*Proceedings of the 2014 Winter Simulation Conference*
*A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.*

# ON A LEAST ABSOLUTE DEVIATIONS ESTIMATOR OF A MULTIVARIATE CONVEX FUNCTION

Eunji Lim

Kean University
1000 Morris Ave
Union, NJ 07083, USA

Yao Luo

Office Depot, Inc
6600 N. Military Trail
Boca Raton, FL 33496, USA

## ABSTRACT

When estimating a performance measure $f_*$ of a complex system from noisy data, the underlying function $f_*$ is often known to be convex. In this case, one often uses convexity to better estimate $f_*$ by fitting a convex function to data. The traditional way of fitting a convex function to data, which is done by computing a convex function minimizing the sum of squares, takes too long to compute. It also runs into an "out of memory" issue for large-scale datasets. In this paper, we propose a computationally efficient way of fitting a convex function by computing the best fit minimizing the sum of absolute deviations. The proposed least absolute deviations estimator can be computed more efficiently via a linear program than the traditional least squares estimator. We illustrate the efficiency of the proposed estimator through several examples.

## 1 INTRODUCTION

Various performance measures that arise in inventory systems, queueing networks, or integrated circuits have a certain shape characteristic such as convexity in the control parameters (Shanthikumar and Yao 1991, Meester and Shanthikumar 1990, Wolff and Wang 2002, del Mar Hershenson, Boyd, and Lee 2001). When estimating such performance measures from noisy data, one often uses the shape characteristic by fitting a convex function to the observed data. This paper is concerned with providing a numerically efficient way of computing the best fit of a convex function and proving the consistency of the proposed method.

Our goal is to estimate the unknown function $f_* : [0,1]^d \to \mathbb{R}$ from the observed data $(X_1,Y_1),\ldots,(X_n,Y_n)$, where

$$Y_i = f_*(X_i) + \varepsilon_i$$

for $i \geq 1$, the $X_i$s are continuous $[0,1]^d$–valued independent and identically distributed (iid) random vectors, and the $\varepsilon_i$s are iid random variables with zero median and $\mathbb{E}(|\varepsilon_1|) < \infty$.

When $f_*$ is known to be convex, a natural way to estimate $f_*$ is to minimize the sum of squares

$$\psi_n(g) \triangleq \frac{1}{n}\sum_{i=1}^{n}(Y_i - g(X_i))^2$$

or the sum of absolute deviations

$$\varphi_n(g) \triangleq \frac{1}{n}\sum_{i=1}^{n}|Y_i - g(X_i)|$$

over the set of convex functions

$$\mathscr{C} = \left\{ g : [0,1]^d \to \mathbb{R} \text{ such that } g \text{ is convex} \right\}.$$

When $\psi_n$ is used as a goodness–of–fit criterion, the fitted function is referred to as "the convex regression estimator". While statistical properties of the convex regression estimator are well–established (Hildreth 1954, Hanson and Pledger 1976, Wu 1982, Fraser and Massam 1989, Mammen 1991, Groeneboom, Jongbloed, and Wellner 2001, Turlach 2005, Birke and Dette 2007, Chang, Chien, Hsiung, C.-C.Wen, and Wu 2007, Meyer 2008, Kuosmanen 2008, Shively, Walker, and Damien 2011, Seijo and Sen 2011, Lim and Glynn 2012, Lim 2014), the convex regression estimator suffers from computational inefficiency. Minimization of $\psi_n$ over $\mathscr{C}$ can be formulated as a quadratic program (QP) with $(d+1)n$ decision variables and $n^2$ constraints (Kuosmanen 2008). The computational burden of solving this QP becomes heavy especially when $dn$ exceeds a few hundred. However, recent studies show that the idea of fitting a convex function can be applied to large–scale data. For example, the power, gain, or bandwidth of integrated circuits is often approximated as a convex or concave function in the sizes of the transistors contained in integrated circuits (del Mar Hershenson, Boyd, and Lee 2001). In this context, more than a few thousand data points can be used to estimate the performance measure of the integrated circuits as a convex function. Thus, there is a growing need of fitting a convex function to large–scale data. Recently, there has been an effort to develop computationally efficient methods to estimate convex functions. Magnani and Boyd (2009) and Hannah and Dunson (2013) use approximate piecewise linear approaches that can scale to large data sets. Aguilera, Forzani, and Morin (2011) uses a two-stage approach that consists of kernel smoothing and then finding the convex hull of the smoothed estimate. But this method requires specification of the smoothing parameter.

