Efficient Algorithms for General Isotone Optimization

Xiwen Wang, Jiaxi Ying, José Vinícius de M. Cardoso, Daniel P. Palomar

The Hong Kong University of Science and Technology
{xwangew, jx.ying, jvdmc}@connect.ust.hk, palomar@ust.hk

Abstract

Monotonicity is often a fundamental assumption involved in the modeling of a number of real-world applications. From an optimization perspective, monotonicity is formulated as partial order constraints among the optimization variables, commonly known as isotone optimization. In this paper, we develop an efficient, provable convergent algorithm for solving isotone optimization problems. The proposed algorithm is general in the sense that it can handle any arbitrary isotonic constraints and a wide range of objective functions. We evaluate our algorithm and state-of-the-art methods with experiments involving both synthetic and real-world data. The experimental results demonstrate that our algorithm is more efficient by one to four orders of magnitude than the state-of-the-art methods.

Introduction

We consider a family of optimization problems under partial orders, also known as isotonic constraints. This so-called isotone optimization problem can be regarded as a generalization of the *isotonic regression* (Ayer et al. 1955; Ubhaya 1974; Durot 2008), a fundamental problem in statistics and machine learning.

As the simplest form of isotone optimization, isotonic regression is formulated as

$$\min_{\mathbf{x}} \ \sum_{i=1}^{p} (x_i - y_i)^2, \quad \text{s.t. } x_1 \le x_2 \le \cdots \le x_p, \qquad (1)$$

where the optimization variables $\{x_i\}_{i=1}^{p}$ are linearly ordered and $\{y_i\}_{i=1}^{p}$ are the data samples.

Isotonic constraints can be represented using a directed acyclic graph (DAG) $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_p\}$ denotes the set of vertices associated with each element of $\mathbf{x} = (x_1, \ldots, x_p)$, and $\mathcal{E} = \{(v_i, v_j), \ldots\}$ is the edge set that defines a partial order over the vertices. For instance, the edge set for problem (1) is expressed as a chain graph with $\mathcal{E} = \{(v_1, v_2), \ldots, (v_{p-1}, v_p)\}$. Let $\mathbf{C} \in \mathbb{R}^{p \times p}$ denote the connectivity matrix of the graph $\mathcal{G}$, where

$$[\mathbf{C}]_{ij} = 1, \text{ if } (v_i, v_j) \in \mathcal{E}; \quad [\mathbf{C}]_{ij} = 0, \text{ if } (v_i, v_j) \notin \mathcal{E}.$$

Namely, a constraint $x_i \le x_j$ exists iff $[\mathbf{C}]_{ij} = 1$.

The isotone optimization of interests in this paper is formulated as

$$\min_{\mathbf{x}} f(\mathbf{x}), \text{s.t. } [\mathbf{C}]_{ij}(x_i - x_j) \le 0, \forall i, j \in \{1, \ldots, p\}.$$

It generalizes the isotonic regression in two aspects. On the one hand, it allows the constraints to be defined on arbitrary DAG. On the other hand, instead of the ℓ_2 norm, the objective function f can be more general.

Monotonicity is a common assumption in many real-world applications, in which we impose isotonic constraints to formulate isotone optimization problems. For instance, in machine learning, isotonic constraints are applied to probability calibration (Niculescu-Mizil and Caruana 2005; Guo et al. 2017; Naeini, Cooper, and Hauskrecht 2015), as the reliability curve is assumed to be monotonically increasing. In statistics, isotonic constraints are also called shape constraints and are widely used in shape-restricted non-parametric estimation (Feelders and Van der Gaag 2006; Groeneboom and Jongbloed 2014; Horowitz and Lee 2017). In genetics, monotonicity is viewed as a key feature of genotype-phenotype mappings Gjuvsland et al. (2013); thus it is suitable to model genetic interactions in heritability (Luss et al. 2012). In addition, isotone optimization also appears in many other modeling problems, such as dose-response (Hu et al. 2005) and psychology models (Kruskal 1964).

Isotonic constraints may be categorized as chain, tree, and arbitrary, whose respective DAGs are shown in Figure 1. However, most existing methods available in the literature focus on the case of chains (Ahuja and Orlin 2001). Algorithms that can address arbitrary isotonic constraints are often restricted to a small class of loss functions (Stout 2013).

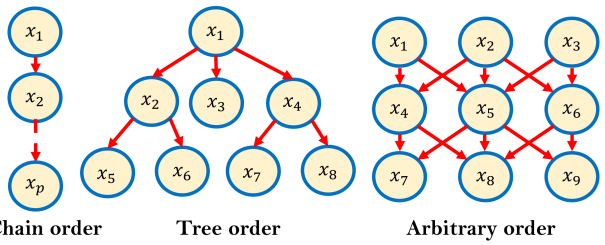

Figure 1: Graph representation of isotonic constraints. Arrow from x_i to x_j implies a partial order $x_i \le x_j$.

The contributions of this paper are listed as follows:

(1) We propose a unified and efficient algorithm that can handle arbitrary isotonic constraints. Instead of directly find-

ing a primal-dual point that satisfies the whole KKT conditions, this novel algorithm decomposes the KKT system, each of which contains a subset of variables thus can be handled more efficiently. This strategy results in higher efficiency and superior flexibility over numerous loss functions.

(2) We prove that our proposed algorithm converges to the optimal solution without oscillations.

(3) We show that our algorithm is one to four orders of magnitude faster than state-of-the-art solvers, evaluated on both synthetic and real-world data.

The remaining part of this paper is organized as follows. We first briefly review the state-of-the-art methods and discuss their limitations. Then, we design an efficient algorithm for isotone optimization and analyze its convergence. Finally, we evaluate the performance of our algorithms and the state-of-the-art ones on both synthetic and real-world data.

Related Works

Algorithms for isotone optimization with chain isotonic constraints have been broadly studied. A well-known one is the pool adjacent violators algorithm (PAVA) (Ayer et al. 1955; Robertson 1988; Grotzinger and Witzgall 1984), which can be viewed as a dual active set method (Best and Chakravarti 1990) and thus can be extended to the ℓ_p loss (Best, Chakravarti, and Ubhaya 2000; Kyng, Rao, and Sachdeva 2015), Huber loss (Lim 2018), and a number of other separable convex losses (Ahuja and Orlin 2001; Luss and Rosset 2014). In practice, PAVA manifests prominent efficiency. However, it fails to obtain optimality when the constraints are not chains.

