Department of Mathematics

A smoothed monotonic regression via L2 regularization

Oleg Sysoev and Oleg Burdakov

LiTH-MAT-R--2016/01--SE

Department of Mathematics
Linköping University
S-581 83 Linköping

A smoothed monotonic regression via L2 regularization

Oleg Sysoev[1]
Oleg Burdakov[2]

LiTH-MAT-R–2016/01–SE

[1]Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
[2]Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden

A smoothed monotonic regression via L2 regularization

Oleg Sysoev[1] and Oleg Burdakov[2]

Abstract Monotonic Regression (MR) is a standard method for extracting a monotone function from non-monotonic data, and it is used in many applications. However, a known drawback of this method is that its fitted response is a piecewise constant function, while practical response functions are often required to be continuous. The method proposed in this paper achieves monotonicity and smoothness of the regression by introducing an L2 regularization term, and it is shown that the complexity of this method is $O(n^2)$. In addition, our simulations demonstrate that the proposed method normally has higher predictive power than some commonly used alternative methods, such as monotonic kernel smoothers. In contrast to these methods, our approach is probabilistically motivated and has connections to Bayesian modeling.
Keywords: Monotonic regression, Kernel smoothing, Penalized regression, Bayesian modeling.

1 Introduction

There is a variety of applications in which it can be believed that one or more predictors have a monotonic relation to the response variable, i.e. the response function is non-decreasing (or non-increasing) when one or more of these predictors increase. Such applications can be found in psychology, biology, signal processing, economics and many other disciplines [1, 14] and [2, 13]. Perhaps the most widely known approach which allows for extracting a monotonic dependence is Monotonic Regression (MR) [4, 24].

In this paper, we consider the case of a model with one predictor $x \in R$. The corresponding MR problem can be formulated as follows. Let $f(x)$ be an unknown monotonic true response function. Given n observations $(X_1, Y_1), \ldots, (X_n, Y_n)$, where $X_i \in R$ and $Y_i \in R$ are observed predictor and response values, respectively. The value of Y_i are random with mean $f(X_i)$. The predictor values are supposed to be sorted in the way that $X_1 < X_2 < \ldots < X_n$. It is required that the fitted response values inherit this monotonicity. They are obtained by solving in $\mu \in R^n$ the optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} (\mu_i - Y_i)^2 \\ \text{s.t.} \quad & \mu_i \le \mu_{i+1}, \ i = 1, \ldots, n-1, \end{aligned} \tag{1}$$

It can be easily shown that a solution to this problem is a maximum likelihood estimator of the parameters μ_i, where μ_i are assumed to be normally distributed and, at the same time, non-decreasing with respect to the predictor.

There exists a simple algorithm, called Pool-Adjacent-Violators (PAV), that solves problem (1) in $O(n)$ steps [3]. It is widely used due to its computational efficiency, and yet some practitioners are skeptic about using the PAV technique in their applications because the fitted response returned by the MR resembles a step function while the expected response in these applications is believed to be continuous and smooth. When there is more than one predictor involved, MR can also be applied, and there are computationally heavy exact algorithms for solving MR problem, such as [16], or computationally inexpensive approximate MR algorithms, such as [7, 25]. However, the fitted response still resembles a piecewise constant function even in the model with many predictors.

In order to handle the lack of smoothness of the MR, various techniques were developed for the model with one predictor. One group of methods ([18, 15, 10, 12]) combines the MR approach with kernel smoothers, and the fitted response appears to be monotonic and smooth. A particularly simple ad-hoc approach is discussed in [15], where a smooth monotonic estimator m_{IS} leans upon applying first

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

the monotonic (in [15], isotonic) regression and then a kernel smoother; in another estimator m_{SI}, the smoothing is followed by the monotonization. Since the complexity of the kernel smoothing is $O(n^2)$ [9] and the complexity of the MR computation is $O(n)$, the computational cost of m_{IS} or m_{SI} is $O(n^2)$. Other methods in this group seem to be more computationally heavy: they either are based on a numerical integration or apply quadratic programming.

Another group of methods that produce a smooth monotonic fit is based on smoothing spline techniques, see for example [23, 28] and also [17]. Therefore, their computational complexity is at least $O(n^3)$ if, like in the ordinary splines [9], the knots are located in the observations.

In the literature, there are also some amount of works based on Bayesian methodology (see for ex [20]) that involve Markov chain Monte Carlo sampling or other type of stochastic optimization, which makes these methods computationally heavy compared to the aforementioned frequentist alternatives.

The purpose of this work is to develop a method that a) is fast and well scalable, i.e. able to efficiently handle large data sets and can be efficiently applied in iterative algorithms like GAM or Gradient Boosting; b) is statistically motivated rather than ad-hoc; c) enables a user to select a desired degree of smoothness, as it is with splines or kernel smoothers and d) has a reasonably good predictive power.

For achieving these goals, we modify the MR formulation (1) by penalizing the difference between adjacent fitted response values. This is aimed at eliminating sharp 'jumps' of the response function. In this paper, we show that such a simple idea has a rigorous probabilistic background motivated by the Bayesian theory. Our Smoothed Monotonic Regression (SMR) problem is formulated as

$$\begin{array}{ll} \min & \sum_{i=1}^{n} (\mu_i - Y_i)^2 + \sum_{i=1}^{n-1} \lambda_i (\mu_{i+1} - \mu_i)^2 \\ \text{s.t.} & \mu_i \le \mu_{i+1},\ i = 1, \ldots, n-1, \end{array} \tag{2}$$

where $\lambda_i > 0$, $i = 1, \ldots, n-1$, are preselected penalty parameters whose choice is discussed in this paper.

The paper is organized as follows. Section 2 presents an algorithm that computes the optimal solution to problem (2) and contains estimates of the complexity of this algorithm. In section 3, a connection of the SMR problem to the Bayesian theory is demonstrated, an appropriate choice of the penalty parameters is discussed and a predictive model is presented. In particular, this section introduces a generalized cross validation for the model corresponding to (2). Section 4 considers a problem of inconsistency of the SMR on the boundaries and introduces a correction strategy. In section 5, results of numerical simulations and a comparative study of our SMR algorithm and some alternative algorithms are presented. Section 6 contains conclusions.

