Volume 16 Issue 24
Originally issued as F5

Global Journal of Science Frontier Research: F
Mathematics and Decision Sciences
Volume 16 Issue 5 Version 1.0 Year 2016
Type : Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

A New Method for Estimating Smooth Regression Functions

By Eunji Lim & Annerys Matos
Kean University, United States

Abstract- We propose a new method for estimating a regression function from noisy data when the underlying function is known to satisfy a certain smoothness condition. The proposed method fits a function to the data set so that the roughness of the fitted function is minimized while ensuring that the sum of the absolute deviations of the fitted function from the data points does not exceed a certain limit. It is shown that the fitted function exists and can be computed by solving a quadratic program. Numerical results demonstrate that the proposed method generates more efficient estimates than its alternative in terms of the mean square error and the amount of time required to compute the fit.

Keywords: *nonparametric regression, smoothing spline, quadratic programming, penalized least, squares regression.*

GJSFR-F Classification : *FOR Code : 62J05*

ANEWMETHODFORESTIMATINGSMOOTHREGRESSIONFUNCTIONS

Strictly as per the compliance and regulations of :

© 2016. Eunji Lim & Annerys Matos. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

A New Method for Estimating Smooth Regression Functions

Eunji Lim [α] & Annerys Matos [σ]

Abstract- We propose a new method for estimating a regression function from noisy data when the underlying function is known to satisfy a certain smoothness condition. The proposed method fits a function to the data set so that the roughness of the fitted function is minimized while ensuring that the sum of the absolute deviations of the fitted function from the data points does not exceed a certain limit. It is shown that the fitted function exists and can be computed by solving a quadratic program. Numerical results demonstrate that the proposed method generates more efficient estimates than its alternative in terms of the mean square error and the amount of time required to compute the fit.

Keywords: nonparametric regression, smoothing spline, quadratic programming, penalized least, squares regression.

I. Introduction

When estimating an unknown function from noisy data, the underlying function is often assumed to be smooth in the independent variables. For example, the price of a stock option and its second derivative are often assumed to be smooth in the underlying stock price over a domain of interest: see Section 3.1 of Lim and Attallah (2016) for an example. In this paper, we consider the problem of estimating a function $f_* : \mathbb{R} \to \mathbb{R}$ over a domain $[a, b)$ of interest, which is known to have a square integrable kth derivative ($k \geq 2$), by observing a data set $((x_i, Y_{ij}) : 1 \leq i \leq n, 1 \leq j \leq m)$ satisfying

$$Y_{ij} = f_*(x_i) + \epsilon_{ij},$$

where the x_i's satisfy $a < x_1 < \cdots < x_n < b$, and the ϵ_{ij}'s are independent and identically distributed (iid) random variables with a mean of 0 and a variance of $\sigma^2 < \infty$.

One of the most popular approaches to estimating the underlying function f_* from the data set is to find a smooth function that is close to the data set by solving the following optimization problem:

$$\text{Minimize} \qquad (1/n)\sum_{i=1}^{n}\left(g(x_i) - \overline{Y}_i\right)^2 + \lambda \int_a^b \left\{g^{(k)}(x)\right\}^2 dx \tag{1}$$

over $g \in \mathcal{D}^k$, where $\overline{Y}_i = \sum_{j=1}^{n} Y_{ij}/m$ for $1 \leq i \leq n$, λ is a non-negative constant,

$$\mathcal{D}^k = \left\{ f : \mathbb{R} \to \mathbb{R} : f \text{ is } k \text{ times differentiable and } \int_a^b \{f^{(k)}(x)\}^2 dx < \infty \right\},$$

Author α σ: Kean University. e-mail: eunji96@gmail.com

© 2016 Global Journals Inc. (US)

and $g^{(k)}$ is the kth derivative of g. The term $(1/n)\sum_{i=1}^{n}(g(x_i)-\overline{Y}_i)^2$ in (1) measures the closeness of the fitted function g to the data set, and the term $\int_a^b\{g^{(k)}(x)\}^2dx$ in (1) measures the "roughness" of the fitted function g. The parameter λ controls the trade-off between the closeness to the data set and the roughness of the fitted function.

Problem (1) appears to be an infinite–dimensional optimization problem at first glance, but the solution to (1) is known to be a piecewise polynomial function of degree $2k-1$ with knots $x_1,\ldots,x_n$ (Theorem 20.1 on page 412 of Györfi et al., 2002). Since the set of piecewise polynomial functions of degree $2k-1$ with knots $x_1,\ldots,x_n$ is finite dimensional, (1) can be reduced to an optimization problem over a finite-dimensional space. In fact, the solution to (1) can be obtained by solving a system of linear equations: see (20.6) on page 412 of Györfi et al. (2002) for details.