To overcome the computational inefficiency of the convex regression estimator, we propose to use $\varphi_n$ instead of $\psi_n$ as a goodness–of–fit criterion. Using $\varphi_n$ may be beneficial from a computational point of view because minimization of $\varphi_n$ over $\mathscr{C}$ can be formulated as an linear program (LP) rather than a QP. Another advantage of using $\varphi_n$ is that the least absolute deviations estimators can provide more robust results because they are not sensitive to outliers in the dataset (Bassett and Koenker 1978, Wagner 1959).

In this paper, we use $\varphi_n$ instead of $\psi_n$ as a goodness–of–fit criterion and investigate the least absolute deviations estimator $\hat{g}_n$, which is the minimizer of $\varphi_n$ over $\mathscr{C}$. We observe that $\hat{g}_n$ can be computed by solving an LP and the LP has a dual problem that can be solved more efficiently. We further discover that the dual problem has a block–angular form in its constraints, and hence, allows application of decomposition techniques such as Dantzig–Wolfe decomposition. Dantzig–Wolfe decomposition then enables one to compute $\hat{g}_n$ for large–scale data. Our numerical reaults in Section 4 show that $\hat{g}_n$ can be computed for a dataset that contains more than $10,000$ datapoints when $d=1$ while the least squares estimator can only be computed for a dataset containing a few hundred data points. In most of our numerical examples, $\hat{g}_n$ was computed much faster than the least squares estimators.

This paper is organized as follows. Section 2 introduces the mathematical framework for our analysis. In Section 3, we provide a numerically efficient LP formulation for computing $\hat{g}_n$ while Section 4 discusses the numerical behavior of the least absolute deviations estimator compared to that of the least squares estimator.

## 2 MATHEMATICAL FRAMEWORK

When $f_*$ is known to be convex, a natural way of estimating it from data is to minimize the sum of absolute deviations

$$\varphi_n(g) = \frac{1}{n}\sum_{i=1}^{n}|Y_i - g(X_i)|$$

over the set of convex functions $\mathscr{C} = \left\{g : [0,1]^d \to \mathbb{R} \text{ such that } g \text{ is convex}\right\}$. Since there are infinitely many convex functions, this minimization may appear to be computationally intractable. However, the following proposition reveals that this minimization can be formulated as an LP with $(d+3)n$ decision variables and $n^2+3n$ constraints.

**Proposition 1** Consider the minimization problem in the decision variables $(g_1,\xi_1),\ldots,(g_n,\xi_n)$

$$\begin{array}{ll} \min & \frac{1}{n}\sum_{i=1}^{n}|Y_i-g_i| \\ \text{s/t} & g_j \geq g_i+\xi_i^T(X_j-X_i), \quad 1\leq i,j\leq n, \end{array} \tag{1}$$

where $g_i\in\mathbb{R}$ and $\xi_i\in\mathbb{R}^d$ for $1\leq i\leq n$. Then, the problem (1) has a minimizer $(\hat{g}_1,\hat{\xi}_1),\ldots,(\hat{g}_n,\hat{\xi}_n)$ and $\hat{g}_n:[0,1]^d\to\mathbb{R}$, defined by

$$\hat{g}_n(x) \quad = \quad \max_{1\leq i\leq n}(\hat{g}_i+\hat{\xi}_i^T(x-X_i)) \tag{2}$$

for $x\in[0,1]^d$, minimizes $\varphi_n$ over $\mathscr{C}$.

Furthermore, problem (1) has a minimizer $(\hat{g}_1,\hat{\xi}_1),\ldots,(\hat{g}_n,\hat{\xi}_n)$ if and only if $(\hat{g}_1,(Y_1-\hat{g}_1)^+,(-Y_1+\hat{g}_1)^+,\hat{\xi}_1),\ldots,(\hat{g}_n,(Y_n-\hat{g}_n)^+,(-Y_n+\hat{g}_n)^+,\hat{\xi}_n)$ is a solution to the following LP in the decision variables $(g_1,p_1,m_1,\xi_1),\ldots,(g_n,p_n,m_n,\xi_n)$:

$$\begin{array}{lll} \min & \frac{1}{n}\sum_{i=1}^{n}(p_i+m_i) & \\ \text{s/t} & g_j\geq g_i+\xi_i^T(X_j-X_i), & 1\leq i,j\leq n \\ & Y_i-g_i=p_i-m_i, & 1\leq i\leq n \\ & p_i,m_i\geq 0, & 1\leq i\leq n, \end{array}$$

where $g_i\in\mathbb{R},p_i\in\mathbb{R},m_i\in\mathbb{R}$, and $\xi_i\in\mathbb{R}^d$ for $1\leq i\leq n$.

