

Strict L_∞ Isotonic Regression

Quentin F. Stout^{1 2 3}

Abstract

Given a function f and weights w on the vertices of a directed acyclic graph G , an isotonic regression of (f, w) is an order-preserving real-valued function that minimizes the weighted distance to f among all order-preserving functions. When the distance is given via the supremum norm there may be many isotonic regressions. One of special interest is the strict isotonic regression, which is the limit of p -norm isotonic regression as p approaches infinity. Algorithms for determining it are given. We also examine previous isotonic regression algorithms in terms of their behavior as mappings from weighted functions over G to isotonic functions over G , showing that the fastest algorithms are not monotonic mappings. In contrast, the strict isotonic regression is monotonic.

Keywords: isotonic regression; shape-constrained optimization; nonparametric; supremum norm; mini-max

AMS Classifications: 62G08, 68Q25, 68W25

1 Introduction

This paper considers a form of shape-constrained nonparametric regression known as *isotonic regression*, an area of study going back at least to 1955 [1–6]. It is of increasing importance as researchers are less willing to impose strong assumptions in their modeling. For example, they may be willing to make the weak assumption that the expected height of a woman is an increasing function of the height of her father and of her mother, but be unwilling to make parametric assumptions such as linearity. The development of faster algorithms for determining isotonic regressions have helped make this a practical approach, and recent applications include very large data sets from the web [7, 8] and from microarrays [9].

A function is *isotonic* iff it is nondecreasing, and an *isotonic regression* is a regression minimizing the regression error among all isotonic functions. When the error norm is the supremum norm the isotonic regression is not necessarily unique, and a given regression may have undesirable properties. For example, for consecutive data values 2, -2, 1, any isotonic regression must have initial values 0, 0, with regression error 2. The supremum norm remains the same, and the isotonic property is preserved, as long as the third value is between 0 and 3. One would generally prefer that it be 1 since there is little reason to increase the error unnecessarily. Despite this, no previously studied algorithm for finding isotonic regression on arbitrary ordered sets, using the supremum norm, results in 1.

In Section 3 the strict isotonic regression is introduced. It is the limit, as p tends to infinity, of the p -norm isotonic regression. In the example, it would result in the third regression value being 1. Some of its uniqueness properties are determined, and it is shown that it can be determined using a simple, but slow, approach. Section 4 gives a faster algorithm for computing it, and yet faster algorithms are given for linear and tree orderings.

¹Communicated by Panos M. Pardalos

²The author thanks the referee for helpful comments.

³Computer Science and Engineering, University of Michigan. Ann Arbor, MI. qstout@umich.edu

Section 5 examines properties of the previously studied isotonic regression algorithms. The focus is on the regressions they produce, rather than on the time taken to produce them, and it is shown that they have undesirable behavior that the strict regression does not have. Final remarks appear in Section 6.

2 Basics

We use (f, w) to denote a real-valued function f and associated non-negative weight function w , and also use (v, x) to denote the value v with associated weight x . The context will always be clear as to whether it is functions or individual values that are being referred to. We use $(f, w)(u)$ to denote $(f(u), w(u))$. Given (f, w) and a regression value r at $v \in V$, the weighted *regression error at v* , denoted $\text{err}(r, v)$, is $w(v) \cdot |f(v) - r|$.

A directed acyclic graph (DAG) $G = (V, E)$ induces a partial order \prec on V , where $v_1 \prec v_2$, $v_1 \neq v_2$, iff there is a path in G from v_1 to v_2 . A real-valued function g on V is *isotonic* if it is weakly order-preserving, that is, if $v \prec w$ then $g(v) \leq g(w)$. An isotonic function g is an L_p *isotonic regression of (f, w)* , $1 \leq p \leq \infty$, iff it minimizes the regression error

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

among all isotonic functions on G . For $1 < p < \infty$ the L_p isotonic regression is unique, but, as was shown in the Introduction, for L_∞ it need not be. Unless otherwise specified, isotonic regression is with respect to the L_∞ metric, which is also known as the supremum, Chebyshev, uniform, or chessboard metric.

The weighted mean of values (y_1, w_1) and (y_2, w_2) , $\text{wmean}((y_1, w_1), (y_2, w_2))$, is $(w_1 y_1 + w_2 y_2) / (w_1 + w_2)$. The error in using the weighted mean at each value is the same, namely $w_1 w_2 |y_1 - y_2| / (w_1 + w_2)$. This will be denoted $\text{err-mean}((y_1, w_1), (y_2, w_2))$. A *violating pair* is a pair of vertices $u, v \in V$ where the data violates the isotonic condition, i.e., $u \prec v$ and $f(u) > f(v)$. For a violating pair, for any isotonic function g , $\max\{\text{err}(g(u), u), \text{err}(g(v), v)\} \geq \text{err-mean}((f, w)(u), (f, w)(v))$.

Given a set S of weighted values $\{(y_1, w_1), \dots, (y_n, w_n)\}$, the L_∞ weighted mean of S , $\text{wmean}(S)$, is the value C that minimizes $\max_{i=1}^n w_i \cdot |y_i - C|$. It is straightforward to show that

$$\begin{aligned} \text{wmean}(S) &= \text{wmean}((y_k, w_k), (y_\ell, w_\ell)), & \text{where} \\ \text{err-mean}((y_k, w_k), (y_\ell, w_\ell)) &= \max\{\text{err-mean}((y_i, w_i), (y_j, w_j)) : 1 \leq i, j \leq n\}, \end{aligned}$$

and that if two pairs maximize err-mean then the pairs have the same mean. Since the weight function will always be obvious, we use $\text{err-mean}(f(u), f(v))$ to denote $\text{err-mean}((f, w)(u), (f, w)(v))$ and $\text{wmean}(f(u), f(v))$ to denote $\text{wmean}((f, w)(u), (f, w)(v))$. For a weighted function (f, w) and set $V' \subset V$, $\text{wmean}(f|V')$ denotes the weighted mean of $\{(f, w)(v) : v \in V'\}$. A set $L \subseteq V$ is a *level set* of a regression g iff it is a maximal connected set such that for every $u, v \in L$, $g(u) = g(v)$. The *regression value* of the level set L will always be $\text{wmean}(f|L)$.