Despite the fact that the solution to (1) can be obtained easily, the performance of the solution to (1) is highly sensitive to the choice of λ. Several authors have proposed the method of cross-validation for choosing λ (Wahba and Wold, 1975). In the method of cross-validation, λ is chosen so that it minimizes the average squared error, which is defined by

$$\mathrm{CV}(\lambda)=\sum_{i=1}^{n}\left(\tilde{g}_\lambda^i(x_i)-\overline{Y}_i\right)^2/n$$

for $\lambda\geq 0$, where $\tilde{g}_\lambda^i$ is the solution to (1) with the ith data point, $(x_i,\overline{Y}_i)$, omitted. In order to select the right value of λ, one needs to find the minimizer of $\mathrm{CV}(\lambda)$ over $\lambda\geq 0$. $\mathrm{CV}(\lambda)$ is a nonlinear function in λ in general. Thus, it takes a significant amount of time in practice to find the minimizer of $\mathrm{CV}(\lambda)$.

To overcome the issue of selecting the right value of λ, the solution $\tilde{g}_n$ to the following alternative formulation is preferred in the numerical analysis community:

$$\text{Problem (A):}\quad \text{Minimize}\quad \int_a^b\left\{g^{(k)}(x)\right\}^2dx$$

$$\text{subject to}\quad (1/n)\sum_{i=1}^{n}\left(g(x_i)-\overline{Y}_i\right)^2\leq u_0$$

for some constant u_0 over $g\in\mathcal{D}^k$, which was first proposed by Reinsch (1967). Problem (A) is preferred in the numerical analysis community because a good estimate of u_0 can be easily computed from the data set $((x_i,Y_{ij}):1\leq i\leq n,1\leq j\leq m)$ by using $\sum_{i=1}^{n}S_i^2/(nm)$, where $S_i^2=\sum_{j=1}^{m}\left(Y_{ij}-\overline{Y}_i\right)^2/(m-1)$ for $1\leq i\leq n$: see page 151 of Lim and Attallah (2016) for details.

Recently, Lim and Attallah (2016) showed that the solution to Problem (A) can be computed by solving a convex program. Several efficient algorithms exist for solving convex programming problems, and they provide guaranteed convergence to the global solution; see the Lagrangian method on page 217 of Zangwill (1969) for an example of methods that solve convex programs. Thus, the formulation in Lim and Attallah (2016) enables one to compute the solution to Problem (A) with guaranteed convergence. However, the amount of time required to solve a convex program increases rapidly as $n\to\infty$, and hence, a computationally more efficient formulation is desired.

In this paper, we propose a new formulation that is designed for better computational efficiency. The new formulation replaces the constraint $(1/n)\sum_{i=1}^{n}(g(x_i)-\overline{Y}_i)^2\leq u_0$ in Problem (A) with $(1/n)\sum_{i=1}^{n}|g(x_i)-\overline{Y}_i|\leq g_0$ for some constant g_0. Thus, our proposed estimator is the solution $\hat{g}_n$ to the following optimization problem:

© 2016 Global Journals Inc. (US)

$$\text{Problem (B):} \quad \text{Minimize} \quad \int_a^b \left\{g^{(k)}(x)\right\}^2 dx$$

$$\text{subject to} \quad (1/n)\sum_{i=1}^{n} \left|g(x_i) - \overline{Y}_i\right| \leq g_0$$

Notes

over $g \in \mathcal{D}^k$. Problem (B) can be further transformed into a quadratic program (see Proposition 1 of this paper), so the solution to Problem (B) can be obtained by solving a quadratic program. Quadratic programs are special cases of convex programs with quadratic objective functions and linear constraints, so they can be solved more efficiently than convex programs. Thus, the solution to our new formulation can be computed more efficiently than the solution to Problem (A). Moreover, the constant g_0 appearing in Problem (B) can be estimated from the data set $((x_i, Y_{ij}) : 1 \leq i \leq n, 1 \leq j \leq m)$ readily, so the performance of the proposed estimator $\hat{g}_n$ does not rely on any unnatural parameters; see Section 2.1 of this paper for a discussion of how to estimate g_0. Even though Problem (B) was motivated by the need for better computational efficiency, the numerical experiments in Section 3 show that the proposed estimator not only is computed faster than the solution to Problem (A), but also achieves better mean squared errors, and hence, is a better estimate of the underlying function f_*. Furthermore, the convergence of the proposed estimator to the true function f_* as $n \to \infty$ is demonstrated empirically in Section 3 by showing that the empirical integrated mean square error between $\hat{g}_n$ and f_* converges to 0 as $n \to \infty$. The numerical results in Section 3.2 show that our formulation is successfully applied to a problem of estimating the sensitivities of option prices as functions of the underlying stock price.

