Regularized monotonic regression

Oleg Burdakov[1] and Oleg Sysoev[2]

Abstract Monotonic (isotonic) Regression (MR) is a powerful tool used for solving a wide range of important applied problems. One of its features, which poses a limitation on its use in some areas, is that it produces a piecewise constant fitted response. For smoothing the fitted response, we introduce a regularization term in the MR formulated as a least distance problem with monotonicity constraints. The resulting Smoothed Monotonic Regrassion (SMR) is a convex quadratic optimization problem. We focus on the SMR, where the set of observations is completely (linearly) ordered. Our Smoothed Pool-Adjacent-Violators (SPAV) algorithm is designed for solving the SMR. It belongs to the class of dual active-set algorithms. We proved its finite convergence to the optimal solution in, at most, n iterations, where n is the problem size. One of its advantages is that the active set is progressively enlarging by including one or, typically, more constraints per iteration. This resulted in solving large-scale SMR test problems in a few iterations, whereas the size of that problems was prohibitively too large for the conventional quadratic optimization solvers. Although the complexity of the SPAV algorithm is $O(n^2)$, its running time was growing in our computational experiments almost linearly with n.

Keywords: Monotonic regression, regularization, quadratic penalty, convex quadratic optimization, dual active-set method, large-scale optimization.

1 Introduction

The Monotonic Regression (MR) is aimed at learning monotonic dependence from a given data set [2, 21]. The enormous amount of publications related to the MR, as well as a growing variety of its application areas, testifies to its exceptional importance. Examples are found in such areas as operations research [18], genetics [11], environmental science [15], meteorology [22], psychology [16] and many others. One can find very large-scale MR problems, e.g., in machine learning [8, 13, 28] and computer simulations [7].

The applications of the MR are often related to a monotonic data fitting, where it is assumed that there exists an unknown monotonic response function $\chi(t)$ of an explanatory variable t. In this paper we focus on the univariate case and suppose that $\chi(t)$ is monotonically increasing, i.e.

$$\chi(t') \leq \chi(t''), \quad \forall t' \leq t''.$$

For a linearly ordered sequence of observed values of the explanatory variable $t_1 < t_2 < \ldots < t_n$, the corresponding sequence of observed response values

$$a_i = \chi(t_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

[1]Department of Mathematics, Linköping University, Linköping, Sweden.
E-mail: oleg.burdakov@liu.se
[2]Department of Computer and Information Science, Linköping University, Linköping, Sweden.
E-mail: oleg.sysoev@liu.se

is supposed to be available, where ϵ_i is an observation error. Because of the errors, the expected monotonicity $a_i \leq a_{i+1}$ may be violated for some indexes i. The MR problem is aimed at restoring the perturbed monotonicity by finding a least-change correction to the observed values. It can be stated formally as a quadratic optimization problem in the following way:

$$\min_{x \in R^n} \sum_{i=1}^{n} w_i (x_i - a_i)^2, \qquad \text{s.t.} \quad x_1 \leq x_2 \leq \ldots \leq x_n, \tag{2}$$

where $w \in R^n_{++}$ is a vector of weights.

The most efficient algorithm used for solving this problem is a, so-called, *pool adjacent violators* (PAV) algorithm [1, 17, 19]. It can be viewed as a dual active set algorithm [3, 10]. The efficiency and popularity of the PAV algorithm is explained mainly by its linear computational complexity, $O(n)$.

Let x^* be the solution to the MR problem. The active constraints suggest that the components of x^* are partitioned into blocks of consecutive components of equal values. Let $x(t)$ be a monotonic function which satisfies the interpolation condition:

$$x(t_i) = x_i^*, \quad \forall i \in [1, n].$$

Here and later, the set of indexes $\{i, i+1, \ldots, j-1, j\}$ is denoted by $[i, j]$ and referred to as a *segment of indexes*. Because of the block structure of x^*, the shape of $x(t)$ resembles a step function, suggesting that it may have sharp changes on certain intervals of t, where the response function $\chi(t)$ does not increase so rapidly. This feature of the MR problem is often criticized, and it motivates the necessity of smoothing the MR solution.

Consider the following regularized monotonic regression problem:

$$\min_{x \in R^n} \sum_{i=1}^{n} w_i (x_i - a_i)^2 + \sum_{i=1}^{n-1} \mu_i (x_i - x_{i+1})^2, \qquad \text{s.t.} \quad x_i \leq x_{i+1}, \ \forall i \in [1, n-1], \tag{3}$$

where $\mu \in R^{n-1}_{+}$ is a vector of penalty parameters. The penalty term in (3) is aimed at smoothing functions which interpolate the solution to this problem. As an example, consider the choice

$$\mu_i = \frac{\mu}{(t_{i+1} - t_i)^2}\ , \tag{4}$$

in which case the penalty term

$$\mu \sum_{i=1}^{n-1} \left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i} \right)^2$$

involves a finite-difference approximation of the first derivative. We shall refer to (3) as *smoothed monotonic regression* (SMR) problem. Since it is a quadratic optimization problem with strictly convex objective function, its solution exists and unique. When $\mu = 0$, problem (3) is obviously reduced to (2).

In the accompanying paper [26], we present a statistical analysis of the SMR problem. In particular, it is shown how to properly choose the values of the penalty parameters μ_i by making use of bayesian modelling and a cross-validation technique. The numerical results in that paper reveal that the solution to the problem (3) provides a better predictive power in comparison to the commonly used alternative approaches of a similar computational complexity used for monotonic smoothing and prediction. It is also shown in [26] that the computational complexity of our *smoothed pool-adjacent-violators* (SPAV) algorithm aimed at solving the SMR problem is $O(n^2)$, under the assumption that it converges to the optimal solution.

Here, we present a detailed analysis of this optimization algorithm viewed as a dual active-set algorithm. Its important feature is that the active set is always expanding by making active, typically, more than one constraint per iteration. The version of the SPAV algorithm considered her is more general than in [26] because it allows for starting not only from an empty active set.

The paper is organized as follows. In the next section, a subproblem determined by the set of active constraints is considered, and the SPAV algorithm is described. It is also shown that, when $\mu = 0$, the SPAV is reduced to the algorithm of complexity $O(n)$ developed in [14], where the primal-dual active-set (PDAS) algorithm [9] was tailored to solving the MR problem. Section 3 is devoted to studying some properties of the SPAV algorithm. We show, in particular, that the Lagrange multiplies do not decrease at each iteration, which allows us to prove that the SPAV algorithm converges to the optimal solution in, at most, n iterations. In section 4, results of numerical experiments are presented. They illustrate how the desired smoothing is performed by the SPAV algorithm. In our experiments, its running time was growing almost linearly with n, rather than in proportion to n^2 as suggested by the worst-case analysis. Finally, we close with concluding remarks in section 5 where, in particular, we discuss an extension of the SMR problem from complete to partially ordered set of observations.

