

# Multivariate Convex Approximation and Least-Norm Convex Data-Smoothing

Alex Y.D. Siem<sup>1</sup>, Dick den Hertog<sup>1</sup>, and Aswin L. Hoffmann<sup>2</sup>

<sup>1</sup> Department of Econometrics and Operations Research/Center for Economic Research (CentER), Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands

{a.y.d.siem, d.denhertog}@uvt.nl

<sup>2</sup> Department of Radiation Oncology, Radboud University Nijmegen Medical Centre, Geert Grooteplein 32, 6525 GA Nijmegen, The Netherlands

a.hoffmann@rther.umcn.nl

**Abstract.** The main contents of this paper is two-fold. First, we present a method to approximate multivariate convex functions by piecewise linear upper and lower bounds. We consider a method that is based on function evaluations only. However, to use this method, the data have to be convex. Unfortunately, even if the underlying function is convex, this is not always the case due to (numerical) errors. Therefore, secondly, we present a multivariate data-smoothing method that smooths nonconvex data. We consider both the case that we have only function evaluations and the case that we also have derivative information. Furthermore, we show that our methods are polynomial time methods. We illustrate this methodology by applying it to some examples.

## 1 Introduction

In the field of discrete approximation, we are interested in approximating a function  $y : \mathbb{R}^q \rightarrow \mathbb{R}$ , given a discrete dataset  $\{(x^i, y^i) : 1 \leq i \leq n\}$ , where  $x^i \in \mathbb{R}^q$  and  $y^i = y(x^i) \in \mathbb{R}$ , and  $n$  is the number of data points. It may happen that we know beforehand that the function  $y(x)$  is convex. However, many approximation methods do not make use of the information that  $y(x)$  is convex and construct approximations that do not preserve the convexity. For the univariate case there is some literature on convexity preserving functions; see e.g. [1] and [2]. In [1], Splines are used, and in [2], polynomial approximation is considered. For the multivariate case, in [3], convex quadratic polynomials are used to approximate convex functions. Furthermore, there is a lot of literature on so-called Sandwich algorithms; see e.g. [4], [5], [6], [7], and [8]. In these papers, upper and lower bounds for the function  $y(x)$  are constructed, based on the discrete dataset, and based on the knowledge that  $y(x)$  is convex.

A problem that may occur in practice is that one may have a dataset that is subject to noise, i.e., instead of the data  $y^i$  we have  $\tilde{y}^i = y(x^i) + \varepsilon_y^i$ , where  $\varepsilon_y^i$  is (numerical) noise. There may also be noise in the input data, i.e.,  $\tilde{x}^i = x^i + \varepsilon_x^i$ , and if derivative information is available, it could also be subject to noise, i.e.,  $\tilde{\nabla}y^i = \nabla y^i + \varepsilon_g^i$ , where  $\nabla y^i = \nabla y(x^i)$ . Note that we assume  $y(x)$  to be convex. However, due to the noise, the perturbed data might loose the convexity of

$y(x)$ , i.e., the noise could be such that it is not possible to fit a convex function through the perturbed data. Therefore, we are interested in data-smoothing, i.e., in shifting the data points, such that they obtain convexity, and such that the amount of movement of the data is minimized. This problem has already been tackled in literature for the univariate case; see e.g. [9], [10], and [11]. Also in isotonic regression, this problem is dealt with for the univariate case; see [12].

In this paper, we will consider two problems. First, we consider how to construct piecewise linear upper and lower bounds to approximate the output for the multivariate case. This extends the method in [7] to the multivariate case. If derivative information is available it is easy to construct upper and lower bounds. However, derivative information is not always available, e.g., in the case of black-box functions. In this paper, it turns out that these upper and lower bounds can be found by solving linear programs (LPs).

Second, we will consider the multivariate data-smoothing problem. We consider both the case that we have only function evaluations and the case that we also have derivative information. We will show that, if we only consider errors in the output data, the first problem can be solved by using techniques, which are from linear robust optimization; see [13]. It turns out that this problem can be tackled by solving an LP. If we also have derivative information, we can also consider errors in the gradients and in the input variables. We then obtain a nonlinear optimization problem. However, if we assume that there are only errors in the gradients and in the output data, we obtain an LP. Also, if we assume that there are only errors in the input data and in the output data, we also obtain an LP.

