

# Weighted $L_\infty$ Isotonic Regression

Quentin F. Stout

Computer Science and Engineering  
University of Michigan  
Ann Arbor, MI 48109–2121

## Abstract

Algorithms are given for determining weighted  $L_\infty$  weighted isotonic regressions satisfying order constraints given by a directed acyclic graph (DAG) with  $n$  vertices and  $m$  edges. An algorithm is given taking  $\Theta(m \log n)$  time for the general case.  $L_\infty$  isotonic regressions are not unique, so we examine properties of the regressions an algorithm produces, in addition to the time it takes. An approach based on calculating prefix solutions is introduced, having better properties than the fastest algorithm in several respects. Prefix algorithms are used for determining isotonic and unimodal regressions over linear and tree orderings. Algorithms are also given for determining isotonic regressions when the values are constrained to a specified set of values, such as the integers, and for penalty functions that are not a metric.

**Keywords:** isotonic regression, monotonic,  $L_\infty$ , minimax, nonparametric, unimodal

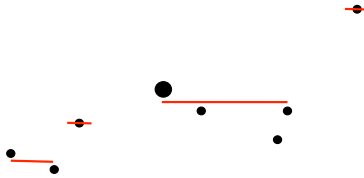
## 1 Introduction

A directed acyclic graph (DAG)  $G = (V, E)$  with  $n$  vertices and  $m$  edges defines a partial order over the vertices, where  $u \prec v$  if and only if there is a path from  $u$  to  $v$ . A function  $g$  on  $V$  is *isotonic* iff it is a weakly order-preserving mapping into the real numbers, i.e., iff for all  $u, v \in V$ , if  $u \prec v$  then  $g(u) \leq g(v)$ . By *weighted data* on  $G$  we mean a pair of real-valued functions  $(f, w)$  on  $V$ , where  $w$ , the *weights*, is non-negative, and  $f$ , the *values*, is arbitrary. Given weighted data  $(f, w)$  on  $G$ , an  $L_p$  *isotonic regression* is an isotonic function  $g$  on  $G$  that minimizes

$$\begin{aligned} & \left( \sum_{v \in V} w(v) \cdot |f(v) - g(v)|^p \right)^{1/p} & \text{if } 1 \leq p < \infty \\ & \max_{v \in V} w(v) \cdot |f(v) - g(v)| & p = \infty \end{aligned}$$

among all isotonic functions. The  $L_p$  *regression error* is the value of this expression. For  $1 < p < \infty$  the regression values are unique, but for  $L_1$  and  $L_\infty$  they may not be. For example, on the vertices  $\{1, 2, 3\}$ , if  $f = (3, 1, 2.5)$  and  $w = (2, 2, 1)$ , then  $L_1$  isotonic regressions are of the form  $g(3) = 2.5$  and  $g(1) = g(2) = x$  for  $x \in [1, 2.5]$ , while  $L_\infty$  isotonic regressions are of the form  $g(1) = g(2) = 2$  and  $g(3) \in [2, 4.5]$ . For  $1 < p < \infty$  the  $L_p$  isotonic regression is  $g(1) = g(2) = 2$ ,  $g(3) = 2.5$ . Figure 1 shows an example of an isotonic regression. Note that the regression values form level sets, and that the regression is undefined in some regions.

Shape-constrained regressions are of increasing importance as researchers reduce their assumptions concerning the underlying systems they are studying, and as computing power increases and algorithms improve. Isotonic regressions have long been applied to a wide range of problems in statistics [5, 15, 33, 40],



Dots represent data, size represents their weight, lines are regression values

Figure 1: An isotonic regression

optimization [6, 22, 26] and classification [11, 12, 37]. Recent applications involving learning from, and data mining on, large data sets [21, 28, 32, 41] and analyzing biological data from microarrays [4]. Rather than assuming a parametric form, such as a polynomial of specified degree, isotonic regressions merely assume that there is an underlying direction. For example, for the shrinkage of tumors being treated by radiation and chemotherapy, it might be assumed that at any given radiation level the shrinkage increases with dose, and at any given dose the shrinkage increases with radiation. However, there may be no assumptions about the ordering between a low dose and high radiation combination vs. a high dose and low radiation. In some settings the independent variables are ordered but do not have a metric, such as  $S < M < L < XL$  sizes.

Isotonic functions are also known as monotonic, monotonic increasing, or order-preserving functions, and the  $L_\infty$  metric is also known as supremum norm, chessboard distance, Chebyshev distance, uniform metric, minimax optimization, or bottleneck criterion.

We develop efficient algorithms for finding  $L_\infty$  isotonic regressions for weighted data.  $L_\infty$  isotonic is not unique and the algorithms have significantly different behavior in terms of the regressions they produce. Section 2 defines several isotonic regression algorithms and looks at properties such as being monotonic (defined in Section 2.2). Section 2.3 evaluates algorithms in terms of the number of vertices with large regression errors, looking beyond just minimizing the worst error. In Section 3 we show how to improve Kaufman and Tamir’s algorithm for general DAGs [22], resulting in one taking  $\Theta(m \log n)$  time. A prefix approach is then introduced which directly determines regression values as opposed to the indirect search used in Section 3, resulting in better behavior. In Section 5 it is used to find isotonic and unimodal regressions on linear and tree orders.

Section 6 considers isotonic regression where the regression values are restricted to a specific set, such as the integers. It is shown that in  $\Theta(m)$  time one can convert an unrestricted regression into a restricted one, or, if the set is small, one can quickly determine the restricted regression without starting with an unrestricted one. Section 7 discusses extensions where the  $L_\infty$  metric is replaced by more general penalty functions, such as those arising in some classification problems. Section 8 contains some final remarks.

## 2 Background

Throughout we assume that  $G$  is a single connected component, and hence  $m \geq n - 1$ . If it has more than one component then isotonic regressions can be found for each component separately.

The complexity of determining an isotonic regression depends on the regression metric and the partially ordered set. For example, when the order is a linear order, it is well-known that a simple left-right scanning approach using pair adjacent violators, PAV, can be used to determine the  $L_2$  isotonic regression in  $\Theta(n)$  time,  $L_1$  in  $\Theta(n \log n)$  time, and  $L_\infty$  on unweighted data in  $\Theta(n)$  time. In Section 5 an algorithm taking

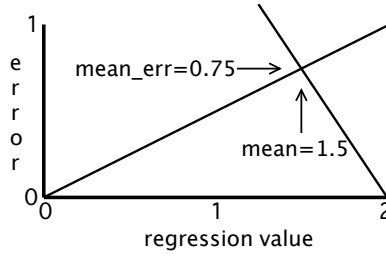


Figure 2: The mean and mean\_err of (0, 0.5) and (2, 1.5)

$\Theta(n \log n)$  time is given for  $L_\infty$  isotonic regression on weighted data.

For a general partial order the fastest known algorithm for  $L_1$ , due to Angelov, Harb, Kannan, and Wang [4], takes  $\Theta(nm + n^2 \log n)$  time. For  $L_2$  the algorithm of Maxwell and Muckstadt [26], with a modest correction by Spouge, Wan, and Wilber [34], takes  $\Theta(n^4)$  time. For sparse graphs this can be reduced to  $\Theta(n^2m + n^3 \log n)$  [37]. For  $L_\infty$  the best previously known is  $\Theta(n \log^2 n + m \log n)$ , due to Kaufman and Tamir [22]. Theorem 1 shows that this can be reduced to  $\Theta(m \log n)$ .

Given weighted data  $(f, w)$  on DAG  $G = (V, E)$ , for vertex  $v \in V$ , the *error of using  $r$  as the regression value at  $v$* ,  $\text{err}(v, r)$ , is  $w(v) \cdot |f(v) - r|$ . Given vertices  $u$  and  $v$ , the weighted mean of their values,  $\text{mean}(f, w : u, v)$ , is  $[w(u)f(u) + w(v)f(v)] / [w(u) + w(v)]$ . Note that  $\text{mean}(f, w : u, v)$  has equal error for the two vertices. Let  $\text{mean\_err}(f, w : u, v)$  denote this error.

There is a simple geometric interpretation of mean and mean\_err: when regression values are plotted horizontally and the error is plotted vertically, the ray with x-intercept  $f(u)$  and slope  $-w(u)$  gives the error for using a regression value below  $f(u)$  at  $u$ , and the ray with x-intercept  $f(v)$  and slope  $w(v)$  gives the error for using a regression value above  $f(v)$  at  $v$ . The regression value where these lines intersect is  $\text{mean}(f, w : u, v)$ , and the error of the intersection point is  $\text{mean\_err}(f, w : u, v)$ . See Figure 2.

For  $u, v \in V$ , if  $u \preceq v$  and  $f(u) \geq f(v)$  then  $u, v$  are a (weakly) *violating pair* and  $\text{mean\_err}(f, w : u, v)$  is a lower bound on the regression error. The only way the error at  $u$  can be less than  $\text{mean\_err}(f, w : u, v)$  is if the regression value is larger than their mean. However,  $u \prec v$  implies that the regression value at  $u$  is  $\leq$  that at  $v$ , so the regression at  $v$  would have an error larger than the error for the mean. More generally, given a set  $S \subset V$ , the  $L_\infty$  mean of the weighted data on  $S$  is  $\text{mean}(f, w : u', v')$ , where

$$\text{mean\_err}(f, w : u', v') = \max\{\text{mean\_err}(f, w : u, v) : u, v \in S\}$$

A straightforward approach to calculating this would take  $\Theta(|S|^2)$  time. In Section 4 it is shown that this can be reduced to  $\Theta(|S| \log |S|)$ .