2 SPAV algorithm

We shall refer to $x_i \leq x_{i+1}$ in the SMR as constraint i. Each iteration of our algorithm is related to solving the subproblem

$$\min_{x \in R^n} \sum_{i=1}^{n} w_i (x_i - a_i)^2 + \sum_{i=1}^{n-1} \mu_i (x_i - x_{i+1})^2, \qquad \text{s.t.} \quad x_i = x_{i+1}, \ \forall i \in S, \tag{5}$$

where $S \subseteq [1, n-1]$ is an *active set*. We denote its unique optimal solution by $x(S)$.

For presenting an efficient way of solving this subproblem, we consider the optimality conditions. They will also be used in the next section for studying the convergence properties of the algorithm.

The active set S suggests that there exist sets of consecutive indices of the form $[\ell, r] \subseteq [1, n]$ such that

$$\begin{cases} \ell - 1 \notin S, \\ j \in S, \ \forall j \in [\ell, r-1], \\ r \notin S. \end{cases}$$

We call these sets *blocks*. Note that a block may be a singleton when $\ell = r$. Then the total number of blocks, denoted here by m, is equal to $n - |S|$. The block partitioning (segmentation) of $[1, n]$ associated with S can be represented as

$$[1, n] = [\ell_1, r_1], [\ell_2, r_2], \ldots, [\ell_m, r_m],$$

where $\ell_1 = 1$, $r_m = n$ and $r_i + 1 = \ell_{i+1}$ for all $i \in [1, m-1]$.

Each block i is characterized by its common value

$$y_i = x_{\ell_i} = x_{\ell_i+1} = \ldots = x_{r_i},$$

common weight

$$w'_i = w_{\ell_i} + w_{\ell_i+1} + \ldots + w_{r_i}$$

and average observed value

$$a'_i = \frac{1}{w'_i} \sum_{j=\ell_i}^{r_i} w_j a_j.$$

Denoting $\mu'_i = \mu_{r_i}$, we can write the subproblem (5) in the notation introduced above as

$$\min_{y \in R^m} c + \sum_{j=1}^{m} w'_j (y_j - a'_j)^2 + \sum_{j=1}^{m-1} \mu'_j (y_j - y_{j+1})^2, \tag{6}$$

where the scalar c does not depend on y. The optimality conditions for (6) are given by the system of linear equations

$$\begin{cases} w'_1(y_1 - a'_1) + \mu'_1(y_1 - y_2) = 0, \\ \cdots \\ w'_j(y_j - a'_j) + \mu'_{j-1}(y_j - y_{j-1}) + \mu'_j(y_j - y_{j+1}) = 0, \\ \cdots \\ w'_m(y_m - a'_m) + \mu'_{m-1}(y_m - y_{m-1}) = 0. \end{cases} \tag{7}$$

Its solution, denoted by $y(S)$, is unique because the objective function in (6) is strictly convex.

The algorithm starts with any active set such that $S \subseteq S^*$. The simplest of the valid choices is $S = \emptyset$. At each iteration, it solves the tridiagonal system of linear equations (7), and then it extends the set S by additionally making active the constraints in (3) for which the strict monotonicity $y_i(S) < y_{i+1}(S)$ is violated. This, like in the PAV algorithm, assumes merging the corresponding adjacent blocks, which explains why we call our algorithm SPAV (smoothed pool-adjacent-violators). The merging is associated with updating the coefficients that define the linear system (7). The corresponding number of arithmetic operations is proportional to the number of new active constraints. In contrast to the conventional active set algorithms, SPAV may enlarge the active set with more than one element at once. It operates with the block common values y_i, whereas the values of x_i are computed only at its terminal stage. The outlined algorithm can be formally expressed as follows.

Algorithm 1. SPAV
input $a \in R^n$, $w \in R^n_{++}$, $\mu \in R^{n-1}_+$, $S \subseteq S^*$
compute a', w' and μ'
find $y(S)$ that solves (7)
while $y(S)$ is not strictly monotone **do**
  set $S \leftarrow S \cup \{r_i : \; y_i(S) \geq y_{i+1}(S)\}$
  update a', w' and μ'
  find $y(S)$ that solves (7)
end while
return $x(S)$

It is shown in [26] that the computational complexity of the SPAV algorithm is $O(n^2)$. This estimate is based on the following two observations. First, the active set S is extended in the while loop by including, at leased, one index, which means that the number of the while loop iterations does not exceed $n-1$. Second, the computational complexity of solving the tridiagonal linear system (7) is $O(n)$. The cost of updating a', w' and μ' is a small multiple of the number of blocks merged at the same iteration, which means that the total sum of operations, associated with updating these vectors, over all iterations is $O(n)$.

As it will follow from the results of the next section, the active set S produced by the SPAV is such that, at each iteration the inclusion $S \subseteq S^*$ is maintained, and after the final iteration it turns out that $S = S^*$. The algorithm can start from any set $S \subseteq S^*$, even though some of the Lagrange multipliers in (5) may be negative. This enables the algorithm to be warm-started by providing a good initial point. If there is no guarantee that $S \subseteq S^*$ holds for the initial S, the recursive calculation of the Lagrange multiplies, as described in the next section, allows for attaining the desired inclusion $S \subseteq S^*$ by splitting certain blocks. The negative Lagrange multipliers indicate how to split the blocks, namely, by making inactive the corresponding monotonicity constraints. Given S, if all the Lagrange multipliers are non-negative, this guarantees that $S \subseteq S^*$.

Note that, when $\mu_i = 0$ for all $i \in [1, n-1]$, the SMR reduces to the MR problem. This permits us to apply the SPAV algorithm to solving the latter problem. In this case, the complexity of the algorithm, that we shall refer to as SPAV$_0$, reduces to $O(n)$, which is the same as for the PAV algorithm. This follows from the facts that (7) becomes a diagonal linear system whose solution is $y_j = a'_j$ for all $j \in [1, m]$, and that the merging of blocks changes only those components of y that correspond to the new blocks. The MR version of the SPAV algorithm can be formally expressed for the initial $S = \emptyset$ as follows.

Algorithm 2. SPAV$_0$
input $a \in R^n$, $w \in R^n_{++}$
set $S \leftarrow \emptyset$, $a' \leftarrow a$ and $w' \leftarrow w$
set $y(S) \leftarrow a'$
while $y(S)$ is not strictly monotone **do**
  set $S \leftarrow S \cup \{r_i : \ y_i(S) \geq y_{i+1}(S)\}$
  update a' and w'
  set $y(S) \leftarrow a'$
end while
return $x(S)$

It should be mentioned that the iterates generated by SPAV$_0$ are not the same as those generated by the PAV algorithm, but they are identical with those generated for the initial $S = \emptyset$ by the PDAS-type algorithm proposed in [14].