2.1 Regression Mappings

Let $G = (V, E)$ be a DAG with n vertices and m edges. An L_∞ *isotonic regression mapping on G* , R , is a mapping from weighted functions on V to isotonic functions on G such that, for any weighted function (f, w) on V , $R(f, w)$ is an L_∞ isotonic regression of (f, w) . Informally, R will be merely called a *regression*

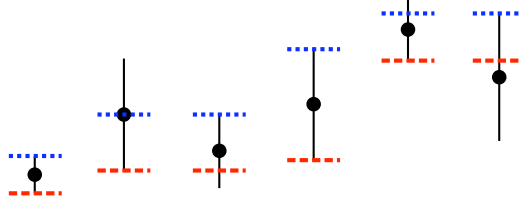


Figure 1: Max and Min: vertical bars represent windows

mapping. The most widely used regression mapping is Basic, apparently first studied by Barlow and Ubhaya [10]:

$$\begin{aligned} \text{Basic}(f, w)(x) &= \text{wmean}(f(u_1), f(u_2)), \quad \text{where} \\ \text{err-mean}(f(u_1), f(u_2)) &= \max\{\text{err-mean}(f(u), f(v)) : u \preceq x \preceq v, f(u) \geq f(v)\}. \end{aligned}$$

It is straightforward to show that this is isotonic and minimizes the error at the violating pair(s) of maximal err-mean, and hence Basic minimizes the overall regression error. For unweighted data,

$$\text{Basic}(f, 1)(x) = \frac{\max\{f(v) : u \preceq x\} + \min\{f(v) : x \preceq v\}}{2}$$

which can be computed by topological sort in $\Theta(m)$ time. Efficient calculations for weighted data are discussed at the end of Section 4.2.

A related regression mapping is Prefix, introduced in [11], which is defined as follows: let

$$g(u) = \max\{\text{wmean}(f(v), f(u)) : v \preceq u\}$$

Then

$$\text{Prefix}(f, w)(x) = \min\{g(u) : x \preceq u\}$$

The prefix property that this has is that for all $u \in V$, $g(u) = \text{Basic}_u(f, w)(u)$, where Basic_u is Basic on G restricted to u and its predecessors.

To see that $\text{Prefix}(f, w)$ is an L_∞ isotonic regression, the minimization over all successors insures that it is isotonic. The regression error of $\text{Prefix}(f, w)$ will be compared to that of $\text{Basic}(f, w)$. To simplify notation, let $P = \text{Prefix}(f, w)$ and $B = \text{Basic}(f, w)$. Since $g(x) \geq B(x)$ for all $x \in V$, $P(x) \geq B(x)$ for all $x \in V$. Therefore, if $P(x) \leq f(x)$, then $\text{err}(P(x), x) \leq \text{err}(B(x), x)$. If $P(x) > f(x)$ then, since $g(x) \geq \text{Prefix}(x)$, $\text{err}(g(x), x) \geq \text{err}(P(x), x)$. Since $g(x) = \text{Basic}(x)$ on the subgraph restricted to x and its predecessors, the error of g at x is no more than the regression error of Basic on this subgraph, which is at least as large as the regression error of Basic on G .

Some important regression mappings are based on a quite different approach. A set W of *windows* on V is an interval $[L(v), U(v)]$, $L(v) \leq U(v)$, at each $v \in V$. An isotonic function g fits through W iff $L(v) \leq g(v) \leq U(v)$ for all $v \in V$. There is an isotonic function that fits through W if and only if the function $g(v) = \max\{L(u) : u \preceq v\}$ fits through it. Further, for any isotonic function h fitting through W , $h(v) \geq g(v)$ for all $v \in V$. Note that g can be computed in $\Theta(m)$ time by using topological sort.

Windows can be used to efficiently find an L_∞ isotonic regression if the optimal regression error, ϵ , is known. One merely uses windows that contain all regression values with error no more than ϵ , i.e., $L(v) = f(v) - \epsilon/w(v)$ and $U(v) = f(v) + \epsilon/w(v)$. The above construction of g gives the Min regression

mapping, and a similar process, using $g(v) = \min\{U(u) : v \preceq u\}$, gives Max. See Figure 1. Define Avg by $\text{Avg}(v) = (\text{Min}(v) + \text{Max}(v))/2$. For unweighted functions Basic = Avg, but this is not necessarily true for weighted functions.

Kaufman and Tamir [12] use parametric search to determine the minimal regression error. Coupling this with the windows approach, one can determine Min, Max and Avg in $\Theta(m \log n)$ time [11]. These are the fastest known algorithms for arbitrary DAGs, but, as will be shown in Section 5, the regressions lack some desirable properties.

3 Strict Isotonic Regression

We introduce a new isotonic regression mapping, Strict. For a DAG G and weighted function (f, w) on G , the *strict L_∞ isotonic regression* of (f, w) is

$$\text{Strict}(f, w)(x) = \lim_{p \rightarrow \infty} \hat{f}_p(x)$$

where \hat{f}_p is the L_p isotonic regression of (f, w^p) . It is the regression of (f, w^p) , not (f, w) , since the L_p means of a weighted set $S = \{(y_1, w_1), \dots, (y_n, w_n)\}$ converge to the L_∞ mean of the unweighted set $\{y_1, \dots, y_n\}$, while the L_p mean of $S_p = \{(y_1, w_1^p), \dots, (y_n, w_n^p)\}$ converges to the L_∞ mean of S .

To show that Strict is well-defined, we introduce a partial order on isotonic functions. Given a DAG $G = (V, E)$ and weighted function (f, w) on V , one way to define the L_∞ isotonic regressions of (f, w) is to consider a partial order on isotonic functions on V . For isotonic functions g_1 and g_2 , say that g_1 precedes g_2 iff there is a $C > 0$ such that g_1 has no vertices with regression error $\geq C$ while g_2 has at least one, i.e., $\|g_1 - f\|_{w, \infty} < C$ and $\|g_2 - f\|_{w, \infty} \geq C$, where $\|\cdot\|_{w, \infty}$ denotes the weighted L_∞ distance. The L_∞ isotonic regressions of (f, w) are the minimal elements of this partial order.