To address more general isotonic constraints (Han et al. 2019; Deng, Zhang et al. 2020), relaxations of the optimization problems by reformulating the isotonic constraints as regularization terms could perform as alternatives (Luss et al. 2012; Burdakov and Sysoev 2017; Sysoev and Burdakov 2019). Though this may yield more robust estimators (Luss, Rosset et al. 2017), the models would be invalid in case strict monotonicity is requisite.

As constrained convex programming, general isotone optimization can be tackled by the interior point method (IPM) or the active set method (ASM). The IPM framework for isotonic regression, investigated in (Kyng, Rao, and Sachdeva 2015), is more scalable than the ASM. However it inevitably inherits some shortcomings from the IPM. For instance, each iterate is computationally expensive, and Newton steps may not be trivial to compute for some objectives. Another framework is the ASM, which is highly efficient in many applications. There are some implementations of ASM in the field of isotone optimization, like the works of (Bonnans et al. 2006; Mair, Hornik, and de Leeuw 2009), which apply primal ASM framework to handle arbitrary isotonic constraints. Despite their ability to handle various loss functions, the computational cost is too prohibitive to be of practical use (Cimini and Bemporad 2017; Arnström and Axehill 2019) because they do not fully exploit the unique properties of isotonic constraints. The most efficient implementation of ASM in isotone optimization is the generalized PAVA (Yu and Xing 2016). However, it is designed for chain constraints, and the extension to general isotonic constraints is nontrivial.

Apart from these frameworks, one can generalize the ideas underpinning the PAVA into more elaborate cases. Motivated by the fact that PAVA iteratively solves the Karush-Kuhn-Tucker (KKT) conditions, the partitioning algorithms (Luss et al. 2012; Luss and Rosset 2014) iteratively find an optimal cut for the constraint graph by solving linear programming. The limitation is that the objectives are restricted to simple least squares.

In this paper, we propose an active-set-like block-merging algorithm. The proposed method can be seen as an extension of PAVA via decomposing the KKT conditions.

Proposed Algorithms

We propose an efficient algorithm to solve the general isotone optimization problems under arbitrary isotonic constraints. This 'block-merging' algorithm extends PAVA with a primal-dual strategy and is designed in an outer/inner loop structure. The details and the theoretical convergence will be established in this section.

Algorithm Architecture and Outer Loop

In this paper, we assume the following assumption holds for the objective functions.

Assumption 1. *The objective function $f : \mathbb{R}^p \to \mathbb{R}$ is convex, differentiable, and separable, which means that f can be written as $f(\mathbf{x}) = \sum_{i=1}^{p} f_i(x_i)$. The solution of $\partial f_i(x) = 0$ exists for each $i \in \{1, \dots, p\}$.*

Given the connectivity matrix $\mathbf{C}$ of a directed acyclic graph, the problem is formulated as

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \sum_{i=1}^{p} f_i(x_i), \\ \text{s.t.} & [\mathbf{C}]_{ij}(x_i - x_j) \le 0, \forall i, j \in \{1, \dots, p\}. \end{array} \tag{2}$$

Then, the Lagrangian is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{p} f_i(x_i) + \sum_{i=1}^{p} \sum_{j=1}^{p} [\mathbf{C}]_{ij} \lambda_{ij} (x_i - x_j),$$

where $\{\lambda_{ij}\}$ multiplied by $[\mathbf{C}]_{ij} = 1$ are the dual variables, and the KKT conditions are

$$\begin{aligned} \partial f_i(x_i) + \textstyle\sum_{s=1}^{p} ([\mathbf{C}]_{is} \lambda_{is} - [\mathbf{C}]_{si} \lambda_{si}) &= 0, \\ [\mathbf{C}]_{ij} \lambda_{ij} &\ge 0, \\ [\mathbf{C}]_{ij}(x_i - x_j) &\le 0, \\ [\mathbf{C}]_{ij} \lambda_{ij} (x_i - x_j) &= 0, \end{aligned} \tag{3}$$

for any indexes $i, j \in \{1, \dots, p\}$. Our algorithm aims to find a sequence of $(\mathbf{x}, \boldsymbol{\lambda})$ that converges to the optimal solution of the KKT system.

The design is motivated by the observations that the optimal solution is composed of several blocks of equalities, i.e., $[\mathbf{C}]_{ij}(x_i - x_j) = 0$, indicating that the constraint graph $\mathcal{G}$ would ultimately be partitioned into several sub-graphs, denoted as *blocks*.

Definition 1. A block B_k refers to a set of indexes that correspond to a set of primal variables sharing the same value, i.e.,

$$x_i = x_j \quad \forall i, j \in B_k. \tag{4}$$

Similarly, we define the set of primal and dual variables associated with B_k

$$\begin{aligned} X_k &= \{ x_i \mid i \in B_k \}, \\ \Lambda_k &= \{ \lambda_{ij} \mid i, j \in B_k, [\mathbf{C}]_{ij} = 1 \}. \end{aligned}$$

An example is illustrated in Figure 2. The graph is divided into three subgraphs $\{\mathcal{G}_k\}_{k=1}^3$, represented as $B_1 = \{1,3\}$, $B_2 = \{2,4\}$ and $B_3 = \{5\}$. Note that $\{\Lambda_k\}_{k=1}^3$ does not involve all dual variables because $\{\mathcal{G}_k\}_{k=1}^3$ only covers the edges inside any $\mathcal{G}_k$. Specifically, $\Lambda_1 = \{\lambda_{13}\}$, $\Lambda_2 = \{\lambda_{24}\}$, and $\Lambda_3 = \emptyset$, while λ_{12}, λ_{34}, and λ_{25} do not belong to any Λ_k.

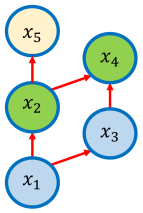

Figure 2: Example of blocks.