3 Convergence of SPAV

Note that the SMR is a strictly convex optimization problem, because the objective function in (3) is strictly convex and the constraints are linear. It has a unique optimal solution determined by the Karush-Kuhn-Tucker (KKT) conditions [20]. For deriving these conditions, we use the Lagrangian function

$$L(x,\lambda) = \sum_{i=1}^{n} w_i(x_i - a_i)^2 + \sum_{i=1}^{n-1} \mu_i(x_i - x_{i+1})^2 + \sum_{i=1}^{n-1} \lambda_i(x_i - x_{i+1}). \tag{8}$$

The condition $\nabla_x L(x,\lambda) = 0$ is written as

$$\begin{cases} 2w_1(x_1 - a_1) + 2\mu_1(x_1 - x_2) + \lambda_1 = 0, \\ 2w_i(x_i - a_i) + 2\mu_{i-1}(x_i - x_{i-1}) + 2\mu_i(x_i - x_{i+1}) + \lambda_i - \lambda_{i-1} = 0, \ \forall i \in [2, n-1], \\ 2w_n(x_n - a_n) + 2\mu_{n-1}(x_n - x_{n-1}) - \lambda_{n-1} = 0. \end{cases} \tag{9}$$

The rest of the KKT conditions has the form

$$x_i \le x_{i+1}, \quad \forall i \in [1, n-1], \tag{10}$$

$$\lambda_i \ge 0, \quad \forall i \in [1, n-1], \tag{11}$$

$$\lambda_i(x_i - x_{i+1}) = 0, \quad \forall i \in [1, n-1]. \tag{12}$$

Cosider now the subproblem (5). Not only its solution $x(S)$ is unique, but also the optimal Lagrange multipliers because the gradients of the constraints in (5) are linearly independent. If to set

$$\lambda_i = 0, \quad \forall i \in [1, n-1] \setminus S, \tag{13}$$

in (8), the resulting function

$$L_S(x,\lambda) = \sum_{i=1}^{n} w_i(x_i - a_i)^2 + \sum_{i=1}^{n-1} \mu_i(x_i - x_{i+1})^2 + \sum_{i \in S} \lambda_i(x_i - x_{i+1}).$$

can serve as the Lagrangian function for (5). Let $\lambda(S)$ be the vector in R^{n-1} whose components $\lambda_i(S)$, $i \in S$, are the optimal Lagrange multipliers for (5), and the rest of them are defined by (13). This property of the $L_S(x,\lambda)$ will later be used for proving optimality of the solution produced by the SPAV algorithm.

The condition (9) establishes a dependence of the Lagrange multipliers on x, and hence, on the common block values y. We will study now monotonic properties of this dependence. Given an active set S, consider any of the corresponding blocks, say, block i. Let the block be non-singleton, i.e. $\ell_i < r_i$. If the left neighbor of the block exists, i.e. $i > 1$, then by (13), we have $\lambda_{\ell_i - 1} = 0$. For its right neighbor, if $i < m$, we similarly have $\lambda_{r_i} = 0$. As it will be shown below, the part of the linear equations (9):

$$\nabla_{x_k} L(x,\lambda) = 0, \tag{14}$$

where $k = \ell_i, \dots, r_i - 1$, uniquely determines a dependence of each λ_j, $j \in [\ell_i, r_i - 1]$, on y_i and also on the value of $y_i - y_{i-1}$, provided that y_{i-1} exists. We denote this function by $\lambda_j(y_i, y_i - y_{i-1})$, assuming for $i = 1$ that λ_j does not change with $y_1 - y_0$, as if $\mu_0 = 0$. For $k = \ell_i + 1, \dots, r_i$, the system of linear equations (14) uniquely determines a dependence of each λ_j, $j \in [\ell_i, r_i - 1]$, on y_i and also on the value of $y_i - y_{i+1}$, provided that y_{i+1} exists. Like above, this dependence is conventionally denoted by $\lambda_j(y_i, y_i - y_{i+1})$, assuming that λ_j does not change with $y_m - y_{m+1}$. A monotonic dependence of the Lagrange multiplies as a function of the block common values is presented by the following result. It will later be used for showing that, at every iteration of the SPAV algorithm, each component of the vector $\lambda(S)$ does not decrease.

Lemma 1 *Let a non-singleton block i be defined by an active set S. Then, for any $j \in [\ell_i, r_i - 1]$, the functions $\lambda_j(y_i, y_i - y_{i-1})$ and $\lambda_j(y_i, y_i - y_{i+1})$ are uniquely determined by the corresponding parts of (9). Further, $\lambda_j(y_i, y_i - y_{i-1})$ decreases with y_i, and it does not increase with $y_i - y_{i-1}$. Finally, $\lambda_j(y_i, y_i - y_{i+1})$ is an increasing and non-decreasing function of y_i and $y_i - y_{i+1}$, respectively.*

Proof. For simplicity, we drop the index i in ℓ_i and r_i. For $k = \ell, \dots, r-1$, the linear system (14) recursively defines the Lagrange multipliers as

$$\begin{aligned}
\lambda_\ell &= -2w_\ell(y_i - a_\ell) - 2\mu_{\ell-1}(y_i - y_{i-1}),\\
\lambda_j &= \lambda_{j-1} - 2w_j(y_i - a_j), \quad j = \ell+1, \ell+2, \dots, r-1,
\end{aligned}$$

where the term $-2\mu_{\ell-1}(y_i - y_{i-1})$ it to be omitted when $i = 1$. This recursion indicates that each $\lambda_j(y_i, y_i - y_{i-1})$, $\ell \le j < r$, decreases with y_i, and it does not increase with $y_i - y_{i-1}$, because $w > 0$ and $\mu \ge 0$. The reverse recursion

$$\begin{aligned}
\lambda_{r-1} &= 2w_r(y_i - a_r) + 2\mu_r(y_i - y_{i+1}),\\
\lambda_{j-1} &= \lambda_j + 2w_j(y_i - a_j), \quad j = r-1, r-2, \dots, \ell+1,
\end{aligned}$$

derived from the linear system that corresponds to $k = \ell+1, \dots, r$ in (14), proves the last statement of the lemma. This completes the proof. ∎

For an arbitrary index $k \in [1, m-1]$, consider the problem obtained from (6) by excluding from the objective function the terms

$$w'_k(y_k - a'_k)^2 + w'_{k+1}(y_{k+1} - a'_{k+1})^2 + \mu'_k(y_k - y_{k+1})^2$$

and viewing y_k and y_{k+1} as parameters. The resulting problem is decoupled into the following two subproblems:

$$\min_{y_1,\dots,y_{k-1}} \sum_{j=1}^{k-1} \left[w'_j(y_j - a'_j)^2 + \mu'_j(y_j - y_{j+1})^2 \right] \tag{15}$$

and

$$\min_{y_{k+2},\dots,y_m} \sum_{j=k+2}^{m} \left[w'_j(y_j - a'_j)^2 + \mu'_{j-1}(y_j - y_{j-1})^2 \right]. \tag{16}$$

We denote the unique solutions to these subproblems by

$$y_1(y_k), \dots, y_{k-1}(y_k) \quad \text{and} \quad y_{k+2}(y_{k+1}), \dots, y_m(y_{k+1}), \tag{17}$$

respectively. In the next result, their monotonic dependence on y_k and y_{k+1} are studied.