## 2.1 Regression Mappings

$L_\infty$  isotonic regression is not always unique, so we examine properties of the regressions produced by various algorithms. An  $L_\infty$  *isotonic regression mapping*, informally called a regression mapping, takes a DAG  $G = (V, E)$  and weighted function  $(f, w)$  on  $G$  and maps them to an  $L_\infty$  isotonic regression of  $(f, w)$ . In this section we present several regression mappings.

### 2.1.1 Basic and Prefix

A simple approach to computing a regression value at vertex  $x$  is to find a pair of vertices  $u', v'$  that maximize  $\text{mean\_err}(f, w : u, v)$  among all violating pairs  $u \preceq x \preceq v$ , and then use  $\text{mean}(f, w : u', v')$  as the regression value. (If there are two such pairs then they have the same mean.) The values determined this way are easily seen to be isotonic, and the regression error is  $\max\{\text{mean\_err}(f, w : u, v) : u \preceq v, f(u) \geq f(v)\}$ , which is optimal. This regression mapping, denoted Basic, has been studied at least to the 1970's [40].

We introduce a closely related regression, denoted Prefix, defined as follows: let

$$\begin{aligned} \text{pre}(v) &= \max\{\text{mean}(f, w : u, v) : u \preceq v \text{ and } f(u) \geq f(v)\} \\ \text{Prefix}(u) &= \min\{\text{pre}(v) : u \preceq v\} \end{aligned}$$

For any  $x \in V$ ,  $\text{pre}(x) = \text{Basic}_x(f, w)(x)$ , where  $\text{Basic}_x$  is the Basic isotonic regression on the DAG induced by the vertices  $V_x = \{v \in V : v \preceq x\}$ . The  $L_\infty$  isotonic regression error on  $V_x$  is  $\max\{\text{err}(v, \text{pre}(v)) : v \in V_x\}$ . The construction of Prefix guarantees that it is isotonic, and optimality follows from that of Basic.

For unweighted data the calculation for Basic reduces to  $(\max\{f(u) : u \preceq x\} + \min\{f(v) : x \preceq v\})/2$  and thus it can be determined in  $\Theta(m)$  time by using topological sorting and reverse topological sorting. Prefix also can be computed in  $\Theta(m)$  time.

For weighted data, suppose the transitive closure of  $G$  is given. It is straightforward to compute Prefix in  $\Theta(n^2)$  time since the calculation of  $\text{pre}(v)$  is linear in the number of predecessors of  $v$ . In contrast, Basic at  $v$  involves pairs of predecessors and successors, and hence a straightforward computation at  $v$  could take  $\Theta(n^2)$  time and  $\Theta(n^3)$  overall. As noted in [4, 22], the calculations at  $v$  can be made linear in the number of predecessors and successors by using techniques developed by Megido [27], giving  $\Theta(n^2)$  total time. However, this is significantly more complicated than determining Prefix.

Section 5 shows that Prefix can be efficiently calculated for some important DAGs, such as linear and tree orderings, without using the transitive closure. In addition, it easily extends to more general penalty functions, discussed in Section 7. Penalty functions need not be linear, and hence the linear programming approach in Megido does not seem to apply, complicating the computation of Basic.

### 2.1.2 Min, Max, and Avg

The Min  $L_\infty$  isotonic regression of weighted data  $(f, w)$  is the pointwise minimum of all  $L_\infty$  isotonic regressions of  $(f, w)$ , and Max is the pointwise maximum. The pointwise minimum of isotonic functions is isotonic, and since at any vertex Min has an error the same as at least one  $L_\infty$  isotonic regression its maximum error is optimal. Hence it, and similarly Max, is an  $L_\infty$  isotonic regression. Avg is the pointwise average of Min and Max. It is easy to show that for unweighted data  $\text{Avg} = \text{Basic}$ , but this need not hold for weighted data. The fastest  $L_\infty$  isotonic regression algorithm computes Min, Max, and Avg, see Section 3.

While Min and Max can be rapidly computed, their behavior is not always desirable. This is discussed in Sections 2.2 and 2.3.

### 2.1.3 Strict

The *strict  $L_\infty$  isotonic regression* (Strict) is the limit, as  $p \rightarrow \infty$ , of  $L_p$  isotonic regression. The limit exists for all weighted data and DAGs, and the limit process is known as the ‘‘Polya algorithm’’ [31]. It has been called the ‘‘best best’’  $L_\infty$  isotonic regression [23]. Algorithms to compute Strict, and proofs of some of its properties, appear in [36].

## 2.2 Monotonic Regression Mappings

There is a natural pointwise ordering on functions, denoted  $\leq^p$ , where  $f \leq^p g$  iff  $f(v) \leq g(v)$  for all  $v \in V$ . A regression mapping  $M$  is *monotonic* iff for any DAG  $G$  and weight function  $w$  on  $G$ , for functions  $f$  and  $g$  on  $G$ , if  $f \leq^p g$  then  $M(f, w) \leq^p M(g, w)$ . For isotonic regression, monotonicity is a natural property to expect. It is straightforward to show that for  $1 < p < \infty$ ,  $L_p$  isotonic regressions are monotonic mappings, and thus Strict is as well. Basic and Prefix are also monotonic, as can be seen from their construction.

For unweighted data Avg = Basic, and hence it is monotonic, but for weighted data it isn't. E.g., for vertices  $\{1, 2, 3\}$ , let  $w = (4, 4, 1)$ ,  $f = (2, 0, 2)$ , and  $g = (3, 3, 3)$ . Then  $f <^p g$ , but  $\text{Avg}(f) = (1, 1, 3.5) \not\leq^p (3, 3, 3) = \text{Avg}(g)$ . Min and Max are not monotonic even for unweighted data. E.g., if  $f = (1, 3, 1)$  and  $g = (1, 3, 3)$  then  $f \leq^p g$  but  $\text{Max}(f) = (2, 2, 2) \not\leq^p (1, 3, 3) = \text{Max}(g)$ . Similar examples hold for Min.

## 2.3 Large Regression Errors

Let  $G$  be a DAG and  $(f, w)$  weighted data on  $G$ . While all  $L_\infty$  isotonic regressions have the same maximum error, regressions can differ considerably in the number of vertices they have with large regression errors.

One of the important properties of Strict is that it minimizes large errors [36]. That is, if  $g \neq \text{Strict}(f, w)$  is an isotonic function on  $G$ , then there is a  $C > 0$  such that  $g$  has more vertices with regression error  $\geq C$  than does  $\text{Strict}(f, w)$ , and for any  $D > C$ ,  $g$  and  $\text{Strict}(f, w)$  have the same number of vertices with regression error  $\geq D$  (it is not necessarily true that if  $c < C$  then Strict has fewer vertices than  $g$  with regression error  $\geq c$ , but the concern is about the number of large errors). For example, for the unweighted function  $(3, 1, 2)$ , Strict is  $(2, 2, 2)$  while Basic and Prefix are  $(2, 2, 2.5)$ , i.e., their last value has an unnecessary error. Max is even worse, being  $(2, 2, 3)$ .

Let  $D$  be the  $L_\infty$  isotonic regression error. Prefix, Basic and Avg have a weaker property: if any of them has regression error  $D$  at vertex  $v$ , then all  $L_\infty$  isotonic regressions have the same value at  $v$ . For Avg this is obvious. For Basic and Prefix, the definition of Basic insures that the error at  $v$  is  $\leq \max\{\text{mean\_err}(f, w : x, y) : x \preceq v \preceq y\}$ , and the same is true for Prefix. This is a lower bound on the larger of the regression errors at  $x$  and  $y$ , and hence if the regression error is  $D$  then any isotonic function having a different value would have larger regression error at  $x$  or  $y$  and would not be an  $L_\infty$  isotonic regression.

Min and Max are at the opposite end of the spectrum, in that for any vertex  $v$ , one or both of them has the largest regression error at  $v$  among all  $L_\infty$  isotonic regressions.

## 3 Isotonic Regression for Arbitrary Partial Orders

As mentioned above, for arbitrary DAGs the Prefix and Basic regressions can be calculated in  $\Theta(n^2)$  time if the transitive closure is given. If it isn't given, then the fastest approach known is to first determine it. For a DAG with  $n$  vertices and  $m$  edges the transitive closure can be determined in  $\Theta(\min\{nm, n^\omega\})$  time, where  $\omega$  is such that matrix multiplication can be performed in  $O(n^\omega)$  time. It is known that  $\omega < 2.376$ . For Strict, given the transitive closure, the fastest known algorithm takes  $\Theta(n^2 \log n)$  time [36].

A faster algorithm was provided by Kaufman and Tamir [22], taking  $\Theta(n \log^2 n + m \log n)$  time without needing the transitive closure. It is based on the inverse problem of determining if there is a regression with error  $\epsilon$ . They decide this using a feasibility test that takes  $\Theta(n \log n + m)$  time. Here it will be decided by using topological sorting, taking  $\Theta(m)$  time. We will show:

**Theorem 1** *Given weighted data  $(f, w)$  on a DAG  $G = (V, E)$ , Algorithm A determines the Min  $L_\infty$  isotonic regression in  $\Theta(m \log n)$  time. Similarly, Max, and hence Avg, can be determined in the same time.*

```

initialize  $\epsilon$ 
loop {parametric search determines new  $\epsilon$  values}
  if Min fits through the windows then
    if  $\epsilon$ -windows tight then exit
    else decrease  $\epsilon$ 
  else increase  $\epsilon$ 
end loop

```

---

Algorithm A: Determining Min  $L_\infty$  isotonic regression for an arbitrary DAG

---

A *window* at  $v$  is a real-valued interval  $[a_v, b_v]$ . Given a set of windows  $W = \{[a_v, b_v] : v \in V\}$ , an *isotonic function through  $W$*  is an isotonic function  $g$  for which  $a_v \leq g(v) \leq b_v$  for all  $v \in V$ .