Owing to the definition of blocks, at optimal point, the KKT conditions reduce to

(a) **Optimal block partitions**:

$$\{1,\dots,p\} = B_1 \cup \cdots \cup B_K. \tag{5}$$

(b) **The decomposed KKT systems** at each block:

- $\forall i \in B_k$,

$$\partial f_i(x_i^\star) + \textstyle\sum_{s\in B_k} \left([\mathbf{C}]_{is}\lambda_{is}^\star - [\mathbf{C}]_{si}\lambda_{si}^\star\right) = 0. \tag{6}$$

- $\forall i,j \in B_k$,

$$[\mathbf{C}]_{ij}\left(x_i^\star - x_j^\star\right) = 0, \tag{7a}$$

$$[\mathbf{C}]_{ij}\lambda_{ij}^\star \geq 0, \tag{7b}$$

$$[\mathbf{C}]_{ij}\lambda_{ij}^\star\left(x_i^\star - x_j^\star\right) = 0. \tag{7c}$$

(c) **Primal feasibility** over block partitions:

- $\forall i \in B_k, j \in B_l$,

$$[\mathbf{C}]_{ij}\left(x_i^\star - x_j^\star\right) \leq 0, \tag{8a}$$

$$[\mathbf{C}]_{ij}\lambda_{ij}^\star = 0, \tag{8b}$$

$$[\mathbf{C}]_{ij}\lambda_{ij}^\star\left(x_i^\star - x_j^\star\right) = 0. \tag{8c}$$

Therefore, each of local KKT systems only contains a subset of the variables and can be solved with significantly reduced computational cost. Among these equations, the equalities (7a), (7c), (8b), and (8c) hold according to the block definitions. We specifically emphasize the inequalities (7b) and (8a) as they are associated with the inner loop's trigger condition, and the whole algorithm's stopping criteria, respectively.

The algorithm starts by a guess of block partition, with each of the decomposed KKT system solved. The most heuristic initialization is to denote every index as an one-element block. Two steps are executed at each outer iteration:

1. Merging blocks. Define B_o along with X_o and Λ_o by merging two blocks via finding the maximum violation in the primal feasibility (8a):

$$(i,j) = \arg\max_{i,j} [\mathbf{C}]_{ij}\left(x_i - x_j\right), \tag{9}$$

$$B_o = B_k \cup B_l, \text{ where } i \in B_k, j \in B_l. \tag{10}$$

2. Solving the decomposed KKT system at B_o. We first compute $\mathbf{x}_o = z\mathbf{1}$ via solving the summation of (6)

$$\textstyle\sum_{i\in B_o}\left[\partial f_i(z) + \sum_{s\in B_o}\left([\mathbf{C}]_{is}\lambda_{is} - [\mathbf{C}]_{si}\lambda_{si}\right)\right] = \sum_{i\in B_o} 0 = \sum_{i\in B_o}\partial f_i(z) = 0,$$

then obtain the dual variables $\boldsymbol{\lambda}_o = (\lambda_{ij})$, $\lambda_{ij} \in \Lambda_o$ via this system of linear equations (6).

- If $\min(\boldsymbol{\lambda}_o) \geq 0$, (7b) holds and $(\mathbf{x}_o, \boldsymbol{\lambda}_o)$ already solves the decomposed KKT system.
- If $\min(\boldsymbol{\lambda}_o) < 0$, meaning that $x_i = x_j\ \forall i,j \in B_o$ does not hold, then the block should be split and an inner loop, introduced in the next subsection, is required to solve the decomposed KKT system.

When the primal feasibility (8a) is reached, i.e.,

$$\max_{i,j}\left([\mathbf{C}_{ij}\left(x_i - x_j\right)]\right) \leq 0,\ \forall i,j \in \{1,\dots,p\}, \tag{11}$$

we establish (6), (7), and (8) for all primal and dual variables, therefore the algorithm reaches the optimal. Due to the nature in which the violators are found and merged sequentially, our algorithm is named Sequential Block Merging (SBM) algorithm.

To better illustrate the relationship between the outer and inner loop, we show one example in Figure 3, in which we iteratively merge blocks at outer iterations, while we split the block at the edge (v_5, v_6) in the inner loop when we find $x_1 = x_2 = x_5 = x_6$ could not construct the solution to the decomposed KKT system at $B_o = \{1,2,5,6\}$.

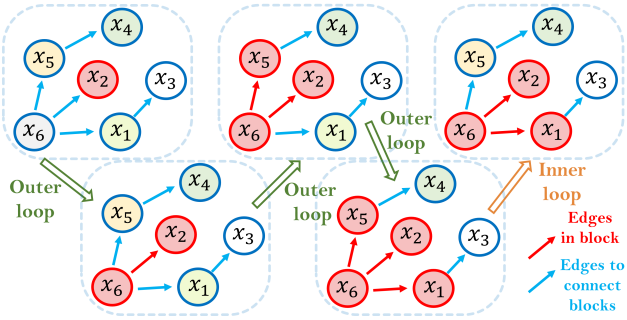

Figure 3: Illustrations of the outer/inner loop design.

Interestingly, the block-merging procedures are reminiscent of the PAVA method, which in fact, is a special case of our SBM algorithm applied to chain isotonic constraints. PAVA does not check $\boldsymbol{\lambda}_o \geq \mathbf{0}$ at each outer loop iteration because this inequality can be guaranteed when $\mathcal{G}$ is a chain. In other words, the inner loop will never appear in the chain isotone optimization problems.

Theorem 1. *If $\mathcal{G}$ is a chain, the proposed SBM algorithm specializes to the generalized PAVA.*

Another difficulty that will not appear in the chain graph resides in the system of linear equations (6), which could be over-determined due to the redundancy of dual variables, resulting in an infinite number of solutions to $\boldsymbol{\lambda}_o$. One intuitive idea is to reduce the local graph $\mathcal{G}_o$ to a tree such that (6) would be just determined. Such construction can be characterized with the help of an auxiliary matrix $\mathbf{C}' \in \mathbb{R}^{p\times p}$. Whenever two blocks B_k and B_l are merged via edge (v_i, v_j), we set $[\mathbf{C}']_{ij} = 1$. Similarly, let $[\mathbf{C}']_{ij} = 0$ if the block is split via (v_i, v_j). Then the new connectivity matrix $\mathbf{C}'$ would refer to a tree.