Lemma 2 *The components (17) of the optimal solutions to subproblems (15) and (16) are linearly non-decreasing functions of y_k and y_{k+1}, respectively. Moreover, the differences*

$$y_{j+1}(y_k) - y_j(y_k), \quad j = 1, \dots, k-1,$$

and

$$y_j(y_{k+1}) - y_{j+1}(y_{k+1}), \quad j = k+1, \dots, m-1,$$

are also non-decreasing functions of y_k and y_{k+1}, respectively.

Proof. The optimality conditions for (15) are represented by the first $k-1$ equations in (7). Since the left-hand side of this system of equations is a linear functional of $y_1, \dots, y_k$, its solution $y_1(y_k), \dots, y_{k-1}(y_k)$ linearly depends on y_k. The linearity of $y_{k+2}(y_{k+1}), \dots, y_m(y_{k+1})$ is obtained in a similar way.

The first $k-1$ equations in (7) can be represented as

$$M\bar{y}(y_k) = y_k \mu'_{k-1} e_{k-1}, \tag{18}$$

where $\bar{y}(y_k) = (y_1(y_k), \dots, y_{k-1}(y_k))^T$, $e_{k-1} = (0, \dots, 0, 1)^T$ and

$$M = \operatorname{diag}(w'_1, \dots, w'_{k-1}) + \begin{bmatrix} \mu'_1 & -\mu'_1 & & & \\ -\mu'_1 & (\mu'_1 + \mu'_2) & -\mu'_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -\mu'_{k-3} & (\mu'_{k-3} + \mu'_{k-2}) & -\mu'_{k-2} \\ & & & -\mu'_{k-2} & (\mu'_{k-2} + \mu'_{k-1}) \end{bmatrix}.$$

It can be easily seen that the tridiagonal matrix $M \in R^{(k-1)\times(k-1)}$ is positive definite. Indeed, for any non-zero vector $v \in R^{k-1}$, we have

$$v^T M v = \sum_{j=1}^{k-1} w'_j v_j^2 + \sum_{j=1}^{k-2} \mu'_j (v_j - v_{j+1})^2 + \mu'_{k-1} v_{k-1}^2 > 0,$$

because $w' \in R^m_{++}$ and $\mu' \in R^{m-1}_{+}$. Thus, M is a real symmetric positive definite matrix with non-positive off-diagonal entries. Then it is a Stieltjes matrix whose property is that all entries of its inverse are non-negative [12], which implies that $M^{-1} e_{k-1} \geq 0$. Hence, each component of the solution $\bar{y}(y_k)$ to the linear system (18) is a non-decreasing function of y_k.

Note that if there exists an index $k' < k$ such that $\mu'_{k'} = 0$, then $y_1(y_k), \dots, y_{k'}(y_k)$ do not change with y_k. Consequently, $y_{j+1}(y_k) - y_j(y_k)$ for all $j < k'$ are non-decreasing functions of y_k. The same refers to the function $y_{k'+1}(y_k) - y_{k'}(y_k)$ because, as it was shown above, $y_{k'+1}(y_k)$ does not decrease with y_k. If $\mu'_{k-1} > 0$, that is $k' \neq k-1$, we need to show that the same refers also to all $j \in [k'+1, k-1]$, where

$$k' = \min_{0 \leq i < k-1} \{ i : \ \mu'_j > 0, \ \forall j \in [i+1, k-1] \}.$$

Here either $k' = 0$, or $\mu'_{k'} = 0$. In order not to separately consider $k' = 0$ as a special case, we introduce artificial $\mu'_0 = 0$ and constant-valued function $y_0(y_k)$. This does not change the relations established by (7). The equations of this linear system that correspond to $j \in [k'+1, k-1]$ can be represented in the form of the following recursive relation

$$\mu'_j (y_{j+1}(y_k) - y_j(y_k)) = w'_j (y_j(y_k) - a'_j) + \mu'_{j-1} (y_j(y_k) - y_{j-1}(y_k)),$$

where $\mu'_j > 0$. Then since $y_{k'+1}(y_k) - y_{k'}(y_k)$ is a no-decreasing function of y_k, it can be easily shown by induction in $j = k'+1, \dots, k-1$ that $y_{j+1}(y_k) - y_j(y_k)$ does not decrease with y_k.

The dependence on y_{k+1} established by the last $m-k-1$ equations in (7) is studied in an analogous way. Like above, we represent them as

$$M\bar{y}(y_{k+1}) = y_{k+1} \mu'_k e_1,$$

where $\bar{y}(y_{k+1}) = (y_1(y_{k+2}), \ldots, y_m(y_{k+1}))^T$, $e_1 = (1, 0, \ldots, 0)^T$, and although here M is not the same as in (18), it is also a Stieltjes matrix of a similar structure. This allows us to prove that each component of $\bar{y}(y_{k+1})$ is a non-decreasing function of y_{k+1}. Then to prove that the functions $y_{j-1}(y_{k+1}) - y_j(y_{k+1})$ decrease with y_{k+1}, we use a reverse induction in decreasing order of j based on the backward recursion:

$$\mu'_{j-1}(y_{j-1}(y_{k+1}) - y_j(y_{k+1})) = w'_j(y_j(y_{k+1}) - a'_j) + \mu'_j(y_j(y_{k+1}) - y_{j+1}(y_{k+1}))$$

This concludes the proof. □

Let y^* be the optimal solution to problem (6). Suppose that the monotonicity is violated for some of its components. Let $k \in [1, m-1]$ be such that the inequality

$$y^*_k > y^*_{k+1} \tag{19}$$

holds. Consider a strictly convex quadratic programming problem which have only one constraint $y_k \leq y_{k+1}$ and the same objective function as in (6). Let y^{**} stand for the optimal solution to this constrained problem. Clearly, $y^{**}_k = y^{**}_{k+1}$.

After skipping the constant c in (6), the constrained problem can be written as

$$\min_{y_k \leq y_{k+1}} w'_k(y_k - a'_k)^2 + w'_{k+1}(y_{k+1} - a'_{k+1})^2 + \mu'_k(y_k - y_{k+1})^2 + \varphi_1(y_k) + \varphi_2(y_{k+1}), \tag{20}$$

where

$$\varphi_1(y_k) = \sum_{j=1}^{k-1} \left[w'_j(y_j(y_k) - a'_j)^2 + \mu'_j(y_j(y_k) - y_{j+1}(y_k))^2 \right]$$

and

$$\varphi_2(y_{k+1}) = \sum_{j=k+2}^{m} \left[w'_j(y_j(y_{k+1}) - a'_j)^2 + \mu'_{j-1}(y_j(y_{k+1}) - y_{j-1}(y_{k+1}))^2 \right]$$

are optimal objective function values in problems (15) and (16), respectively. The next result presents a relation between y^* and y^{**}.

Lemma 3 *Let inequality (19) hold. Then*

$$y^*_k > y^{**}_k = y^{**}_{k+1} > y^*_{k+1}. \tag{21}$$

Proof. The equality $y^{**}_k = y^{**}_{k+1}$ is a straightforward implication from the strict convexity of the objective function in problem (20).