*Observation:* A set of windows  $W = \{[a_v, b_v]\}$  has an isotonic function through them if and only if the following is an isotonic function through  $W$ :  $h(v) = \max\{a_u : u \preceq v\}$ .

The isotonic constraint forces  $h(v)$  to be at least this large, and thus if it is not an isotonic function through  $W$  then there must be an  $v$  for which  $h(v) > b_v$ . Since  $h(v)$  is the smallest possible value at  $v$  among all isotonic functions through  $W$ , this implies that there is no isotonic function through  $W$ . The  $h$  values can be computed via topological sorting, and thus, given a set of windows, in  $\Theta(m)$  time it can be decided if there is an isotonic function through them.

A set of windows is *tight* if there is an isotonic function through them and for at least one vertex  $v$ , the above procedure results in  $h(v) = b_v$ . Given any  $\epsilon > 0$ , the set of  $\epsilon$ -*windows* is the set of windows where  $w(v)(f(v) - a_v) = w(v)(b_v - f(v)) = \epsilon$ . The minimum  $L_\infty$  regression error among isotonic functions is the  $\epsilon$  for which the  $\epsilon$ -windows are tight, and an isotonic function is an  $L_\infty$  isotonic regression iff it fits through these windows. At any vertex  $h$  has the minimum possible value among all isotonic functions fitting through the windows, and hence is Min.  $\text{Max}(x) = \min\{b_v : v \succeq x\}$ .

*Proof of Theorem 1:* As noted in Section 2, the optimal  $\epsilon$  is of the form  $\text{mean\_err}(f, w : u, v)$  for some  $u, v \in V$ . Kaufman and Tamir utilized parametric search among these values to locate determine the values of  $\epsilon$  considered. This results in a logarithmic number of iterations, so the total time is proportional to  $\log n$  times the time to check a given value of  $\epsilon$ . We have shown that this can be done in  $\Theta(m)$  time.  $\square$

Linear and rooted tree orderings have  $m = \Theta(n)$  and hence for them Algorithm A is faster than Kaufman and Tamir's algorithm. Another important class with such sparsity are  $d$ -dimensional grids. That is, given  $d$  positive integers  $n_1, \dots, n_d$ , let  $V$  be all points in  $d$ -dimensional space of the form  $(i_1, \dots, i_d)$ , where  $i_j \in \{1, \dots, n_j\}$  for all  $1 \leq j \leq d$ .  $V$  has  $n = \prod_{i=1}^d n_i$  points. If  $p = (p_1, \dots, p_d)$  and  $q = (q_1, \dots, q_d)$ , then  $p \preceq q$  iff  $p_i \leq q_i$  for all  $1 \leq i \leq d$ . Note that this ordering can be represented by a standard grid where vertex  $p$  has an edge to all points of the form  $q = (q_1, \dots, q_d)$  where  $p \preceq q$  and  $\sum_{i=1}^d q_i - p_i = 1$ . This DAG has  $\Theta(dn)$  edges. Isotonic regression on  $d$ -dimensional points, which corresponds to having  $d$  independent variables, has been extensively studied [5, 7, 15, 33, 34, 37].

**Corollary 2** *Given weighted data  $(f, w)$  on a linear, tree, or  $d$ -dimensional grid ordering, the Min, Max and Avg  $L_\infty$  isotonic regressions can be determined in  $\Theta(n \log n)$  time, where for  $d$ -dimensional grids the implied constant is a function of  $d$ .  $\square$*

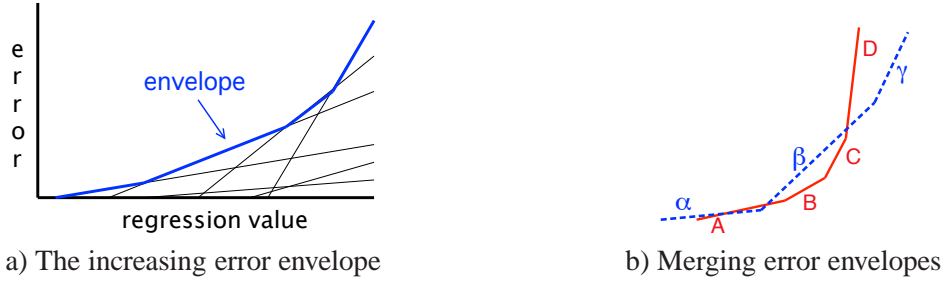


Figure 3: Error envelopes

## 4 Error Envelopes

We will exploit properties of intersecting line segments to find maximum violating pairs.

Given data  $(f, w)$  on a set  $S$ , for  $x \geq \min\{f(v) : v \in S\}$  let  $g(x)$  be the  $L_\infty$  regression error of using  $x$  as the regression value for  $\{v \in S : f(v) \leq x\}$ . This has a simple geometric interpretation: for each  $v \in S$ , plot the ray with x-intercept  $f(v)$  and slope  $w(v)$ .  $g$  is the set of line segments with no points above them. See Figure 3 a). We call this upper envelope the *increasing error envelope*, and  $\max\{\text{err}(v, x) : v \in S, f(v) \geq x\}$  is the *decreasing error envelope*. If the envelopes intersect at regression value  $r$ , then  $r$  is the  $L_\infty$  mean of  $S$ .

It is straightforward to use balanced search trees, ordered by slope, so that the increasing and decreasing error envelopes can be maintained and in  $\Theta(\log |S|)$  time one can insert a new weighted point, determine the value of the envelope at a given regression value, determine the regression value having a given error, and determine the point of intersection of a ray with an envelope. Deleting points is more difficult and discussed in Section 2.1.3. Given the error envelopes it is trivial to determine their intersection in  $\Theta(|S|)$  time and thus one can determine the mean in  $\Theta(|S| \log |S|)$  time.

A slight variation on this will prove useful later. Given the decreasing error envelope  $E$  of a set  $S$  of vertices, and given a vertex  $v$ , an *error query of  $E$*  determines the intersection of the error envelope with the ray with x-intercept  $f(v)$  and slope  $w(v)$ , i.e., the ray representing the error in using a regression value  $\geq f(v)$  at  $v$ . The maximum error of these queries is the regression error of using the  $L_\infty$  mean on  $S$ .

Another operation involving envelopes is merger. The only special aspect is that after two envelopes are merged, using standard merging of balanced binary search trees, some of the segments may need to be removed. E.g., in Figure 3 b), when the envelope with Roman labeling and the one Greek labeling are merged, segments B, C, and  $\gamma$  will be removed. The endpoints of some of the remaining segments also change. After the merger the removal can be performed in time linear in the size of the smaller envelope plus the number of segments removed as follows: for all segments from the smaller envelope (Greek), determine if they should be removed or which (if any) of the segments of the other envelope should be removed. The process initiated by a Greek segment starts by examining its closest predecessor and closest successor in the Roman ordering. E.g., in the figure, the merger will initially have the segments in the order  $\alpha, A, B, \beta, \gamma, C, D$  (recall that they are stored in increasing order of their slope). If one considers  $\beta$ , then it is above the line segment in the Roman envelope which immediately precedes it in the ordering (B) and B is removed. Then the predecessor of B in the Roman envelope is examined, but  $\beta$  is not over all of it so A remains. Then the successor of B is examined, and it too is removed. Then its successor (D) is examined, and the process initiated by  $\beta$  is completed. (The process initiated by  $\alpha$  immediately terminates). Then  $\gamma$  is examined. The

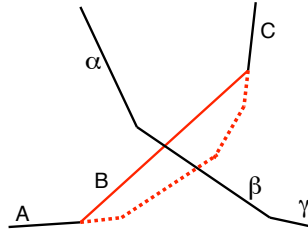


Figure 4: Dynamic envelope intersections

remaining Roman immediate predecessor is A, which remains, and the immediate successor is D, which dominates  $\gamma$  and hence  $\gamma$  is removed.

Given sets  $S_1, S_2$  of vertices, a *bipartite maximum violators* operation determines a pair of vertices  $u_1 \in S_1, u_2 \in S_2$  that maximize  $\text{mean\_err}(f, w : v_1, v_2)$  among all pairs where  $v_1 \in S_1, v_2 \in S_2$  and  $f(v_1) \geq f(v_2)$ . If there are no such pairs then it returns the empty set. Note that  $S_1$  and  $S_2$  might overlap, in which case the operation does not correspond to an acyclic ordering.

**Lemma 3 (Repeated Violators)** *Given sets  $S_1$  and  $S_2$  of weighted values, in  $\Theta(N \log N)$  time one can do an arbitrary sequence of  $O(N)$  deletion and bipartite maximum violators operations, where  $N = |S_1| + |S_2|$ .*

*Proof:* A decreasing error envelope for  $S_1$ , and an increasing one for  $S_2$ , will be used to determine the violators. The envelopes will be threaded so that one can determine the segments adjacent to a given segment in constant time. They are “semi-dynamic” since only deletions change them once they are initially constructed. Hershberger and Suri [20] showed how to create the initial envelopes, and perform all the deletions, in  $O(N \log N)$  time (they solved the semi-dynamic problem of maintaining the convex hull of a set of planar points which, by duality, can be used to solve the error envelope problem).