Lemma 1. *The solution of $\boldsymbol{\lambda}_o$ to the following system of linear equations within $i \in B_o$ is unique*

$$\partial f_i(z) + \textstyle\sum_{s\in B_o}\left([\mathbf{C}']_{is}\lambda_{is} - [\mathbf{C}']_{si}\lambda_{si}\right) = 0. \tag{12}$$

Suppose the solution to (12) is $z \in \mathbb{R}$ and $\boldsymbol{\lambda}'_o = \{\lambda'_{ij}\} \in \mathbb{R}^{|B_o|-1}$, we set $\lambda_{ij} = \lambda'_{ij}$ if $[\mathbf{C}]'_{ij} = 1$ and $\lambda_{ij} = 0$ otherwise. Obviously, $(z, \boldsymbol{\lambda}_o)$ is one solution to (6). Regardless of $\boldsymbol{\lambda}_o \geq \mathbf{0}$, later these procedures are abbreviated in pseudo-code as 'update $(\mathbf{x}_o, \boldsymbol{\lambda}_o)$'.

Inner Loop Design

The outer loop decomposes the problem into many local KKT systems with subset of variables and assumes the primal variables in one block would be equal. However, this may not be satisfied when $\min(\boldsymbol{\lambda}) < 0$ (the sub-script o is omitted in the subsection). The role of the inner loop is to solve the decomposed KKT system when such cases appear. Before we introduce the concepts and detailed procedures, we first show the diagram of the inner loop design in Figure 4.

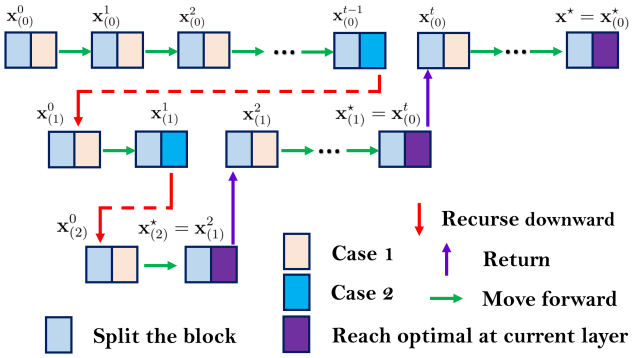

Figure 4: Diagram of inner loop procedures.

The inner loop is in a recursive fashion. The idea is illustrated in the 0-th layer with the sub-script (0) omitted and could be easily generalized to other layers.

Move forward: At 0-th layer, a sequence of primal feasible points $\mathbf{x}^0, \mathbf{x}^1, \ldots$ is generated such that

$$\mathbf{x}^t = q\left(G^t, \bar{G}^t\right) = \begin{array}{l} \arg\min_{\mathbf{x}} \sum_{i\in B} f_i(x_i), \\ \text{s.t.} \begin{cases} x_i - x_j = 0, & \forall (i,j) \in G^t, \\ x_i - x_j \leq 0, & \forall (i,j) \in \bar{G}^t. \end{cases} \end{array}$$

where the index pair G^t, a set of equality constraints, and $\bar{G}^t$, a set of inequality constraints, should satisfy

$$\bar{G} = G^t \cup \bar{G}^t = \left\{(i,j) \,\middle|\, i,j \in B, [\mathbf{C}']_{ij} = 1\right\}, \forall t.$$

We initialize $G^0 = \left\{(i,j) \,\middle|\, i,j \in B, [\mathbf{C}']_{ij} = 1\right\}$ and $\bar{G}^0 = \emptyset$, such that $\mathbf{x}^0 = q\left(G^0, \bar{G}^0\right)$. Their union $\bar{G}$ refers to the target inequality constraints set at this layer. Obviously, if we keep increasing $\bar{G}^t$ by removing elements from G^t, the value of $\sum_{i\in B} f_i(x_i)$ will strictly decrease. Eventually, $\mathbf{x}^t$ would converge to $q\left(\emptyset, \bar{G}\right)$, the optimal of the local KKT system.

Split the block: We apply $\bar{G}^t = \bar{G}^{t-1} \cup \{(u,v)\}$ where $\lambda_{uv} = \min(\boldsymbol{\lambda})$ corresponds to the most negative dual variables. It means that we can achieve lower objective values via converting equality constraint $x_u = x_v$ into inequality constraint $x_u \leq x_v$, resulting in the split of the block. Note that the primal feasibility should be restored immediately if found violated because every $\mathbf{x}^t$ in the inner loop should be primal feasible.

Algorithm 1 Sequential block merging (SBM).

Input: $\{f_i\}$, $\mathbf{C}$, and initialized $\mathbf{x}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$, $\mathbf{C}'$.
while (11) not met **do**
  Outer loop: Compute B_o and update $(\mathbf{x}_o, \boldsymbol{\lambda}_o)$
  if $\min(\boldsymbol{\lambda}_o) < 0$ **then**
    $\bar{G} = \left\{(i,j) \,\middle|\, i,j \in B_o, [\mathbf{C}']_{ij} = 1\right\}$
    Inner loop: SOLVE$\left(\bar{G}, \mathbf{x}_o, \boldsymbol{\lambda}_o\right)$
  end if
end while

function <SOLVE>(<$\bar{G}$, $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$>) ▷ $\bar{G}$ is the set of inequality contraints, $\mathbf{x}^0, \boldsymbol{\lambda}^0$ are initial points
  $t = 0$, $\bar{G}^0 = \left\{(i,j) \,\middle|\, x_i \neq x_j, i,j \in \bar{G}\right\}$
  while $\lambda_{uv} = \min_{(i,j)\in\bar{G}}(\lambda_{ij}) < 0$ **do**
    $\bar{G}^t = \bar{G}^{t-1} \cup \{u,v\}, (u \in B_k, v \in B_l)$
    update $(\mathbf{x}_k, \boldsymbol{\lambda}_k)$, update $(\mathbf{x}_l, \boldsymbol{\lambda}_l)$
    while $\exists (i \in B_k, j \in B_l) \in \bar{G} : x_i - x_j > 0$ **do**
      $B_s \leftarrow B_k \cup B_l$, update $(\mathbf{x}_s, \boldsymbol{\lambda}_s)$
    end while
    if $\exists (i,j) \in \bar{G}^t, \lambda_{ij} < 0$ **then**
      Recursion: SOLVE $\left(\bar{G}^t, \mathbf{x}^t, \boldsymbol{\lambda}^t\right)$
    end if
    $t \leftarrow t + 1$
  end while
end function

Recurse downward: Whenever we 'split the block', there would be two possible outcomes. If $\forall (i,j) \in \bar{G}^t, \lambda_{ij} \geq 0$, obviously $\mathbf{x}^t = q\left(G^t, \bar{G}^t\right)$, a move forward step is completed, and we call it *case 1* in Figure 4. If $\exists (i,j) \in \bar{G}^t, \lambda_{ij} < 0$, then current $\mathbf{x}^t \neq q\left(G^t, \bar{G}^t\right)$ marked as *case 2* in Figure 4. Then, in order to solve $q\left(G^t, \bar{G}^t\right)$, we define an new inequality-constrained sub-problem in terms of a simpler version of the original problem, with a smaller set of target inequality constraints $\bar{G}$ and different starting points. We illustrate this idea in Figure 5.