By Lemma 2, the functions (17) are linear, which means that $\varphi_1(y_k)$ and $\varphi_2(y_{k+1})$ are convex quadratic functions. Then problem (20) can be reduced to the two dimensional problem

$$\min_{y_k \leq y_{k+1}} w''_k(y_k - a''_k)^2 + w''_{k+1}(y_{k+1} - a''_{k+1})^2 + \mu'_k(y_k - y_{k+1})^2, \tag{22}$$

where we skip the terms that do not depend on y_k or y_{k+1}. Here, as it can be easily verified, the multipliers w''_k and w''_{k+1} are strictly positive. From the optimality conditions for this problem, we obtain

$$\begin{cases} w''_k(y_k - a''_k) + \mu'_k(y_k - y_{k+1}) + \lambda/2 = 0, \\ w''_{k+1}(y_{k+1} - a''_{k+1}) - \mu'_k(y_k - y_{k+1}) - \lambda/2 = 0, \end{cases} \tag{23}$$

where $\lambda \geq 0$ is a Lagrange multiplier. Taking into account that $y_k^{**} = y_{k+1}^{**}$, we denote this value by $\bar{y}$. Notice that y_k^* and y_{k+1}^* solve the unconstrained version of problem (22), and they correspond in (23) to the case of $\lambda = 0$. Then simple manipulations with (23) yield the relation

$$\alpha y_k^* + (1-\alpha) y_{k+1}^* = \bar{y},$$

where

$$\alpha = w_k''/(w_k'' + w_{k+1}'').$$

Since $\alpha \in (0,1)$, this implies (21) and completes our proof. □

In the next result, we study some important properties of merging two adjacent blocks, say, blocks k and $k+1$. If S is a current active set, the merging assumes making active the constraint $x_{r_k} \leq x_{r_k+1}$ in addition the active constraints determined by S.

Lemma 4 *Let S be an active set such that there exists a block index $k \leq m-1$ for which $y_k(S) \geq y_{k+1}(S)$. Then*

$$\lambda(S') \geq \lambda(S), \tag{24}$$

where $S' = S \cup \{r_k\}$. Moreover,

$$x_i(S) \geq x_{i+1}(S) \;\Rightarrow\; x_i(S') \geq x_{i+1}(S'). \tag{25}$$

Proof. The statement of this lemma trivially holds when $y_k(S) = y_{k+1}(S)$ because the corresponding merging does not change any block common value.

Consider the case when $y_k(S) > y_{k+1}(S)$. The vector of new block common values $y(S')$ is obtained from problem (20). Then Lemma 3 yields

$$y_k(S) > y_k(S') > y_{k+1}(S).$$

These inequalities together with Lemma 2 imply

$$x_i(S') \leq x_i(S), \quad \forall i \in [1, r_k], \tag{26}$$

$$x_i(S') \geq x_i(S), \quad \forall i \in [r_k+1, n], \tag{27}$$

$$x_{j+1}(S') - x_i(S') \leq x_{j+1}(S) - x_i(S), \quad \forall i \in [1, n-1], i \neq r_k. \tag{28}$$

Statement (25) imideatly followas from (28). By Lemma 1, we conclude from (26)-(28) that

$$\lambda_i(S') \geq \lambda_i(S), \quad \forall i \in [1, n-1], i \neq r_k.$$

Recalling that $x_{r_k}(S) \geq x_{r_k+1}(S)$, and that $x(S')$ solves the problem obtained from (5) by adding the constraint $x_{r_k} = x_{r_k+1}$, we get $\lambda_{r_k}(S') \geq 0$. Then the inequality

$$\lambda_{r_k}(S') \geq \lambda_{r_k}(S)$$

holds because $\lambda_{r_k}(S) = 0$, so the proof is complete. □

The statement (25) says, in particular, that a monotonicity constraint, if violated, remains violated after making active another violated monotonicity constraint. This property will allow us to justify the potentially massive enlargement of the active set at each iteration of the SPAV algorithm, when more than one violated constraint may be simultaneously turned into active.

We are now in a position to prove a finite-step convergence of the algorithm.

Theorem 5 *For any initial $S \subseteq S^*$, the SPAV algorithm converges to the optimal solution of the SMR problem in, at most, $n-1-|S|$ iterations. Moreover, after the final iteration, $S = S^*$.*

Proof. At each iteration of the algorithm, the active set S is extended by adding at least one index of the set $[1, n-1]$ which is not contained in S. The SPAV terminates when $y(S)$ becomes monotone. This happens when either $|S| < n-1$ or $S = [1, n-1]$. In the latter case, $m = 1$ and, therefore, there is no violation of the monotonicity. Hence the number of iterations is less than $n - |S|$, where $|S|$ is the number of constraints in the initial active set.

We need now to prove that the algorithm returns the optimal solution to problem (3). To this end, we will, first, show that

$$\lambda(S) \geq 0 \;\Rightarrow\; \lambda(S \cup \Delta S) \geq 0, \tag{29}$$

where

$$\Delta S = \{r_i : \; y_i(S) \geq y_{i+1}(S)\}.$$

Clearly, the result of merging the set of blocks determined by ΔS is equivalent to the result of successively merging the same blocks one-by-one. By Lemma 4, as the result of making active any next monotonicity constraint $r_k \in \Delta S$, the Lagrange multipliers do not decrease and the statement (25) remains true for any $i \in \Delta S$. Consequently, we obtain the inequality

$$\lambda(S \cup \Delta S) \geq \lambda(S), \tag{30}$$

which proves (29).

Consider, first, the case when the initial active set is empty, which means that initially $\lambda(S) = 0$. Then from inequality (30), we get $\lambda(S) \geq 0$ for the terminal active set S. Moreover, the $x(S)$ returned by the algorithm is feasible in the SMR problem. Since the block common values $y(S)$ satisfy equation (7), the corresponding $x(S)$ and $\lambda(S)$ satisfy equation (9). Thus, all the KKT conditions (9)-(12) for problem (3) hold, which means for the initial $S = \emptyset$ that the $x(S)$ returned by the SPAV algorithm solves the SMR problem. Consequently, after the final iteration, we have $S = S^*$.

Consider now a more general case of the initial active set such that $S \subseteq S^*$. Notice that problem (3) is equivalent to

$$\min_{x \in R^n} \sum_{i=1}^{n} w_i (x_i - a_i)^2 + \sum_{i=1}^{n-1} \mu_i (x_i - x_{i+1})^2, \quad \text{s.t. } x_i \leq x_{i+1}, \; \forall i \in [1, n-1] \backslash S, \quad x_i = x_{i+1}, \; \forall i \in S.$$

This problem can be rewritten in terms of $m = n - |S|$ initial block common values as

$$\min_{y \in R^m} \sum_{i=1}^{m} w'_i (y_i - a'_i)^2 + \sum_{i=1}^{m-1} \mu'_i (y_i - y_{i+1})^2, \qquad \text{s.t.} \quad y_i \leq y_{i+1}, \; \forall i \in [1, m-1], \tag{31}$$

where the vectors a', w' and μ' are defined by the initial S. It is clearly an SMR problem. As it was shown above, the SPAV algorithm started with empty set of active constraints solves this problem, and consequently problem (3). It can be easily seen that the initial vector y produced in this case by the SPAV is the same as the $y(S)$ produced by the SPAV for the initial $S \subseteq S^*$. Then the subsequent values of y generated by the SPAV algorithm for problems (31) and (3) are identical. This finally proves that the SPAV algorithm returns the optimal solution to problem (3) for any initial $S \subseteq S^*$. ∎

It follows from the proof of Theorem 5 that if $\lambda(S) \geq 0$, then $S \subseteq S^*$. The converse of this statement is not true in general. This is shown by the following counterexample.