Here the partial order is slightly refined: given isotonic functions g_1 and g_2 , then $g_1 <_{(f, w)} g_2$ iff there is a $C > 0$ such that g_2 has more vertices with regression error $\geq C$ than does g_1 , and for any $D > C$, g_1 and g_2 have the same number of vertices with regression error $\geq D$. It may be that there is a $C' < C$ such that g_1 has more vertices with error $\geq C'$ than does g_2 , but since the emphasis is in minimizing large errors, the behavior at C trumps that at C' . Theorem 3.1 proves that Algorithm A generates the unique minimum of $<_{(f, w)}$, and Theorem 3.2 shows that this is $\text{Strict}(f, w)$.

3.1 The Minimum Regression

Given a DAG G and a weighted function (f, w) on G , Algorithm A gives an explicit construction of an L_∞ isotonic regression, which we call $R(f, w)$. Figure 2 illustrates the stages of Algorithm A. Lemma 3.1 shows that the algorithm always terminates, and Theorem 3.1 proves that it is the unique minimum of the $<_{(f, w)}$ ordering.

Lemma 3.1 *For any DAG $G = (V, E)$ and weighted function (f, w) , Algorithm A terminates with $s \leq n - 1$ and $R(f, w)(v)$ being defined for all $v \in V$.*

Proof: It suffices to show that for each stage s , ϵ_s exists, and that there is a v for which $R(f, w)(v)$ is defined during stage s . The first iteration of the while-loop results in ϵ_1 being the regression error of R . If this is 0 then the algorithm is finished. Otherwise, there is a level set L with this as its regression error. L must have at least one vertex with a value less than L 's and at least one with a value greater than L 's, each with

$R(v)$ is initially undefined for all $v \in V$.

$s = 0$

while there exist $v \in V$ for which $R(v)$ is undefined

$s = s + 1$

$\epsilon_s =$ minimum ϵ for which \exists an isotonic function through the windows $[L(v, \epsilon), H(v, \epsilon)]$, where

 if $R(v)$ is defined then $L(v, \epsilon) = H(v, \epsilon) = R(v)$

 else $L(v, \epsilon) = f(v) - \epsilon/w(v)$ and $H(v, \epsilon) = f(v) + \epsilon/w(v)$

 for all $v \in V$

$B(v) = \max\{L(u, \epsilon_s) : u \preceq v\}$

$T(v) = \min\{H(u, \epsilon_s) : v \preceq u\}$

 if $B(v) = T(v)$ and $R(v)$ is undefined then $R(v) = B(v)$

 end for

end while

Algorithm A: L_∞ Isotonic Regression R for a Weighted Function (f, w) on DAG $G = (V, E)$

regression error ϵ_1 . Both of these vertices will have their B and T values equal, so at least two vertices have R defined during the first stage. Thus the final s is no more than $|V| - 1$.

Suppose that by the end of stage s , $\epsilon_1 > \epsilon_1 \dots > \epsilon_s$. We will show that ϵ_{s+1} exists and $\epsilon_{s+1} < \epsilon_s$. Let W be the elements of V for which R is still undefined at the end of stage s . For $u \in W$, at the end of stage s , $B(u) < T(u)$. $B(u)$ is a continuous non-decreasing function of ϵ , $T(u)$ is a continuous non-increasing function of ϵ , and $B(u) \geq T(u)$ when $\epsilon = 0$. Therefore for each u there is a minimal $0 \leq \delta_u < \epsilon_s$ such that $B(u) = T(u)$ when $\epsilon = \delta_u$. Then $\epsilon_{s+1} = \max\{\delta_u : u \in W\}$, and for at least one $v \in W$, $\delta_v = \epsilon_{s+1}$ and $R(f, w)(v)$ is defined at stage $s + 1$. \square

Theorem 3.1 *Given a DAG G and a weighted function (f, w) on G , $R(f, w)$ is the unique minimum of the $\prec_{(f, w)}$ ordering.*

Proof: Let $\hat{f} = R(f, w)$ and let $g \neq \hat{f}$ be an isotonic function on G . Let s be such that $g(v) = \hat{f}(v)$ for all v for which $\hat{f}(v)$ was defined at stage $s - 1$ or earlier, and there is a $u \in V$ such that $\hat{f}(u)$ was defined at stage s and $g(u) \neq \hat{f}(u)$.

Claim: there is a $v \in V$ such that $\hat{f}(v)$ was defined at stage s and $\text{err}(g(v), v) > \epsilon_s$. This implies that $\hat{f} \prec_{(f, w)} g$ since they agree on all vertices with greater error. To prove the claim, since $g = \hat{f}$ on vertices defined for R at earlier stages, at stage s , for any regression error ϵ , g and \hat{f} must fit through the same ϵ windows. When $\epsilon = \epsilon_s$ the vertices u where \hat{f} is defined at stage s have $B(u) = T(u)$, i.e., any isotonic function through the ϵ_s windows must have value $\hat{f}(u)$. Therefore g cannot fit through the ϵ_s windows, proving the claim. \square

3.2 $R = \text{Strict}$

In this section we will prove that $R = \text{Strict}$, a side effect of which is a proof that Strict is well-defined. This will be accomplished by examining the structure of the level sets of R . A subset S of a set $L \subseteq V$ is a *lower subset* of L iff for every $v \in S$, if $u \in L$ and $u \prec v$ then $u \in S$. It is an *upper subset* of L iff for every $v \in S$, if $u \in L$ and $u \succ v$ then $u \in S$. For $1 < p < \infty$, L_p regression satisfies properties 1), 2), and 3) in