*Proof.* Let $\mathscr{G}_n=\{(g_1,\ldots,g_n)\in\mathbb{R}^n$ such that there exists a convex function $g:[0,1]^d\to\mathbb{R}$ satisfying $g(X_i)=g_i$ for $1\leq i\leq n\}$. Then, $\mathscr{G}_n$ is nonempty ($(0,\ldots,0)\in\mathscr{G}_n$), closed and convex by Lemma 2.3 of Seijo and Sen (2011). Note that $\varphi_n$ is continuous and coercive (i.e., $|\varphi_n(g_1,\ldots,g_n)|\to\infty$ as $\|(g_1,\ldots,g_n)\|\to\infty$). Thus, $\varphi_n$ has a minimizer $(\hat{g}_1,\ldots,\hat{g}_n)$ over $\mathscr{G}_n$; see Proposition 7.3.1 and Theorem 7.3.7 in pp. 216 and 217 of Kurdila and Zabarankin (2005). Since $(\hat{g}_1,\ldots,\hat{g}_n)\in\mathscr{G}_n$, there exist vectors $\hat{\xi}_1,\ldots,\hat{\xi}_n$ in $\mathbb{R}^d$ satisfying $\hat{g}_j\geq\hat{g}_i+\hat{\xi}_i^T(X_j-X_i)$ for $1\leq i,j\leq n$, and hence, $(\hat{g}_1,\hat{\xi}_1),\ldots,(\hat{g}_n,\hat{\xi}_n)$ is a feasible solution of (1). Furthermore, $(\hat{g}_1,\hat{\xi}_1),\ldots,(\hat{g}_n,\hat{\xi}_n)$ becomes a minimizer of (1) by p. 337 of Boyd and Vandenberghe (2004). The rest of the proposition follows trivially. $\square$

**Remark 1** While Proposition 1 asserts that $(\hat{g}_1,\ldots,\hat{g}_n)$ exists, it should be noted that $(\hat{g}_1,\ldots,\hat{g}_n)$ may not be unique. A simple example that illustrates the non-uniqueness of $(\hat{g}_1,\ldots,\hat{g}_n)$ is the following: When $d=1,n=4,(X_1,Y_1)=(0.2,0),(X_2,\ Y_2)=(0.4,1),(X_3,Y_3)=(0.6,1)$, and $(X_4,Y_4)=(0.8,0)$, any point from the set

$$\begin{aligned}\mathscr{T}=\big\{&(\hat{g}_1,\hat{\xi}_1)=(a,b),(\hat{g}_2,\hat{\xi}_2)=(a,0),(\hat{g}_3,\hat{\xi}_3)=(a,0),(\hat{g}_4,\hat{\xi}_4)=(a,c): \\ &a\in[0,1],b\in(-\infty,0],c\in[0,\infty)\big\}\end{aligned}$$

is a minimizer of (1). So, $(\hat{g}_1,\hat{g}_2,\hat{g}_3,\hat{g}_4)$ is not unique.

Throughout this paper, we will work with the set of minimizers of $\varphi_n$ over $\mathscr{C}$:

$$\mathscr{S}_n=\{g_n\in\mathscr{C}:\varphi_n(g_n)\leq\varphi_n(g)\text{ for all }g\in\mathscr{C}\}$$

for $n\geq 1$. By Proposition 1, $\mathscr{S}_n$ is nonempty for all $n\geq 1$ a.s. and Proposition 1 suggests a way of computing an element $\hat{g}_n$ in $\mathscr{S}_n$ by using (1) and (2). The convex function $\hat{g}_n$ is our estimator for $f_*(\cdot)$.

The next section provides an efficient way of computing $\hat{g}_n$.