The remainder of this paper is organized as follows. In Sect. 2, we consider the problem of constructing upper and lower bounds. In Sect. 3, we consider multivariate data-smoothing, and in Sect. 4, we give some examples of the application of the data-smoothing techniques, considered in Sect. 3. Finally, in Sect. 5, we present possible directions for further research.

## 2 Bounds Preserving Convexity

In this section we assume that  $y(x)$  is convex and that the data  $(x^i, y(x^i))$  for  $i = 1, \dots, n$  are convex as well, i.e., there are no (numerical) errors, and there exists a convex function that fits through the data points.

### 2.1 Upper Bounds

We are interested in finding the smallest upper bound for  $y(x)$ , given convexity, and the data  $(x^i, y(x^i))$ , for  $i = 1, \dots, n$ . Let  $x = \sum_{i=1}^n \alpha_i x^i$ , where  $\sum_{i=1}^n \alpha_i = 1$ , and  $0 \leq \alpha_i \leq 1$ , i.e.,  $x$  is a convex combination of the input data  $x^i$ . Then, it is well-known that convexity gives us the following inequality:

$$y(x) = y\left(\sum_{i=1}^n \alpha_i x^i\right) \leq \sum_{i=1}^n \alpha_i y(x^i) .$$

This means that  $\sum_{i=1}^n \alpha_i y(x^i)$  is an upper bound for  $y(x)$ . To find the smallest upper bound we should therefore solve

$$\begin{aligned}
 u(x) &:= \min_{\alpha_1, \dots, \alpha_n} \sum_{i=1}^n \alpha_i y(x^i) \\
 \text{s.t. } x &= \sum_{i=1}^n \alpha_i x^i \\
 0 &\leq \alpha_i \leq 1 \\
 \sum_{i=1}^n \alpha_i &= 1,
 \end{aligned} \tag{1}$$

where we put the decision variables underneath 'min'.

### 2.2 Lower Bounds

If we have derivative information, it is easy to construct a lower bound. It is well-known that if  $y(x)$  is convex, we have that

$$y(x) \geq y(x^i) + \nabla y(x^i)^T (x - x^i), \quad \forall x \in \mathbb{R}^q, \forall i = 1, \dots, n .$$

Therefore,  $\ell(x) = \max_{i=1, \dots, n} (y(x^i) + \nabla y(x^i)^T (x - x^i))$  is a lower bound.

If we do not have derivative information, we have to do something else. We are interested in finding the largest lower bound for  $y(x)$ , given convexity and the data  $(x^i, y(x^i))$ , for  $i = 1, \dots, n$ . Let  $x^k = \sum_{i \neq k} \alpha_i^k x^i + \alpha^k x$ , where  $\sum_{i \neq k} \alpha_i^k + \alpha^k = 1$ , with  $0 \leq \alpha_i^k \leq 1$ , and  $0 < \alpha^k \leq 1$ , for all  $k = 1, \dots, n$ , i.e.,  $x^k$  is a convex combination of  $x^i$ ,  $i \neq k$ , and  $x$ . Then the following holds for all  $k \in \{1, \dots, n\}$ :

$$y(x^k) = y \left( \sum_{i \neq k} \alpha_i^k x^i + \alpha^k x \right) \leq \sum_{i \neq k} \alpha_i^k y(x^i) + \alpha^k y(x) . \tag{2}$$

Without loss of generality we may assume that  $\alpha^k > 0$ . Then we can rewrite (2) as

$$y(x) \geq \frac{y(x^k) - \sum_{i \neq k} \alpha_i^k y(x^i)}{\alpha^k}, \text{ for } k = 1, \dots, n .$$

This inequality gives us a lower bound for  $y(x)$ . To obtain the largest lower bound we should solve the following problem:

$$\max_{k=1, \dots, n} \left\{ \begin{array}{l} \max_{\alpha^k, \alpha_i^k} \frac{y(x^k) - \sum_{i \neq k} \alpha_i^k y(x^i)}{\alpha^k} \\ \text{s.t. } x^k = \sum_{i \neq k} \alpha_i^k x^i + \alpha^k x \\ \sum_{i \neq k} \alpha_i^k + \alpha^k = 1 \\ 0 \leq \alpha_i^k \leq 1 \\ 0 < \alpha^k \leq 1 \end{array} \right\} . \tag{3}$$

This comes down to solving  $n$  nonlinear optimization problems, and taking the value of the largest solution. Note that the nonlinear optimization problems have

linear constraints and a fractional objective with linear numerator and denominator. These kinds of optimization problems can be rewritten into an LP; see [14].