2 SPAV algorithm

On can see that the objective function in (2) is strictly convex, because the first term is a strictly convex function and the other one is convex. The constraints are linear. Therefore, the SMR problem (2) is a strictly convex quadratic programming (QP) problem. In principle, it can be solved by the conventional QP algorithms [21]. However, these algorithms do not make use of a special structure of the objective function or constraints in (2), whereas taking this structure into account, as one can see below, allows for substantially speeding up the solution process. It should also be noted that the general purpose QP algorithms are not of the polynomial complexity.

Owing to the strict convexity of the SMR problem, its solution is unique and it is determined by the Karush-Kuhn-Tucker (KKT) optimality conditions [21]. To derive them, consider the Lagrangian function

$$L(\mu, \rho) = \sum_{i=1}^{n} (\mu_i - Y_i)^2 + \sum_{i=1}^{n-1} \lambda_i (\mu_{i+1} - \mu_i)^2 - \sum_{i=1}^{n-1} \rho_i (\mu_{i+1} - \mu_i),$$

where $\rho \in R^{n-1}$ is the vector of non-negative Lagrange multiplies. Its first derivative is calculated by the formula

$$\nabla_{\mu_i} L(\mu, \rho) = \begin{cases} 2(\mu_1 - Y_1) + 2\lambda_1 (\mu_1 - \mu_2) + \rho_1, & i = 1, \\ 2(\mu_i - Y_i) + 2\lambda_{i-1} (\mu_i - \mu_{i-1}) + & \\ \qquad +2\lambda_i (\mu_i - \mu_{i+1}) - (\rho_{i-1} - \rho_i), & i = 2, \ldots, n-1, \\ 2(\mu_n - Y_n) + 2\lambda_{n-1} (\mu_n - \mu_{n-1}) - \rho_{n-1}, & i = n. \end{cases} \tag{3}$$

To represent the KKT conditions in a compact form, we shall use the following notations. Let

$$I(a) = \begin{cases} 1, & \text{if } a \text{ is true,} \\ 0, & \text{if } a \text{ is false} \end{cases}$$

stand for the indicator function. We denote by $A \in R^{n\times n}$ and $B \in R^{n\times(n-1)}$ matrices whose elements are defined as

$$A_{ij} = \begin{cases} 1 + \lambda_{i-1} I(i > 1) + \lambda_i I(i < n), & \text{if } j = i, \\ -\lambda_{i-1} I(i > 1), & \text{if } j = i - 1, \\ -\lambda_i I(i < n), & \text{if } j = i + 1, \\ 0, & \text{otherwise} \end{cases}$$

and

$$B_{ij} = \begin{cases} -I(i > 1), & j = i - 1, \\ I(i < n), & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Note that A is a tridiagonal matrix. Denote also $Y = (Y_1, \ldots, Y_n)^T$.

Using these notations and the formula for $\nabla_\mu L(\mu, \rho)$, we can represent the KKT conditions as

$$A \cdot \mu = Y + \frac{1}{2} B \cdot \rho, \tag{4}$$

$$\mu_i \le \mu_{i+1}, \quad i = 1, \ldots, n-1, \tag{5}$$

$$\rho_i \ge 0, \quad i = 1, \ldots, n-1, \tag{6}$$

$$\rho_i(\mu_{i+1} - \mu_i) = 0, \quad i = 1, \ldots, n-1. \tag{7}$$

The algorithm that we introduce here for solving the SMR problem belongs to the class of dual active set algorithms [5, 11]. Its correctness is theoretically justified in the accompanying paper [6] which is focused on optimization aspects of our approach. At each iteration, our algorithm solves the problem

$$\begin{array}{ll} \min & \sum_{i=1}^{n} (\mu_i - Y_i)^2 + \sum_{i=1}^{n-1} \lambda_i (\mu_{i+1} - \mu_i)^2 \\ \text{s.t.} & \mu_i = \mu_{i+1}, \ i \in S, \end{array} \tag{8}$$

for a set of active constraints $S \subseteq \{1, 2, \ldots, n-1\}$. The optimal solution to (8) may violate some of those constraints in (5) which are not in the active set. The algorithm starts with $S = \emptyset$, which is enlarging from iteration to iteration, and it terminates when the vector μ that solves this problem becomes feasible in the original problem (2).

The Lagrangian function is defined for problem (8) as

$$L(\mu, \rho') = \sum_{i=1}^{n} (\mu_i - Y_i)^2 + \sum_{i=1}^{n-1} \lambda_i (\mu_{i+1} - \mu_i)^2 - \sum_{i \in S} \rho'_i (\mu_{i+1} - \mu_i).$$

As shown in [6], the algorithm ensures that, at each iteration, the solution to problem (8) is such that the corresponding Lagrange multiplies ρ'_i, $i \in S$, are non-negative. This means that this optimal solution would not change if the equality constraints were substituted in (8) by the inequalities

$$\mu_i \le \mu_{i+1}, \ i \in S. \tag{9}$$

This also means that the solution to problem (8) along with

$$\rho_i = \begin{cases} \rho'_i, & \text{if } i \in S, \\ 0, & \text{otherwise} \end{cases}$$

satisfy (4), (6) and (7), which is a part of the KKT conditions in the original problem (2). The rest of the conditions, namely (5), are guaranteed to be satisfied for the active constraints as equalities. However, for some $i \notin S$, the conditions (5) may be violated. It should be emphasized that the optimal solution to problem (8) is obtained as a unique solution to the system of linear equations (4), in which all ρ_i, $i \notin S$ are set equal zero.

A proof of the correctness of our algorithm is based in [6] on the following general statement. Let S be an active set such that the Lagrange multipliers ρ'_i, $i \in S$, obtained in (8) are all non-negative. Then the constraints (9) are among those active in the optimal solution to problem (2). Moreover, if the optimal solution to problem (8) is such that $\mu_i \ge \mu_{i+1}$ for some $i \notin S$, then this constraint also belongs to the optimal active set in (2). In accordance with this, the algorithm makes use of the solution μ to problem (8) for adding new constraints to the active set by the formula

$$S \leftarrow S \cup \{i \notin S : \ \mu_i \ge \mu_{i+1}\}. \tag{10}$$

This implies that, at each iteration, the active set S remains a subset of the optimal active set in (2), and also that the algorithm finds the optimal solution to (2) in at most n iterations.