Example 1. Consider the three-dimensional problem, in which

$$a = (0, 30, -45)^T, \quad w = (0.5, 0.5, 0.5)^T, \text{ and } \mu = (0.5, 0.5)^T.$$

For the optimal solution, we have

$$S^* = \{1, 2\}, \quad x(S^*) = (-5, -5, -5)^T, \text{ and } \lambda(S^*) = (5, 40)^T.$$

In the case of $S = \{1\}$, the system of linear equations (7) gives $y_1 = 3$ and $y_2 = -21$. Substituting $x(S) = (3, 3, -21)^T$ and $\lambda_2(S) = 0$ into (9), we finally obtain $\lambda_1(S) = -3$. Thus, in this example, one of the components of $\lambda(S)$ is negative, whereas $S \subset S^*$.

As it was shown above, the SPAV algorithm generates at each iteration an active set S such that $x(S)$ and $\lambda(S)$ satisfy conditions (9), (11) and (12) of all the KKT conditions, but not (10). Since it aims for attaining the primal feasibility while maintaining the dual feasibility and complementary slackness, the SPAV can be viewed as a dual active-set algorithm, even though the Lagrange multipliers are not calculated. According to (30), the sequence of the generated active sets is such that the corresponding sequence $\lambda(S)$ is non-decreasing. Note that the same property is inherit in the primal-dual active-set (PDAS) algorithms developed in [9] and also in the version of the PDAS tailored in [14] for solving the MR problem.

4 Numerical experiments

In our experiments, the data sets were generated by formula (1). The following two response functions were used:

$$\chi_1(t) = t, \quad \text{and} \quad \chi_3(t) = t^3.$$

Our choice of functions, was motivated by the intention to study the case of a linear function, $\chi_1(t)$, and the case of a non-linear function which combines slow and rapid changes, $\chi_3(t)$. For these two cases, the observed values of explanatory variables t_i were uniformly distributed on the intervals $[0, 1]$ and $[-2, 2]$, respectively. In the both cases, the observation error ϵ_i was normally distributed with zero mean and standard deviation 0.3. The penalty parameters μ_i in the SMR problem were calculated by formula (4), where the value of μ was produced, for each data instance, by the cross-validation-based technique specially designed in [26] for the SMR problem. All components of the vector of weights w were ones.

The algorithms discussed in this section were implemented in R, version 3.2.3. For solving the tridiagonal system of linear equations (7), function Solve.tridiag of package limSolve was used. Function solve.QP of package quadprog was used as an alternative solver for the SMR problems to compare it with our SPAV algorithm. The numerical

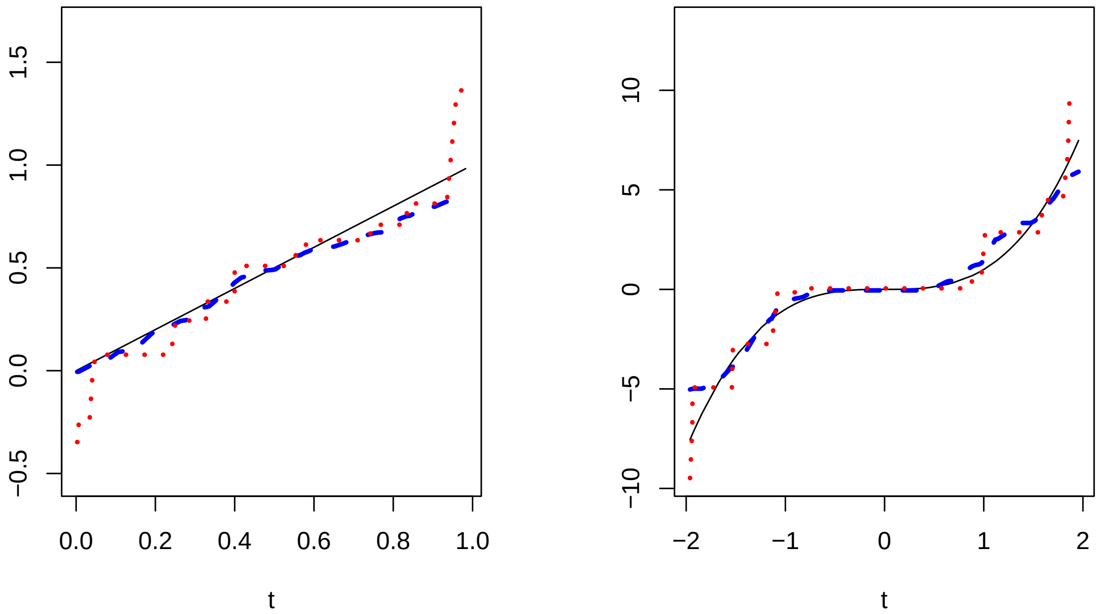

Figure 1: MR (dotted red) and SMR (dashed blue) solutions for the data sets of the size $n = 100$ generated for $\chi_1(t)$ (solid black, left) and $\chi_3(t)$ (solid black, right).

experiments were performed on a Windows PC with 16 GB RAM and Intel Xeon processor (3,30 GHz).

Figure 1 illustrates the ability of the SMR problem to smooth the solution to the MR problem. The values $\mu = 0.022$ and $\mu = 0.024$ were produced by the cross validation [26] for the two data sets, each of the size $n = 100$, that were generated for $\chi_1(t)$ and $\chi_3(t)$, respectively. The interpolation of the fitted values was performed by making use of a prediction model proposed in [26]. In what follows, we report results only for the linear response function, because the results for $\chi_3(t)$ were very similar. The value $\mu = 0.02$ was used.

The SMR serves not only for smoothing, but it also breaks blocks of the MR solution. Figure 2 shows how the number of blocks grows with μ starting from $\mu = 0$. Recall that the MR problem is a special case of the SMR problem which corresponds to the zero value of μ.