This can be done as follows. Define  $t^k := 1/\alpha^k$ . We can now rewrite the inner optimization problem in (3) as

$$\begin{aligned} & \max_{\alpha^k, \alpha_i^k, t^k} t^k y(x^k) - \sum_{i \neq k} \alpha_i^k t^k y(x^i) \\ & \text{s.t. } x^k t^k = \sum_{i \neq k} \alpha_i^k t^k x^i + \alpha^k t^k x \\ & \quad \sum_{i \neq k} \alpha_i^k t^k + \alpha^k t^k = t^k \\ & \quad \alpha_i^k t^k \geq 0 \\ & \quad \alpha^k t^k = 1, \end{aligned}$$

where we multiplied all constraints by  $t^k$ . Now we define  $z_i^k := \alpha_i^k t^k$  and  $z^k := \alpha^k t^k$ . We then get

$$\begin{aligned} & \max_{z^k, z_i^k, t^k} t^k y(x^k) - \sum_{i \neq k} z_i^k y(x^i) \\ & \text{s.t. } x^k t^k = \sum_{i \neq k} z_i^k x^i + z^k x \\ & \quad \sum_{i \neq k} z_i^k + z^k = t^k \\ & \quad z_i^k \geq 0 \\ & \quad z^k = 1. \end{aligned} \tag{4}$$

Note that (4) is an LP. Therefore, for the lower bound  $\ell(x)$  we obtain the following:

$$\ell(x) := \max_{k=1, \dots, n} \left\{ \begin{array}{l} \max_{z^k, z_i^k, t^k} t^k y(x^k) - \sum_{i \neq k} z_i^k y(x^i) \\ \text{s.t. } x^k t^k = \sum_{i \neq k} z_i^k x^i + z^k x \\ \sum_{i \neq k} z_i^k + z^k = t^k \\ z_i^k \geq 0 \\ z^k = 1. \end{array} \right\}. \tag{5}$$

Note that the number of constraints in (5) is  $q + 1$ . The number of variables in (5) is also  $q + 1$ . Therefore it takes polynomial time to find the lower bound.

### 3 Convex Data-Smoothing

If the dataset is not convex, we first have to smooth the data such that it becomes convex. We distinguish between the case that we only have function evaluations and the case that we also have derivative information.

#### 3.1 Function Value Information

We only consider movement of the output data  $\tilde{y}^i$ . So, we want to minimally shift the perturbed output data  $\tilde{y}^i$  such that they become convex. In the following

optimization problem, we minimize the upward shifts  $(\delta_y^+)^i$  and the downward shifts  $(\delta_y^-)^i$  such that the new shifted output data points  $y_s^i$  become convex:

$$\begin{aligned}
 \min_{\delta_y^+, \delta_y^-, y_s} & \sum_{i=1}^n ((\delta_y^+)^i + (\delta_y^-)^i) \\
 \text{s.t. } & y_s^i = \tilde{y}^i + (\delta_y^+)^i - (\delta_y^-)^i \quad \forall i = 1, \dots, n \\
 & y_s^i \leq \sum_{k \neq i} \lambda_k^i y_s^k \quad (\forall \lambda_k^i \in [0, 1] | x^i = \sum_{k \neq i} \lambda_k^i x^k, \sum_{k \neq i} \lambda_k^i = 1), \quad (6) \\
 & \hspace{15em} \forall i = 1, \dots, n \\
 & \delta_y^+ \in \mathbb{R}_+^n, \delta_y^- \in \mathbb{R}_+^n.
 \end{aligned}$$