We will show now how to efficiently solve problem (8). Let $[i, j]$ denote a segment of indexes $\{i, i+1, \ldots, j-1, j\}$. Suppose that $[p, q-1] \subseteq S$. Then these adjacent active constraints imply $\mu_p = \mu_{p+1} = \ldots = \mu_q$. If, in addition, $p-1 \notin S$ and $q \notin S$, then we call $[p, q]$ a *block*. A singleton set $\{p\} \subseteq [1, n]$ is also called a block if $p-1 \notin S$ and $p \notin S$. Thus, any active set S generates a partitioning of $[1, n]$ into a set of blocks $\Omega = \{B_1, B_2, \ldots, B_{n_B}\}$, where n_B is the number of blocks. Each block $B_i = [p_i, q_i]$ is characterized by its number of elements $n_i = q_i - p_i + 1$, *observed mean*

$$\bar{Y}_i = \frac{1}{n_i} \sum_{j=p_i}^{q_i} Y_j$$

and a *common block value* $\hat{\mu}_i$ such that

$$\mu_j = \hat{\mu}_i, \quad \forall j \in B_i. \tag{11}$$

Denote $\bar{Y} = (\bar{Y}_1, \ldots, \bar{Y}_{n_B})^T$ and $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n_B})^T$. The blocks are supposed to be sorted in increasing order of the indexes they contain, which means that $p_i - 1 = q_{i-1}$. We denote by $\hat{A}$ a $n_B \times n_B$ tridiagonal matrix whose elements are defined as

$$\hat{A}_{ij} = \begin{cases} 1 + \frac{\lambda_{p_i-1}}{n_i} I(p_i > 1) + \frac{\lambda_{q_i}}{n_i} I(q_i < n), & \text{if } j = i, \\ -\frac{\lambda_{p_i-1}}{n_i} I(p_i > 1), & \text{if } j = i-1, \\ -\frac{\lambda_{q_i}}{n_i} I(q_i < n), & \text{if } j = i+1, \\ 0, & \text{otherwise.} \end{cases}$$

The next result shows how to easily compute the solution to problem (8).

Theorem 1 *The common block values of the optimal solution to problem (8) are uniquelly defined by the system of linear equations*

$$\hat{A} \cdot \hat{\mu} = \bar{Y}. \tag{12}$$

Proof. As it was mentioned above, the solution to problem (8) is uniquely defined by the system of linear equations (4), where (11) holds and $\rho_i = 0$ for all $i \notin S$. Consider a block $B_i = [p_i, q_i]$. The corresponding equations in (4) are of the form:

$$\begin{cases} \hat{\mu}_i + \lambda_{j-1}(\hat{\mu}_i - \hat{\mu}_{i-1}) \cdot I(j > 1) = Y_j - \frac{\rho_j}{2} I(j < n) & \text{for } j = p_i \\ \hat{\mu}_i = Y_j + \frac{\rho_{j-1}}{2} I(j > 1) - \frac{\rho_j}{2} I(j < n) & \text{for } j = p_i + 1 \\ \ldots \\ \hat{\mu}_i + \lambda_j(\hat{\mu}_i - \hat{\mu}_{i+1}) \cdot I(j < n) = Y_j + \frac{\rho_{j-1}}{2} I(j > 1) & \text{for } j = q_i \end{cases}$$

By summing these equations, we obtain

$$n_i \hat{\mu}_i + \lambda_{p_i-1}(\hat{\mu}_i - \hat{\mu}_{i-1}) \cdot I(p_i > 1) + \lambda_{q_i}(\hat{\mu}_i - \hat{\mu}_{i+1}) \cdot I(q_i < n) = \sum_{j=p_i}^{q_i} Y_j$$

A simple algebraic manipulation of this equation results in equation (12). ∎

The important feature of system (12) is that the matrix $\hat{A}$ is tridiagonal. It can be efficiently solved, e.g., by the Thomas algorithm [8] of complexity $O(n)$. Moreover, since $\hat{A}$ does not depend on Y, (12) can be viewed as a linear smoother because it is aimed at making the values $\hat{\mu}$ more smooth then the observed response Y.

We call the outlined algorithm for solving the SMR problem (2) a *Smoothed Pool Adjacent Violators* (SPAV) algorithm. It employs Theorem 1.

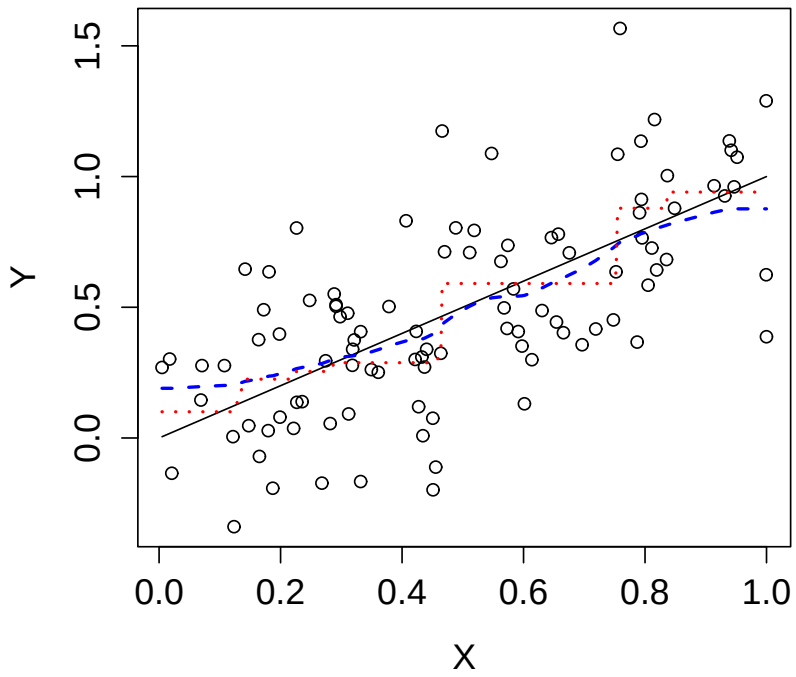

Figure 1: The solid line is $f(x) = x$; the circles are artificially generated data $Y = f(X)+\epsilon$, where $\epsilon \sim N(0, 0.3)$; the dashed line is the SPAV solution (with linear kernel and λ selected by cross-validation, introduced below); the dotted line is the PAV solution.

