

L_0 Isotonic Regression With Secondary Objectives

Quentin F. Stout

qstout@umich.edu

www.eecs.umich.edu/~qstout/

Abstract: We provide algorithms for isotonic regression minimizing L_0 error (Hamming distance). This is also known as monotonic relabeling, and is applicable when labels have a linear ordering but not necessarily a metric. There may be exponentially many optimal relabelings, so we look at secondary criteria to determine which are best. For arbitrary ordinal labels the criterion is maximizing the number of labels which are only changed to an adjacent label (and recursively apply this). For real-valued labels we minimize the L_p error. For linearly ordered sets we also give algorithms which minimize the sum of the L_p and weighted L_0 errors, a form of penalized (regularized) regression. We also examine L_0 isotonic regression on multidimensional coordinate-wise orderings.

Keywords: isotonic regression, monotonic relabeling, L_0 , Hamming distance, ordinal response, distance to monotonicity

1 Introduction

There are many scenarios when one expects that an attribute of objects increases as a function of other attributes. For example, if people’s weight is categorized as {underweight, normal, overweight, obese}, one would expect that if the daily intake of fat is held constant then the categorization increases as the carbohydrate intake increases, and similarly it increases if the daily carbohydrate intake is held constant and the fat intake increases. No assumptions are made about the weight of a person with higher fat, but lower carbohydrate, vs. one with lower fat but higher carbohydrates. However, noisy datasets may have data that violates these assumptions. Constructing a representation of the data which obeys the assumptions and has minimal changes to the data is generically known as isotonic regression. In this example the dependent variable, weight categorization, is a label with no assumed metric, only an ordering, and the problem is more commonly known as monotonic relabeling.

As datasets become increasingly complex, often with only ordinal ordering on some attributes, such non-parametric regressions become increasingly more useful, but are often nonunique and difficult to compute. We will use optimizations based on secondary criteria to choose among the possible regressions, and give algorithms to find the resulting regressions.

More precisely, let V be a set with a partial order \prec , and \mathcal{L} be a linearly ordered set. A *label function* f is a mapping $f : V \rightarrow \mathcal{L}$. Given label functions f, g , the L_0 distance between them, $\|f - g\|_0$, is

$$\sum_{v \in V} \mathbf{1} \cdot (f(v) \neq g(v))$$

This is also known as the *Hamming distance*, 0-1 loss, or Kronecker delta loss. It is similar to the well-known L_p distance when the labels are real values:

$$\|f - g\|_p = \begin{cases} (\sum_{v \in V} |f(v) - g(v)|^p)^{1/p} & 1 \leq p < \infty \\ \max_{v \in V} |f(v) - g(v)| & p = \infty \end{cases}$$

Here we consider two classes of label values: one is when $\mathcal{L} = \{\lambda(1), \dots, \lambda(\ell)\}$ for a positive integer ℓ , where $\lambda(1) < \dots < \lambda(\ell)$. The other is where \mathcal{L} is the real numbers. The former is used in papers which describe the problem as “relabeling”, such as [26], while the latter is used in papers on function approximation and penalized regression. Our results are applicable in both settings.

A label function g is *monotonic (isotonic)* if whenever $u \prec v$, for $u, v \in V$, then $g(u) \leq g(v)$, i.e., it is a weakly order-preserving mapping from V to \mathcal{L} . Let $\Delta_p(f) = \min\{\|f - g\|_p : g \text{ is monotonic}\}$. This is the L_p distance of f to monotonicity. The term *distance to monotonicity*, without mention of p , typically means L_0 distance, and often the goal is to quickly estimate it, rather than determine it exactly [5, 15, 17, 19]. Such estimation, or the mere decision if a function is monotonic, has often been used in property testing [4, 5, 6, 10, 13, 15, 27]. Here we are interested in the exact value of $\Delta_0(f)$.

A monotonic function g is an L_p -optimal monotonic relabeling, or an L_p isotonic regression, of label function f iff $\|f - g\|_p = \Delta_p(f)$. For $p \geq 1$ this requires that the labels are real values. L_0 -optimal monotonic relabelings need not be unique, e.g., on a sequence of just 2 points, if f is `overweight`, `normal`, then `normal`, `normal` and `overweight`, `overweight` are both optimal monotonic relabelings. Since optimal L_0 monotonic relabelings are not always unique, how should one decide which to use? Here we decide by using L_1 , L_2 , or L_∞ distances to select among them. This question was also raised in [25], where a similar selection approach was used, but theirs required the regression values to be in a fixed finite set \mathcal{L} , while we consider arbitrary real-valued regressions as well. We also give more efficient algorithms for the cases they considered.