Thus the only time remaining to be analyzed is the time needed to find the maximum violators. Given the envelopes, the initial maximum violators can easily be found in  $\Theta(N)$  time. For subsequent violators, the only case of interest is when the segment deleted was one of the ones in the intersection. For example, in Figure 4, suppose initially B and  $\beta$  intersect, and then B is deleted, resulting in new segments in the increasing error envelope represented by dashed lines. The new intersection can involve  $\beta$  or a successor, but not  $\alpha$ . Further, on the increasing envelope it might involve C since that segment may have lengthened, but cannot involve any successor of C. It might, however, involve some predecessor of C. The intersection is found by checking C, then its predecessor, etc., until it is found. During this process it may also be discovered that the intersection will involve a successor of  $\alpha$ , etc.

While a single deletion may take  $\Theta(N)$  time, if the new intersection does not involve C then no future intersection can involve C until its predecessor has been deleted. Therefore the total number of segments examined over all iterations is  $O(N)$ .  $\square$

The fact that only the semi-dynamic scenario is needed, without any insertions, is fortunate since it is not known how solve the fully dynamic convex hull problem in  $O(N \log N)$  time.

## 5 Linear and Tree Orderings

Isotonic regression on linear and tree orders has been used for decades. Recent applications of isotonic regression on trees include problems involving web data [9, 28]. For linear orders it is widely known that

$\{\text{DecErrEnv}(v)$  is the decreasing error envelope of  $\{u : u \preceq v\}$  for  $v \in V\}$

Using a topological sort traversal,

$\text{DecErrEnv}(v) = \cup\{\text{DecErrEnv}(u) : (u, v) \in E\}$

Insert  $(f(v), w(v))$  into  $\text{DecErrEnv}(v)$

$\text{pre}(v) =$  result of error query of  $\text{DecErrEnv}(v)$

Using topological sort traversal in reverse order,

$\text{Prefix}(v) = \min\{\text{pre}(u) : v \preceq u\}$

Algorithm B: Computing Prefix using topological sorting on DAG  $G = (V, E)$

---

the  $L_1$  and  $L_2$  isotonic regressions can easily be found in  $\Theta(n \log n)$  and  $\Theta(n)$  time, respectively. For rooted trees, where the ordering is towards the root, the fastest isotonic regression algorithms use a bottom-up pair adjacent violators (PAV) approach [38], taking  $\Theta(n \log n)$  time for the  $L_1$  [37] and  $L_2$  [29] metrics.

Corollary 2 shows that for the  $L_\infty$  metric, Min, Max and Avg isotonic regression on a tree can be determined in  $\Theta(n \log n)$  time. Here we show how to compute Prefix in the same time. These  $L_\infty$  algorithms do not use PAV, but the fastest known for Strict does [36].

**Theorem 4** *Given weighted data  $(f, w)$  on a tree  $T = (V, E)$ , Algorithm B computes the Prefix  $L_\infty$  isotonic regression in  $\Theta(n \log n)$  time.*

*Proof:* Envelope merging was described in Section 4. Using a careful merge, all of the initial union operations, including insertions, can be performed in  $\Theta(n \log n)$  time [8]. The time of the segment removal during each merge is linear in the number of elements in the smaller envelope, plus the time needed to remove each of the intervals. An interval is removed at most once throughout all of the mergers, so the total time for removals is  $O(n \log n)$ , and since the processes are initiated by segments in the smaller tree, the total time for the initiation is  $O(n \log n)$ . This completes the proof of Theorem 4.  $\square$

## 5.1 Unimodal Regression

Given a weighted function on a linear order  $v_1 < \dots < v_n$ , a *unimodal regression*  $g$  is a regression that minimizes the regression error subject to the unimodal constraint

$$g(v_1) \leq g(v_2) \dots \leq g(v_k) \geq g(v_{k+1}) \dots \geq g(v_n)$$

for some  $k$ ,  $1 \leq k \leq n$ . Unimodal orderings are also known as umbrella orderings. Optimal modes may not be unique, which can be seen by considering the unweighted function  $(1, 0, 1)$ . Several unimodal regression algorithms have appeared [10, 30, 35, 39] and applied to problems such as dose-response [13] and tree growth [17]. For example, in dose-response settings with “competing failure modes”, as the dose increases the efficacy presumably increases, but toxicity increases as well. The goal is to find a dose that maximizes the probability of being both efficacious and nontoxic.

A prefix approach was used in [35] to find the unimodal regression of a linear order for the  $L_2$  metric in  $\Theta(n)$  time, for the  $L_1$  metric in  $\Theta(n \log n)$  time, and the  $L_\infty$  metric with unweighted data in  $\Theta(n)$  time. For the prefix approach the information that needs to be computed at each vertex  $v_k$  is the regression error of an isotonic regression on  $v_1 \dots v_k$ . This is the maximum of the error of using  $\text{pre}(i)$  at  $v_i$ ,  $1 \leq i \leq k$ , which

can be computed as pre is being computed. Once this is known, along with the corresponding information for decreasing isotonic regressions on  $v_k \dots v_n$ , an optimal mode vertex for  $L_\infty$  is one that minimizes the largest of these two errors. Thus, using Algorithm B gives:

**Corollary 5** *Given weighted data  $(f, w)$  on a linear order of  $n$  vertices, an  $L_\infty$  unimodal regression can be determined in  $\Theta(n \log n)$  time.  $\square$*

This result was also obtained by Lin et al. [24].

The properties of unimodal regression are slightly different than for isotonic regression. For example, as the unweighted data  $(1, 0, 1)$  shows, there need not be a unimodal regression corresponding to Max, i.e., one which is pointwise as large as any other.

For an undirected tree the unimodal problem is to find an optimal root and the isotonic regression on the resulting rooted tree. Agarwal, Phillips and Sadri [1] gave an example motivating interest in this problem, and showed that the  $L_2$  unimodal regression of a tree with unweighted data can be found in  $\Theta(n \log^3 n)$  time even when a Lipschitz condition is imposed. Yu and Mannor [42] gave algorithms for locating the mode of a tree for a multi-arm bandit model with a bandit at each vertex, where the expected reward is unimodal. Given the observations at the vertices, unimodal regression can be used to estimate the location of the mode and decide which vertex to sample from next. This is similar to the use of unimodal regression in dose-response settings with competing failure modes [13, 18]. Unimodal bandit models have been examined at least since the 1980's [19].

For the  $L_\infty$  metric and unweighted data, unimodal regression on a tree is quite easy.

**Proposition 6** *Given unweighted data  $f$  on an undirected tree  $T = (V, E)$ , in  $\Theta(n)$  time an  $L_\infty$  unimodal regression can be found.*

*Proof:* Let  $x$  be a vertex where  $f$  achieves its maximum and let  $g$  be an optimal  $L_\infty$  unimodal regression on  $T$ . Let  $g'$  be the function on  $T$  given by  $g'(v) = \min\{g(v), g(x)\}$  for all  $v \in V$ . If  $\epsilon = |f(x) - g(x)|$ , then  $\|f - g\|_\infty \geq \epsilon$ . Since  $f(v) \leq f(x)$  for all  $v \in V$ ,  $|g'(v) - f(v)| \leq \max\{\epsilon, |g(v) - f(v)|\}$ . Thus  $\|g' - f\|_\infty \leq \|g - f\|_\infty$ , and, since  $g$  was optimal, so is  $g'$ . Further,  $g'$  is isotonic when the tree is directed with  $x$  as its root.

Thus, to find a unimodal regression it suffices to locate a vertex with maximum data value, orient the tree with this as the root, and use Prefix to determine the isotonic regression.  $\square$

For weighted functions it is not immediately obvious which vertices can be optimal roots, but there are edges for which one can quickly determine their correct orientation. The *path from  $u$  to  $v$* , for  $u, v \in V$ , means the simple path from  $u$  to  $v$ . For the weighted data  $(f, w)$  on the undirected tree  $T = (V, E)$ , let  $u, v$  be such that  $\text{mean\_err}(f, w : u, v) = \max\{\text{mean\_err}(f, w : p, q) : p, q \in V\}$ . Suppose  $f(u) < f(v)$ . If in the directed tree  $u$  is on the path from  $v$  to the root then they are violators and the regression error is at least  $\text{mean\_err}(f, w : u, v)$ . This is the error of the regression where all values are  $\text{mean}(f, w : u, v)$ , and hence every vertex is an optimal root. To obtain a smaller regression error it must be that  $u$  is not on the path from  $v$  to the root of the directed tree, in which case the edge incident to  $u$  on the simple path from  $u$  to  $v$  is directed towards the root. This will be used to show the following:

**Theorem 7** *Given weighted data  $(f, w)$  on an undirected tree  $T = (V, E)$ , in  $\Theta(n \log n)$  time an  $L_\infty$  unimodal regression can be found.*

*Proof:* The algorithm proceeds via recursion. Locate a pair of vertices  $u, v$  which maximize  $\text{mean\_err}$ . If  $f(u) < f(v)$  then let  $e$  be the edge incident to  $u$  on the path towards  $v$ . Remove  $e$ , which partitions  $T$  into trees  $T_1 = (V_1, E_1), T_2 = (V_2, E_2)$ , where  $u \in V_1$ . Recursively find an optimum root  $r$  for a unimodal regression of  $T_2$ . Lemma 8 proves that  $r$  is an optimal root for unimodal regression on  $T$ . Once an optimal root is known the Prefix isotonic regression can be determined in  $\Theta(n \log n)$  time.