We minimize the  $\ell_1$ -norm. It is easy to see that in the optimum either  $(\delta_y^+)^i = 0$  or  $(\delta_y^-)^i = 0$ . The second constraint forces the shifted output data points  $y_s^i$  to become convex. Note that (6) is an LP with infinitely many constraints, i.e., it is a semi-infinite LP, which can also be seen as a robust linear programming problem. We can solve this problem with methods from [13]. Since the "uncertainty region":  $\{\forall \lambda_k^i \in [0, 1] | x^i = \sum_{k \neq i} \lambda_k^i x^k, \sum_{k \neq i} \lambda_k^i = 1\}$  of the second constraint in (6) is a polytope, we can rewrite this semi-infinite programming constraint as a collection of linear constraints without an uncertainty region. We follow the reasoning of the proof of Theorem 1 in [13] to show this. Let us consider the second constraint for a certain value of  $i$ . We can write this constraint as

$$\sum_{k \neq i} \lambda_k^i y_s^k - y_s^i \geq 0 \quad \forall (\lambda_k^i \in [0, 1] | x^i = \sum_{k \neq i} \lambda_k^i x^k, \sum_{k \neq i} \lambda_k^i = 1). \quad (7)$$

Note that this constraint is satisfied if and only if the solution of the minimization problem

$$\begin{aligned}
 \min_{\lambda_k^i} & \sum_{k \neq i} \lambda_k^i y_s^k - y_s^i \\
 \text{s.t. } & x^i = \sum_{k \neq i} \lambda_k^i x^k \\
 & \sum_{k \neq i} \lambda_k^i = 1 \\
 & \lambda_k^i \geq 0
 \end{aligned} \quad (8)$$

is nonnegative. The dual of this LP is given by:

$$\begin{aligned}
 \max_{r^i, v^i} & (x^i)^T r^i + v^i - y_s^i \\
 \text{s.t. } & (x^k)^T r^i + v^i \leq y_s^k \quad \forall k \neq i \\
 & r^i \in \mathbb{R}^q, v^i \in \mathbb{R},
 \end{aligned} \quad (9)$$

where  $e_{n-1}$  is the  $(n - 1)$ -dimensional all-one vector. Since (9) is the dual of (8), both LP's have the same optimal solution. Note that the optimal value of (9) is nonnegative if and only if there *exists* a feasible solution for (9) such that the objective function of (9) is nonnegative. We can now conclude that (7) is satisfied if and only if there exist  $r^i$  and  $v^i$ , which are feasible for (9) and have a nonnegative objective, i.e. if the following inequalities are satisfied:

$$\begin{cases} (x^i)^T r^i + v^i \geq y_s^i \\ (x^k)^T r^i + v^i \leq y_s^k \quad \forall k \neq i \\ r^i \in \mathbb{R}^q, v^i \in \mathbb{R} \end{cases} \tag{10}$$

We can now finally rewrite the second constraint in (6) as (10) for every  $i = 1, \dots, n$ . This means that we can rewrite (6) as

$$\begin{aligned} \min_{\delta_y^+, \delta_y^-, y_s, r^i, v^i} & \sum_{i=1}^n ((\delta_y^+)^i + (\delta_y^-)^i) \\ \text{s.t.} & y_s^i = \tilde{y}^i + (\delta_y^+)^i - (\delta_y^-)^i \quad \forall i = 1, \dots, n \\ & (x^i)^T r^i + v^i \geq y_s^i \quad \forall i = 1, \dots, n \\ & (x^k)^T r^i + v^i \leq y_s^k \quad \forall k \neq i, \forall i = 1, \dots, n \\ & \delta_y^+ \in \mathbb{R}_+^n, \delta_y^- \in \mathbb{R}_+^n, \\ & r^i \in \mathbb{R}^q, v^i \in \mathbb{R} \quad \forall i = 1, \dots, n, \end{aligned} \tag{11}$$

which is an LP. Note that, after substituting the equality constraints for  $y_s^i$ , the number of constraints in (11) is  $n(n - 1) + n = n^2$ . The number of variables in (11) is  $(q + 3)n$ .

Above, we minimized the sum of the absolute values of the shifts, i.e. the  $\ell_1$ -norm. However, we can also choose to minimize other norms, such as e.g., the  $\ell_\infty$ -norm or the  $\ell_2$ -norm. Using the  $\ell_\infty$ -norm, we also obtain an LP, which is similar to (11).