We compared the running time of the SPAV algorithm and function SOLVE.QP by studying the growth of each individual time with n. To minimize the impact of error in estimating the CPU time, 100 data instances were generated for each n as described above, and then the mean CPU time was calculated for each solver. The sequence of n was $100, 300, 500, \ldots, 5\,000$ followed by $6\,000, 7\,000, \ldots, 20\,000$. The SOLVE.QP failed to solve a fairly large number of the generated problems for numerical reasons related to the too small values of the denominator in (4). For instance, for $n = 100, 500$ and $7\,000$, it solved 95, 52 and 18 of 100 problems, respectively. It failed in all runs for $n \geq 8\,000$. The average CPU time (in seconds) of solving the generated SMR problems is plotted in Figure 3, where the average time of the SOLVE.QP is calculated excluding the failures. It shows that the too rapid increase of the running time of the conventional quadratic

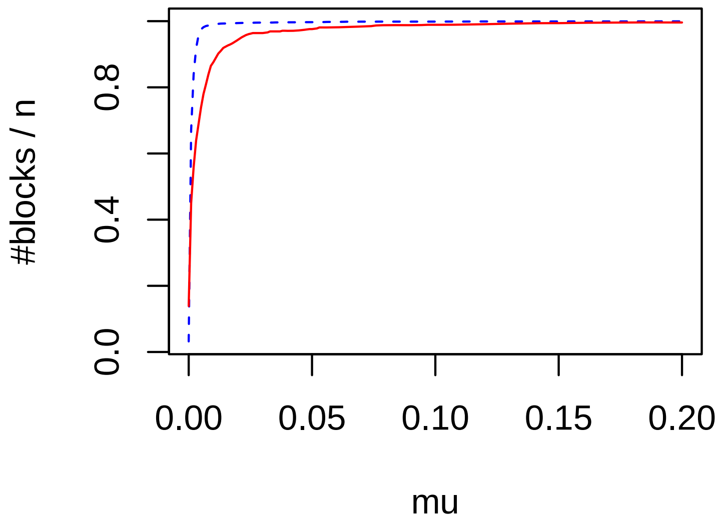

Figure 2: The number of blocks as a portion of n vs μ for $n = 100$ (solid red) and $n = 1\,000$ (dashed blue).

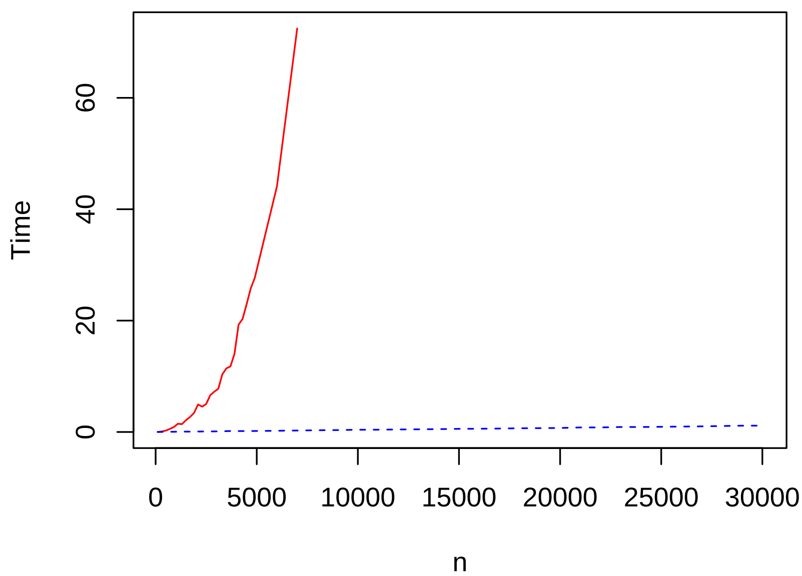

Figure 3: The CPU time of the SPAV algorithm and SOLVE.QP vs n.

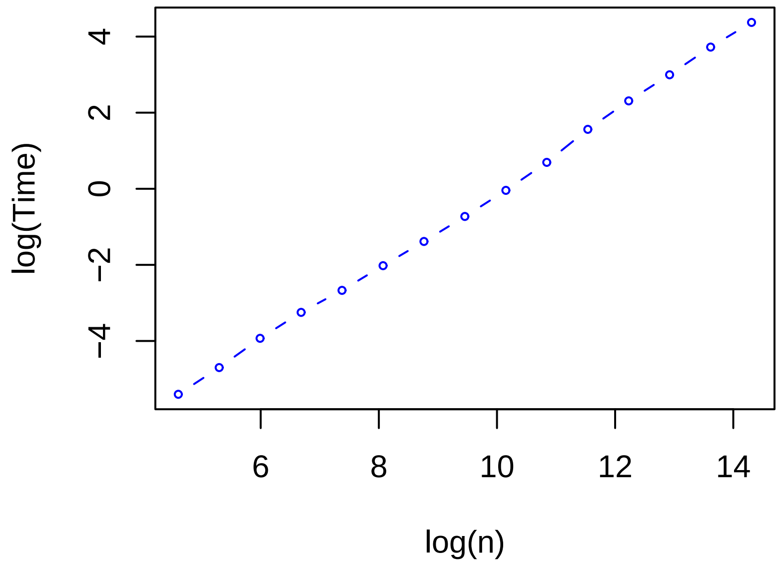

Figure 4: The CPU time of the SPAV algorithm vs n in the natural logarithmic scale.

optimization algorithm does not allow it to solve large-scale SMR problems, whereas the SPAV scales pretty well with the increasing data size.

Figure 4 represents for the SPAV the same relation as in Figure 3, but in the logarithmic scale and for $n = 100 \cdot 2^i$, where $i = 0, 1, 2, \ldots, 14$. The linear least square estimate of the slope of this graph suggests that the running time of the SPAV grows in our experiments in proportion to $n^{1.06}$, which is much slower compared to the growth in proportion to n^2 that follows from the worst-case analysis. For the response function $\chi_3(t)$, the fitted slope was 0.995, which is indicative, to within the experimental error, of an almost linear growth.

In the worst-case, the number of SPAV iterations equals n, while the observed number of iterations was far less than n. Moreover, the size of the linear tridialgonal system (7) to be solved at each iteration decreases in the process of solving SMR problem. It is essential that the most significant drop of its size usually occurs after the fist iteration. All this explains why the SPAV algorithm is so fast in practice. To study how the number of iterations changes with n, we generated 10 data instances for each n, and then calculated the mean number of iterations. Figure 5 shows that just a few iterations were typically required for the SPAV algorithm to solve the generated SMR problems. Observe that this number remains very small even for very large values of n. The maximal number of iterations over all 500 data instances was five, in which case n was 25 000. This counts in favor of the robustness of our algorithm. One can also see in Figure 5 that the ratio of the actual number of SPAV iterations to the worst-case number tends to zero as n increases.

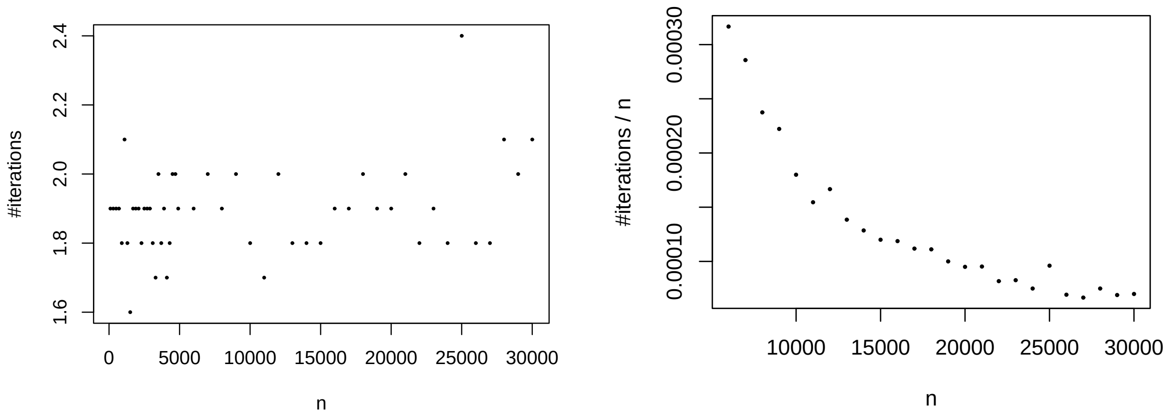

Figure 5: The number of SPAV iterations (left) and the same divided by n (right) vs n.