This paper is organized as follows. In Section 2.1, we describe how g_0 can be estimated from the data set in more detail. In Section 2.2, we prove that the solution to Problem (B) exists and can be obtained by solving a quadratic program. In Section 3, we compare the performance of the proposed estimator to that of the solution to Problem (A) through numerical experiments. Concluding remarks are included in Section 4.

II Problem Formulation

a) How to Estimate g_0 from the Data Set?

In this section, we provide a heuristic argument on how g_0 can be estimated from the data set $((x_i, Y_{ij}) : 1 \leq i \leq n, 1 \leq j \leq m)$. We start by noticing that $\sum_{j=1}^{m} \epsilon_{ij}/\sqrt{m}$ converges in distribution to $N(0, \sigma^2)$ as $m \to \infty$ by the weak law of large numbers, where $N(0, \sigma^2)$ denotes a normal random variable with a mean of 0 and a variance of σ^2. Hence, if we denote $\sum_{j=1}^{m} \epsilon_{ij}/m$ by $\bar{\epsilon}_i$ for $1 \leq i \leq n$, then $|\bar{\epsilon}_i|$ can be approximated by $\left|N(0, \sigma^2)\right|/\sqrt{m}$ for m sufficiently large. Therefore, the following approximation is possible:

$$\frac{1}{n}\sum_{i=1}^{n} \left|\overline{Y}_i - f_*(x_i)\right| = \frac{1}{n}\sum_{i=1}^{n} |\bar{\epsilon}_i| \approx \frac{1}{n\sqrt{m}}\sum_{i=1}^{n} \left|N(0, \sigma^2)\right|$$

for m sufficiently large. The symbol $\approx$ is used to express "approximate equality" informally. When the ϵ_i's are assumed to be normally distributed, $\sum_{i=1}^{n} \left|N(0, \sigma^2)\right|/n$ converges to $\mathbb{E}\,|\epsilon_{11}|$ as $n \to \infty$ by the strong law of large numbers. Thus, the following approximation is appropriate:

© 2016 Global Journals Inc. (US)

$$\frac{1}{n}\sum_{i=1}^{n}\left|\overline{Y}_i - f_*(x_i)\right| \approx \frac{\mathbb{E}\left|\epsilon_{11}\right|}{\sqrt{m}}$$

for n and m sufficiently large. Furthermore, $\mathbb{E}\left|\epsilon_{11}\right|$ can be estimated from the data set via $\sum_{i=1}^{n}\sum_{j=1}^{m}\left|Y_{ij}-\overline{Y}_i\right|/(mn)$, and hence, a good estimate of g_0 is

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left|Y_{ij}-\overline{Y}_i\right|/(m^{3/2}n). \tag{2}$$

Notes

b) Quadratic Programming Representation of the Proposed Formulation

In this section, we describe how the proposed estimator can be obtained by solving a quadratic program. To make this paper self-contained, we present some preliminary results.

A spline function with degree $r > 1$ with knots $x_1, \ldots, x_n$, where $a < x_1 < \ldots, < x_n < b$, is a function $s : [a, b) \to \mathbb{R}$ having the following two properties: (a) In each of the intervals $[a, x_1), [x_1, x_2), \ldots, [x_{n-1}, x_n), [x_n, b)$, $s(x)$ is given by some polynomial of degree r or less, and (b) $s(x)$ is $r-1$ times continuously differentiable on $[a, b)$. We denote the set of spline functions with degree r by $\mathcal{S}_r([a, b))$. $\mathcal{S}_r([a, b))$ can be spanned by a finite number of elements in $\mathcal{S}_r([a, b))$, so we introduce one of the bases for $\mathcal{S}_r([a, b))$, which is the set of B-splines; the B-splines have bounded supports and produce well–conditioned numerical settings. We need to introduce additional knots $x_{-r}, \ldots, x_0, x_{n+1}, \ldots, x_{n+r+1}$ so that

$$x_{-r} < x_{-r+1} < \cdots < x_0 < a < x_1 < \cdots < x_n < b < x_{n+1} < \cdots < x_{n+r+1}.$$

The B-spline $B_{i,r}$ of degree r is defined recursively by

$$B_{i,0}(x) = \begin{cases} 1, & \text{if } x_i \le x < x_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

for $i = -r, \ldots, n + r$ and $x \in \mathbb{R}$ and

$$B_{i,l}(x) = \frac{x - x_i}{x_{i+l} - x_i} B_{i,l-1}(x) + \frac{x_{i+l+1} - x}{x_{i+l+1} - x_{i+1}} B_{i+1,l-1}(x) \tag{4}$$

for $i = -r, \ldots, n + r - l$, $l = 1, \ldots, r$, and $x \in \mathbb{R}$. By Theorem 14.1 on page 262 of Györfi et al. (2002), $\{B_{i,r} : i = -r, \ldots, n\}$ restricted to $[a, b)$ is a basis of $\mathcal{S}_r([a, b))$. Proposition 1 below proves the existence of the solution to Problem (B) and describes how Problem (B) can be solved through a quadratic program.