Algorithm 1 *SPAV algorithm*

Given $Y = (Y_1, ...Y_n)$ *and* $\lambda = (\lambda_1, \ldots, \lambda_n)$.
Set $S = \emptyset$, $n_B = n$, $B_j = \{j\}$ $(\forall j \in [1, n])$ *and* $\Omega = \{B_1, \ldots, B_n\}$.
Smoothing step
1. *Compute* $\hat{\mu}$ *and* μ *according to (12) and (11), respectively.*
2. *If* $\mu_i \leq \mu_{i+1}$ *for all* $i = 1, \ldots, n-1$ *then stop and return* μ *as the optimal solution, else*

PAV step
3. *Enlarge* S *by formula (10), and update* Ω *and* n_B *accordingly.*
4. *Go to step 1.*

Step 3 is called PAV step because it pools adjacent violators, i.e. indexes i and $i+1$ satisfying $\mu_i > \mu_{i+1}$ as well as the two corresponding blocks, in order to compose one block. The result of the SPAV algorithm and the traditional PAV algorithm are compared in Figure 1. The step-wise response produced by PAV algorithm can sometimes be unrealistic and deviate quite a lot from the true model. One can see this in Figure 1, where the SPAV response looks much more natural and closer to the true model.

The following theorem provides an estimate of the complexity of the SPAV algorithm.

Theorem 2 *The complexity of the SPAV algorithm is* $O(n^2)$

Proof. Smoothing step followed by PAV step can be viewed as one iteration. As it was mentioned above, the number of iterations does not exceed n. The complexity of step 1 is determined by the complexity of composing and solving a tridiagonal system of equations, which is $O(n)$. Step 2 involves

$O(n)$ comparison operations. Step 3 involves a search that requires $O(n)$ operations plus adding an element to a list and updating the list of blocks which requires $O(n)$ steps. Therefore, the total complexity of one iteration is $O(n)$, which means that the total complexity of the algorithm is $O(n^2)$. □

Note that the quadratic complexity is a worst-case estimate, which assumes that the number of iterations is n. In practice, we observed that this number was fewer than n, and the CPU time of SPAV algorithm was growing with n much slower than as n^2.

3 Connections to Bayesian theory

It is a known fact that penalization of the objective can often be thought of putting a regularization prior on the model, as it is for instance with Ridge regression and LASSO [19]. Consider now how the problem (2) can be represented from the Bayesian perspective. Assume the following model on the means:

$$Y_i \sim N(\mu_i, \alpha^2), \tag{13}$$

where the prior on the components of μ are assumed to have the following form:

$$\mu_{i+1} \sim \mu_i + N_+(0, \sigma_i^2),\ i = 1, \ldots, n-1, \tag{14}$$

$$p(\mu_1) \propto 1. \tag{15}$$

Here N_+ is the truncated normal distribution that ensures that the sequence of μ_i's is not decreasing and σ_i^2 are constants that can be chosen in some way. Since we assume that the expected response should be a continuous function, it is natural to let σ_i^2 depend on how close X_{i+1} is to X_i: the smaller the distance, the lesser the variance should be; and when $X_{i+1} = X_i$, σ_i^2 should turn into zero.

We introduce a family of functions K, called here *kernel* (by analogy to the Gaussian Process model), which meet the requirements mentioned above and are defined by

$$\sigma_i^2 = \frac{1}{\beta K\left(X_i, X_{i+1}\right)}, \tag{16}$$

where, for instance,

$$K(X, X') = \frac{1}{|X - X'|^p}. \tag{17}$$

Here β is some constant that regulates the level of smoothness: the higher the β, the lesser the variance is and the higher is the degree of smoothness; p is a constant that also regulates the flexibility of the model: higher p allows for larger variation between the nearest points. We shall call the function K with $p = 1$ a *linear kernel*, and $p = 2$ a *quadratic kernel*. In principle, one may select some expression for the kernel $K(X, X')$ other than (17); the important is that this function should be non-decreasing and $1/K(X, X')$ should turn into zero for $X = X'$. Given our choice of the model, the following result holds.

Theorem 3 *The solution to problem (2), where*

$$\lambda_i = \frac{\alpha^2}{\sigma_i^2}, \quad i = 1, \ldots\ n-1, \tag{18}$$

is the maximum aposteriori estimate provided by the model (13)-(15).

Proof. According to Bayes' theorem, the posterior distribution

$$p(\mu|Y, X) \propto p(Y|\mu, X) \cdot p(\mu|X).$$

The likelihood is the product of normal distributions

$$p(Y|\mu, X) \propto exp\left(-\frac{\sum_{i=1}^n (Y_i - \mu_i)^2}{2\rho^2}\right).$$

The prior $p(\mu|X)$ can be computed via the chain rule:

$$p(\mu|X) = p(\mu_1|X) \prod_{i=1}^{n-1} p(\mu_{i+1}|\mu_i, X),$$

Assuming that $\mu_i \leq \mu_{i+1}$ for all $i = 1, \ldots, n-1$, we have

$$p(\mu|X) \propto exp\left(-\frac{\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2}{2\sigma_i^2}\right) = exp\left(-1/(2\alpha^2)\sum_{i=1}^{n-1}\lambda_i(\mu_{i+1}-\mu_i)^2\right),$$

where λ_i is defined by (18).

By computing the product of the likelihood and the prior, and then taking the logarithm, we get

$$log\ p(\mu|Y,X) \propto -\sum_{i=1}^{n}(Y_i-\mu_i)^2 - \sum_{i=1}^{n-1}\lambda_i(\mu_{i+1}-\mu_i)^2,$$

which implies that the maximum of $p(\mu|Y,X)$ is assured by the solution to problem (2). □

Usung this theorem and assuming that the variance is given by (16), we obtain

$$\lambda_i = \frac{\alpha^2}{\sigma_i^2} = \lambda K(X_i, X_{i+1}), \tag{19}$$

where $\lambda = \alpha^2 \cdot \beta$. Given that the noise variance α^2 is fixed, the smoothing parameter λ plays the same role as β: the higher λ is, the greater is the degree of smoothness of the fitted SPAV response.

For the kernel defined by (17) with $p = 2$, formula (19) indicates that the penalty terms in (2) of the form

$$\lambda_i(\mu_{i+1}-\mu_i)^2 = \lambda\left(\frac{\mu_{i+1}-\mu_i}{X_{i+1}-X_i}\right)^2$$

perform smoothing by preventing from too large values of the finite-difference approximation of the first derivative of the response function.

Our analysis enables us to conclude that, using the Bayesian approach, it is possible to select an appropriate structure of λ_i in such a way that it reflects the location of the X components. Without using Bayesian modeling, this structure would perhaps be difficult to determine. Another advantage of using Bayesian methodology is that it provides a strategy for determining appropriate predictions for some new values of X, as it is discussed below.