## 3 A MORE EFFICIENT LP FORMULATION

In this section, we present an efficient LP formulation for computing $\hat{g}_n$. By Proposition 1, $\hat{g}_n$ can be computed by solving the following LP in the decision variables $(g_1, p_1, m_1, \xi_1), \ldots, (g_n, p_n, m_n, \xi_n)$

$$\begin{array}{ll} \min & \displaystyle\frac{1}{n}\sum_{i=1}^{n}(p_i + m_i) \\ \text{s/t} & g_j \geq g_i + \xi_i^T (X_j - X_i), \quad 1 \leq i, j \leq n \\ & Y_i - g_i = p_i - m_i, \quad 1 \leq i \leq n \\ & p_i, m_i \geq 0, \quad 1 \leq i \leq n. \end{array} \tag{3}$$

We notice that the dual problem of (3) is the following LP:

$$\begin{array}{llllllll} \max & Y^T t \\ \text{s/t} & A_1 s_1 + & A_2 s_2 + & \cdots + & A_n s_n + & I_n t & = & 0_n \\ & b_1^T s_1 & & & & & = & 0 \\ & & b_2^T s_2 & & & & = & 0 \\ & & & \ddots & & & & \vdots \\ & & & & b_n^T s_n & & = & 0 \\ & & & & & I_n t & \leq & 1_n \\ & & & & & I_n t & \geq & -1_n \\ & & & & & s_i & \geq & 0_n, \quad 1 \leq i \leq n \end{array} \tag{4}$$

with the decision variables $(s_{ij} : 1 \leq i, j \leq n)$ and $(t_i : 1 \leq i \leq n)$, where $Y = (Y_1, \ldots, Y_n)$, $t = (t_1, \ldots, t_n)^T \in \mathbb{R}^n, s_i = (s_{i1}, \ldots, s_{in})^T \in \mathbb{R}^n$ for $1 \leq i \leq n$, $A_i = (a_{jk} : 1 \leq j, k \leq n)$ with

$$a_{jk} = \begin{cases} 1, & j = k, j \neq i \\ -1, & j = i, k \neq i \\ 0, & \text{otherwise,} \end{cases}$$

$b_i = (X_i - X_1, \ldots, X_i - X_n)^T$ for $1 \leq i \leq n$, $I_n$ is an $n$ by $n$ identity matrix, $1_n$ is an $n$ by 1 vector of all ones, and $0_n$ is an $n$ by 1 vector of all zeros. The $s_{ij}$s and $t_i$s are dual variables corresponding to the first and second sets of constraints of (3), respectively.

The dual problem (4) has two sets of decision variables $(s_{ij} : 1 \leq i, j \leq n)$ and $(t_i : 1 \leq i \leq n)$. The two sets of variables are related only through the first constraint in (4). Thus, (4) has a block structure in its constraints and hence allows application of decomposition techniques such as Dantzig–Wolfe decomposition.

The reason that formulation (4) can be solved more efficiently than formulation (3) is two–fold. When solving an LP problem using the simplex method, it is more efficient to solve the dual problem than the primal problem if the primal problem has much more constraints than the decision variables; see p. 147 of Bradley, Hax, and Magnanti (1977) and p. 234 of Grover (2004) for details. It is the case with the primal problem (3) and the dual problem (4) because (3) has $O(n)$ decision variables and $O(n^2)$ constraints while (4) has $O(n^2)$ decision variables and $O(n)$ constraints. Second, one can apply Dantzig–Wolfe decomposition to solve (4) and thus can compute $\hat{g}_n$ for larger datasets than (3) can handle.

In the next section, we compare formulations (3) and (4) through numerical examples.

## 4 NUMERICAL RESULTS

In this section, we investigate how fast $\hat{g}_n$ can be computed by solving (4) through common LP solving techniques such as the simplex method and the interior point method. We further illustrate how $\hat{g}_n$ can be computed for large datasets by solving (4) with Dantzig–Wolfe decomposition.

We are particularly interested in the relative performance of $\hat{g}_n$ compared to that of the least squares estimator. The least squares estimator is defined as the minimizing values $\tilde{g}_n(X_1),\ldots,\tilde{g}_n(X_n)$ of the following QP in the decision variables $(g_1,\xi_1),\ldots,(g_n,\ \xi_n)$

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n}(Y_i-g_i)^2 \\ \text{s/t} \quad & g_j \geq g_i+\xi_i^T(X_j-X_i), \quad 1\leq i,j\leq n; \end{aligned} \tag{5}$$

see Kuosmanen (2008) for details.