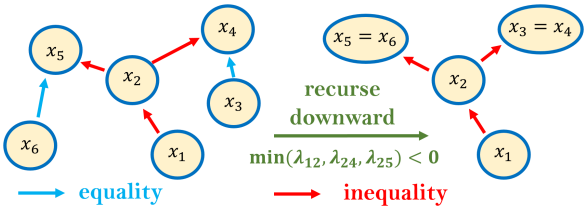

Figure 5: Example of 'recurse downward'. The sub-problem is defined on a smaller set of target inequality constraints $\bar{G} = \{(1,2),(2,4),(2,5)\}$. Therefore, the inequalities $x_6 \leq x_5$ and $x_3 \leq x_4$ are not considered anymore in subsequent sub-problems.

Return: When $\forall (i,j) \in \bar{G}, \lambda_{ij} \geq 0$, together with a primal feasible $\mathbf{x}$, all the KKT conditions hold. If we are in 0-th layer, the inner loop terminates, otherwise it returns to the upper layer, resulting in a 'move forward' step. Inevitably, every layer will return because the number of 'move forward' steps

is bounded and the objective value in each layer is strictly decreasing.

More technical details and convergence analysis are deferred to the Supplementary Material. Note that our method has some similarities with the primal active set method (PASM). However, the conventional PASM is only provably convergent under limited cases, like quadratic programming (Bonnans et al. 2006), while our method is provably convergent under any f that satisfies Assumption 1. With the details of the inner loop, the pseudo-code for the SBM algorithm is summarized in Algorithm 1.

Convergence Analysis

To establish the theoretical convergence of our algorithm, we first show the convergence of the inner loop, which can be seen as a convergent implementation of the active set strategy, in Lemma 2.

Lemma 2. *The inner loop converges to a solution of (6), (7), and (8) for all variables in X_o and Λ_o.*

With Lemma 2, the convergence of the outer loop can be also established, shown in Theorem 2. In the Supplementary Material we show that the outer loop can be seen as a variant of the dual active set method. Therefore, the SBM algorithm is a primal-dual strategy.

Theorem 2. *The proposed SBM algorithm converges to an optimal solution of problem (2).*

Though the SBM algorithm is designed in loop, the inner loop is needed only when $\boldsymbol{\lambda}_o \geq \mathbf{0}$ is not satisfied. The number of inner loops depends on the local graph structure and the strategies used in the outer loop. For example, if we do not find the maximum violator in each outer loop, then the inner loop would be more likely to happen. In practice, the number of inner loop is small when the data already has some monotonicity, which is usually the case in the practical isotone optimization problems.

Extension to Non-separable Objectives

In numerous real-world applications, most of the methods in the literature could not be applied if the objective function f is not separable. Here we introduce some optimization frameworks that can decompose the problem into a sequence of sub-problems with separable objective functions.

One is the successive convex approximation (Scutari et al. 2013), which solves the problem by iteratively solving the following isotone optimization problem

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\|\mathbf{x} - \left(\mathbf{x}^k - \eta \mathbf{d}^k\right)\right\|_2^2, \text{ s.t. } \mathbf{x} \in \mathcal{X}, \quad (13)$$

in which η is the step size and $\mathbf{d}^k$ refers to the descent direction. The problem (13) can be solved via our SBM algorithm and many other state-of-the-art solvers. Note that when $\mathbf{d}^k = \partial f\left(\mathbf{x}^k\right)$, which is the gradient or sub-gradient of f, this method specializes to the projected gradient descent (PGD), or the projected sub-gradient descent (PGSD).

Another framework – Majorization Minimization (MM) (Hunter and Lange 2004; Sun, Babu, and Palomar 2016) minimizes $f(\mathbf{x})$ by iteratively optimizing its upper bound surrogate function $\tilde{f}\left(\mathbf{x}; \mathbf{x}^k\right)$

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \tilde{f}\left(\mathbf{x}; \mathbf{x}^k\right), \text{ s.t. } \mathbf{x} \in \mathcal{X}, \quad (14)$$

in which the separable function $\tilde{f}$ should satisfy

$$\tilde{f}\left(\mathbf{x}; \mathbf{x}^k\right) \geq f(\mathbf{x}), \partial \tilde{f}\left(\mathbf{x}; \mathbf{x}^k\right)\big|_{\mathbf{x}=\mathbf{x}^k} = \partial f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}^k}.$$

Then, our SBM algorithm can tackle (14). This MM approach can be very efficient if $\tilde{f}$ is elegantly designed, but many state-of-the-art methods can not solve (14) as $\tilde{f}$ do not meet their requirements.

Numerical Simulations

In this section, we conduct numerical experiments on both synthetic data and real-world data to verify the performance of the proposed SBM algorithm.

Synthetic Data

We evaluate the performance of our algorithm on synthetic data. The benchmarks isotone (Mair, Hornik, and de Leeuw 2009), quadprog (Goldfarb and Idnani 1983), IRP (Luss and Rosset 2014), and IPM (Kyng, Rao, and Sachdeva 2015) are summarized in Table 1.

Table 1: Benchmarks methods for isotone optimization.

Name for short	Method	Objective
Pkg isotone	Primal Active Set	Separable
Pkg quadprog	Dual Active Set	Quadratic
IRP	Recursive Partitioning	ℓ_2 norm
IPM	Interior Point Method	ℓ_p norm

We consider two representative isotonic constraints, binary tree and 2d-grid constraints, shown in Figure 6 with the problem size $p = 10^3$.

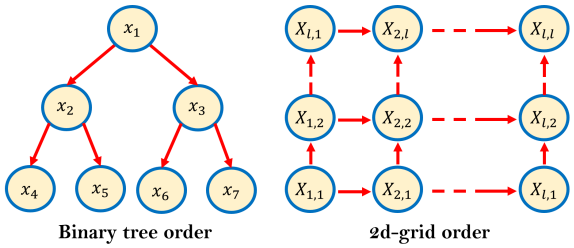

Figure 6: Graphs of binary tree and 2d-grid.