### 3.2 Derivative Information

Next, we consider the case in which we also have gradient information. Suppose that the underlying function is convex, but the data are not convex, due to (numerical) errors. Again, we are interested in shifting the data such that they become convex. We consider perturbed output values  $\tilde{y}^i$ , perturbed gradients  $\tilde{\nabla}y(x^i)$ , and perturbed input values  $\tilde{x}^i$ . Therefore in this case we want to minimize the shifts in the output values, in the gradients, and in the inputs. So, in the following optimization problem, we minimize the sum of upward and downward shifts  $(\delta_y^+)^i$  and  $(\delta_y^-)^i$  of the output values, the upward and downward shifts  $(\delta_g^+)^i$  and  $(\delta_g^-)^i$  of the gradient, and the upward and downward shifts  $(\delta_x^+)^i$  and  $(\delta_x^-)^i$  of the input values such that the data become convex:

$$\begin{aligned} \min_{\substack{(\delta_y^+)^i, (\delta_y^-)^i, (\delta_g^+)^i, \\ (\delta_g^-)^i, (\delta_x^+)^i, (\delta_x^-)^i, \\ x_s^i, y_s^i, (\nabla y^i)_s}} & \sum_{i=1}^n ((\delta_y^+)^i + (\delta_y^-)^i + e_q^T (\delta_g^+)^i + e_q^T (\delta_g^-)^i + e_q^T (\delta_x^+)^i + e_q^T (\delta_x^-)^i) \\ \text{s.t.} & (\nabla y^i)_s = \tilde{\nabla}y^i + (\delta_g^+)^i - (\delta_g^-)^i \quad \forall i = 1, \dots, n \\ & x_s^i = \tilde{x}^i + (\delta_x^+)^i - (\delta_x^-)^i \quad \forall i = 1, \dots, n \\ & y_s^i = \tilde{y}^i + (\delta_y^+)^i - (\delta_y^-)^i \quad \forall i = 1, \dots, n \\ & (\nabla y^i)_s^T (x_s^j - x_s^i) + y_s^i \leq y_s^j \quad \forall i, j = 1, \dots, n, i \neq j \\ & (\delta_y^+)^i \in \mathbb{R}_+, (\delta_y^-)^i \in \mathbb{R}_+, (\delta_g^+)^i \in \mathbb{R}_+^q \quad \forall i = 1, \dots, n \\ & (\delta_g^-)^i \in \mathbb{R}_+^q, (\delta_x^+)^i \in \mathbb{R}_+^q, (\delta_x^-)^i \in \mathbb{R}_+^q \quad \forall i = 1, \dots, n, \end{aligned} \tag{12}$$

where  $\nabla y^i = \nabla y(x^i)$ , and  $e_q$  is the  $q$ -dimensional all-one vector. The 4-th constraint in (12) is a necessary and sufficient condition for convexity of the data; see page 338 in [15]. However, (12) is a nonconvex optimization problem, and therefore not tractable.

However, if there is no uncertainty in the input values  $x^1, \dots, x^n$ , we can omit the variables  $(\delta_x^+)^i$  and  $(\delta_x^-)^i$  in (12), and we obtain an LP. Similarly, if there is no uncertainty in the values of the gradients, we can omit  $(\delta_g^+)^i$  and  $(\delta_g^-)^i$  in (12), and we also obtain an LP.

An example of a problem, where the gradient information is exact, and we only have errors in the input variables and output variables is in the field of multiobjective optimization. In the so-called weighted sum method, to determine a point on the Pareto curve/surface the weights determine the exact value of the gradient, whereas due to numerical errors of the solver, the input value and the output value might be subject to noise.

Note that in the formulation of (12) we have minimized the shifts  $(\delta_y^+)^i, (\delta_y^-)^i, (\delta_g^+)^i, (\delta_g^-)^i, (\delta_x^+)^i, (\delta_x^-)^i$ , and have given them all equal importance. However, we might want to give one type of the error more weight than the other type.

### 4 Numerical Examples

In this section we will consider some examples of the theory discussed in Sect. 3.