In the next three subsections, we compare formulations (4) and (5) in three numerical examples: 1) a stylized model, 2) an inventory control system, and 3) a tandem queueing network. In each of these examples, we consider four different computational strategies. Estimator 1 is the least squares estimator and Estimators 2, 3, and 4 are our proposed estimator that are computed using different algorithms. Estimator 1, $\tilde{g}_n(X_1),\ldots,\tilde{g}_n(X_n)$, is computed by solving (5) through the interior point method in CPLEX. Estimator 2 is our proposed estimator $\hat{g}_n$ and it is computed by solving (4) through the simplex method in CPLEX. Estimator 3 is also our proposed estimator $\hat{g}_n$ and it is computed by solving (4) through the interior point method in CPLEX. Estimator 4 is our proposed estimator $\hat{g}_n$. To compute estimator 4, we implement the Dantzig–Wolfe decomposition algorithm presented in p. 243 of Bertsimas and Tsitsiklis (1997). We also implemented a stabilization technique introduced in du Merle, Villeneuve, Desrosiers, and Hansen (1999) while implementing the Dantzig–Wolfe decomposition algorithm.

In each of the three examples, the proposed estimator is computed faster and for larger datasets than the least squares estimator. All the numerical experiments are conducted on a computer with a processor of 2.33 GHz and a RAM of 12 GB.

### 4.1 One–Dimensional Case: A Stylized Model

We consider the case where $f_*:[0,1]\to\mathbb{R}$ is defined by $f_*(x)=(x-0.5)^2$ for $x\in[0,1], X_i=i/n$ for $1\leq i\leq n$. Using $(X_i,Y_i)$ for $1\leq i\leq n$, we compute estimators 1, 2, 3, and 4.

Table 1 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory and could not execute the procedure.

Table 2 reports the averages of the the empirical integrated mean squared error (EIMSE) based on 30 independent copies for each of the estimators 1, 2, 3, and 4 and for each $n$. The EIMSE is computed as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\text{Estimator evaluated at } X_i\right)^2.$$

Tables 1 and 2 show that our proposed estimator, estimator 4, is computed faster and more accurately. For example, it takes estimator 1 (the least squares estimator) an average of 53.73 seconds to produce an EIMSE of 0.0038 ($n=400$) whereas it takes estimator 4 (our proposed estimator) an average of 32.21 seconds to produce an EIMSE of 0.0017 ($n=1400$).

### 4.2 Two–Dimensional Case: $(Q,r)$ Inventory System

We consider a single-item continuous-review $(Q,r)$ inventory system, where we place an order with a fixed quantity $Q$ whenever the inventory position (= on hand stock minus backorders plus any outstanding orders) drops below a prespecified quantity $r$. The replenishment lead time is assumed to be one unit of time. When an order is placed, a fixed setup cost of \$100 is incurred. A holding cost of \$10 or a penalty cost of \$25 per unit per unit time is charged against any inventory or backorder. We further assume that demand follows a Poisson process with a rate of 50 per unit time. Any unfilled demand is backordered. Our goal is to estimate the steady–state mean total costs per unit time $C(Q,r)$, which is proven to be

Table 1: Time required to compute estimators 1, 2, 3, and 4 for a quadratic function

| | CPU Time (sec) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Estimator 1 | | Estimator 2 | | Estimator 3 | | Estimator 4 | |
| $n$ | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| 5 | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.01 | 0.00 | 0.01 |
| 50 | 2.00 | 0.09 | 0.05 | 0.01 | 0.09 | 0.03 | 0.07 | 0.01 |
| 400 | 53.73 | 5.47 | 5.85 | 0.16 | 15.16 | 0.96 | 2.27 | 0.24 |
| 600 | - | - | 18.65 | 0.58 | 55.24 | 2.14 | 5.28 | 0.47 |
| 1000 | - | - | 41.67 | 1.29 | 264.97 | 11.12 | 15.32 | 1.45 |
| 1400 | - | - | 114.79 | 5.71 | 681.80 | 21.48 | 32.21 | 3.35 |
| 1600 | - | - | - | - | - | - | 44.76 | 4.75 |
| 2000 | - | - | - | - | - | - | 71.36 | 7.20 |
| 5000 | - | - | - | - | - | - | 679.40 | 74.05 |
| 10000 | - | - | - | - | - | - | 4177.68 | 532.36 |

Table 2: EIMSE of estimators 1, 2, 3, and 4 for a quadratic function

| | EIMSE | | | |
|---|---|---|---|---|
| $n$ | Estimator 1 | Estimator 2 | Estimator 3 | Estimator 4 |
| 5 | 0.1473 | 0.1980 | 0.1664 | 0.1908 |
| 50 | 0.0206 | 0.0245 | 0.0249 | 0.0240 |
| 400 | 0.0038 | 0.0049 | 0.0050 | 0.0050 |
| 600 | - | 0.0043 | 0.0043 | 0.0043 |
| 1000 | - | 0.0026 | 0.0028 | 0.0027 |
| 1400 | - | 0.0017 | 0.0017 | 0.0017 |
| 1600 | - | - | - | 0.0016 |
| 2000 | - | - | - | 0.0018 |
| 5000 | - | - | - | 0.0007 |
| 10000 | - | - | - | 0.0003 |