Prediction for monotonic regression is a special problem, as it is discussed in [26]. In general, a way of making prediction is not unique. However, for the smoothed MR, we can define an appropriate prediction model for some new predictor value X^* as follows.

Assume the following model for the expected response μ^* that corresponds to predictor value X^* which is located between X_i and X_{i+1}:

$$\mu^*|\mu_i \sim \mu_i + N_+(0, \sigma_{i,*}^2), \tag{20}$$

$$\mu_{i+1}|\mu^* \sim \mu^* + N_+(0, \sigma_{*,i+1}^2), \tag{21}$$

where the σ values are defined by (16). If x^* is outside the observed X interval, i.e. $X^* > X_n$ or $X^* < X_1$, then we simply define the predicted value for μ^* as μ_n or μ_1, respectively.

The prediction model presented by (20) and (21) is a natural extension of the model given by (14) because the two models have the same structure. Accordingly, we can derive the prediction in the same way as it was done for (14), i.e. by computing a maximum aposteriori estimate of $\mu^*|\mu_i, \mu_{i+1}$.

Theorem 4 *Given the model (20), (21), the maximum of $p(\mu^*|\mu_i, \mu_{i+1})$ is attained for*

$$\mu^* = \frac{K(X_i, X^*)\mu_i + K(X^*, X_{i+1})\mu_{i+1}}{K(X_i, X^*) + K(X^*, X_{i+1})}. \tag{22}$$

Proof. By Bayes' formula and the chain rule, we get:

$$p(\mu^*|\mu_i, \mu_{i+1}) = \frac{p(\mu^*, \mu_i, \mu_{i+1})}{p(\mu_i, \mu_{i+1})} = \frac{p(\mu_i)}{p(\mu_i, \mu_{i+1})} p(\mu^*|\mu_i) p(\mu_{i+1}|\mu^*).$$

Therefore,

$$logp(\mu^*|\mu_i, \mu_{i+1}) \propto -\frac{(\mu^*-\mu_i)^2}{2\sigma_{i,*}^2} - \frac{(\mu_{i+1}-\mu^*)^2}{2\sigma_{*,i+1}^2}.$$

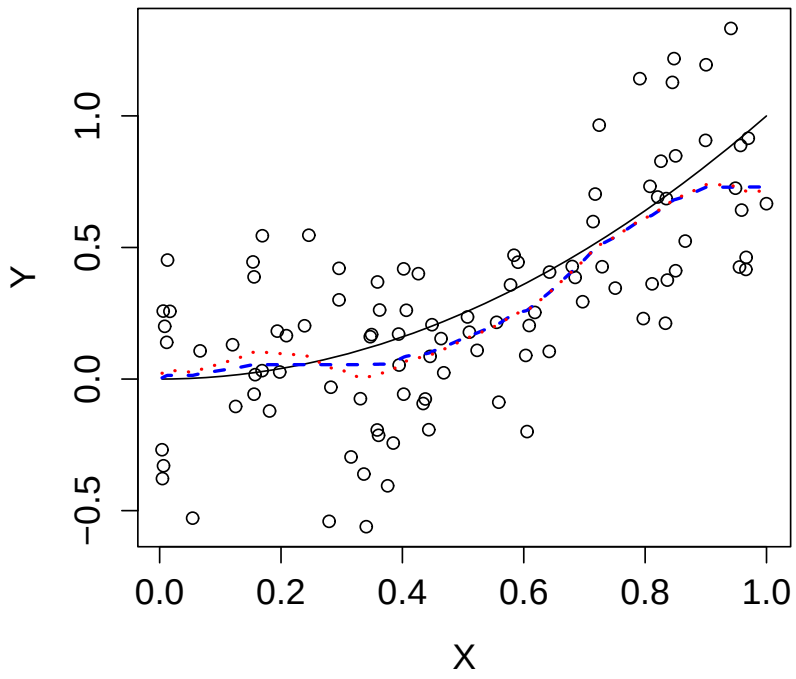

Figure 2: An artificially generated data $Y = f(X) + \epsilon, \epsilon \sim N(0, 0.3)$, where the solid line is $f(x) = x^2$, the dashed line is the response obtained by the SPAV algorithm (cross-validated, linear kernel) and Smoothing algorithm with the same λ as in the SPAV algorithm (dotted line).

By maximizing the expression above and using (16), we get

$$2\beta K(X_i, X^*)(\mu^* - \mu_i) + 2\beta K(X^*, X_{i+1})(\mu^* - \mu_{i+1}) = 0,$$

which leads to (22). □

A huge advantage of the structure of λ_i presented by (19) is that instead of n unknown parameters in (2), we need to estimate only one parameter λ. A possible way of doing this is cross-validation. However, the standard cross-validation implies repeated application of the SMR algorithm and, depending on the number of λ values selected for comparison, this may become computationally expensive.

We suggest a procedure called here *generalized cross-validation*. It is based on the following idea supported by empirical observations. The expected (true) response should normally be increasing, and there is a little chance of observing a region where the response is constant. Since in our model the response is constant for the observations that were merged into one block by the PAV step of the SPAV algorithm, we can conclude that in an ideal model there should be very few executions of the PAV step. Accordingly, if we remove the PAV steps from this algorithm, it should not affect much this ideal model, although the monotonicity may be violated. The algorithm can be presented as follows.

Algorithm 2 *Smoothing algorithm*

Given $Y = (Y_1, ...Y_n)$ *and* $\lambda = (\lambda_1, \ldots, \lambda_n)$.
Solve (4).

The fitted responses produced by the SPAV and Smoothing algorithms are presented in Figure 2, which demonstrates that these responses are quite similar.

