | | <u>Locpoly(4)</u> | |
|--------|------------|------|------|---------------|------|--------------|------|-----------------|------|-------------------|------|
| | 1-s | Ite | Dir | 1-s | Ite | 1-s | Dir | 1-s | Ite | 1-s | Ite |
| mean | 0.47 | 0.56 | 0.56 | 0.52 | 0.64 | 0.95 | 0.89 | 0.58 | 0.62 | 0.46 | 0.59 |
| median | 0.39 | 0.41 | 0.43 | 0.44 | 0.53 | 0.82 | 0.76 | 0.47 | 0.50 | 0.38 | 0.44 |
| std | 0.35 | 0.55 | 0.52 | 0.40 | 0.58 | 0.73 | 0.69 | 0.45 | 0.56 | 0.35 | 0.56 |
| MSPE | 0.34 | 0.62 | 0.58 | 0.43 | 0.74 | 1.43 | 1.26 | 0.54 | 0.70 | 0.33 | 0.66 |

| | <u>FAR(4,1)</u> | | | <u>FAR(4,2)</u> | | | <u>FAR(4,3)</u> | | | <u>FAR(4,4)</u> | | |
|--------|-----------------|------|------|-----------------|------|------|-----------------|------|------|-----------------|------|------|
| | 1-s | Ite | Dir | 1-s | Ite | Dir | 1-s | Ite | Dir | 1-s | Ite | Dir |
| mean | 0.51 | 1.00 | 0.62 | 0.59 | 0.97 | 0.63 | 0.56 | 1.03 | 0.58 | 0.57 | 1.00 | 0.59 |
| median | 0.41 | 0.82 | 0.48 | 0.49 | 0.81 | 0.51 | 0.44 | 0.92 | 0.46 | 0.45 | 0.81 | 0.48 |
| std | 0.39 | 0.78 | 0.63 | 0.47 | 0.77 | 0.53 | 0.55 | 0.77 | 0.50 | 0.45 | 0.75 | 0.55 |
| MSPE | 0.42 | 1.61 | 0.79 | 0.57 | 1.54 | 0.67 | 0.62 | 1.66 | 0.58 | 0.53 | 1.56 | 0.65 |

(b) Under model (6.19)

| | <u>HRM</u> | | | <u>AAR(4)</u> | | <u>AR(4)</u> | | <u>Loess(4)</u> | | <u>Locpoly(4)</u> | |
|--------|------------|------|------|---------------|------|--------------|------|-----------------|------|-------------------|------|
| | 1-s | Ite | Dir | 1-s | Ite | 1-s | Dir | 1-s | Ite | 1-s | Ite |
| mean | 0.52 | 0.49 | 0.52 | 0.65 | 0.53 | 0.82 | 0.71 | 1.53 | 1.12 | 0.52 | 0.48 |
| median | 0.40 | 0.37 | 0.36 | 0.48 | 0.39 | 0.57 | 0.52 | 0.61 | 0.54 | 0.40 | 0.38 |
| std | 0.81 | 0.50 | 0.74 | 1.08 | 0.52 | 1.08 | 0.69 | 5.71 | 2.71 | 0.90 | 0.45 |
| MSPE | 0.92 | 0.48 | 0.82 | 1.59 | 0.54 | 1.85 | 0.97 | 34.85 | 8.54 | 1.09 | 0.43 |

| | <u>FAR(4,1)</u> | | | <u>FAR(4,2)</u> | | | <u>FAR(4,3)</u> | | | <u>FAR(4,4)</u> | | |
|--------|-----------------|------|------|-----------------|------|------|-----------------|------|------|-----------------|------|------|
| | 1-s | Ite | Dir | 1-s | Ite | Dir | 1-s | Ite | Dir | 1-s | Ite | Dir |
| mean | 0.64 | 0.61 | 0.62 | 0.80 | 0.56 | 0.52 | 0.61 | 0.54 | 0.60 | 0.67 | 0.58 | 0.56 |
| median | 0.47 | 0.43 | 0.42 | 0.46 | 0.39 | 0.40 | 0.44 | 0.41 | 0.42 | 0.48 | 0.42 | 0.38 |
| std | 1.00 | 0.94 | 0.98 | 3.33 | 0.80 | 0.55 | 0.89 | 0.60 | 0.65 | 1.10 | 0.60 | 0.63 |
| MSPE | 1.42 | 1.25 | 1.35 | 11.69 | 0.94 | 0.57 | 1.16 | 0.65 | 0.77 | 1.65 | 0.70 | 0.71 |

Table 4: Prediction Errors for Sunspot Data.