Table 3: Time required to compute estimators 1, 2, 3, and 4 for a $(Q,r)$ inventory system

| | CPU Time (sec) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Estimator 1 | | Estimator 2 | | Estimator 3 | | Estimator 4 | |
| $n$ | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| 64 | 1.98 | 0.15 | 0.19 | 0.02 | 0.10 | 0.02 | 0.48 | 0.08 |
| 100 | 3.29 | 0.19 | 0.32 | 0.02 | 0.25 | 0.03 | 1.80 | 0.31 |
| 625 | - | - | 56.61 | 1.72 | 62.28 | 1.38 | 34.39 | 4.71 |
| 1600 | - | - | - | - | - | - | 265.30 | 42.47 |
| 2500 | - | - | - | - | - | - | 733.65 | 91.31 |
| 6400 | - | - | - | - | - | - | 12244.76 | 1701.47 |

Table 4: EIMSE of estimators 1, 2, 3, and 4 for a $(Q,r)$ inventory system

| | EIMSE | | | |
|---|---|---|---|---|
| $n$ | Estimator 1 | Estimator 2 | Estimator 3 | Estimator 4 |
| 64 | 0.3408 | 0.4201 | 0.4185 | 0.4106 |
| 100 | 0.2022 | 0.2715 | 0.2654 | 0.2633 |
| 625 | - | 0.2235 | 0.2241 | 0.2247 |
| 1600 | - | - | - | 0.1907 |
| 2500 | - | - | - | 0.1952 |
| 6400 | - | - | - | 0.1105 |

jointly convex in $Q$ and $r$ (p. 89 of (Zheng 1992)). To compute $C(Q,r)$, we select the values for $(Q,r)$ at $X_{ij} = (35+10i/(n^{1/2}), 35+10j/(n^{1/2}))$ for $1 \le i,j \le n^{1/2}$, simulate the inventory system up to time 100 at each $X_{ij}$, compute the average $Y_{ij}$ of all the costs up to time 100 at each $X_{ij}$, and obtain the average of 20 independent copies of $Y_{ij}$, say $\overline{Y}_{ij}$. Using $(X_{ij}, \overline{Y}_{ij})$ for $1 \le i,j \le n^{1/2}$, we compute estimators 1, 2, 3, and 4.

Table 3 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory could not execute the procedure.

Table 4 reports the averages of the the empirical integrated mean squared error (EIMSE) based on 30 independent copies for each of the estimators 1, 2, 3, and 4 and for each $n$. The EIMSE is computed as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left(C(X_{ij}) - \text{Estimator evaluated at } X_{ij}\right)^2.$$

We estimated $C(X_{ij})$ by computing the average of 1,000 independent copies of $Y_{ij}$.

Tables 3 and 4 show that our proposed estimator, estimator 4, is computed faster and more accurately. For example, it takes estimator 1 (the least squares estimator) an average of 1.98 seconds to produce an EIMSE of 0.3408 ($n = 64$) whereas it takes estimator 4 (our proposed estimator) an average of 1.80 seconds to produce an EIMSE of 0.2633 ($n = 100$).

### 4.3 Three–Dimensional Case: Tandem Queue

We consider a queueing system of three single-server stations connected in tandem, where the interarrival times follow a uniform distribution over $[2.43, 3.43]$ and the service times at server $i$ follow a uniform distribution over $[x_i - 0.5, x_i + 0.5]$ for $1 \le i \le 3$. The interarrival times and service times are independent of each other and the first in/first out queueing discipline is used at each server. Each server has unlimited

Table 5: Time required to compute estimators 1, 2, 3, and 4 for a tandem queue

| | CPU Time (sec) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Estimator 1 | | Estimator 2 | | Estimator 3 | | Estimator 4 | |
| $n$ | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| 64 | 0.17 | 0.05 | 0.06 | 0.01 | 0.07 | 0.01 | 0.39 | 0.01 |
| 216 | 50.06 | 47.92 | 2.09 | 0.09 | 1.09 | 0.08 | 7.51 | 0.87 |
| 512 | - | - | 51.22 | 2.43 | 15.81 | 1.51 | 27.51 | 4.50 |
| 1000 | - | - | 340.89 | 19.86 | 187.18 | 16.33 | 121.81 | 24.29 |
| 1728 | - | - | - | - | - | - | 862.55 | 127.83 |
| 4096 | - | - | - | - | - | - | 11410.52 | 2895.74 |