When f is real-valued and $p \in \{1, 2, \infty\}$ three scenarios are considered:

1. Find an isotonic function g such that $g \in \arg \min\{\|f - h\|_p : h \text{ isotonic and } \|f - h\|_0 = \Delta_0(f)\}$. Note that g is an L_0 optimal isotonic relabeling of f . g will be called $L_{0,p}$ optimal.
2. Given a subset C of vertices on which f is isotonic, where $|C| = n - \Delta_0(f)$, find an isotonic function g such that $g \in \arg \min\{\|f - h\|_p : h \text{ isotonic and } h = f \text{ on } C\}$. Note that for any function g' optimizing the criterion in 1) there is a C' of size $n - \Delta_0(f)$ where f is isotonic on C' and g' optimizes the criterion here for C' . However, there may be C for which a function optimizing the criterion here does not optimize that in 1). g will be called $L_{0,p|C}$ optimal, or *weakly $L_{0,p}$ optimal*.
3. Given $\alpha > 0$, find an isotonic function g such that $g \in \arg \min\{\|f - h\|_p + \alpha\|f - h\|_0 : h \text{ isotonic}\}$. Note that it may be that $\|f - g\|_0 > \Delta_0(f)$, while if α is sufficiently large g is in $L_{0,p}$.

When the labels \mathcal{L} are a finite set of real numbers we consider variants of 1) and 2) where all isotonic regressions must be \mathcal{L} -valued. We also introduce strong and weak versions of \mathcal{L} -valued functions with which iteratively optimize L_0 for a decreasing set of vertices. In this case \mathcal{L} merely needs to be ordered, not a subset of the reals.

We give algorithms for all of these when the dag is a linear order, but only for some of them when the dag is more complex.

As will be shown in Section 2, for an arbitrary dag most of the computational work involves a “violator dag”, a directed acyclic graph that represents where the monotonicity condition is violated. To help keep track of which ordering or dag is being used, we use $G = (V, E)$ to denote the initial dag and $\widehat{G} = (\widehat{V}, \widehat{E})$ to denote the violator dag. We assume V is connected, and if it isn’t then the construction is used on each component independently. Let $n = |V|$, $m = |E|$, $\widehat{n} = |\widehat{V}|$ and $\widehat{m} = |\widehat{E}|$.

By *multidimensional ordering* or *d-dimensional ordering* we mean that the vertices are from some d -dimensional space, where each dimension has a linear ordering, and the vertices are ordered by component-wise ordering, i.e., for $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$, $x \prec y$ iff $x \neq y$ and $x_i \leq y_i$ for all $1 \leq i \leq d$. This is also known as the product order of linear orders. While in general violator graphs may have $\widehat{m} = \Theta(n^2)$, in Section 4 we use a Steiner dag (a dag with additional vertices) as the violator graph

for d -dimensional points, with $\widehat{m} = \widetilde{\Theta}(n)$. The size of the violator graph, and time required to construct it, are significantly smaller than previous algorithms, and these orderings seem to be the ones of most interest in many applications.

In many applications, for a label set \mathcal{L} of size ℓ , ℓ might be quite small and independent of n , such as in the {underweight, normal, overweight, obese} example, and so some of the results are stated in terms of ℓ . To simplify exposition we assume that the labels are $1 \dots \ell$, but for L_p secondary optimization where the labels are real numbers we use the true values.

We do not examine any specific application, concentrating instead on the algorithms and reducing the time so that they can be applied to large problems. Previous algorithms [16, 21, 26] are reviewed in Section 2. Examples and applications of L_0 isotonic regressions and monotonic relabeling appear in many papers, such as [14, 16, 24, 25, 26, 33]. There are many applications of it to monotonic learning and classification, see [7, 9] and the extensive references therein. However, in some classification settings one must be careful about how monotonicity is applied [3].