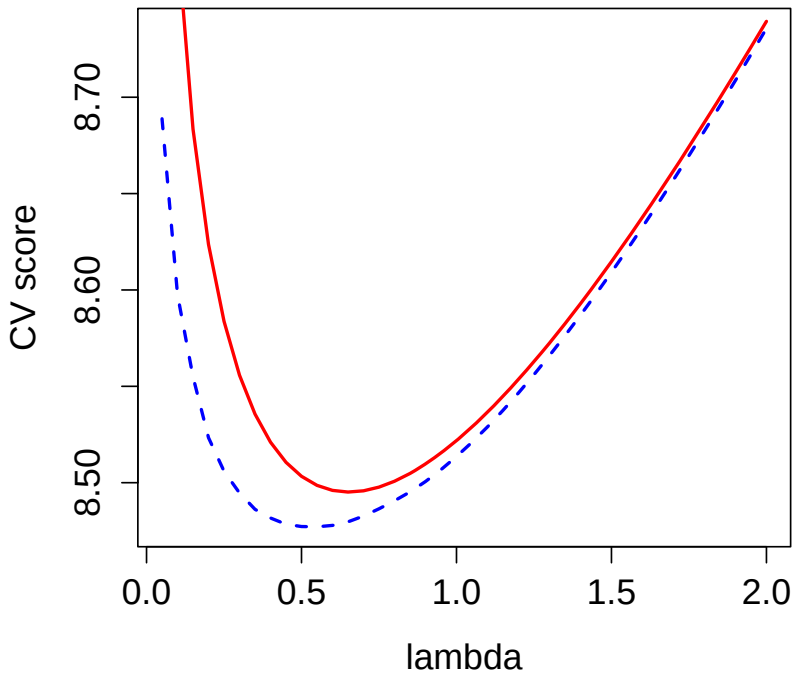

Figure 3: Cross-validation curve (solid) and the generalized cross-validation scores (dashed) for some artificially generated data, $n = 100$

We define the generalized cross-validation as a standard cross-validation in which the algorithm 2 is applied instead of algorithm 1 in order to determine an optimal penalty λ. This strategy allows for substantially reducing the computational time. Our numerical experiments confirm that the local optimum of the cross-validation curve and the generalized cross-validation curves are typically located close to each other, as it is demonstrated by Figure 3.

4 Correction on the boundaries

The MR is known to be inconsistent on the boundaries, which means that the MR estimates for μ_1 and μ_n do not converge to the true values when n tends to infinity, and some solutions to this problem were suggested [27, 22]. This problem is also known as spiking problem. An intuitive explanation of this issue is the asymmetry of the PAV algorithm on the boundary: the observation with index n can only be merged into a block with the observations $i < n$, while the observation with index j located in the middle of the domain of X can be merged into block with neighbors $i > j$ and $i < j$. In special cases, it may happen that $Y_n > Y_i$ for all $i < n$. In this case, the PAV algorithm does not make any correction of the error and returns $\hat{\mu}_n = Y_n$.

A similar kind of asymmetry can be observed in the SMR problem: the objective function in (2) contains two penalty factors (involving $i + 1$ and $i - 1$) for all $i = 2, \ldots, n - 1$, while for $i = n$ or $i = 1$ there is only one penalty factor. Figure 1 demonstrates that the SPAV fitted response can sometimes behave badly on the boundaries, i.e. it can be shifted towards the observed values of Y_1 and Y_n.

In order to resolve this issue, we apply the idea presented in [27]. More specifically, we add two extra terms aimed at penalizing too high values of μ_n and too low values of μ_1:

$$\min \quad \sum_{i=1}^{n} (\mu_i - Y_i)^2 + \sum_{i=1}^{n-1} \lambda_i (\mu_{i+1} - \mu_i)^2 + \phi\mu_n - \phi\mu_1 \tag{23}$$

$$\text{s.t.} \qquad \mu_i \le \mu_{i+1},\ i = 1, \ldots, n-1,$$

where $\phi > 0$ is a penalty factor.

It can be easily verified that

$$(\mu_1 - Y_1)^2 + (\mu_n - Y_n)^2 + \phi\mu_n - \phi\mu_1 = (\mu_1 - Y_1 - \frac{\phi}{2})^2 + (\mu_n - Y_n + \frac{\phi}{2})^2 + c,$$

where c does not depend on μ_1 or μ_n. Like in [27], this observation allows for reducing problem (23) to (2) by shifting the first and the last observed response values as follows:

$$Y_1 \leftarrow Y_1 + \frac{\phi}{2} \quad \text{and} \quad Y_n \leftarrow Y_n - \frac{\phi}{2}.$$

Then the problem (2) modified in this way can be solved by means of the SPAV algorithm. The only requirement is to use the values $Y' = Y + \phi \cdot e$ instead of Y, where $e = (1/2, 0, \ldots, 0, -1/2)^T$.

Consider now how to choose parameter ϕ. One possibility is to use the cross-validation, as it was suggested for λ. However, selecting an optimal combination of two parameters might be computationally expensive, and we describe an alternative approach that appeared to work well in practice.

The SPAV algorithm solves problem (12) repeatedly, and if Y is replaced by Y', the equation becomes

$$\hat{A} \cdot \hat{\mu} = \bar{Y}' = \bar{Y} + \phi\bar{e} \tag{24}$$

or

$$\hat{\mu} = \hat{A}^{-1}\bar{Y} + \phi\hat{A}^{-1}\bar{e} \tag{25}$$

where $\bar{e} = \left(\frac{1}{2n_1}, 0, \ldots, 0, \frac{-1}{2n_m}\right)$ and n_1 and n_m are sizes of block 1 and $m = n_B$, respectively. We suggest to select ϕ in such a way that $\hat{\mu} = \hat{\mu}(\phi)$ approximates the observed data $\bar{Y}$ in the best way in the least-squares sense:

$$\min_{\phi}(\hat{\mu}(\phi) - \bar{Y})^T(\hat{\mu}(\phi) - \bar{Y}).$$

This yields the optimal estimate

$$\hat{\phi} = \frac{(\bar{Y} - \mu')^T e'}{\|e'\|^2},$$

where $\mu' = \hat{A}^{-1}\bar{Y}$ and $e' = \hat{A}^{-1}\bar{e}$. Therefore, the fitted value $\hat{\mu}$ at each Smoothing Step of the SPAV algorithm can be found as follows:

Algorithm 3 *Step 1 of SPAV with correction*

1. *Solve* $\hat{A}\mu' = \bar{Y}$
2. *Solve* $\hat{A}e' = \bar{e}$
3. *Compute* $\hat{\phi} = \frac{(\bar{Y} - \mu')^T e'}{\|e'\|^2}$
4. *Compute* $\hat{\mu} = \mu' + \hat{\phi} \cdot e'$

Note that the linear equations in Algorithm 3 involve tridiagonal matrices only, and the rest of the algorithm contains simple vector operations. Therefore, the complexity of computing the correction is $O(n)$, and the overall $O(n^2)$ complexity of the SPAV algorithm is not affected by introducing the correction.