Table 6: EIMSE of estimators 1, 2, 3, and 4 for a tandem queue

| | EIMSE | | | |
|---|---|---|---|---|
| $n$ | Estimator 1 | Estimator 2 | Estimator 3 | Estimator 4 |
| 64 | 0.1030 | 0.1340 | 0.1270 | 0.1296 |
| 216 | 0.0640 | 0.0815 | 0.0772 | 0.0782 |
| 512 | - | 0.0446 | 0.0448 | 0.0454 |
| 1000 | - | 0.0363 | 0.0366 | 0.0377 |
| 1728 | - | - | - | 0.0286 |
| 4096 | - | - | - | 0.0224 |

buffer space. We wish to compute the expected sojourn time $s_{600}(x_1,x_2,x_3)$ of the 600th customer. Even though there is no explicit formula for $s_{600}$, the convexity of $s_{600}$ has been proven; see p. 141 of (Shanthikumar and Yao 1991) for details. To compute $s_{600}$, we simulate the tandem queue at $X_{ijk} = (2.85+0.06i/(n^{1/3}), 2.85+0.06j/(n^{1/3}), 2.85+0.06k/(n^{1/3}))$ for $1 \le i,j,k \le n^{1/3}$ and compute the sojourn time $Y_{ijk}$ of the 600th customer. We then obtain the average of 30 independent copies of $Y_{ijk}$, say $\overline{Y}_{ijk}$. Using $(X_{ijk}, \overline{Y}_{ijk})$ for $1 \le i,j,k \le n^{1/3}$, we compute estimators 1, 2, 3, and 4.

Table 5 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory could not execute the procedure.

Table 6 reports the averages of the the empirical integrated mean squared error (EIMSE) based on 30 independent copies for each of the estimators 1, 2, 3, and 4 and for each $n$. The EIMSE is computed as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left(s_{600}(X_{ijk}) - \text{Estimator evaluated at } X_{ijk}\right)^2.$$

We estimated $s(X_{ijk})$ by computing the average of 1,000 independent copies of $Y_{ijk}$.

Tables 5 and 6 show that our proposed estimator, estimator 4, is computed faster and more accurately. For example, it takes estimator 1 (the least squares estimator) an average of 50.06 seconds to produce an EIMSE of 0.0640 ($n = 216$) whereas it takes estimator 4 (our proposed estimator) an average of 27.51 seconds to produce an EIMSE of 0.0454 ($n = 512$).

In each of the three examples, the proposed estimator is computed faster and for larger datasets than the least squares estimator.

## 5 CONCLUSION

In this paper, we proposed a least absolute deviations estimator for fitting a convex function to a dataset. Numerical examples illustrate the computational superiority of the proposed estimator over the traditional least squares estimator. Statistical properties of the proposed estimator such as the consistency or the rate of convergence are important research issues that deserve further study.

## REFERENCES

Aguilera, N., L. Forzani, and P. Morin. 2011. "On Uniform Consistent Estimators for Convex Regression". *Journal of Nonparametric Statistics* 23:897–908.

Bassett, G., and R. Koenker. 1978. "Asymptotic Theory of Least Absolute Error Regression". *Journal of the American Statistical Association* 73 (363): 618–622.

Bertsimas, D., and J. N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific.

Birke, M., and H. Dette. 2007. "Estimating a Convex Function in Nonparametric Regression". *Scandinavian Journal of Statistics* 34 (2): 384–404.

Boyd, S., and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge, UK: Cambridge University Press.

Bradley, S. P., A. C. Hax, and T. L. Magnanti. 1977. *Applied Mathematical Programming*. Addison-Wesley Pub. Co.

Chang, I.-S., L.-C. Chien, C. A. Hsiung, C.-C.Wen, and Y.-J. Wu. 2007. "Shape Restricted Regression with Random Bernstein Polynomials". *Lecture Notes-Monograph Series* 54:187–202.

del Mar Hershenson, M., S. P. Boyd, and T. H. Lee. 2001. "Optimal Design of a CMOS Opamp via Geometric Programming". *IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems* 20 (1): 1–21.

du Merle, O., D. Villeneuve, J. Desrosiers, and P. Hansen. 1999. "Stabilized column generation". *Discrete Mathematics* 194 (1–3): 229–237.