5 Conclusions

The SMR problem was designed for smoothing the solution to the MR problem, and it was statistically motivated in [26]. Here and in [26], we developed a fast dual active-set algorithm for solving the SMR. In the present paper, its finite-step convergence to the optimal solution has been proved. Our computational experiments has verified several important advantages of the SPAV algorithm, in particular, its scalability, which allows for regarding it as a practical algorithm for solving large-scale SMR problems. The efficiency of our algorithm originates from its ability to enlarge the active set by adding a large portion of constraints at once. Another advantage is that it admits a warm-starting.

Here and in [26], we focused on the SMR problem associated with a complete (linear) order of observations. Problem (3) admits a natural extension to the case of partial order. Indeed, let a partial order of n observations be defined by a set of pairs $E \subset [1, n] \times [1, n]$. Then the corresponding SMR problem can be formulated as follows:

$$\min_{x \in R^n} \sum_{i=1}^{n} w_i (x_i - a_i)^2 + \sum_{(i,j) \in E} \mu_{ij} (x_i - x_j)^2, \qquad \text{s.t.} \quad x_i \leq x_j, \ \forall (i, j) \in E. \tag{32}$$

From the computational point of view, this quadratic optimization problem is much more complicated than (3). In evidence of this, it is suffice to compare their simplified versions corresponding to $\mu = 0$. As it was mentioned earlier, the MR problem (2), can be solved in $O(n)$ arithmetic operations. However, the non-regularized version of (32) is much more computationally demanding, because the best known complexity of algorithms able to solve (32) for $\mu = 0$ is $O(n^2|E| + n^3 \log n)$ [18, 23, 24, 25]. Even its approximate solution requires $O(n^2)$ operations [4, 5, 6, 27]. Therefore, the development of efficient exact and approximate algorithms for solving problem (32) should be viewed as a challenging task for the future research.

References

[1] M. Ayer, H.D. Brunk, G.M. Ewing, W.T. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647, 1955.

[2] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical inference under order restrictions*. Wiley, New York, NY, 1972.

[3] Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

[4] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian. An $O(n^2)$ algorithm for isotonic regression problems. In G. Di Pillo and M. Roma, editors, *Nonconvex Optimization and Its Applications*, pages 25–33. Springer-Verlag, 2006.

[5] Oleg Burdakov, Anders Grimvall, and Mohamed Hussian. A generalised pav algorithm for monotonic regression in several variables. In *COMPSTAT, Proceedings of the 16th Symposium in Computational Statistics*, pages 761–767, 2004.

[6] Oleg Burdakov, Anders Grimvall, and Oleg Sysoev. Data preordering in generalized PAV algorithm for monotonic regression. *Journal of Computational Mathematics*, 24(6):771–790, 2006.

[7] Oleg Burdakov, Ivan Kapyrin, and Yuri Vassilevski. Monotonicity recovering and accuracy preserving optimization methods for postprocessing finite element solutions. *Journal of computational physics*, 231(8):3126–3142, 2012.

[8] Ramaswamy Chandrasekaran, Young U. Ryu, Varghese S. Jacob, and Sungchul Hong. Isotonic separation. *INFORMS Journal on Computing*, 17(4):462–474, 2005.

[9] Frank E. Curtis, Zheng Han, and Daniel P. Robinson. A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization. *Computational Optimization and Applications*, 60(2):311–341, 2015.

[10] Jan de Leeuw, Kurt Hornik, and Patrick Mair. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32(5):1–24, 2009.

[11] Arne B. Gjuvsland, Yunpeng Wang, Erik Plahte, and Stig W. Omholt. Monotonicity is a key feature of genotype-phenotype maps. *Frontiers in Genetics*, 2013. doi: 10.3389/fgene.2013.00216.

[12] Anne Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.

[13] Pedro Antonio Gutierrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. Ordinal regression methods: survey and experimental study. *Knowledge and Data Engineering, IEEE Transactions on*, 28(1):127–146, 2016.

[14] Zheng Han and Frank E. Curtis. Primal-dual active-set methods for isotonic regression and trend filtering. *arXiv preprint arXiv:1508.02452v2*, 2016.

[15] Mohamed Hussian, Anders Grimvall, Oleg Burdakov, and Oleg Sysoev. Monotonic regression for the detection of temporal trends in environmental quality data. *MATCH Commun. Math. Comput. Chem*, 54:535–550, 2005.

[16] M.L. Kalish, J.C. Dunn, O. Burdakov, and O. Sysoev. A statistical test of the equality of latent orders. *Journal of Mathematical Psychology*, 70:1–11, 2016.

[17] Joseph B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[18] W.L. Maxwell and J.A. Muckstadt. Establishing consistent and realistic reorder intervals in production-distribution systems. *Operations Research*, 33(6):1316–1341, 1985.

[19] R.E. Miles. The complete amalgamation into blocks, by weighted means, of a finite set of real numbers. *Biometrika*, 46(3/4):317–327, 1959.

[20] J. Nocedal and S. Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[21] T. Robertson, F.T. Wright, and R.L. Dykstra. *Order Restricted Statistical Inference.* Wiley, New York, NY, 1988.

[22] M. Roth, T.A. Buishand, and G. Jongbloed. Trends in moderate rainfall extremes: A regional monotone regression approach. *Journal of Climate*, 28(22):8760–8769, 2015.

[23] J. Spouge, H. Wan, and W.J. Wilbur. Least squares isotonic regression in two dimensions. *Journal of Optimization Theory and Applications*, 117(3):585–605, 2003.

[24] Quentin F. Stout. Isotonic regression via partitioning. *Algorithmica*, 66(1):93–112, 2013.

[25] Quentin F. Stout. Isotonic regression for multiple independent variables. *Algorithmica*, 71(2):450–470, 2015.

[26] O. Sysoev and O. Burdakov. A smoothed monotonic regression via l_2 regularization. Technical Report LiTH-MAT-R–2016/01–SE, Department of Mathematics, Linköping University, 2016. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-125398.

[27] O. Sysoev, O. Burdakov, and A. Grimvall. A segmentation-based algorithm for large-scale monotonic regression problems. *Journal of Computational Statistics and Data Analysis*, 55:2463–2476, 2011.

[28] Marina Velikova. *Monotone Prediction Models in Data Mining.* VDM Verlag, 2008.