To find  $u, v$ , set  $S_1 = S_2 = V$  and perform a bipartite maximum violator operation. Once  $T_1$  is determined, all weighted values corresponding to it are removed from  $S_1$  and  $S_2$  and the resulting sets are used for the subproblem on  $T_2$ . The repeated violators lemma (Lemma 3) shows that one can find pairs of maximum violators during all recursive steps in  $\Theta(n \log n)$  total time.

Lemma 9 shows that, given  $u$  and  $v$ , one can find  $e$  in  $\Theta(|V_1|)$  time. Using this,  $T_1$  can be removed from  $T$ , creating  $T_2$ , in  $\Theta(|V_1|)$  time. Thus the total time for recursively shrinking the subtrees is  $\Theta(n)$ .  $\square$

**Lemma 8** *For an undirected tree  $T = (V, E)$  with weighted data  $(f, w)$ , let  $u_1, u_2$  be a pair which maximizes  $\text{mean\_err}$  and for which  $f(u_1) < f(u_2)$ . Let  $\{u_1, x\}$  be an edge on the path from  $u_1$  to  $u_2$ . Let  $T_1 = (V_1, E_1), T_2 = (V_2, E_2)$  be the subtrees of  $T$  induced by removing this edge, where  $u_1 \in T_1$  and  $x \in T_2$ . Then an optimal root for  $L_\infty$  unimodal regression of  $(f, w)$  restricted to  $T_2$  is an optimal root for  $L_\infty$  unimodal regression of  $(f, w)$  on  $T$ .*

*Proof:* If the minimal regression error of a unimodal regression on  $T$  is  $\text{mean\_err}(f, w : u_1, u_2)$  then all vertices are optimal roots. Otherwise the regression value at  $u_2$  must be  $> \text{mean}(f, w : u_1, u_2)$  and at  $u_1$  is  $< \text{mean}(f, w : u_1, u_2)$ , and therefore any optimal root is in  $T_2$ . Let  $r$  be an optimal root of unimodal regression on  $T_2$ , with  $g_r$  being an isotonic regression on  $T_2$  with  $r$  as its root. Let  $s$  be an optimal root of unimodal regression on  $T$ , with  $h_s$  being an isotonic regression on  $T$  with  $s$  as its root. Define  $h_r$  on  $T$  as follows:

$$h_r(v) = \begin{cases} h_s(v) & \text{if } v \in T_1 \\ \max\{h_s(u_1), g_r(v)\} & \text{if } v \text{ is on the path from } r \text{ to } x \text{ or the path from } r \text{ to } u_2 \\ g_r(v) & \text{otherwise} \end{cases}$$

Since  $g_r$  is isotonic with root  $r$  and  $h_s$  is isotonic with root  $s$ ,  $h_r$  is isotonic on  $T$  with root  $r$ . To show that  $h_r$  is an optimal unimodal regression of  $T$ , since the regression error of  $g_r$  on  $T_2$  is no more than the regression error of  $h_s$  on  $T$ , it suffices to show that pointwise  $h_r$  is always equal to  $g_r$  or  $h_s$  or between their values. Only vertices on the paths from  $r$  to  $x$  or  $r$  to  $u_2$  are in doubt.

If  $h_s$  has a regression error  $< \text{mean\_err}(f, w : u_1, u_2)$  then  $h_s(u_2) > \text{mean}(f, w : u_1, u_2) > h_s(u_1)$ . Since  $h_s$  is non-increasing on the path from  $s$  to  $u_2$ , its value on all these vertices is  $> h_s(u_1)$  and on the path from  $s$  to  $u_1$   $h_s$  is nonincreasing and is always  $\geq h_s(u_1)$ .  $T$  is a tree and hence there is a unique path from  $u_2$  to  $u_1$ , and all edges on this path are on the path from  $s$  to  $u_2$  or the path from  $s$  to  $u_1$ . Thus on the path from  $u_2$  to  $u_1$ , all  $h_s$  values are  $\geq h_s(u_1)$ , and therefore for any vertex  $v$  on this path,  $h_r(v) \in [\min\{g_r(v), h_s(v)\}, \max\{g_r(v), h_s(v)\}]$ .

$g_r$  also has regression error  $< \text{mean\_err}(f, w : u_1, u_2)$  and hence  $g_r(u_2) > \text{mean}(f, w : u_1, u_2) > h_s(u_1)$ .  $g_r$  is nonincreasing on the path from  $r$  to  $u_2$  with all values  $\geq g_r(u_2)$ , and hence  $h_r = g_r$  on this path. As before, all vertices on the path from  $r$  to  $x$  are in the path from  $r$  to  $u_2$ , where  $h_r = g_r$ , or the path from  $u_2$  to  $u_1$ . Therefore for any vertex  $v$  in  $T_2$ ,  $h_r(v) \in [\min\{g_r(v), h_s(v)\}, \max\{g_r(v), h_s(v)\}]$ , as was to be shown.  $\square$

**Lemma 9** *Given an undirected tree  $T = (V, E)$  and vertices  $u, v \in V, u \neq v$ , let  $x \in V$  be such that the edge  $\{u, x\}$  is on the path from  $u$  to  $v$ . Then  $x$  can be determined in  $O(|V'|)$  time where  $V' \subset V$  is the set of vertices in the maximal subtree of  $T$  containing  $u$  but not containing  $x$ .*

*Proof:* Let  $T'$  denote the maximal subtree of  $T$  containing  $u$  but not containing  $x$ . Let  $p_1, \dots, p_k$  be the neighbors of  $u$ . Viewing  $u$  as the root, let  $T_i$  denote the subtree with root  $p_i, i = 1, k$ . On each  $T_i$  a preorder traversal will be used. A global traversal is performed as follows: visit the first node in the traversal of  $T_1$ , then the first node in the traversal of  $T_2, \dots$ , first node in  $T_k$ , then the 2nd node in the traversal of  $T_1$ , 2nd node in the traversal of  $T_2, \dots$ . If  $v$  is reached while traversing  $T_j$ , or for every  $T_i, i \neq j$ , the final node in the traversal of  $T_i$  has been reached, then  $x = p_j$  and  $T' = \{u\} \cup \{T_i : i = 1 \dots k, i \neq j\}$ . In all cases,  $x$  is identified and the number of nodes examined in the traversal of  $T_j$  is no more than the number of nodes traversed in the other subtrees, i.e., in  $T'$ .  $\square$

## 6 Restricted Regression Values

Isotonic regressions with restricted values have been studied by several authors (see [3] and the references therein), including their use in various classification problems, e.g. [14]. For integer-valued isotonic regression on a linear order, Goldstein and Kruskal [16] gave a  $\Theta(n)$  time algorithm for the  $L_2$  metric and Liu and Ubhaya [25] gave a  $\Theta(n^2)$  algorithm for the  $L_\infty$  metric. Theorem 10 below, coupled with Theorem 4, shows that for  $L_\infty$  the time can be reduced to  $\Theta(n \log n)$ . Further, coupled with Theorem 1 it shows that for any DAG an integer-valued  $L_\infty$  regression can be found in  $\Theta(m \log n)$  time.

We use an approach applicable to arbitrary DAGs and regression values restricted to a set  $S$  of real numbers. An  $S$ -valued  $L_p$  isotonic regression is an  $S$ -valued isotonic function which minimizes the  $L_p$  regression error among all  $S$ -valued isotonic functions.  $S$  need not be finite, but does need to be a closed discrete set, which implies that for any real number  $x$ , if any elements of  $S$  are  $\geq x$  then there is a minimal such one, denoted  $\lceil x \rceil_S$ , and correspondingly if any elements are  $\leq x$  then there is a maximal one, denoted  $\lfloor x \rfloor_S$ . Further, for  $x \in S$  there is a unique successor if there are any elements  $> x$ , and a unique predecessor if there are any elements  $< x$ . If  $x \leq \min\{s \in S\}$  define  $\lfloor x \rfloor_S = \lceil x \rceil_S = \min\{s \in S\}$ , and if  $x \geq \max\{s \in S\}$  define  $\lceil x \rceil_S = \lfloor x \rfloor_S = \max\{s \in S\}$ .

One can quickly convert an unrestricted isotonic regression into an  $S$ -valued one. We assume that one can determine the predecessor, successor, floor, and ceiling functions in constant time.

**Theorem 10** *Given weighted data  $(f, w)$  on a DAG  $G = (V, E)$ , a closed discrete set  $S$  of real numbers, and an unrestricted  $L_\infty$  isotonic regression of  $f$ , an  $S$ -valued  $L_\infty$  isotonic regression can be determined in  $\Theta(m)$  time.*

We first establish a straightforward property connecting values of restricted and unrestricted isotonic regressions. For linear orders, Goldstein and Kruskal [16] showed that for  $L_2$  integer isotonic regression, rounding the unrestricted isotonic regression gives the correct regression, and Liu and Ubhaya [25] showed that this is true for the  $L_\infty$  metric when the data is unweighted and the original regression is Basic. While these results were for integer isotonic regression, the proofs carry over to arbitrary closed discrete sets. However, for weighted data, or when the original regression is not Basic, integer-valued  $L_\infty$  isotonic regression is a bit more complicated. For example, on vertices  $\{0, 1\}$ , suppose the data values are  $(0.6, 0)$  and weights are  $(2, 1)$ . Then the unique unrestricted  $L_\infty$  isotonic regression is  $(0.4, 0.4)$  yet the unique integer-valued  $L_\infty$  regression is  $(1, 1)$ , not  $(0, 0)$ . While rounding does not always give the optimal  $S$ -valued isotonic regression, it nearly does. For linear orders the following was proven in [16] (for  $L_2$ ) and [25] (for  $L_\infty$ ).