Fraser, D. A. S., and H. Massam. 1989. "A Mixed Primal-dual Bases Algorithm for Regression under Inequality Constraints: Application to Concave Regression". *Scandinavian Journal of Statistics* 16:65–74.

Groeneboom, P., G. Jongbloed, and J. A. Wellner. 2001. "Estimation of a Convex Function: Characterizations and Asymptotic Theory". *Annals of Statistics* 29:1653–1698.

Grover, W. D. 2004. *Mesh-based Survivable Networks: Options and Strategies for Optical, Mpls, Sonet, and ATM Networking*. Prentice Hall.

Hannah, L. A., and D. B. Dunson. 2013. "Multivariate Convex Regression with Adaptive Partitioning". *Journal of Machine Mearning Research* 14:3261–3294.

Hanson, D. L., and G. Pledger. 1976. "Consistency in Concave Regression". *Annals of Statistics* 4 (6): 1038–1050.

Hildreth, C. 1954. "Point Estimates of Ordinates of Concave Functions". *Journal of the American Statistical Association* 49 (267): 598–619.

Kuosmanen, T. 2008. "Representation Theorem for Convex Nonparametric Least Squares". *Econometrics Journal* 11 (2): 308–325.

Kurdila, A. J., and M. Zabarankin. 2005. *Convex Functional Analysis (Systems & Control: Foundations & Applications)*. Switzerland: Birkhäuser.

Lim, E. 2014. "On Convergence Rates of Convex Regression in Multiple Dimensions". *INFORMS Journal on Computing*. Articles in Advance. http://dx.doi.org/10.1287/ijoc.2013.0587.

Lim, E., and P. W. Glynn. 2012. "Consistency of Multidimensional Convex Regression". *Operations Research* 60 (1): 196–208.

Magnani, A., and S. P. Boyd. 2009. "Convex Piecewise-linear Fitting". *Optimization and Engineering* 10:117.

Mammen, E. 1991. "Nonparametric Regression under Qualitative Smoothness Assumptions". *Annals of Statistics* 19:741–759.

Meester, L. E., and J. G. Shanthikumar. 1990. "Concavity of the Throughput of Tandem Queueing Systems with Finite Buffer Storage Space". *Advances in Applied Probability* 22:764–767.

Meyer, M. C. 2008. "Inference Using Shape-Restricted Regression Splines". *Annals of Applied Statistics* 2 (3): 1013–1033.

Seijo, E., and B. Sen. 2011. "Nonparametric Least Squares Estimation of a Multivariate Convex Regression Function". *Annals of Statistics* 39:1633–1657.

Shanthikumar, J. G., and D. D. Yao. 1991. "Strong Stochastic Convexity: Closure Properties and Applications". *Journal of Applied Probability* 28:131–145.

Shively, T. S., S. G. Walker, and P. Damien. 2011. "Nonparametric Function Estimation Subject to Monotonicity, Convexity and Other Shape Constraints". *Journal of Econometrics* 161 (2): 166–181.

Turlach, B. 2005. "Shape Constrained Smoothing Using Smoothing Splines". *Computational Statistics* 20:81–104.

Wagner, H. M. 1959. "Linear Programming Techniques for Regression Analysis". *Journal of the American Statistical Association* 54 (285): 206–212.

Wolff, R. W., and C.-L. Wang. 2002. "On the Convexity of Loss Probabilities". *Journal of Applied Probability* 39:402–406.

Wu, C. F. 1982. "Some Algorithms for Concave and Isotonic Regression". *TIMS Studies in Management Science* 19:105–116.

Zheng, Y.-S. 1992. "On Properties of Stochastic Inventory Systems". *Management Science* 38 (87–103): 1.

## AUTHOR BIOGRAPHIES

**EUNJI LIM** is an Assistant Professor at the Department of Management, Marketing, and International Business at Kean University. Her research interest includes simulation-based optimization and stochastic models. Her email address is elim@kean.edu.

**YAO LUO** is a Project Manager of Supply Chain Analytics and Optimization at Office Depot, Inc. He received his Ph.D. in Industrial Engineering from University of Miami (2013) and B.En. in Industrial Engineering from Shanghai Jiao Tong University (2009). His research interests are in simulation optimization and supply chain management. His email address is ly1987510@gmail.com.