Proposition 1 *Assume $2 \le k \le n$. There exists a solution $\hat{g}_n \in \mathcal{D}^k$ to Problem (B). Furthermore, $\hat{g}_n$ has the following representation:*

$$\hat{g}_n(x) = \sum_{i=-(2k-1)}^{n} \hat{c}_i B_{i,2k-1}(x)$$

for $x \in [a, b)$, where $\hat{c}_{-(2k-1)}, \ldots, \hat{c}_n, \hat{y}_1, \ldots, \hat{y}_n, \hat{p}_1, \ldots, \hat{p}_n, \hat{m}_1, \ldots, \hat{m}_n$ is the solution to the following quadratic program in the decision variables $c_{-(2k-1)}, \ldots, c_n, y_1, \ldots, y_n, p_1, \ldots, p_n, m_1, \ldots, m_n \in \mathbb{R}$:

© 2016 Global Journals Inc. (US)

Notes

$$\textit{Minimize} \quad \int_a^b \left(\sum_{i=-(2k-1)}^{n} c_i B_{i,2k-1}^{(k)}(x) \right)^2 dx$$

$$= \sum_{i=-(2k-1)}^{n} \sum_{j=-(2k-1)}^{n} c_i c_j \int_a^b B_{i,2k-1}^{(k)}(x) B_{j,2k-1}^{(k)}(x) dx$$

$$\textit{subject to} \quad \overline{Y}_i - y_i = p_i - m_i, \quad 1 \le i \le n, \tag{5}$$

$$\sum_{i=1}^{n} (p_i + m_i)/n \le g_0,$$

$$\sum_{i=-(2k-1)}^{n} c_i B_{i,2k-1}(X_j) = y_j, \quad j = 1, \ldots, n,$$

$$p_i, m_i \ge 0, \quad 1 \le i \le n.$$

Proof. The existence of the solution to Problem (B) is proven by an argument similar to the proof of Proposition 1 in Lim and Attallah (2016). Next, to show that the solution to (5) exists, we notice that (i) for any $c_{-(2k-1)}, \ldots, c_n$, the objective function of (5) is greater than or equal to 0, and (ii) Problem (5) has a feasible solution because there exist $\overline{c}_{-(2k-1)}, \ldots, \overline{c}_n$ satisfying

$$\sum_{i=-(2k-1)}^{n} \overline{c}_i B_{i,2k-1}(x_j) = \overline{Y}_j$$

for $1 \le j \le n$ by Lemmas 20.2 and 20.3 on pages 415 and 416 of Györfi et al. (2002). By Frank and Wolfe (1956), there exists a solution to (5).

Let $\hat{g}_n$ be a solution to Problem (B). Let $\hat{y}_i = \hat{g}_n(x_i)$ for $1 \le i \le n$. By Lemmas 20.2 and 20.3 on pages 415 and 416 of Györfi et al. (2002), there exists a unique solution $(\hat{c}_{-(2k-1)}, \ldots, \hat{c}_n)$ to the following linear system:

$$\sum_{i=-(2k-1)}^{n} \hat{c}_i B_{i,2k-1}(x_j) = \hat{y}_j \text{ for } 1 \le j \le n,$$

$$\sum_{i=-(2k-1)}^{n} \hat{c}_i B_{i,2k-1}^{(l)}(a) = 0,$$

$$\sum_{i=-(2k-1)}^{n} \hat{c}_i B_{i,2k-1}^{(l)}(b) = 0.$$

Let $\hat{p}_i = \max(\overline{Y}_i - \hat{y}_i, 0)$ and $\hat{m}_i = \max(\hat{y}_i - \overline{Y}_i, 0)$ for $1 \le i \le n$. We will show that $\hat{c}_{-(2k-1)}, \ldots, \hat{c}_n, \hat{y}_1, \ldots, \hat{y}_n, \hat{p}_1, \ldots, \hat{p}_n, \hat{m}_1, \ldots, \hat{m}_n$ is a solution to (5). Let $\overline{c}_{-(2k-1)}, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n, \overline{p}_1, \ldots, \overline{p}_n, \overline{m}_1, \ldots, \overline{m}_n$ be any feasible solution to (5). Without loss of generality, we may assume that either $\overline{p}_i = 0$ or $\overline{m}_i = 0$ for each $1 \le i \le n$. (Otherwise, we replace $\overline{p}_i$ with $\overline{p}_i - \min(\overline{p}_i, \overline{m}_i)$ and $\overline{m}_i$ with $\overline{m}_i - \min(\overline{p}_i, \overline{m}_i)$ for each $1 \le i \le n$.) We notice that $\left|\overline{Y}_i - \overline{y}_i\right| = \overline{p}_i + \overline{m}_i$ for each $1 \le i \le n$. So, if we define $\overline{g}_n : \mathbb{R} \to \mathbb{R}$ by