**Lemma 11** *For  $1 \leq p \leq \infty$ , given weighted data  $(f, w)$  on a DAG  $G = (V, E)$ , a closed discrete set  $S$  of real numbers, and an  $L_p$  isotonic regression  $g$  of  $(f, w)$ , there is an  $S$ -valued  $L_p$  isotonic regression  $\hat{g}$  such that  $\hat{g}(v) \in \{\lfloor g(v) \rfloor_S, \lceil g(v) \rceil_S\}$  for all  $v \in V$ .*

*Proof:* Given  $p$ , let  $g$  be an unrestricted  $L_p$  isotonic regression, and let  $\hat{g}$  be an  $S$ -valued  $L_p$  isotonic regression. If there are vertices  $v$  such that  $\hat{g}(v) > \lceil g(v) \rceil_S$  then let  $u$  be one that maximizes the difference (a similar proof holds if there are vertices  $v$  such that  $\hat{g}(v) < \lfloor g(v) \rfloor_S$ ). Let  $A = \{v : g(v) = g(u) \text{ and } \hat{g}(v) = \hat{g}(u), v \in V\}$ . For any vertex  $v \notin A$  which is a predecessor of an element of  $A$ ,  $\hat{g}(v) < \hat{g}(u)$  because the fact that it isn't in  $A$  means either  $\hat{g}(v) < \hat{g}(u)$  or  $g(v) < g(u)$ , but if only the latter holds then  $u$  does not optimize the difference. Similarly, if  $v \notin A$  is a successor of a vertex in  $A$  then  $g(v) > g(u)$ .

Let  $x \in S$  be the predecessor of  $\hat{g}(u)$ . Note that  $x \geq \lceil g(u) \rceil_S$ . Let  $\hat{g}'$  be the  $S$ -valued isotonic function which is  $\hat{g}$  on  $V \setminus A$  and  $x$  on  $A$ . If the  $L_p$  regression error of  $\hat{g}'$  is the same as that of  $\hat{g}$  then recursively consider  $\hat{g}'$ . If the regression error of  $\hat{g}'$  is less than that of  $\hat{g}$  then  $\hat{g}$  was not optimal. If the regression error of  $\hat{g}'$  is larger then the weighted mean of  $A$  is greater than  $x$ . Let  $y = \min\{x, \min\{g(v) : g(v) > g(u), v \in V\}\}$ . Raising the value of  $g$  on  $A$  to  $y$ , and keeping it the same elsewhere, would decrease the  $L_p$  error and maintain the isotonic property, contradicting the optimality of  $g$ .  $\square$

To prove Theorem 10, let  $g$  be an unrestricted  $L_\infty$  isotonic regression and for  $r \in S$  let  $\mathcal{L}_r$  be the set  $\{v : \lfloor g(v) \rfloor_S = r, v \in V\}$ . Note that this is a union of level sets in  $g$ . No matter what choices are made for the regression values within  $\mathcal{L}_r$ , since these choices are either  $r$  or its successor  $r^+$  in  $S$ , they do not impose any isotonic restrictions on the choices for other level sets since the values used on  $\mathcal{L}_r$  are at least as large as the upper values in any predecessor and no larger than the lower values in any successor. Therefore we can consider each  $\mathcal{L}_r$  separately.

For  $v \in \mathcal{L}_r$ , define functions `down_err` and `up_err` by

$$\begin{aligned} \text{down\_err}(v) &= \max\{\text{err}(u, r) : u \preceq v, u \in \mathcal{L}_r\} \\ \text{up\_err}(v) &= \max\{\text{err}(u, r^+) : u \succeq v, u \in \mathcal{L}_r\} \end{aligned}$$

`down_err`( $v$ ) is a lower bound on the  $L_\infty$  isotonic regression error if the regression value at  $v$  is  $r$  since all predecessors in  $\mathcal{L}_r$  must also have regression value  $r$ , and similarly `up_err`( $v$ ) is a lower bound if it is  $r^+$ . Let  $\hat{g}$  be defined by

$$\hat{g}(v) = \begin{cases} r & \text{if } \text{down\_err}(v) \leq \text{up\_err}(v) \\ r^+ & \text{otherwise} \end{cases}$$

Since `down_err` is monotonic increasing on  $\mathcal{L}_r$  and `up_err` is monotonic decreasing,  $\hat{g}$  is isotonic increasing. At each vertex  $v$ , changing the choice for  $\hat{g}(v)$  can never decrease the overall regression error, and hence an optimal  $S$ -valued  $L_\infty$  isotonic regression has been found.

The `down_err` and `up_err` values can be computed by topological sorting in  $\Theta(m)$  time, completing the proof of the theorem.  $\square$

The properties of Prefix, Basic and Strict are not preserved by the procedure in the lemma. For example, consider integer-valued regression of the unweighted functions (1, 1, 0) and (1, 0, 0). The Strict regression of each is (0.5, 0.5, 0.5), and the `up_err` and `down_err` values at every vertex are 1 for both. However, the correct integer-valued Strict regression of the first, i.e., the limit, as  $p \rightarrow \infty$  of integer-valued  $L_p$  isotonic regression, is (1, 1, 1), while for the second is (0, 0, 0). To obtain the correct values one needs a more sophisticated method of breaking ties when `up_err`( $i$ ) = `down_err`( $i$ ).

Independent of how the choice is made in the lemma, the result might not satisfy the properties of Max or Min. For example, with function values (0.5, 1) and weights (2, 1), Max is (0.5, 1), which, when converted to integer-valued regressions, results in (0, 1) or (1, 1). However, the integer-valued  $L_\infty$  isotonic regression which is the pointwise maximum of all integer-valued  $L_\infty$  isotonic regressions is (1, 2). Of course, this could be generated by first generating the unrestricted Max, converting it to an integer-valued  $L_\infty$  regression, determining the regression error, and then using  $\epsilon$ -windows with this error.

For small sets one can efficiently find an  $S$ -valued regression without starting with an unrestricted one.

**Theorem 12** *Given weighted data  $(f, w)$  on a DAG  $G = (V, E)$ , and given a finite set  $S$  of real numbers, in  $\Theta(m \log |S|)$  time an  $S$ -valued  $L_\infty$  isotonic regression can be determined.*

*Proof:* Let  $S = \{s_1, \dots, s_d\}$ , let  $[s_a, s_b]$ ,  $1 \leq a \leq b \leq d$  denote the interval  $s_a \dots s_b$  of values in  $S$ , and for  $v \in V$  let  $\text{err}_S(v, a, b)$  denote  $\min\{\text{err}(v, t) : t \in [s_a, s_b]\}$ . The algorithm determines intervals of  $S$  in which the regression value for  $v$  will be assigned, where the intervals are progressively halved until they are a single value, which is the regression value of  $v$ . At each stage the assignment is isotonic in that if  $u \prec v$  then either  $u$  and  $v$  are assigned to the same interval or the intervals have no overlap and  $u$ 's interval has smaller values than  $v$ 's. Initially each vertex is assigned to all of  $S$ , i.e., to  $[s_1, s_d]$ .

Suppose  $x$  is currently assigned to  $[s_a, s_b]$ , and let  $c = \lfloor (a + b)/2 \rfloor$ . Define

$$\begin{aligned} \text{down\_err}(x) &= \max\{\text{err}_S(u, a, c) : u \preceq x, u \text{ currently assigned to } [s_a, s_b]\} \\ \text{up\_err}(x) &= \max\{\text{err}_S(v, c+1, b) : v \succeq x, v \text{ currently assigned to } [s_a, s_b]\} \end{aligned}$$

$\text{down\_err}(x)$  is a lower bound on the  $L_\infty$  regression error for vertices currently assigned to  $[s_a, s_b]$  if  $x$  is assigned to  $[s_a, s_c]$  in the next stage, and  $\text{up\_err}(x)$  is a lower bound if it is assigned to  $[s_{c+1}, s_b]$ . Assign  $x$  to  $[s_a, s_c]$  if  $\text{down\_err}(x) \leq \text{up\_err}(x)$  and to  $[s_{c+1}, s_b]$  otherwise. As before, this assignment is isotonic and can be determined in  $\Theta(m)$  time.

To see that the final assignments form an  $S$ -valued  $L_\infty$  isotonic regression, suppose they don't. Let  $\epsilon$  be the regression error of an  $S$ -valued  $L_\infty$  isotonic regression, and consider the first iteration at which some vertex was assigned to a half for which its error is  $> \epsilon$ . Let  $x$  be a minimal such a vertex, initially in  $[s_a, s_b]$ , and assume it is assigned to the lower half (a similar argument holds if it is assigned to the upper half). Thus  $\epsilon < \text{down\_err}(x) \leq \text{up\_err}(x)$ . No predecessor of  $x$  has error  $> \epsilon$ , therefore  $\text{down\_err}(x) = \text{err}_S(x, a, c)$ . Since the error bound was not exceeded in previous iterations it must be that  $f(x) > s_c$  and  $\text{err}_S(x, 1, c) > \epsilon$ . Because  $\text{up\_err}(x) \geq \text{down\_err}(x)$  there must be an  $x' \succ x$  initially in  $[s_a, s_b]$  for which  $\text{err}_S(x', c+1, b) = \text{up\_err}(x) > \epsilon$ . Let  $y$  be a maximal such element. Similarly to  $x$ 's properties, it must be that  $f(y) < s_{c+1}$  and  $\text{err}_S(y, c+1, d) > \epsilon$ . Since  $x \prec y$ , any  $S$ -valued isotonic regression must either assign a regression value to  $x$  in  $[s_1, s_c]$  or a regression value to  $y$  in  $[s_{c+1}, s_d]$ , contradicting the assumption that there was an  $S$ -valued isotonic regression with error  $\epsilon$ .  $\square$

## 7 Penalty Functions

One can interpret the goal of  $L_\infty$  isotonic regression as minimizing the maximum ‘‘penalty’’ at the vertices, given the isotonic restriction. This can be generalized to penalty functions that do not correspond to a metric. When the possible regression values are from a specified set (as in Section 6) this can be viewed as a classification problem, with a penalty function giving the error of misclassifying a vertex. Few of the algorithms rely on the fact that  $L_\infty$  is a metric, but rather on inequalities. Here we briefly sketch changes that are needed to obtain results for more general penalty functions. Isotonic regressions which minimize the maximum penalty will be called *minimax isotonic regressions*.