5 Numerical results

In order to evaluate the efficiency of the proposed SPAV approach, we compared it to the monotonic smoothers m_{IS} and m_{SI} described in [15] and also to the PAV solution m_{PAV}. The m_{IS} and m_{SI} are considered because they are clear competitors from the perspective of the computational complexity (in contrast to other known monotonic smoothers). The PAV solution was chosen mostly to demonstrate that there is a clear domination of the smoothed monotonic regression over ordinary monotonic regression in terms of the predictive power, and also to investigate how much the predictive power is improved.

Algorithm	m_4	m_1	m_2	m_3	m_{IS}	m_{SI}	m_{PAV}
Mean rank	1.95	3.02	3.80	3.84	3.91	4.70	6.76

Table 1: Mean rank of the PAV, SPAV, SI and IS algorithms

Our simulation studies involve the data generated as $X \sim Unif[0,1]$, $Y = f(X)+\epsilon$, $\epsilon \sim N(0,v)$ for all combinations of the following settings: $n = 100, 1000$ or 10000; $v = 0.03, 0.1$ or 0.3, the functional shape f is one of the following functions $f_1(X) = X$, $f_2(X) = X^2$, $f_3(X) = X + sin(10X)/10$, $f_4(X) = tanh(3X)$. These functions were chosen in order to represent various possible behaviors of a curve such as linearity, non-linearity, wiggly parts or a sudden change of the level. To measure the uncertainty of our results, we generate 100 instances of the same model.

Each data set generated in our simulations was processed by:

- monotonic smoothers m_{IS} and m_{SI},
- monotonic regression m_{PAV},
- m_1: SPAV algorithm with linear kernel without correction where λ was selected by the generalized cross-validation with 10 folds,
- m_2: SPAV algorithm with linear kernel without correction where λ was selected by the cross-validation with 10 folds,
- m_3: SPAV algorithm with quadratic kernel without correction where λ was selected by the generalized cross-validation with 10 folds,
- m_4: SPAV algorithm with linear kernel with correction where λ was selected by the generalized cross-validation with 10 folds.

The mean squared error (which is a measure of the predictive performance)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(m_*(X_i) - f(X_i)\right)^2$$

is computed, where m_* is a fitted response computed by some method.

For making an overall evaluation of the performance of the algorithms, we compared the MSE values produced by different algorithms given a generated data set. Respectively, each algorithm obtained a rank value from 1 (best) to 7 (worst) according to its predictive performance. The rank values were then averaged out over all generated data sets. The mean rank for each algorithm is presented in Table 1. It indicates that the SPAV with the linear kernel provides better predictions than m_{IS} or m_{SI}. Moreover, the correction improves the prediction accuracy. The model with a quadratic kernel has somewhat lower rank than the one with linear kernel for our data models, but the difference is not dramatic. Our simulations also confirm a previously established fact that m_{IS} has typically a higher predictive power than m_{SI} [15].

To make a more careful comparative analysis of our best algorithm and the existing methods, we have computed the average MSE values for each model given by a combination of N, v and f and the standard deviation per model. It appears that the standard deviation of the MSE values is at most 1% for each model, which means that the MSE estimates are sufficiently reliable. We have then sorted the models first by n, then by f and at last by v, and this order was saved as a model index. The average MSE values of algorithm m_4 versus the model index is presented in Figure 5, and the deviation of the average MSE values of a given method to the model m_4 is illustrated in Figure 4. The first four peaks in both figures correspond to $n = 100$, the following four peaks correspond to $n = 1000$, and the final sequence of peaks corresponds to $n = 10000$. We have also put a confidence band for m_4 (black areas) computed as $\pm$ two standard deviations for the MSE of m_4.

It can be concluded that m_4 gives the lowest MSE values for almost all models we studied, with few exceptions such as when either m_{SI} dominates or the difference between the MSE values of m_{IS} and m_4 is not significantly greater than zero. Another observation is that the average MSE values for all algorithms decrease with n, as it is expected. Comparing in Figure 5 the peaks corresponding to the same n, one can see that there is a difference in MSE values between various functions f, and nonlinear functions are

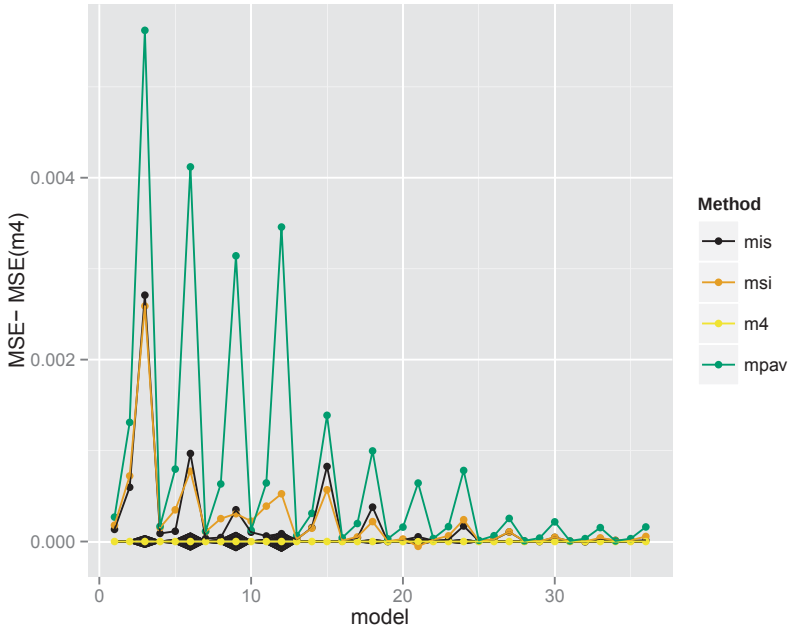

Figure 4: The average difference between MSE of a method and MSE of m_4 for various models. The methods considered are m_4, m_{IS}, m_{SI} and m_{PAV}, and the models are sorted consecutively by n, f and v.

generally harder to fit. By investigating the behavior of the function within each peak in Figures 4 and 5, we can conclude that MSE is an increasing function of the variance for the given n and f. This also looks reasonable since it is harder to fit models with higher error variance. Finally, we observe that the m_{PAV} has largest MSE values for all models, and the reason for this is the lack of smoothness of the PAV fitted models.