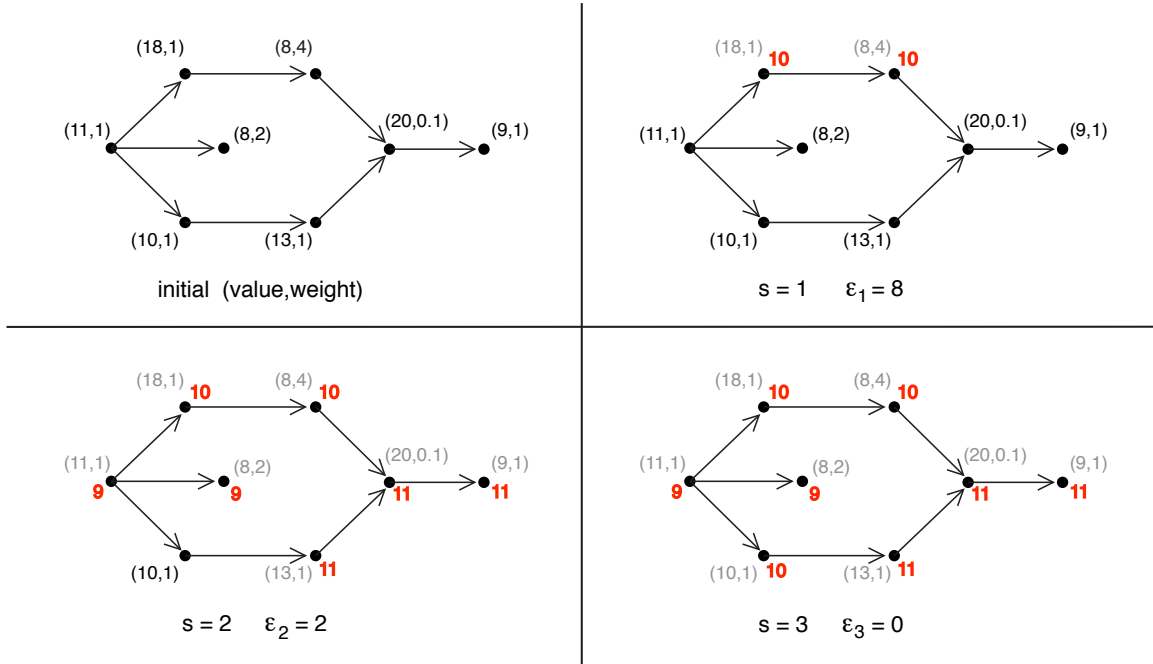


Figure 2: Steps in Algorithm A, boldface = regression value

the theorem below (for L_1 the nonuniqueness of medians makes the situation more complex). Furthermore, once 1), 2), and 3) are met, for any metric, the proof shows that the regression is unique.

Proposition 3.1 *Given DAG $G = (V, E)$ and weighted function (f, w) on G , let g be an isotonic regression such that, for every level set L of g ,*

1. $g(L) = \text{wmean}(f|L)$,
2. for every lower subset L' of L , $\text{wmean}(f|L') \geq \text{wmean}(f|L)$, and
3. for every upper subset L'' of L , $\text{wmean}(f|L'') \leq \text{wmean}(f|L)$.

Then g is $R(f, w)$.

Proof: Let $\hat{f} = R(f, w)$. To see that \hat{f} satisfies the properties, property 1) follows immediately from the construction of R . For 2), suppose there is a level set L with a lower level subset L' such that $\text{wmean}(f|L') < \text{wmean}(f|L)$. Let C be the maximum of $\text{wmean}(f|L')$ and the largest level set value less than that of L , and let the function f' be equal to \hat{f} on $V \setminus L'$ and C on L' . Then f' is isotonic and $f' <_{(f, w)} \hat{f}$, contradicting the minimality of \hat{f} . The proof of 3) is similar.

To show uniqueness, let $g \neq \hat{f}$ be isotonic and obey 1) – 3), and suppose there is a v be such that $g(v) < \hat{f}(v)$ (the proof is the same if $g(v) > \hat{f}(v)$). WLOG v is a maximal element of its level set in g , since otherwise we could replace it with one, which would have the same g value, and, due to the monotonicity of \hat{f} , an \hat{f} at least as large. We construct a sequence $\beta_0 \subseteq \alpha_1 \subseteq \beta_1 \subseteq \alpha_2 \dots$ of subsets of V as follows:

```

 $\beta_0 = \{v\}$ 
 $i = 0$ 
repeat
   $i = i + 1$ 
   $\alpha_i = \{w : w \preceq u, u \in \beta_{i-1}, w \text{ and } u \text{ are in the same level set of } \hat{f}\}$ 
   $\beta_i = \{w : w \succeq u, u \in \alpha_i, w \text{ and } u \text{ are in the same level set of } g\}$ 
until  $\alpha_i = \beta_i$ 

```

Since V is finite and the sets are nondecreasing, eventually the termination condition will be met. Let I denote the final value of i . When a level set A of \hat{f} first had an element included in the α, β sequence it was added in a β step since α never includes elements from level sets of \hat{f} that weren't already in the preceding β . The construction of β only involves successors, so each new A has value at least that of the lower set containing v , i.e., $w\text{mean}(f|A) \geq \hat{f}(v)$. Similarly a level set B of g can only be added in an α step, and $w\text{mean}(f|B) \leq g(v)$.

By the construction of each α , $A \cap \alpha_I$ is a lower set of A , and similarly $B \cap \beta_I$ is an upper set of B . Since it is a lower set, $w\text{mean}(f|\alpha_I) \geq \hat{f}(v)$. Similarly, $w\text{mean}(f|\beta_I) \leq g(v)$, which is $< \hat{f}(v)$. However, $\alpha_I = \beta_I$, so it is impossible for their means to differ. Therefore there is no such g . \square

Theorem 3.2 *Given a DAG G and weighted function (f, w) on G , $R(f, w) = \text{Strict}(f, w)$.*

Proof: For any $V' \subseteq V$, the L_p weighted mean of V' , using weights w^p , converges to its L_∞ weighted mean, using weights w , as p tends to infinity. Since there are only finitely many subsets, this implies that for any $\epsilon > 0$ there is a $p_\epsilon > 1$ so that for any $V' \subseteq V$ and any $p \geq p_\epsilon$, the L_p weighted mean of V' differs from the L_∞ weighted mean of V' by less than ϵ . Let \hat{f}_p be the L_p isotonic regression of (f, w^p) , $1 < p < \infty$.