A modest extension is when the weights can be different for regression values above the data value than for regression values below it. I.e., for a vertex  $v$  there are nonnegative weights  $w_1, w_2$  so that  $w_1(y - z)$  is the error of using  $z$  as the regression value when the data value is  $y$ ,  $y \geq z$ , and  $w_2(z - y)$  is the error when  $y \leq z$ . For example, overestimating the capacity of a bridge is more serious than underestimating it. None

of the algorithms in this paper, nor their time analyses, require that  $w_1 = w_2$ , but rather depend on the fact that errors and means of violating pairs can be found via intersections of rays. Hence this generalization can be easily accommodated for all of the results in the previous sections with no change in asymptotic run time.

A quite general extension is to nonnegative penalty functions  $p_v(x, y)$  which give the penalty of using  $x$  as the regression value at  $v \in V$  given data  $y$ . It may even be that for some  $v$  and  $y$ , the  $x$  minimizing  $p_v(x, y)$  is not  $y$ . For example, in a Bayesian setting there may be prior distribution at  $v$  and the posterior after observing  $y$  has an MLE different than  $y$ . We require that for any  $v \in V$ ,  $y \in \mathfrak{R}$ , and  $\epsilon \geq 0$ , either  $p_v(\cdot, y)$  has no values  $\leq \epsilon$ , or there are  $a \leq b$  such that  $p_v(x, y) \leq \epsilon$  for all  $x \in [a, b]$  and  $p_v(x, y) > \epsilon$  for  $x \notin [a, b]$ , i.e., the  $\epsilon$ -window is well-defined. It may be that  $a = -\infty$  or  $b = \infty$ , which occurs when the penalty has an upper bound. The requirement that  $\epsilon$ -windows be intervals insures that for any  $v \in V$  and  $y$ , the function  $p_v(x, y)$  is a nonincreasing and then nondecreasing function of  $x$ , where one of these two components may be missing. Note that we allow the possibility that any regression value at a vertex has some penalty. Subtracting the minimum error from all others, so that there is at least one regression value with no penalty, does not always result in the same isotonic regression.

We require that for any  $u, v \in V$  and data  $y_u, y_v$  at  $u, v$ , respectively,

$$\text{pmean\_err}((u, y_u), (v, y_v)) = \min_{x \in \mathfrak{R}} \max\{p_u(x, y_u), p_v(x, y_v)\}$$

is well-defined, i.e., that the minimum is obtained.

*Simple* penalty functions behave similarly to the standard error metrics. A penalty function  $p$  is simple if:  $p(y, y) = 0$  for all  $y$ ; for all  $y$ ,  $p(x, y)$  is a continuous decreasing function of  $x$  for  $x \leq y$  and a continuous increasing function of  $x$  for  $x \geq y$ ; and for all  $x$ ,  $p(x, y)$  is a continuous decreasing function of  $y$  for  $y \leq x$  and a continuous increasing function for  $y \geq x$ . These properties insure that  $\text{pmean\_err}$  is well-defined, as is  $\text{pmean}((u, y_u), (v, y_v))$ , which is the unique value between  $y_u$  and  $y_v$  at which  $u$  and  $v$  have error  $\text{pmean\_err}((u, y_u), (v, y_v))$ . We do not require that  $p \rightarrow \infty$  in any direction, and it may be that  $p(x, y) \neq p(y, x)$ . There is a natural extension of unweighted  $L_\infty$  isotonic regression, namely that there is a single simple penalty function  $p$  such that  $p_v = p$  for all  $v \in V$ .

The Basic and Prefix definitions in Section 2.1.1 extend immediately to simple penalty functions, retaining their monotonicity property. Similar comments hold for Strict. However, the time complexity increases because we can no longer use properties of intersecting rays. We assume that  $p_v(x, y)$ , the bounds of an  $\epsilon$ -window,  $\text{pmean\_err}$ , and (for simple penalty functions)  $\text{pmean}$  can all be computed in unit time.

**Theorem 13** *For data  $f$  on a DAG  $G = (V, E)$ , with penalty functions  $p_v$  for  $v \in V$ , a minimax isotonic regression can be determined in*

- a)  $\Theta(n^2 + m \log n)$  time for general penalty functions,
- b)  $\Theta(m)$  time for the Basic and Prefix regressions and an unweighted simple penalty function,
- c)  $\Theta(n^2)$  time for the Prefix regression and simple penalty functions, assuming that the transitive closure of  $G$  has been given,
- d)  $\Theta(n^2 \log n)$  time for the Strict regression and simple penalty functions, assuming that the transitive closure of  $G$  has been given.

*Proof:* For a), one can still use  $\epsilon$ -windows, as in Algorithm A, but cannot use parametric search, which is based on the properties of straight lines, to determine the increments. This can be remedied by computing

$\text{pmean\_err}((u, f(u)), (v, f(v)))$  for all pairs of vertices  $u, v \in V$  and using binary search on the errors. By computing medians of subsets of these errors as the binary search proceeds, instead of presorting, the total time is  $\Theta(n^2 + m \log n)$ . Just as for the regular  $L_\infty$  case one need not restrict the  $\text{pmean\_err}$  calculations to  $u \prec v$ , so the transitive closure need not be determined.

However, there are some additional aspects. There may never be an  $\epsilon$  for which the  $\epsilon$ -windows are tight, though there will be a smallest  $\epsilon$  for which there is an isotonic regression with maximum penalty  $\epsilon$  (this will be one of the  $\text{pmean\_err}$  values). For example, if  $p(x, y) = \min\{1, |x - y|\}$  and there is data  $(2, 0)$  then any isotonic function has maximum penalty 1. This example also shows that Min and Max might not exist. However, simple changes can handle the situation when  $\epsilon$ -windows have infinite upper or lower bounds.

For b), for the unweighted case,

$$\text{Basic}(v) = \text{pmean}(\max\{f(u_1) : u_1 \preceq v\}, \min\{f(u_2) : u_2 \succeq v\})$$

which, as before, can be shown to be isotonic and of minimal maximum penalty, and can be computed by topological sorting. Prefix can be similarly computed.

For c), using  $\text{pmean}$  instead of  $\text{mean}$ , one can compute the pre function in Algorithm B at vertex  $v$  in time linear in the number of predecessors, and once again it is isotonic and of minimal maximum penalty. Basic is not included in c) because, as noted in Section 2.1.1, it seems to require  $\Theta(n^2)$  calculations per vertex in the general case when linearity need not hold.

For d), Algorithm B in [36], which takes  $\Theta(n^2 \log n)$  time if the transitive closure is given, can be modified by replacing the standard mean calculation with  $\text{pmean}$ .  $\square$

The transitive closure of linear and tree orderings can be determined in  $\Theta(n^2)$  time. This is also true for points in  $d$ -dimensional space with componentwise ordering, even if the points do not form a grid.

**Corollary 14** *For data  $f$  on a linear,  $d$ -dimensional, or tree order of  $n$  vertices, with simple penalty functions at the vertices, a Prefix minimax isotonic regression can be determined in  $\Theta(n^2)$  time and a Strict minimax isotonic regression can be determined in  $\Theta(n^2 \log n)$  time. For orders given by  $d$ -dimensional points the implied constants are a function of  $d$ .  $\square$*

## 8 Final Comments

Algorithm A is the fastest known algorithm for  $L_\infty$  isotonic regression on an arbitrary DAG with weighted data, taking  $\Theta(m \log n)$  time. It improves upon the  $\Theta(m \log n + n \log^2 n)$  algorithm of Kaufman and Tamir [22] when  $m = o(n \log n)$ . The improvement is useful because the DAGs that are most commonly used in isotonic regression, namely linear orders, trees, and  $d$ -dimensional grids, all have  $m = \Theta(n)$ .

The paper also introduced the Prefix regression. Basic has been extensively studied, but computationally Prefix seems more useful for weighted data, and has efficient extensions to penalty functions. It also provides a straightforward approach to problems such as unimodal regression on a line. For arbitrary DAGs it is faster to compute than Strict [36], the “best”  $L_\infty$  isotonic regression mapping, though slower than Min, Max and Avg. However, for linear and tree orders it can be determined as quickly as Min, Max and Avg.

Corollary 5 shows that by using Prefix one can directly determine an  $L_\infty$  unimodal regression in  $\Theta(n \log n)$  time, and thus the parametric search used by Chun, Sadakane, and Tokuyama [10] is not necessary. It would be interesting to find a  $\Theta(m \log n)$  algorithm for arbitrary DAGs that is not based on parametric search. In general, finding a replacement for parametric search is viewed favorably (see the

paper by Agarwal and Sharir [2]). Except for Theorems 1 and 13 a), the results presented here utilize direct approaches with smaller constants than those of parametric search. On the other hand, the indirect  $\epsilon$ -windows approach used in Theorem 13 a) allows extensions to quite general penalty functions.