The average computational time for ℓ_2 norm loss is evaluated on 100 randomly generated data sets, with the initial violating rate around $20 - 50\%$. The performance is shown in Figure 7, in which the SBM method outperforms other benchmark methods under both kinds of isotonic constraints by one to four orders of magnitude. Its performance is extraordinary, especially when the number of initial violations is not too huge.

Another advantage of applying the SBM algorithm is the flexibility in tackling various convex separable objective

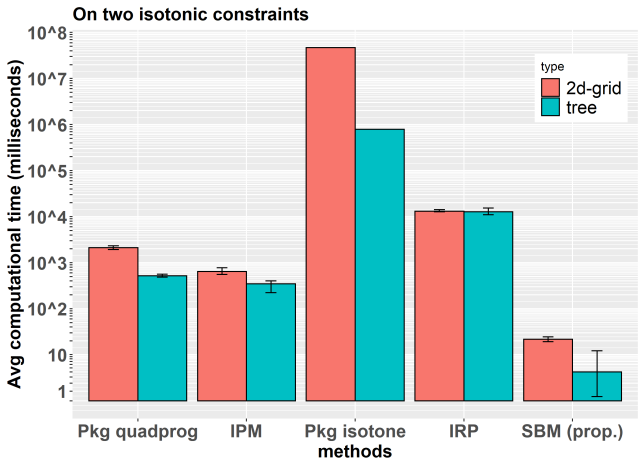

Figure 7: Average computational time of different methods on two different isotonic constraints.

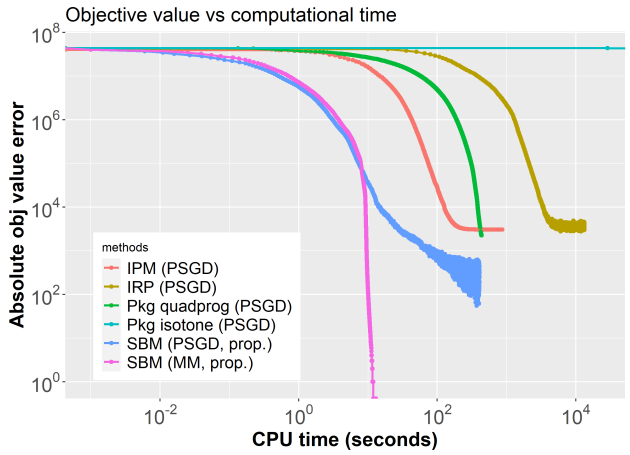

Figure 8: Comparison on solving problem (15). The absolute objective value error is defined as the difference of the objective value at each iteration and the smallest objective value we obtained across all the methods.

functions, which is crucial for algorithm design in practice. Suppose we want to solve a multivariate isotonic regression problem with non-convex weighted ℓ_1 norm regularization, formulated as

$$\begin{array}{ll} \min\limits_{\mathbf{X}} & \|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2+\lambda\sum_{i,j}\log\left(1+\frac{|X_{i,j}|}{\epsilon}\right) \\ \text{s.t.} & \text{vec}\left(\mathbf{X}\right)\in\mathcal{X}, \end{array} \tag{15}$$

where $\mathcal{X}$ represents a 2d-grid isotonic cone, ϵ and λ are given hyper-parameters. The regularization term enforces sparsity on the values of each $X_{i,j}$, while the isotonic constraints restrict the sparsity pattern under some orders. To solve this problem, the projected sub-gradient descent (PSGD) can be applied, with the projection problems solved by IPM, IRP, quadprog, or our SBM method.

Unlike PSGD, the MM algorithm minimizes $f\left(\mathbf{X}\right)$ by iteratively minimizing its upper bounds. The defined surrogate function can be chosen as

$$\tilde{f}\left(\mathbf{X};\mathbf{X}^k\right)=\text{const}+\sum_{i,j}\left[\tilde{f}_{i,j}^{(1)}\left(X_{i,j};\mathbf{X}^k\right)+\tilde{f}_{i,j}^{(2)}\left(X_{i,j};\mathbf{X}^k\right)\right],$$

in which

$$\begin{aligned}\tilde{f}_{i,j}^{(1)}\left(X_{i,j};\mathbf{X}^k\right)=2\left[\mathbf{A}^T\left(\mathbf{A}\mathbf{X}^k-\mathbf{Y}\right)\right]_{i,j}\left(X_{i,j}-X_{i,j}^k\right)\\ +\tfrac{1}{2\eta}\left(X_{i,j}-X_{i,j}^k\right)^2,\end{aligned}$$

upper bounds the term $\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2$, and

$$\tilde{f}_{i,j}^{(2)}\left(X_{i,j};\mathbf{X}^k\right)=\begin{cases}\frac{\lambda}{\epsilon+|X_{i,j}^k|}\text{sign}\left(X_{i,j}\right)X_{i,j} & \text{if } |X_{ij}|\geq\delta,\\ \frac{\lambda}{\epsilon+|X_{i,j}^k|}\left(\frac{1}{2\delta}X_{ij}^2+\frac{1}{2}\delta\right) & \text{if } |X_{ij}|<\delta,\end{cases}$$

upper bounds the term $\lambda\sum_{i,j}\log\left(1+|X_{i,j}|/\epsilon\right)$ with a small coefficient δ to make it smooth. The constant guarantees $\tilde{f}\left(\mathbf{X}^k;\mathbf{X}^k\right)=f\left(\mathbf{X}^k\right)$ and more details are elaborated in the Appendix. As $\tilde{f}\left(\mathbf{X};\mathbf{X}^k\right)$ is convex and separable, minimizing $\tilde{f}\left(\mathbf{X};\mathbf{X}^k\right)$ subject to $\text{vec}\left(\mathbf{X}\right)\in\mathcal{X}$ can be efficiently solved by our SBM algorithm.

In the experiments, all the methods are initialized with the same strictly feasible point. We set $\lambda=20$, $\epsilon=0.1$, $p=30^2$, and step size $\eta=5\times10^{-4}$ for all the methods. As the problem is non-convex, we do not guarantee global optimal. Instead, we compare the objective values given the same computational time. The performance is shown in Figure 8.