Choose a $v \in V$ and an $\epsilon > 0$, and let $p > p_\epsilon$. The proof of uniqueness in Proposition 3.1 shows that if $\hat{f}_p(v)$ and $R(f, w)(v)$ differ by more than ϵ , then there is a set where the means differ by more than ϵ . Since $p > p_\epsilon$ this cannot occur. Thus, as $p \rightarrow \infty$, $\hat{f}_p(v) \rightarrow R(f, w)(v)$. \square

Ubhaya [6] showed that \hat{f}_p converges to an L_∞ isotonic regression, using a somewhat more complicated proof which was also based on level sets. However, no algorithm was given to determine it. For unweighted isotonic functions on the unit interval this convergence has been studied by several authors [13–15]. In these papers it is called the “best best” L_∞ -approximant, and the limiting process is known as the “Polya algorithm” [16]. Note that their concerns were quite different than for regressions on finite sets.

4 More Efficient Algorithms

For a DAG of n vertices, Algorithm A may require $\Theta(n)$ iterations, each of which involves determining an ϵ_s . The fastest algorithm known for this requires $\Theta(m \log m)$ time [11], and thus Algorithm A may take $\Theta(n^3 \log n)$ time. In Section 4.1 a faster algorithm is given for linear or tree orders, and in Section 4.2 one is given for general DAGs. Both directly generate the regression, in contrast to the indirect approach of Algorithm A which is based on determining a regression error and then finding an isotonic function with that error.

4.1 Linear and Tree Orderings

For a DAG which is a linear order, pair adjacent violators (PAV) starts with each vertex being its own level set. If there are two adjacent level sets that are out of order (i.e., the value of the level set to the left is larger

than that of the level set to the right) then they are merged and the regression value of the level set is the weighted mean of the function values. This process is repeated until there are no adjacent level sets out of order. In any step of the process, for a level set L on vertices $v_i \dots v_k$, for any $i \leq j \leq k$, the mean of the function values on $v_i \dots v_j$ is at least as large as the mean of the entire level set, and the mean of the function values on $v_j \dots v_k$ is no larger than the mean of the entire level set. It is easy to show that these invariants are maintained at each pooling, and that the resulting regression is unique, not depending on the order in which level sets were merged. The use of PAV goes back at least to 1955, by Ayer et al. [1], and has been repeatedly rediscovered. Using it, isotonic regression on a linear order can be found in $\Theta(n)$ time for the L_2 metric, and $\Theta(n \log n)$ time for the L_1 [17,18] and L_∞ [11] metrics.

For orderings given by a directed tree the pooling process needs to be done more carefully, as noted by Thompson [19]. We assume that the tree is upward directed and apply PAV bottom-up. (If it is downward directed one can take the negative of the function values, apply the algorithm for an upward directed tree, and then take the negative of the regression.) Suppose the process has been completed for all the children of vertex v . If $f(v)$ is as large as the regression value of any level set below it, then the algorithm is finished at this node. Otherwise, choose the level set beneath with largest value (which must contain one of v 's children) and merge v with that, taking the weighted mean of this set as its regression value. Continue expanding this level set, at each step pooling with the level set beneath it with the largest value, until its regression value is as large as any level set below.

Proposition 4.1 *For DAGs which are linear or tree orderings, PAV = Strict.*

Proof: For trees, all level sets are subtrees, all connected lower subsets of a level set are a subtree of the level set, and all upper subsets of a level set are subtrees with root equal to the root of the level set. Using this, it is easy to see that PAV maintains the invariants that for any level set L and any lower subset L' of L , $w\text{mean}(f|L') \geq w\text{mean}(f|L)$, and for any upper subset L'' , $w\text{mean}(f|L) \geq w\text{mean}(f|L'')$. These are properties 2) and 3) of Proposition 3.1, and 1) is trivially true. Since Strict is unique, PAV = Strict. \square

Proposition 4.2 *For linear or tree orderings, Strict can be determined in $\Theta(n \log n)$ time.*

Proof sketch: For tree orderings, Pardalos and Xue [20] showed that PAV can be computed in $\Theta(n \log n)$ time for the L_2 metric. There are two components: keeping track of the level sets below so that the one of largest value can be located efficiently, and determining the new regression value when two level sets are merged. They showed how to use standard structures, such as mergable heaps, to do the former, taking at most $\Theta(n \log n)$ time for all steps.

In Stout [11] it is shown that for each level set one can keep trees of upper and lower envelopes of lines corresponding to values and their weights, and from this determine the level set L_∞ mean in $\Theta(\log n)$ time. Storing this as a balanced tree, and using a careful merge algorithm, one can merge level sets in arbitrary order and determine the level set regression value at each step in $\Theta(n \log n)$ total time. Combining this with the first component from Pardalos and Xue gives the desired algorithm. \square

4.2 General DAGs

As noted earlier, Algorithm A takes $\Theta(n^3 \log n)$ time in the worst case. Algorithm B shows that the worst-case time can be reduced by first computing the transitive closure. For a DAG with n vertices and m edges it is well-known that the transitive closure can be determined in $O(\min\{nm, n^\alpha\})$ time, where α is such that matrix multiplication can be performed in $O(n^\alpha)$ time. Currently the smallest known value of α is 2.376.