Beyond their computational time, another important aspect of  $L_\infty$  isotonic regression algorithms is the mathematical properties of the regressions they produce. This is not an issue for  $L_p$  isotonic regression when  $1 < p < \infty$ , but the nonuniqueness of  $L_\infty$  regression produces a range of algorithmic behavior. For isotonic regression one would not expect that increasing a data value results in the regression lowering at other vertices, and this does not happen for  $L_p$  regression when  $1 < p < \infty$ , nor for Prefix, Basic and Strict. However, it can happen for Min, Max and Avg. In Section 2.3 it was shown that, in terms of minimizing the number of vertices with large regression errors, Strict is the best possible, and Avg, Basic and Prefix are significantly better than Min and Max. While the primary goal of  $L_\infty$  regression is to minimize the worst error, in some applications the number of large errors might also be relevant. For example, if one is locating service centers to minimize the worst distance to a service center [22], it might be beneficial if few customers had to travel the maximum distance.

Further, the Max, Min, and Avg regressions can have values outside the data range. E.g., for the function on  $\{1, 2, 3, 4\}$  with data values (2, 3, 1, 2) and weights (1, 4, 4, 1), Min is (-2, 2, 2, 2), Max is (2, 2, 2, 6), and Avg is (0, 2, 2, 4). For some purposes an isotonic regression with values outside the data range would not be acceptable. While one could trim the values so that they lie within the data range, Prefix, Basic and Strict have this property automatically, as do  $L_p$  isotonic regressions for  $1 \leq p < \infty$ .

Overall, among the algorithms for general DAGs there is an inverse relationship between an algorithm's worst-case time to determine an  $L_\infty$  isotonic regression and the desirability of the regressions it produces.

## References

- [1] Agarwal, K, Phillips, JM, and Sadri, B (2010), "Lipschitz unimodal and isotonic regression on paths and trees", *LATIN 2010*, pp. 384–396.
- [2] Agarwal, PK and Sharir, M (1998), "Efficient algorithms for geometric optimization", *ACM Computing Surveys*, pp. 412–458,
- [3] Ahuja, RK and Orlin, JB (2001), "A fast scaling algorithm for minimizing separable convex functions subject to chain constraints", *Operations Research* 49, pp. 784–789.
- [4] Angelov, S, Harb, B, Kannan, S, and Wang, L-S (2006), "Weighted isotonic regression under the  $L_1$  norm", *Symp. Discrete Algorithms* 2006, pp. 783–791.
- [5] Barlow, RE, Bartholomew, DJ, Bremner, JM, and Brunk, HD (1972), *Statistical Inference Under Order Restrictions: the Theory and Application of Isotonic Regression*, Wiley.
- [6] Barlow, RE and Brunk, HD (1972), "The isotonic regression problem and its dual", *J. Amer. Stat. Soc.* 67, pp. 140–147.
- [7] Brill, G, Dykstra, R, Pillars, C and Robertson, T (1984), "Algorithm AS 206: Isotonic regression in two independent variables", *J. Royal Stat. Soc. Series C (Applied Stat.)* 33, pp. 352–357.
- [8] Brown, MR and Tarjan, RE (1979), "A fast merging algorithm", *J. Assoc. Comp. Mach.* 26, pp. 211–226.

- [9] Chakrabarti, D, Kumar, R and Punera, K (2007), “Page-level template detection via isotonic smoothing”, *Proc. 16th Int’l. World Wide Web Conf.*
- [10] Chun, J, Sadakane, K, and Tokuyama, T (2006), “Linear time algorithm for approximating a curve by a single-peaked curve”, *Algorithmica* 44, pp. 103–115.
- [11] Chandrasekaran, R, Rhy, YU, Jacob, VS, and Hong, S (2005), “Isotonic separation”, *INFORMS J. Computing* 17, pp. 462–474.
- [12] Dembczynski, K, Greco, S, Kotlowski, W, and Slowinski, R (2007), “Statistical model for rough set approach to multicriteria classification”, *PKDD 2007: 11<sup>th</sup> European Conf. Principles and Practice Knowledge Discovery in Databases*, Springer Lec. Notes Comp. Sci. 4702, pp. 164–175.
- [13] Durham, S, Flournoy N, and Li, W (1998), “A sequential design for maximizing the probability of a favorable response”, *Can. J. Stat.* 26, pp. 479–495.
- [14] Dykstra, R, Hewett, J and Robertson, T (1999), “Nonparametric, isotonic discriminant procedures”, *Biometrika* 86, pp. 429–438.
- [15] Gebhardt, F (1970), “An algorithm for monotone regression with one or more independent variables”, *Biometrika* 57, pp. 263–271.
- [16] Goldstein, AJ and Kruskal, JB (1976), “Least-squares fitting by monotonic functions having integer values”, *J. Amer. Stat. Assoc.* 71, pp. 370–373.
- [17] Haiminen, N, and Gionis, A (2004), “Unimodal segmentation of sequences”, *Proc. 4<sup>th</sup> IEEE Int’l. Conf. Data Mining*, pp. 106–113.
- [18] Hardwick, J, Meyer, M and Stout, QF (2003), “Directed walk designs for dose response problems with competing failure modes”, *Biometrics* 59, pp. 229–236.
- [19] Herkenrath, U. (1983), “The  $N$ -armed bandit with unimodal structure”, *Metrika* 30, pp. 195–210.
- [20] Hershberger, J, and Suri, S (1992), “Applications of a semi-dynamic convex hull algorithm”, *BIT* 32, pp. 249–267.
- [21] Kalai, AT and Sastry, R (2009), “The Isotron algorithm: high-dimensional isotonic regression”, *Proc. Comp. Learning Theory (COLT) 2009*.
- [22] Kaufman, Y and Tamir, A (1993), “Locating service centers with precedence constraints”, *Discrete Applied Math.* 47, pp. 251–261.
- [23] Legg, D., Townsend, D (1984), “Best monotone approximation in  $L_\infty[0,1]$ ”, *J. Approx. Theory* 42, pp. 30–35.
- [24] Lin, T.-C., Kuo, C.-C., Hsieh, Y.-H., and Wang, B.-F. (2009), “Efficient algorithms for the inverse sorting problem with bound constraints under the  $l_\infty$ -norm and the Hamming distance”, *J. Comp. and Sys. Sci.* 75, pp. 451–464.
- [25] Liu, M-H and Ubhaya, VA (1997), “Integer isotone optimization”, *SIAM J. Optimization* 7, pp. 1152–1159.

- [26] Maxwell, WL and Muckstadt, JA (1985), “Establishing consistent and realistic reorder intervals in production-distribution systems”, *Operations Research* 33, pp. 1316–1341.
- [27] Megido, N (1983), “Linear-time algorithms for linear programming in  $\mathfrak{R}^3$  and related problems”, *SIAM J. Computing* 12, pp. 759–776.
- [28] Moon, T, Smola, A, Chang, Y and Zheng, Z (2010), “IntervalRank — isotonic regression with listwise and pairwise constraints”, *Proc. Web Search and Data Mining*, pp. 151–160.
- [29] Pardalos, PM and Xue, G (1999), “Algorithms for a class of isotonic regression problems”, *Algorithmica* 23, pp. 211–222.
- [30] Pehrsson, N-G and Frisén, M (1983), “The UREGR procedure”, Gothenburg Computer Central, Göteborg, Sweden.
- [31] Pólya, G. (1913), “Sur une algorithme toujours convergent pour obtenir les polynomes de meilleure approximation de Tchebycheff pour un fonction continue quelconque”, *Comptes Rendus* 157, pp. 840–843.
- [32] Punera, K and Ghosh, J (2008), “Enhanced hierarchical classification via isotonic smoothing”, *Proc. Int’l. Conf. World Wide Web 2008*, pp. 151–160.
- [33] Robertson, T, Wright, FT, and Dykstra, RL (1988), *Order Restricted Statistical Inference*, Wiley.
- [34] Spouge, J, Wan, H, and Wilber, WJ (2003), “Least squares isotonic regression in two dimensions”, *J. Optimization Theory and Appl.* 117, pp. 585–605.
- [35] Stout, QF (2008), “Unimodal regression via prefix isotonic regression”, *Comp. Stat. and Data Anal.* 53, pp. 289–297.
- [36] Stout, QF (2012), “Strict  $L_\infty$  isotonic regression”, *J. Optimization Theory and Appl.* 152, pp. 121–135.
- [37] Stout, QF (2012), “Isotonic regression via partitioning”, *Algorithmica*, to appear. Available at [www.eecs.umich.edu/~qstout/pap/L1IsoReg.pdf](http://www.eecs.umich.edu/~qstout/pap/L1IsoReg.pdf)
- [38] Thompson, WA Jr. (1962), “The problem of negative estimates of variance components”, *Annals Math. Stat.* 33, pp. 273–289.
- [39] Turner, TR and Wollan, PC (1997), “Locating a maximum using isotonic regression”, *Comp. Stat. and Data Anal.* 25, pp. 305–320.
- [40] Ubhaya, VA (1974), “Isotone optimization, I, II”, *J. Approx. Theory* 12, pp. 146–159, 315–331.
- [41] Velikova, M and Daniels, H (2008), *Monotone Prediction Models in Data Mining*, VDM Verlag.
- [42] Yu, JY and Mannor, S (2011), “Unimodal bandits”, *Proc. ICML: Int’l. Conf. Machine Learn.*, pp. 41–48.