© 2016 Global Journals Inc. (US)

$$\overline{g}_n(x) = \sum_{i=-(2k-1)}^{n} \overline{c}_i B_{i,2k-1}(x)$$

for $x \in \mathbb{R}$, then $\overline{g}_n$ satisfies $(1/n)\sum_{i=1}^{n} \left|\overline{g}_n(x_i) - \overline{Y}_i\right| \leq g_0$, and hence, is a feasible solution to Problem (B). Thus, $\int_a^b \left\{\hat{g}_n^{(k)}(x)\right\}^2 dx \leq \int_a^b \left\{\overline{g}_n^{(k)}(x)\right\}^2 dx$, and hence, $\hat{c}_{-(2k-1)}, \ldots, \hat{c}_n, \hat{y}_1, \ldots, \hat{y}_n, \hat{p}_1, \ldots, \hat{p}_n, \hat{m}_1, \ldots, \hat{m}_n$ is a solution to (5).

Conversely, let $\tilde{c}_{-(2k-1)}, \ldots, \tilde{c}_n, \tilde{y}_1, \ldots, \tilde{y}_n, \tilde{p}_1, \ldots, \tilde{p}_n, \tilde{m}_1, \ldots, \tilde{m}_n$ be a solution to (5). We will show that $\tilde{g}_n : \mathbb{R} \to \mathbb{R}$ defined by

$$\tilde{g}_n(x) = \sum_{i=-(2k-1)}^{n} \tilde{c}_i B_{i,2k-1}(x)$$

for $x \in \mathbb{R}$ is a solution to Problem (B), or equivalently, $\int_a^b \left\{\tilde{g}_n^{(k)}(x)\right\}^2 dx \leq \int_a^b \left\{\hat{g}_n^{(k)}(x)\right\}^2 dx$ for any solution $\hat{g}_n$ to Problem (B). First, we note that $\tilde{g}_n$ is a feasible solution to Problem (B) since we may assume that either $\tilde{p}_i = 0$ or $\tilde{m}_i = 0$ for each $1 \leq i \leq n$. For any solution $\hat{g}_n$ to Problem (B), let $\hat{y}_i = \hat{g}_n(X_i)$ for $1 \leq i \leq n$ and define the $\hat{c}_i$'s, $\hat{p}_i$'s, and $\hat{m}_i$'s as before. The $\hat{c}_i$'s, $\hat{y}_i$'s, $\hat{p}_i$'s, and $\hat{m}_i$'s form a feasible solution to (5), so it follows that $\int_a^b \left\{\tilde{g}_n^{(k)}(x)\right\}^2 dx \leq \int_a^b \left\{\hat{g}_n^{(k)}(x)\right\}^2 dx$. □

III. Numerical Results

In this section, we compare the performance of the proposed estimator to that of the solution to Problem (A). In Section 3.1, we consider the case where f_* is given by a polynomial function of degree 5. In Section 3.2, f_* is the expected payoff of a certain equity-linked security. In both cases, we compute the empirical integrated mean square error and the amount of time required to compute the estimators.

All of the simulations are conducted on a 64-bit computer with an Intel(R) Core(TM) i7-6700K CPU at 4 GHz and a 32GB RAM. All of the simulations are programmed in MATLAB R2010a.

When solving Problems (A) and (B), one needs to evaluate the B-splines and the integration of the product of their kth derivatives. The B-splines can be evaluated recursively through Equations (3) and (4). The kth derivative of the B-spline can be evaluated recursively through the following relation: for a B-spline of degree r,

$$dB_{i,r}(x)/dx = (r/(x_{i+r} - x_i))B_{i,r-1}(x) - (r/(x_{i+r+1} - x_{i+1}))B_{i+1,r-1}(x) \tag{6}$$

for $i = -r, \ldots, n$ and $x \in [a, b)$; see Lemma 14.6 on page 265 of Györfi et al. (2002). Next, we compute $\int_a^b B_{i,2k-1}^{(k)}(x) B_{j,2k-1}^{(k)}(x)dx$ by evaluating $B_{i,2k-1}^{(k)}$ using the recursion in (6) and by numerically evaluating the integration.

a) A Stylized Model

We consider the case where $f_*(x) = x(x-0.5)(x+0.5)(x-1.05)(x+1.05)$ for $x \in \mathbb{R}$, $x_i = i/n - 1/(2n)$ for $1 \leq i \leq n$, $Y_{ij} = f(x_i) + \epsilon_{ij}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$, and the ϵ_{ij}'s are iid random variables, each of which is normally distributed with a mean of 0 and a variance of 4. The proposed estimator $\hat{g}_n$ is computed as the solution to Problem (B) with $m = 100$, $k = 4$, and g_0 estimated from (2).