$S(v)$ is initially undefined for all $v \in V$

$\text{pred}(v)$ = set of predecessors of v in G $\text{succ}(v)$ = set of successors of v in G

1. initialize error_q
2. for all $v \in V$
3. for all $u \in \text{pred}(v)$
4. if $f(u) > f(v)$ then { u and v are a violating pair}
5. value = $w\text{mean}(f(u), f(v))$ error = $\text{dist-mean}(f(u), f(v))$
6. add (error, value, u, v) to error_q add (error, value, v, u) to error_q
7. end if
8. end for
9. add $(0, f(v), v, v)$ to error_q
10. end for

11. for $k=1$ to $|V|$ do
12. repeat
13. extract-max (error, value, u, v) from error_q
14. until ($S(u)$ undefined) and (($S(v)$ undefined) or ($S(v) = \text{value}$))
15. $S(u) = \text{value}$
16. for all $x \in \text{pred}(u)$ { $S(u)$ is an upper bound on $S(x)$ }
17. if $f(x) > S(u)$ then insert ($\text{err}(S(u), x), S(u), x, u$) into error_q
18. end for
19. for all $x \in \text{succ}(v)$ { $S(u)$ is a lower bound on $S(x)$ }
20. if $f(x) < S(u)$ then insert ($\text{err}(S(u), x), S(u), x, u$) into error_q
21. end for
22. end for

Algorithm B: Computing $S = \text{Strict}(f, w)$ on DAG $G = (V, E)$ Using Transitive Closure

Algorithm B uses a priority queue, `error_q`, which stores records of the form $(\text{error}, \text{value}, u, v)$, where $u, v \in V$ and $\text{err}(u, \text{value}) = \text{error}$. The priority is based on `error`. Each record represents a constraint on the value and error of $S(u)$ imposed by v , where u and v are a violating pair. The initial constraints, in lines 4–7, are that if u and v are in the same level set then the error and value are as indicated. If $S(v)$ is determined before $S(u)$ and is different than the original constraint imposed by u , then it must have been in another violating pair with even greater error. Thus either v , with value $S(v)$, and u , with value $f(u)$, are no longer a violating pair, or $S(v)$ is closer to $f(u)$ than $f(v)$ was. If they are still a violating pair then a new constraint, with smaller error, was inserted into the queue in lines 16–21 when $S(v)$ was determined. Whether or not there is such an insertion, the original record will be ignored in line 14.

Theorem 4.1 *For any DAG $G = (V, E)$ and weighted function (f, w) on G , given the transitive closure $G' = (V, E')$ of G , Algorithm B determines $S = \text{Strict}(f, w)$ in $\Theta(m' \log m')$ time, where $m' = |E'|$.*

Proof: The following lemma shows that Algorithm B correctly computes `Strict`. The time analysis is straightforward since an edge can, at most, cause an insertion into `error_q` at line 6, and perhaps once again at 17 or 20. The only operations on `error_q` are to insert, remove, or extract-max, and standard priority queue implementations can do each operation in $O(\log m')$ time. \square

Note that, given G' , the time is $\Theta(m' + m'' \log m'')$, where m'' is the number of violating pairs.

Lemma 4.1 *For any weighted function (f, w) and DAG $G = (V, E)$, $S = \text{Strict}(f, w)$.*

Proof: We use proof by contradiction. Let $v \in V$ be the first vertex during the execution of Algorithm B for which $S(v) \neq \text{Strict}(f, w)(v)$. Let $a = \text{Strict}(f, w)(v)$, $b = S(v)$, and let (err_B, b, v, v') be the record that was retrieved (the “current record”) to determine $S(v)$. Assume $b \geq f(v)$ (the proof for $b < f(v)$ being similar). If $a > b$ then $\text{err}(a, v) > \text{err}(b, v) = \text{err}_B$. There must be a $u \prec v$ such that $\text{Strict}(f, w)(u) = a$, $f(u) > f(v)$, and $w\text{mean}(f(u), f(v)) \geq a$. Since u and v are a violating pair, the records representing this were inserted at line 6, with an error $\text{err}\text{-mean}(f(u), f(v)) > \text{err}_B$. Since $S(v)$ was not determined previously, these records must have been ignored when encountered. This can happen only if $S(u)$ was defined previously, but by assumption it was correct, and line 17 would have inserted a record with error greater than err_B , which would have been encountered earlier and Algorithm B would have computed the correct value.

Similarly, if $a < b$, then the current record cannot be one generated on line 20, since that would imply that an earlier calculation by Algorithm B disagreed with Algorithm A. Thus it was a record corresponding to a violating pair, and since $\text{Strict}(f, w)(v') \leq a$, $\text{err}(\text{Strict}(f, w)(v'), v') > \text{err}_B$. There must be a $u \succ v'$ for which $\text{err}\text{-mean}(f(u), f(v')) \geq \text{err}(\text{Strict}(f, w)(v'), v')$, and hence, as before, Algorithm B would have determined $S(v')$ before the current record was retrieved. \square

Given the transitive closure, `Prefix` and `Basic` can be computed in $\Theta(m')$ time. For `Prefix`, the g function in the definition given in Section 2 can easily be computed in $\Theta(m')$ time, as can the minimum operation. For `Basic` the computations are somewhat more complex. As noted by Kaufman and Tamir [12], for any set S , $w\text{mean}(S)$ can be computed in $\Theta(|S|)$ time by using techniques developed by Megido [21]. Using this, at each vertex one can compute `Basic` in time linear in the number of predecessors and successors, and hence in $\Theta(m')$ total time. Apparently the only previous discussion of the computational complexity of `Basic` for weighted data is by Angelov et al. [9], who also noted that `Basic` can be computed via the transitive closure. Given the increase in the complexity, from an implementation viewpoint `Prefix` seems preferable to `Basic`.

5 Properties of Regression Mappings

In this section we compare regression mappings in terms of the regressions they produce, instead of the time required to compute them. We consider their behavior on closely related functions and show that Strict has properties in common with L_p regression, $1 < p < \infty$, properties that Max, Min, and Avg do not have.

The results below show that only three vertices are needed to exhibit the properties under consideration. One minor fact is that for any connected DAG $G = (V, E)$ of at least three vertices, at least one of the following is true:

- a) There are vertices v_1, v_2, v_3 such that $v_1 \prec v_2 \prec v_3$, i.e., G has a *chain* of length 3.
- b) There is a vertex w such that $v \prec w$ for all $v \in V - w$, i.e., G has an *upward star* with *center* w .
- c) There is a vertex w such that $w \prec v$ for all $v \in V - w$, i.e., G has a *downward star* with *center* w .