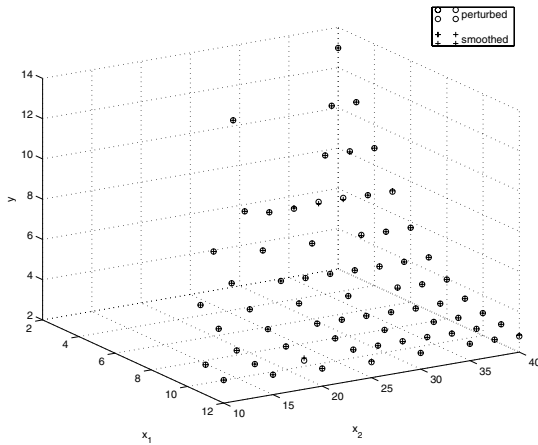
*Example 1 (artificial, no derivative information).* In this example we apply the theory that we developed in Sect. 3.1. We consider the function  $y : \mathbb{R}^2 \rightarrow \mathbb{R}$ , given by  $y(x_1, x_2) = x_1^2 + x_2^2$ . We take a sample of 10 input data points  $x^1, \dots, x^{10}$  from  $[-2, 2] \times [-2, 2]$ , and calculate the output values  $y(x^1), \dots, y(x^{10})$ . Furthermore, we add some noise to it, i.e., we add a noise  $\varepsilon_y^i$ , where the  $\varepsilon_y^i$ 's are independent and uniformly distributed on  $[-2.5, 2.5]$ , such that the data become nonconvex. We obtain values  $\tilde{y}^i = y^i + \varepsilon_y^i$ . The values are given in Table 1. We solved (11) for this problem, and the shifted data  $y_s^i$  are also given in Table 1. The values that are really shifted, are shown in italics. □

**Table 1.** Data and results of smoothing in Example 1

| number | $x_1$   | $x_2$   | $y$    | $\tilde{y}$ | $y_s$         |
|--------|---------|---------|--------|-------------|---------------|
| 1      | -0.0199 | -1.9768 | 3.9081 | 6.1588      | 6.1588        |
| 2      | 0.0925  | 1.3411  | 1.8071 | 0.4628      | 0.4628        |
| 3      | 1.4427  | 0.3253  | 2.1872 | 2.7214      | 2.7214        |
| 4      | -1.8056 | -1.1961 | 4.6908 | 4.6208      | 4.6208        |
| 5      | -0.4435 | -0.3444 | 0.3153 | 2.2718      | <i>2.0578</i> |
| 6      | -1.2952 | 0.8811  | 2.4539 | 3.7644      | 3.7644        |
| 7      | 1.7826  | 1.6795  | 5.9984 | 5.7807      | 5.7807        |
| 8      | 0.8074  | -1.3585 | 2.4974 | 0.0899      | <i>0.4842</i> |
| 9      | -0.8714 | 0.5089  | 1.0183 | 2.6254      | 2.6254        |
| 10     | 0.5779  | -0.7205 | 0.8531 | 0.5766      | 0.5766        |

*Example 2 (radio therapy, no derivative information).* In radiotherapy the main goal is to give the tumour enough radiation dose, such that the surrounding organs do not receive too much radiation dose. This problem can be formulated mathematically by a multiobjective optimization problem. With the tumour and each healthy surrounding organ, an objective function is associated. One of the problems is that the calculation of a point on the Pareto surface can be very time-consuming. Therefore, we are interested in approximating the Pareto surface; see e.g. [16]. Under certain conditions, we may assume that this Pareto surface is convex. However, due to numerical errors, the Pareto points may not be convex. Therefore we should first smooth them to make them convex.

We have data from a patient of the Radboud University Nijmegen Medical Centre, in Nijmegen, the Netherlands. This data is from a multiobjective optimization problem with 3 objectives, and has 69 data points. The data are shown in Fig. 1. The Pareto surface is a convex and decreasing function. However, it turned out that the data is not convex. By solving (11), the data is smoothed such that the data becomes convex. The smoothed data points are also shown in Fig. 1. □



**Fig. 1.** The the perturbed data and the smoothed data of Example 2

## 5 Further Research

As interesting topics for further research we mention several possible applications of the methods developed in this paper.