© 2016 Global Journals Inc. (US)

The solution $\tilde{g}_n$ to Problem (A) is computed from Problem (A) with $m = 100$, $k = 4$, and u_0 estimated from $\sum_{i=1}^{n} S_i^2/(nm)$, where $S_i^2 = \sum_{j=1}^{m}(Y_{ij} - \overline{Y}_i)^2/(m-1)$ for $1 \leq i \leq n$. Both Problems (A) and (B) are solved with CVX, a package for specifying and solving convex programs (Grant and Boyd, 2014). To measure the accuracy of the proposed estimator, we compute the following empirical integrated mean square error (EIMSE):

$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{g}_n(x_i) - f_*(x_i)\right)^2. \tag{7}$$

The EIMSE of $\tilde{g}_n$ is computed similarly by $\sum_{i=1}^{n}\left(\tilde{g}_n(x_i) - f_*(x_i)\right)^2/n$.

Table 1 reports the 95% confidence intervals of the EIMSE and the average amounts of time required to compute $\hat{g}_n$ and $\tilde{g}_n$, based on 300 iid replications, for a variety of n values. The proposed estimator produces lower EIMSE values and is computed in less time than $\tilde{g}_n$.

Table 1 : The 95% confidence intervals of the EIMSE and the amounts of computer time required to compute the proposed estimator and the solution to Problem (A) when $f_*(x) = x(x-0.5)(x+0.5)(x-1.05)(x+1.05)$.

	Solution to Problem (A)		Proposed Estimator	
n	EIMSE	Time (sec)	EIMSE	Time (sec)
10	0.0168 ± 0.0014	0.097	0.0156 ± 0.0013	0.094
20	0.0090 ± 0.0007	0.099	0.0081 ± 0.0007	0.097
40	0.0047 ± 0.0004	0.111	0.0041 ± 0.0003	0.107
80	0.0032 ± 0.0003	0.129	0.0023 ± 0.0002	0.125

Figure 1 displays the graphs of f_*, the $\overline{Y}_i$'s, $\hat{g}_n$, and $\tilde{g}_n$ on the left side and the graphs of $f_*^{(2)}$, $\hat{g}_n^{(2)}$, and $\tilde{g}_n^{(2)}$ on the right side for the case when $n = 40$. The proposed estimator appears to have a smoother second derivative than $\tilde{g}_n$.

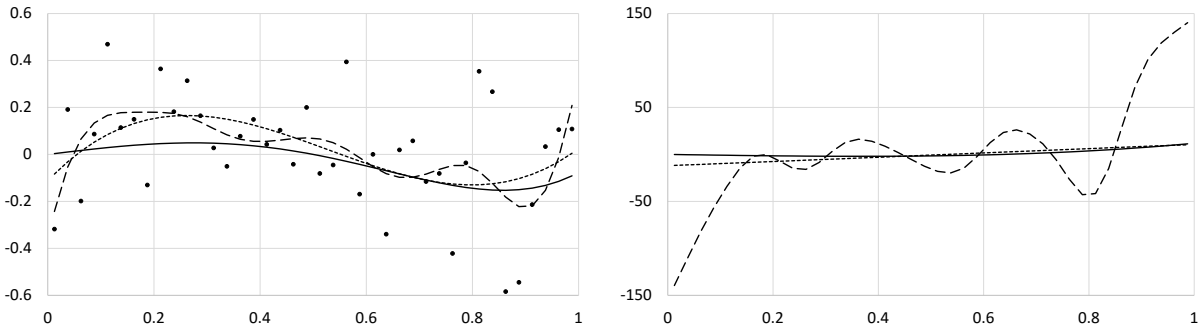

Figure 1 : The horizontal axis is x. On the left side, the solid line is $f_*(x)$, the dots are the $\overline{Y}_i$'s, the dotted line is $\hat{g}_n(x)$, and the dashed line is $\tilde{g}_n(x)$. On the right side, the solid line is $f_*^{(2)}(x)$, the dotted line is $\hat{g}_n^{(2)}(x)$, and the dashed line is $\tilde{g}_n^{(2)}(x)$.