Given a set V , let $<^p$ denote the natural pointwise ordering of real-valued functions over V , i.e., $g <^p h$, $g \neq h$, iff for all $v \in V$, $g(v) \leq h(v)$. A regression mapping T on DAG $G = (V, E)$ is *monotonic* iff for all weight functions w and functions f and g on V , if $f <^p g$ then $T(f, w) \leq^p T(g, w)$.

Proposition 5.1 *For any connected DAG G with at least three vertices:*

1. Strict, Prefix, and Basic are monotonic mappings,
2. Avg is not monotonic, though it is monotonic when restricted to unweighted functions,
3. Max and Min are not monotonic even when restricted to unweighted functions.

Proof: The monotonicity of Prefix, Basic and Strict follows from the fact that all of the operations involved in their calculation are monotonic. Avg is the same as Basic on unweighted functions, and hence is monotonic on them.

To prove that Avg is not monotonic in the general weighted case, suppose G has the chain $v_1 \prec v_2 \prec v_3$ or the upward star $v_1, v_2 \prec v_3$. In this case $w = (1, 2, 2)$, $f = (1, 1, 1)$, and $g = (1, 2, 1)$ shows the nonmonotonic behavior. If there are additional vertices then set their w , f , and g values to 1. If G does not contain a chain or upward star then it must be a downward star with center v_1 and two incomparable vertices v_2 and v_3 . Using $w = (2, 2, 1)$, $f = (1, 0, 1)$ and $g = (1, 1, 1)$ provides an appropriate example. If there are additional vertices then set their w , f , and g values to 1. Similar examples can be found for Max and Min, without the need for unequal weights. \square

For a set V and weighted function (f, w) on V , let $S \subset V$. The *trim of (f, w) on S* is the function h on V where

$$h(v) = \begin{cases} \text{wmean}(f|S) & \text{if } v \in S, \\ f(v) & \text{otherwise.} \end{cases}$$

A regression mapping T on a DAG $G = (V, E)$ *preserves level set trimming* iff for any weighted function (f, w) on V and any level set L of $T(f, w)$, if h is the trim of (f, w) on L then $T(h, w) = T(f, w)$. Level set trimming is preserved by L_p isotonic regression for all $1 < p < \infty$.

Trimming the left level set of the example in the introduction (i.e., using $h(1) = h(2) = 0$ and $h(3) = 1$) shows that Basic, Prefix, Max, Min, and Avg do not preserve level set trimming on chains of length 3 even for unweighted data. In contrast, Algorithm A determines Strict by trimming. That is, the definitions of the

$[L(v, \epsilon), B(v, \epsilon)]$ windows for each s trim all of the level sets defined in earlier stages, i.e., those with larger regression error.

The proof of the following is similar to previous proofs and is omitted.

Proposition 5.2 *For any connected DAG G of at least three vertices,*

1. *Strict preserves level set trimming,*
2. *Prefix preserves level set trimming if G is an upward star, but otherwise does not preserve it even when restricted to unweighted functions,*
3. *Max, Min, Avg, and Basic do not preserve level set trimming even when restricted to unweighted functions.*

□

Note added in proof: one can show that for any DAG, if a regression operator R is monotonic and preserves level set trimming, then $R = \text{Strict}$. See the Appendix.

6 Final Remarks

For a given weighted function, the wide swath of isotonic functions that minimize the L_∞ regression error has helped researchers design algorithms for finding one [6,11,12]. Unfortunately, previous algorithms generate regressions that have unnecessarily large errors at many points, and the regressions for closely related functions, such as ones related via monotonic changes, are not well-behaved. In this sense, the fastest algorithms (Max, Min and Avg) are the least desirable. In contrast, strict L_∞ isotonic regression minimizes the large errors, and, like L_p isotonic regression, $1 < p < \infty$, obeys properties such as monotonicity and preservation of level set trimming. It is the “best best” L_∞ approximation [13–15].

Strict is characterized by its level set structure (Proposition 3.1), and others have considered this structure [6,14,15], but the author is unaware of it leading to efficient algorithms, other than PAV. Determining Strict by refining the L_∞ norm seems to be a new approach which leads to useful algorithms. This approach also extends to more general settings. Suppose at each vertex v there is a nonnegative penalty function $p_v(r)$ for using value r at v , and the goal is to find an isotonic function that minimizes large penalties. As long as each p_v has a minimum at some x_v , is continuous and decreasing on $(-\infty, x_v]$ and continuous and increasing on $[x_v, \infty)$, then for any level set the unique value minimizing the worst penalty is determined by a violating pair. Further, Algorithms A and B can be applied to find the optimal isotonic function.

Algorithm B shows that for general DAGs, Strict can be computed in the same time as the fastest known algorithms for Basic and Prefix, since all are based on first determining a transitive closure. If the graph is given as its own transitive closure then Strict can be computed as quickly as Min, Max and Avg, and only a factor of $\log n$ slower than Basic and Prefix (though one could reduce the time to find Min, Max and Avg by first using Prefix to determine the regression error).

Fortunately, for important classes of DAGs Strict can be computed significantly faster than the size of the transitive closure. Proposition 4.1 showed that for a tree ordering Strict can be computed in $\Theta(n \log n)$ time, and in [11] it is shown that when the vertices are points in d -dimensional space with the natural componentwise ordering, i.e., $(x_1, \dots, x_d) \prec (y_1, \dots, y_d)$ iff $x_i \leq y_i$ for $1 \leq i \leq d$, then Strict can be determined in $\Theta(n \log^{d+1} n)$ time.

Finally, L_1 isotonic regression can also fail to be unique since the L_1 mean (the median) can be an interval rather than a single value. A natural choice is to use strict L_1 isotonic regression, i.e., the limit, as $p \rightarrow 1$, of L_p isotonic regression. A discussion of the appropriate choice of median appears in [22]. Apparently no algorithms have yet been published for determining strict L_1 isotonic regression, though PAV can be used for linear and tree orderings.