| Year | <u>HRM</u> | | <u>FAR-1</u> | | <u>FAR-2</u> | | <u>TAR</u> | |
|------|------------|------|--------------|------|--------------|------|------------|------|
| | 1-s | Ite | 1-s | Ite | 1-s | Ite | 1-s | Ite |
| 1980 | 1.5 | 1.5 | 1.4 | 1.4 | 13.8 | 13.8 | 5.5 | 5.5 |
| 1981 | 7.8 | 9.1 | 11.4 | 10.4 | 0.0 | 3.8 | 1.3 | 0.0 |
| 1982 | 9.6 | 14.1 | 15.7 | 20.7 | 10.0 | 16.4 | 19.5 | 22.1 |
| 1983 | 8.1 | 4.0 | 10.3 | 0.7 | 3.3 | 0.8 | 4.8 | 6.5 |
| 1984 | 3.3 | 1.0 | 1.0 | 1.5 | 3.8 | 5.6 | 14.8 | 15.9 |
| 1985 | 10.3 | 8.1 | 2.6 | 3.4 | 4.6 | 1.7 | 0.2 | 2.7 |
| 1986 | 0.4 | 7.3 | 3.1 | 0.7 | 1.3 | 2.5 | 5.5 | 5.4 |
| 1987 | 9.5 | 9.1 | 12.3 | 13.1 | 21.7 | 23.6 | 0.7 | 17.5 |
| AAPE | 6.3 | 6.8 | 7.2 | 6.5 | 7.3 | 8.3 | 6.6 | 9.5 |

Table 5: Prediction Errors for Blowfly Data.

| obs. | <u>HRM</u> | | <u>TAR(1,3;8)</u> | | <u>FAR-1</u> | | <u>FAR-2</u> | | <u>AR(8)</u> | |
|------|------------|-------|-------------------|-------|--------------|-------|--------------|-------|--------------|-------|
| | 1-s | Ite | 1-s | Ite | 1-s | Ite | 1-s | Ite | 1-s | Ite |
| 196 | 0.040 | 0.196 | 0.048 | 0.175 | 0.112 | 0.236 | 0.089 | 0.227 | 0.003 | 0.266 |
| 197 | 0.052 | 0.008 | 0.035 | 0.031 | 0.012 | 0.174 | 0.009 | 0.139 | 0.075 | 0.144 |
| 198 | 0.004 | 0.076 | 0.014 | 0.033 | 0.062 | 0.081 | 0.057 | 0.071 | 0.087 | 0.078 |
| 199 | 0.010 | 0.003 | 0.015 | 0.035 | 0.029 | 0.125 | 0.023 | 0.113 | 0.031 | 0.117 |
| 200 | 0.078 | 0.061 | 0.092 | 0.113 | 0.097 | 0.142 | 0.090 | 0.126 | 0.066 | 0.033 |
| 201 | 0.136 | 0.261 | 0.163 | 0.290 | 0.129 | 0.266 | 0.138 | 0.257 | 0.271 | 0.241 |
| 202 | 0.049 | 0.240 | 0.063 | 0.287 | 0.060 | 0.215 | 0.071 | 0.232 | 0.322 | 0.410 |
| 203 | 0.000 | 0.040 | 0.081 | 0.004 | 0.045 | 0.024 | 0.047 | 0.033 | 0.121 | 0.442 |
| 204 | 0.108 | 0.107 | 0.071 | 0.040 | 0.082 | 0.032 | 0.063 | 0.011 | 0.051 | 0.416 |
| 205 | 0.046 | 0.158 | 0.036 | 0.134 | 0.031 | 0.114 | 0.012 | 0.077 | 0.088 | 0.199 |
| 206 | 0.180 | 0.130 | 0.175 | 0.124 | 0.177 | 0.146 | 0.159 | 0.147 | 0.208 | 0.167 |
| 207 | 0.234 | 0.016 | 0.298 | 0.059 | 0.186 | 0.018 | 0.192 | 0.038 | 0.040 | 0.039 |
| 208 | 0.007 | 0.286 | 0.033 | 0.441 | 0.028 | 0.176 | 0.019 | 0.190 | 0.210 | 0.078 |
| 209 | 0.081 | 0.090 | 0.107 | 0.061 | 0.099 | 0.139 | 0.055 | 0.081 | 0.167 | 0.174 |
| 210 | 0.111 | 0.210 | 0.154 | 0.301 | 0.134 | 0.290 | 0.148 | 0.235 | 0.350 | 0.114 |
| Ave. | 0.076 | 0.126 | 0.092 | 0.142 | 0.086 | 0.145 | 0.078 | 0.132 | 0.139 | 0.195 |