6 Conclusions

In this work, we have introduced an approach for a smoothed monotonic regression in one predictor variable. It allows for adjusting the degree of smoothness to the data. We have shown that the computational complexity of this method is $O(n^2)$ which makes the method suitable in large-scale settings. This complexity reflects the worst-case estimate of the computational burden. In practice, it is growing with n far less rapidly. It was demonstrated by our numerical simulations that the SMR method has a better predictive performance than some other widely known computationally cheap approaches that produce a monotone response, such as monotonic regression and monotonic kernel smoothers.

The SMR problem was presented as a regularization problem with n regularization parameters. Then a connection was drawn to a Bayesian model that gave the same MAP estimate as the solution to the regularization problem. The Bayesian formulation provided us a clear strategy for reducing n regularization parameters to just one smoothing parameter by setting the structure of the prior variance. The prior variance depends on the kernel structure, and the predictive performance was similar when linear and quadratic kernels were used. In addition, the Bayesian formulation provided us a natural predictive model for the input values which are not present in the training data. Further, we have motivated a generalized cross-validation strategy for the selection of the smoothing parameter, and our preliminary experiments have confirmed that the cross-validation produces similar results but the generalized version is much more computationally efficient. Finally, a boundary effect problem was studied, and a strategy for the inconsistency correction was suggested. The proposed strategy had a clear positive effect on the predictive performance of the SMR model.

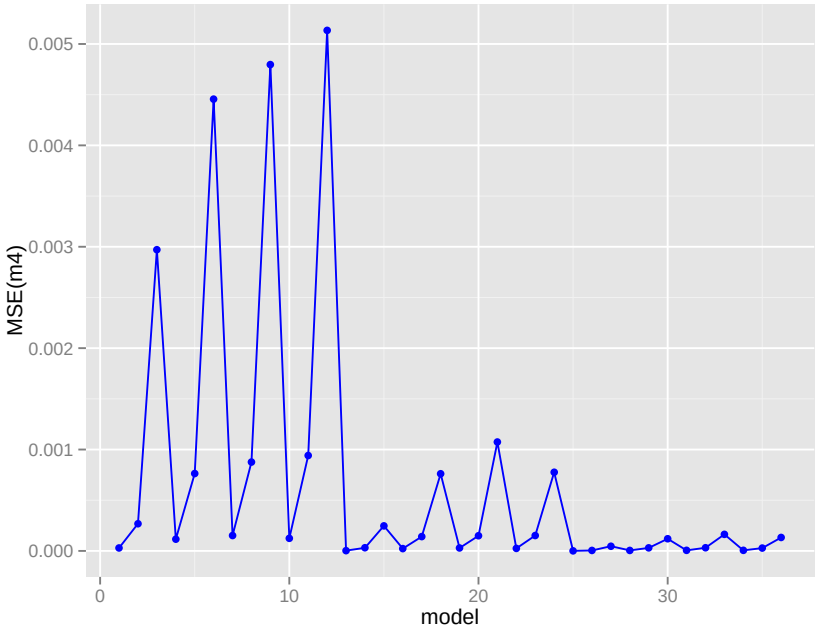

Figure 5: The average MSE values for m_4 for various models. The models are sorted consecutively by n, f and v.

An obvious advantage of the SMR method is that it can treat both smooth and non-smooth responses: for small values of the smoothing parameter, a response close to piecewise-constant is produced, and the increasing smoothing parameter value increases the degree of smoothness. However, our model has some limitations. First, although the fitted response is continuous, it is in general not differentiable. This is because the predictive model uses only two adjacent fitted values to make prediction for a new input value, and this may lead to that the left and right derivatives of the fitted response may be different in some points. Another limitation is that the fitted response and the prediction will depend on the choice of the kernel, but this is also a property of many other methods, such as kernel smoothers.

There are several directions for the future work. The SMR problem was formulated in the setting with one predictor variable, and it would be interesting to extend it to models with multiple predictors, and also incorporate the current model into a generalized additive model setting. Another interesting direction is to state a joint fitted and predicted model and perform a full Bayesian analysis of this model.

References

[1] S.A. Acton and A.C. Bovik. Nonlinear image estimation using piecewise and local image models. *IEEE Transactions on Image Processing*, 7:979–991, 1998.

[2] Y. Ant-Sahalia and J. Duarte. Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, 116:9–47, 2003.

[3] 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.

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

[5] M.J. Best. Equivalence of some quadratic programming algorithms. *Mathematical Programming*, 30(1):71–87, 1984.

[6] O. Burdakov and O. Sysoev. Regularized monotonic regression. Technical Report LiTH-MAT-R–2016/02–SE, Department of Mathematics, Linköping University, 2016.

[7] O. Burdakov, O. Sysoev, Grimvall A., 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, 2006b.

[8] C. de Boor. *Elementary numerical analysis*. McGraw-Hill, 1972.

[9] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[10] J. Friedman and R. Tibshirani. The monotone smoothing of scatterplots. *Technometrics*, 26(3):243–250, 1984.

[11] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[12] P. Hall and L.S. Huang. Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, pages 624–647, 2001.

[13] 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.

[14] C.I.C. Lee. On estimation for monotone dose-response curves. *J. Amer. Statist. Assoc.*, 91:1110–1119, 1996.

[15] E. Mammen. Estimating a smooth monotone regression function. *The Annals of Statistics*, pages 724–740, 1991.

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

[17] M.C. Meyer. Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206, 2012.

[18] H. Mukerjee. Monotone nonparametric regression. *The Annals of Statistics*, pages 741–750, 1988.

[19] K.P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[20] B. Neelon and D. B. Dunson. Bayesian isotonic regression and trend analysis. *Biometrics*, 60(2):398–406, 2004.

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

[22] J.K. Pal. Spiking problem in monotone regression: Penalized residual sum of squares. *Statistics & Probability Letters*, 78(12):1548–1556, 2008.

[23] J. O. Ramsay. Monotone regression splines in action. *Statistical science*, pages 425–441, 1988.

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

[25] 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.

[26] O. Sysoev, A. Grimvall, and O. Burdakov. Bootstrap confidence intervals for large-scale multivariate monotonic regression problems. *Communications in Statistics-Simulation and Computation*, (accepted for publication), 2014. DOI:10.1080/03610918.2014.911899.

[27] J. Wu, M.C. Meyer, and J.D. Opsomer. Penalized isotonic regression. *Journal of Statistical Planning and Inference*, 161:12–24, 2015.

[28] J.T. Zhang. A simple and efficient monotone smoother using smoothing splines. *Journal of Nonparametric Statistics*, 16(5):779–796, 2004.