© 2016 Global Journals Inc. (US)

b) Sensitivity Estimation of Option Prices

We consider the case where $f_*(x)$ is the expected payoff of a certain equity-linked security (ELS) when the underlying stock price is $x \geq 0$. The second derivative of f_* plays an important role when financial portfolio managers try to hedge the risks associated with the ELS; see Section 3.1 of Lim and Attallah (2016) for details. In particular, the second derivative of f_* is often assumed to be smooth over a domain of interest. The smoothness of the second derivative is particularly important because portfolio managers use the second derivative to make a decision on whether to buy or sell their ELS in order to hedge the risks, and a smooth second derivative suggests consistent selling strategies; see Section 3.1 of Lim and Attallah (2016) for a detailed explanation. One of the challenges is that there is no-closed form formula for f_* when the payoff structure of the ELS is complex, so simulation must be used to estimate f_* and its second derivative. Thus, the problem boils down to an estimation of the second derivative of f_* as a smooth function over a domain of interest. Since the roughness of a function $g : [a, b) \to \mathbb{R}$ is measured by $\int_a^b \left\{g^{(2)}(x)\right\}^2 dx$, the roughness of the second derivative of g is measured by $\int_a^b \left\{g^{(4)}(x)\right\}^2 dx$. Therefore, our proposed estimator is the solution to the following optimization problem:

$$\begin{aligned} &\text{Minimize} && \int_a^b \left\{g^{(4)}(x)\right\}^2 dx \\ &\text{subject to} && (1/n)\sum_{i=1}^{n} |g(x_i) - Y_i| \leq g_0, \end{aligned} \tag{8}$$

where g_0 can be estimated from (2).

We assume that the ELS is paid off in the following way: The ELS is issued at time 0 and matures at time $T = 365$. We denote the price of the underlying stock at time $t \in [0, T]$ by S_t. There are six days, $d_1, \ldots, d_6$, when early redemption is possible. On day d_i $(1 \leq i \leq 6)$, the ELS expires with a payoff of $\$r_i$ if S_{d_i}/S_0 exceeds some threshold b_i. Otherwise, the ELS does not expire until maturity. If there is no early redemption and S_t/S_0 does not drop below a lower limit b over the entire lifetime of the ELS, then the ELS expires with a payoff of \$1 at maturity. In the rest of the cases, the ELS expires with a payoff of $\$S_T/S_0$ at maturity.

We let $x_i = 90 + (20)(i/n) - (10/n)$ for $1 \leq i \leq n$. For each x_i, a sample path of a geometric Brownian motion is generated as a trajectory of the stock price between now and maturity, and the corresponding payoff of the ELS is computed. Y_{ij} is the payoff computed this way in the jth replication at x_i. The parameters are $d_1 = 61, d_2 = 122, d_3 = 182, d_4 = 243, d_5 = 304, d_6 = 365, b_1 = 0.9, b_2 = 0.9, b_3 = 0.85, b_4 = 0.85, b_5 = 0.8, b_6 = 0.8, r_1 = 1.05, r_2 = 1.10, r_3 = 1.15, r_4 = 1.20, r_5 = 1.25, r_6 = 1.30$, and $b = 0.7$. The remaining time until maturity is 60 days, the annual volatility is 30%, the annual risk–neutral interest rate is 5%, and the initial stock price at time 0 is \$125.

We set $m = 10$, so 10 sample paths for the geometric Brownian motion are generated at each x_i to compute $Y_{i1}, \ldots, Y_{i10}$ for $1 \leq i \leq n$. We compute $\overline{Y}_i = \sum_{j=1}^{10} Y_{ij}/10$ for $1 \leq i \leq n$ and use $(x_1, \overline{Y}_1), \ldots, (x_n, \overline{Y}_n)$ to compute the proposed estimator $\hat{g}_n$ by solving Problem (B) with g_0 estimated from (2). The solution $\tilde{g}_n$ to Problem (A) is computed from Problem (A) with $m = 10$, $k = 4$, and u_0 estimated from $\sum_{i=1}^{n} S_i^2/(nm)$, where $S_i^2 = \sum_{j=1}^{m}(Y_{ij} - \overline{Y}_i)^2/(m-1)$ for $1 \leq i \leq n$. Both Problems (A) and (B) are solved with CVX. To measure the accuracy of the proposed estimate,

© 2016 Global Journals Inc. (US)

Notes

Table 2 : The 95% confidence intervals of the EIMSE and the amounts of computer time required to compute the proposed estimator and the solution to Problem (A) when f_* is the expected payoff of the ELS.