Possible applications for the construction of the bounds in Sect. 2 are:

- Extend the Sandwich algorithms as exist for the approximation of univariate convex functions to the multivariate case by using the lower and upper bounds. More specifically, this may be useful for approximating convex Pareto surfaces and black-box functions (e.g. deterministic computer simulation).



- Use the lower bounds in convex optimization. For each new candidate proposed by the nonlinear programming solver, we can calculate the lower bound, and if this lower bound is larger than the best known objective value up to now, we reject the candidate before evaluating its function value. This may reduce computation time, especially when the function evaluation is time-consuming. In [17] promising results are shown for the univariate case.

Possible applications for the data-smoothing methods of Sect. 3 are:

- Apply data-smoothing before applying Sandwich type algorithms. This may be necessary because of (numerical) noise. This noise occurs e.g., when we want to estimate a Pareto surface in the field of multiobjective optimization.
- Apply data-smoothing to multivariate isotonic regression problems.
- Apply the data-smoothing techniques to sampling methods to assess the convexity/concavity of multivariate nonlinear functions; see [18].

## References

1. Kuijt, F.: Convexity preserving interpolation – Stationary nonlinear subdivision and splines. PhD thesis, University of Twente, Enschede, The Netherlands (1998)
2. Siem, A.Y.D., de Klerk, E., den Hertog, D.: Discrete least-norm approximation by nonnegative (trigonometric) polynomials and rational functions. CentER Discussion Paper 2005-73, Tilburg University, Tilburg (2005)
3. den Hertog, D., de Klerk, E., Roos, K.: On convex quadratic approximation. *Statistica Neerlandica* **563** (2002) 376–385
4. Burkard, R.E., Hamacher, H.W., Rote, G.: Sandwich approximation of univariate convex functions with an application to separable convex programming. *Naval Research Logistics* **38** (1991) 911–924
5. Fruhwirth, B., Burkard, R.E., Rote, G.: Approximation of convex curves with application to the bi-criteria minimum cost flow problem. *European Journal of Operational Research* **42** (1989) 326–338
6. Rote, G.: The convergence rate of the Sandwich algorithm for approximating convex functions. *Computing* **48** (1992) 337–361
7. Siem, A.Y.D., den Hertog, D., Hoffmann, A.L.: A method for approximating univariate convex functions using only function evaluations. Working paper, Tilburg University, Tilburg (2005)
8. Yang, X.Q., Goh, C.J.: A method for convex curve approximation. *European Journal of Operational Research* **97** (1997) 205–212
9. Cullinan, M.P.: Data smoothing using non-negative divided differences and  $\ell_2$  approximation. *IMA Journal of Numerical Analysis* **10** (1990) 583–608
10. Demetriou, I.C., Powell, M.J.D.: Least squares smoothing of univariate data to achieve piecewise monotonicity. *IMA Journal of Numerical Analysis* **11** (1991) 411–432
11. Demetriou, I.C., Powell, M.J.D.: The minimum sum of squares change to univariate data that gives convexity. *IMA Journal of Numerical Analysis* **11** (1991) 433–448
12. Barlow, R.E., Bartholomew, R.J., Bremner, J.M., Brunk, H.D.: *Statistical inference under order restrictions*. Wiley, Chichester (1972)
13. Ben-Tal, A., Nemirovski, A.: *Robust optimization - methodology and applications*. *Mathematical Programming, Series B* **92** (2002) 453–480

14. Charnes, A., Cooper, W.W.: Programming with linear fractional functional. *Naval Research Logistics Quarterly* **9** (1962) 181–186
15. Boyd, S., Vandenberghe, L.: *Convex optimization*. Cambridge University Press, Cambridge (2004)
16. Hoffmann, A.L., Siem, A.Y.D., den Hertog, D., Kaanders, J.H.A.M., Huizenga, H.: Dynamic generation and interpolation of pareto optimal IMRT treatment plans for convex objective functions. Working paper, Radboud University Nijmegen Medical Centre, Nijmegen (2005)
17. den Boef, E., den Hertog, D.: Efficient line searching for convex functions. CentER Discussion Paper 2004-52, Tilburg University, Tilburg (2004)
18. Chinneck, J.W.: Discovering the characteristics of mathematical programs via sampling. *Optimization Methods and Software* **17**(2) (2002) 319–352