At the early stage, the SBM (PSGD) method outperforms the SBM (MM) approach as its objective function at each iteration is simpler. However, none of the PSGD methods converge well when they are close to local optimal due to the defect of simple projection. The IRP (PSGD) and the SBM (PSGD) methods oscillate around local optimal. One reason is that the sub-gradient method is not a descent method. Another is that the solutions obtained by IPM, quadprog, and IRP are not exact; hence the numerical errors at each iteration would accumulate. Compared to other methods, our proposed SBM algorithm is desirable for the following reasons:

- It is one to fours orders of magnitude faster than other benchmark methods.
- It provides more flexibility to various objective functions. Therefore, we can apply it together with the MM algorithm to achieve the smallest objective value.
- It converges to the exact solution for each sub-problem, while other methods can only have solutions with small errors. This is one reason why SBM could obtain smaller objective values than others under the PSGD framework.

Real Data

Isotone optimization is widely used in non-parametric estimation under shape constraints (Horowitz and Lee 2017; Guntuboyina, Sen et al. 2018), which aims at learning a function $f:\mathbf{x}\in\mathbb{R}^d\to y\in\mathbb{R}$ that best maps the input features $(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)$ to the output regression values $(y_1,y_2,\ldots,y_n)$ by learning the parameters $\mathbf{\Theta}=\left(f\left(\mathbf{x}_1\right),f\left(\mathbf{x}_2\right),\ldots,f\left(\mathbf{x}_n\right)\right)$ with the constraints on the shape of $\mathbf{\Theta}$. For example, we restrict $\mathbf{\Theta}$ in a closed iso-

tonic cone $\mathcal{X}$, thus the estimation can be formulated as isotone optimization.

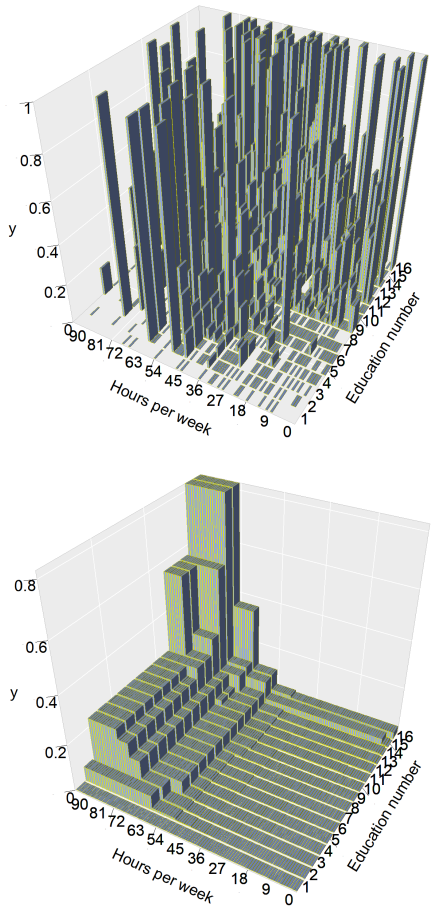

Figure 9: Data visualization before (up) and after (down) imposing the isotonic constraints, which assume that a person should earn more ($y \to 1$) when she/he receives more years of education or works more hours per week.

To illustrate the practicality of our method in real-world applications, we use the Adult data set, available from the UCI Machine Learning repository (Asuncion and Newman 2007). The target is to predict whether the salary of a person is greater or less than $50k$, denoted as $y = 1$ or $y = 0$, respectively, given six continuous and eight nominal attributes. If only two features, the number of years in education (education-num) and working hours per week (hours-per-week), are considered, monotonicity is imposed as it is expected that both years of education and working hours should be positively correlated with income.

Applying a regular uniform grid design, the problem can be formulated as

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta} \in \mathcal{X}} \quad \|\mathbf{W} \odot (\mathbf{\Theta} - \mathbf{Y})\|_F^2 \ , \qquad (16)$$

in which the data matrix $\mathbf{Y} = (y_{i_1,i_2}) \in \mathbb{R}^{p_1 \times p_2}$ ($p_1 = 16$, $p_2 = 99$) is the average of the observations within each grid; the isotonic matrix $\mathbf{\Theta} = (\theta_{i_1,i_2})$ to estimate is restricted on an isotonic cone defined as

$$\mathcal{X} = \left\{ \mathbf{\Theta} \in \mathbb{R}^{p_1 \times p_2} \middle| \, \theta_{j_1,j_2} \le \theta_{k_1,k_2}, \forall j_1 \le k_1, j_2 \le k_2 \right\};$$

the non-negative weights $\mathbf{W} = (w_{i_1,i_2})$ are introduced to counter the imbalance or the incompleteness of the data.

Figure 9 visualizes $\mathbf{Y}$ and $\hat{\mathbf{\Theta}}$. The height of each column indicates the probability of being a positive class ($y = 1$). The isotonic constraints smooth the two-dimensional curve, enforce the monotonicity along each axis and provide an insightful understanding of the pattern: the impact of years of education is negligible, especially when working hours are small. The model (16) not only inherits the advantages of non-linearity from non-parametric estimation but also fully complies with the assumption of monotonicity.

When more features are considered, the problem is to estimate a tensor $\mathbf{\Theta} = (\theta_{i_1,i_2,\ldots,i_d}) \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ under isotonic constraints. Assume that the partial monotonicity is imposed on the first l coordinates ($l = 2$), the underlying isotonic cone is expressed as

$$\mathcal{X} = \{ \mathbf{\Theta} | \, \theta_{j_1,\ldots,j_l,\ldots,j_d} \le \theta_{k_1,\ldots,k_l,\ldots,k_d}, \\ \forall j_1 \le k_1, \ldots, j_l \le k_l, j_{l+1} = k_{l+1}, \ldots, j_d = k_d \} .$$

Then the shape restricted estimation on $\mathbf{\Theta}$ is

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta} \in \mathcal{X}} \sum_{i_1=1}^{p_1} \cdots \sum_{i_d=1}^{p_d} \left[w_{i_1,\ldots,i_d} \left(y_{i_1,\ldots,i_d} - \theta_{i_1,\ldots,i_d} \right) \right]^2 .$$

This is an isotone optimization problem and can be solved by SBM. In our experiment, the following attributes are taken into consideration in sequence: workclass ($p_3 = 3$), occupation ($p_4 = 4$), race ($p_5 = 3$) and sex ($p_6 = 2$). With different numbers of features considered, isotone optimization problems with different problem sizes are solved via SBM and quadprog respectively. The IRP method is not included as it is observed that IRP is not able to converge to the global optimal because of the high conditional number of the Hessian matrix. The experimental results are shown in Table 2.