References

1. Ayer, M., Brunk, H.D., Ewing, G.M., Reid, W.T., Silverman, E.: An empirical distribution function for sampling with incomplete information. *Annals of Math. Stat.* 5, 641–647 (1955).
2. Barlow, R.E., Bartholomew, D.J., Bremner, J.M., Brunk, H.D.: *Statistical Inference Under Order Restrictions: the Theory and Application of Isotonic Regression*. Wiley (1972).
3. Gebhardt, F.: An algorithm for monotone regression with one or more independent variables. *Biometrika* 57, 263–271 (1970).
4. Hanson, D.L., Pledger, G., Wright, F.T.: On consistency in monotonic regression. *Annals of Stat.* 1, 401–421 (1973).
5. Robertson, T., Wright, F.T., Dykstra, R.L.: *Order Restricted Statistical Inference*, Wiley (1988).
6. Ubhaya, V.A.: Isotone optimization, I, II. *J. Approx. Theory* 12, 146–159, 315–331 (1974).
7. Punera, K., Ghosh, J.: Enhanced hierarchical classification via isotonic smoothing. *Proc. Int’l. Conf. World Wide Web*, 151–160 (2008).
8. Zheng, Z., Zha, H., Sun, G.: Query-level learning to rank using isotonic regression. *Proc. 46th Allerton Conf.*, 1108–1115 (2008).
9. Angelov, S., Harb, B., Kannan, S., Wang, L-S: Weighted isotonic regression under the L_1 norm. *Symp. Discrete Algorithms*, 783–791 (2006).
10. Barlow, R.E., Ubhaya, V.A.: Isotonic approximation. In Rustagi, J.S. (ed): *Proc. Optimizing Methods in Stat.*, pp. 77–86, Academic Press (1971).
11. Stout, Q.F.: Algorithms for L_∞ isotonic regression, submitted (2011).
12. Kaufman, Y., Tamir, A.: Locating service centers with precedence constraints. *Discrete Applied Math.* 47, 251–261 (1993).
13. Cuesta, J.A., Matrán, C.: Conditional bounds and best L_∞ -approximations in probability spaces. *J. Approx. Theory* 56, 1–12 (1989).
14. Darst, R.B., Sahab, S.: Approximations of continuous and quasi-continuous functions by monotone functions. *J. Approx. Theory* 38, 9–27 (1983).
15. Legg, D., Townsend, D.: Best monotone approximation in $L_\infty[0,1]$. *J. Approx. Theory* 42, 30–35 (1984).

16. Pólya, G.: Sur une algorithmme toujours convergent pour obtenir les polynomes de meillure approximation de Tchebycheff pour un fonction continue quelconque. *Comptes Rendus* 157, 840–843 (1913).
17. Ahuja, R.K., Olin, J.B.: A fast scaling algorithm for minimizing separable convex functions subject ot chain constraints. *Oper. Res.* 49, 784–789 (2001).
18. Stout, Q.F.: Unimodal regression via prefix isotonic regression. *Comp. Stat. Data Anal.* 53, 289–297 (2008).
19. Thompson, W.A. Jr.: The problem of negative estimates of variance components. *Annals Math. Stat.* 33, 273–289 (1962).
20. Pardalos, P.M., Xue, G.: Algorithms for a class of isotonic regression problems. *Algorithmica* 23, 211–222 (1999).
21. Megido, N.: Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Computing* 12, 759–776 (1983).
22. Jackson, D.: Note on the median of a set of numbers. *Bull. Amer. Math. Soc.* 27, 160–164 (1921).

Appendix

The following shows that monotonicity and trimming completely characterize Strict. It does not appear in the version of the paper appearing in *Journal of Optimization Theory and Applications* since I only noticed it after that paper was in the publication processes.

Theorem 6.1 *For any DAG $G = (V, E)$ and regression operator R on G , if R is monotonic and preserves level set trimming, then $R = \text{Strict}$.*

Proof: We use proof by contradiction. Suppose R is monotonic and preserves level set trimming, and there is a weighted function (f, w) for which $R(f, w) \neq \text{Strict}(f, w)$. Among the level sets of Strict which are not level sets of R , or where the regression values differ, let A be one of maximal error. Let f_1 be f trimmed on all level sets of Strict with error greater than A 's. Then $\text{Strict}(f_1, w) = \text{Strict}(f, w)$, $R(f_1, w) = R(f, w)$, and the regression error of $\text{Strict}(f_1, w)$ is its error on A (there may be other level sets with the same error). Let C be the value of Strict on A . If $R(f_1, w)$ does not equal C on all of A then it has larger regression error than does $\text{Strict}(f_1, w)$, in which case it is not an isotonic regression. Otherwise, it has a level set $B \supsetneq A$ with regression value C . Let $B' = \{u : u \in B, \text{Strict}(f_1, w)(u) < C\}$ (if B' is empty then a similar proof can be applied to the set where $\text{Strict} > C$). Since $\text{Strict}(f_1, w) < C$ on B' , and raising the values to C would not violate the isotonic condition, it must be that $w\text{mean}(B') < C$.

Let f_2 be the function formed by trimming f_1 on all level sets of $R(f_1, w)$ except B . Then $R(f_2, w) = R(f_1, w)$. Define $f_3(u)$ to be $f_2(u)$ if $f_2(u) < C$ or $u \in B'$, and M otherwise, where M is the maximum value of f_2 . Since $R(f_2, w) = C$ on B' and $f_3 \geq^p f_2$, by monotonicity $R(f_3, w) \geq C$ on B' . Let h be the function where $h(u) = f_3(u)$ on $V \setminus B'$, and $h(u) = \max\{w\text{mean}(B'), D\}$ on B' , where D is the largest value of f_3 less than C . Then h is isotonic, and as a regression of f_3 has no error on $V \setminus B'$ and smaller error than $R(f_3, w)$ on B' . Thus $R(f_3, w)$ is not optimal and hence R is not a regression operator on G . \square