	Solution to Problem (A)		Proposed Estimator	
n	EIMSE	Time (sec)	EIMSE	Time (sec)
10	0.0024 ± 0.0002	0.100	0.0024 ± 0.0002	0.098
20	0.0013 ± 0.0001	0.102	0.0012 ± 0.0001	0.096
40	0.0008 ± 0.0000	0.113	0.0005 ± 0.0000	0.104
80	0.0005 ± 0.0000	0.134	0.0003 ± 0.0000	0.129

we compute the following EIMSE between the underlying function and $\hat{g}_n$:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{g}_n(x_i) - f_*(x_i)\right)^2, \tag{9}$$

where $f_*(x_i)$ is estimated from the average of $2,000,000$ iid replications of Y_{ij} at each x_i. The EIMSE of $\tilde{g}_n$ is computed similarly by $\sum_{i=1}^{n}\left(\tilde{g}_n(x_i) - f_*(x_i)\right)^2/n$.

Table 2 reports the 95% confidence intervals of the EIMSE and the average amounts of time required to compute $\hat{g}_n$ and $\tilde{g}_n$, based on 300 iid replications, for a variety of n values. The proposed estimator produces lower IMSE values and is computed in less time than $\tilde{g}_n$.

Figure 2 displays the graphs of f_*, the $\overline{Y}_i$'s, $\hat{g}_n$, and $\tilde{g}_n$ on the left side and the graphs of $f_*^{(2)}$, $\hat{g}_n^{(2)}$, and $\tilde{g}_n^{(2)}$ on the right side for the case when $n = 40$. In the graphs of Figure 2, the proposed estimator shows a smoother second derivative than that of $\tilde{g}_n$. Considering the fact that our goal is to estimate the second derivative of f_* as a smooth function, our proposed estimator appears to have the desired property.

IV. Concluding Remarks

In this paper, we proposed a new method for estimating a smooth regression function f_*. The proposed estimator $\hat{g}_n$ of f_* is designed for better computational efficiency. Numerical results show that the proposed estimator is computed faster and achieves better mean square errors than its alternative. Furthermore, the proposed estimator shows smoother derivatives than its alternative, which is a desired property when estimating a smooth regression function. The convergence of the proposed estimator to f_* as the number of data points increases to infinity was demonstrated

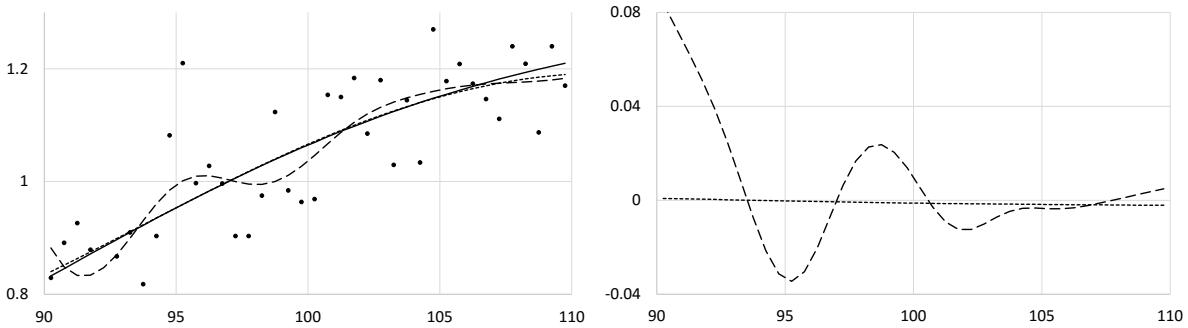

Figure 2 : The horizontal axis is x. On the left side, the solid line is $f_*(x)$, the dots are the $\overline{Y}_i$'s, the dotted line is $\hat{g}_n(x)$, and the dashed line is $\tilde{g}_n(x)$. On the right side, the dotted line is $\hat{g}_n^{(2)}(x)$, and the dashed line is $\tilde{g}_n^{(2)}(x)$.

© 2016 Global Journals Inc. (US)

empirically, so a promising future research topic involves proving the consistency of the proposed estimator theoretically.

References Références Referencias

1. Frank, M and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3, 95110.
2. Grant, M., and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.
3. Györfi, L., Kohler, M, Krzy˙zak, A., and Walk, H. (2002). *A distribution-free theory of nonparametric regression*. New York: Springer.
4. Lim, E., and Attallah, M. (2016). Estimation of smooth functions via convex programs. *International Journal of Statistics and Probability*, 5, 150-155.
5. Reinsch, C. H. (1967). Smoothing by spline functions. *Numer. Math., 10,* 177-183.
6. Wahba, G., and Wold, S. (1975). A completely automatic French curve: fitting spline functions by cross-validation. *Communications in Statistics,* 4, 1-17.
7. Zangwill, W. I. (1969). *Nonlinear programming: a unified approach.* New Jersey: Prentice-Hall Inc.

Notes

© 2016 Global Journals Inc. (US)