Table 2: Number of variables p, number of constraints m, and time costs with different number of features d.

d	2	3	4	5	6
p	1584	4752	19008	57024	114048
m	3057	9459	36636	109908	219816
quadprog	24.1s	859.36s	–	–	–
SBM	1.41s	15.69s	99.36s	277.63s	920.76s

From Table 2, we can observe that the proposed SBM algorithm performs well, whereas the QP solver quadprog fails when $d \ge 4$ as it exceeds the memory limit.

Conclusions

In this paper, we propose a unified algorithm for isotone optimization with convex separable losses under arbitrary isotonic constraints. The algorithm aims at solving the KKT conditions by iteratively manipulating the decomposed KKT system on selected local variables. The reduction on the scale of sub-problems results in high efficiency to the methods. Our method is evaluated on both synthetic and real data sets and outperforms state-of-the-art benchmark methods.

Acknowledgments

This work was supported by the Hong Kong GRF 16207820 research grant.

References

Ahuja, R. K.; and Orlin, J. B. 2001. A fast scaling algorithm for minimizing separable convex functions subject to chain constraints. *Operations Research*, 49(5): 784–789.

Arnström, D.; and Axehill, D. 2019. Exact complexity certification of a standard primal active-set method for quadratic programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 4317–4324.

Asuncion, A.; and Newman, D. 2007. UCI machine learning repository.

Ayer, M.; Brunk, H. D.; Ewing, G. M.; Reid, W. T.; and Silverman, E. 1955. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 641–647.

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

Best, M. J.; Chakravarti, N.; and Ubhaya, V. A. 2000. Minimizing separable convex functions subject to simple chain constraints. *SIAM Journal on Optimization*, 10(3): 658–672.

Bonnans, J.-F.; Gilbert, J. C.; Lemaréchal, C.; and Sagastizábal, C. A. 2006. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media.

Burdakov, O.; and Sysoev, O. 2017. A dual active-set algorithm for regularized monotonic regression. *Journal of Optimization Theory and Applications*, 172(3): 929–949.

Cimini, G.; and Bemporad, A. 2017. Exact complexity certification of active-set methods for quadratic programming. *IEEE Transactions on Automatic Control*, 62(12): 6094–6109.

Deng, H.; Zhang, C.-H.; et al. 2020. Isotonic regression in multi-dimensional spaces and graphs. *Annals of Statistics*, 48(6): 3672–3698.

Durot, C. 2008. Monotone nonparametric regression with random design. *Mathematical Methods of Statistics*, 17(4): 327–341.

Feelders, A.; and Van der Gaag, L. C. 2006. Learning Bayesian network parameters under order constraints. *International Journal of Approximate Reasoning*, 42(1-2): 37–53.

Gjuvsland, A. B.; Wang, Y.; Plahte, E.; and Omholt, S. W. 2013. Monotonicity is a key feature of genotype-phenotype maps. *Frontiers in Genetics*, 4: 216.

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

Groeneboom, P.; and Jongbloed, G. 2014. *Nonparametric estimation under shape constraints*, volume 38. Cambridge University Press.

Grotzinger, S.; and Witzgall, C. 1984. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12(1): 247–270.

Guntuboyina, A.; Sen, B.; et al. 2018. Nonparametric shape-restricted regression. *Statistical Science*, 33(4): 568–594.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. *Proceedings of the 34th International Conference on Machine Learning*, 70: 1321–1330.

Han, Q.; Wang, T.; Chatterjee, S.; Samworth, R. J.; et al. 2019. Isotonic regression in general dimensions. *Annals of Statistics*, 47(5): 2440–2471.

Horowitz, J. L.; and Lee, S. 2017. Nonparametric estimation and inference under shape restrictions. *Journal of Econometrics*, 201(1): 108–126.

Hu, J.; Kapoor, M.; Zhang, W.; Hamilton, S. R.; and Coombes, K. R. 2005. Analysis of dose–response effects on gene expression data with comparison of two microarray platforms. *Bioinformatics*, 21(17): 3524–3529.

Hunter, D. R.; and Lange, K. 2004. A tutorial on MM algorithms. *The American Statistician*, 58(1): 30–37.

Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1): 1–27.

Kyng, R.; Rao, A.; and Sachdeva, S. 2015. Fast, provable algorithms for isotonic regression in all ℓ_p-norms. In *Advances in Neural Information Processing Systems*, 2719–2727.

Lim, C. H. 2018. An efficient pruning algorithm for robust isotonic regression. In *Advances in Neural Information Processing Systems*, 219–229.

Luss, R.; and Rosset, S. 2014. Generalized isotonic regression. *Journal of Computational and Graphical Statistics*, 23(1): 192–210.

Luss, R.; Rosset, S.; Shahar, M.; et al. 2012. Efficient regularized isotonic regression with application to gene–gene interaction search. *The Annals of Applied Statistics*, 6(1): 253–283.

Luss, R.; Rosset, S.; et al. 2017. Bounded isotonic regression. *Electronic Journal of Statistics*, 11(2): 4488–4514.

Mair, P.; Hornik, K.; and de Leeuw, J. 2009. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32(5): 1–24.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, 625–632.

Robertson, T. 1988. *Order restricted statistical inference*. Chichester: John Wiley & Sons.

Scutari, G.; Facchinei, F.; Song, P.; Palomar, D. P.; and Pang, J.-S. 2013. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Transactions on Signal Processing*, 62(3): 641–656.

Stout, Q. F. 2013. Isotonic regression via partitioning. *Algorithmica*, 66(1): 93–112.

Sun, Y.; Babu, P.; and Palomar, D. P. 2016. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3): 794–816.

Sysoev, O.; and Burdakov, O. 2019. A smoothed monotonic regression via ℓ_2 regularization. *Knowledge and Information Systems*, 59(1): 197–218.

Ubhaya, V. A. 1974. Isotone optimization. *Journal of Approximation Theory*, 12(2): 146–159.

Yu, Y.-L.; and Xing, E. P. 2016. Exact algorithms for isotonic regression and related. In *Journal of Physics: Conference Series*, volume 699, 1–9.