2 Background

There are some special cases which have simple solutions. Minimizing the number of changed labels is equivalent to maximizing the number of unchanged ones, and hence if \prec is a simple linear ordering then the problem is the same as finding a nondecreasing subsequence of maximum length. This well-known problem has an easy $\Theta(n \log \ell)$ solution. When \prec is an arbitrary order but $\ell = 2$ then the problem is equivalent to $\{0, 1\}$ -valued L_1 isotonic regression, for which numerous algorithms are known [1, 2, 8, 12, 29, 30]. These algorithms vary widely in their time and construction as a function of the underlying order. If it is linear or a 2-dimensional grid then the regression can be found in $\Theta(n)$ time, an arbitrary set of vertices in d dimensions with d -dimensional ordering in $\Theta(n^{1.5} \log^{d-1} n)$ time, and an arbitrary dag in $\Theta(nm + n^2 \log n)$ time. The first three of these results appear in [29, 30], while that for arbitrary dags appears in [2]. For integer-valued weights recent results have lowered the time for arbitrary dags to nearly linear in m [8, 12], where the time is achieved with high probability.

Similarly there is a range of times and algorithms for L_0 isotonic regression depending on the underlying dag, but an extra aspect, namely secondary criteria, causes yet more complexity. The general case, without such criteria, was analyzed in [16, 23, 25], while here we introduce a faster algorithm for multidimensional vertices (Section 4) and various algorithms optimizing secondary criteria.

Many of the algorithms utilize maximum flow algorithms. The time of our algorithms is analyzed in terms of $\mathcal{T}(\tilde{n}, \tilde{m})$, the time of the flow algorithm on a graph of \tilde{n} vertices and \tilde{m} edges. We use $\widetilde{\Theta}(\cdot)$ to denote that logarithmic factors in terms of \tilde{n} are omitted. Algorithms in [16, 23, 25] had $\mathcal{T} = \Theta(n^3)$. This is what occurs when you apply the faster of Orlin's algorithm and King-Rao-Tarjan's to the standard violator dag since there the worst case has $\tilde{n} = \Theta(n)$ and $\tilde{m} = \Theta(n^2)$. Faster flow algorithms are now known, though with some constraints on the flow capacities and/or probability of attaining the expected time. If all weights and capacities are integers in the range $[1, U]$ then the BLLSSSW algorithm [8] takes $\widetilde{O}(\tilde{m} + \tilde{n}^{\frac{3}{2}} \log U)$ time and the CKLPPS algorithm [12] takes $O(\tilde{m}^{1+o(1)} \log^2 U)$ time, both with high probability. From now on we make the common assumption that U at most polynomial in n , and hence all of the terms involving U are absorbed in the \widetilde{O} notation, and thus currently \mathcal{T} is no worse than $\widetilde{O}(\min\{\tilde{m}^{1+o(1)}, \tilde{n}^{\frac{3}{2}}\})$ with high probability.

Flow algorithms are still undergoing improvement, and the wikipedia page [34] is usually up-to-date with the latest improvements. In practice simpler, but asymptotically slower, algorithms are faster on small datasets, but we are more interested in performance for the ever increasing size of data collections. Since the time of our algorithms is expressed in terms of \mathcal{T} one can interpret them in terms of whatever flow algorithm is used, not necessarily the asymptotically fastest.

1. Create a violator dag $\widehat{G} = (\widehat{V}, \widehat{E})$, where for all $u, v \in V$, there is a path from u to v in \widehat{G} iff $u \prec v$ and $f(u) > f(v)$. {Note: $V \subset \widehat{V}$.}
2. Find a maximum antichain C of \widehat{G}
 - (a) Create a flow graph \widehat{G}_f from \widehat{G} .
 - (b) Find a minimum flow on \widehat{G}_f and use this to determine C .
3. Determine f' :
 - (a) For all $v \in V$ determine the window $[w_{C \prec}(v), w_{C \succ}(v)]$ induced by C .
 - (b) Let f' be any isotonic function on V that goes through the windows. Since $w_{C \prec}(v) = w_{C \succ}(v) = f(v)$ for $v \in C$, $f' = f$ on C .

Figure 1: L_0 -optimal monotonic relabeling f' of label function f on $G = (V, E)$ with order \prec (see [16, 23, 26])

For the general case of L_0 isotonic regression on arbitrary dags, [16, 23, 25] used an approach based on violating pairs. Given a label function f , vertices $p, q \in V$ are a *violating pair* iff $p \prec q$ and $f(p) > f(q)$, i.e., they violate the monotonicity requirement. Let $p \prec_v q$ denote that p, q are a violating pair. Note that \prec_v is transitive, and hence is a partial ordering on the vertices in V . Their approach is outlined in Figure 1. Detailed proofs appear in their papers, but the basic idea is to create a dag based on violating pairs and find an antichain of maximum size in it. Let $\widehat{G} = (V, \widehat{E})$ be a dag where there is a path from p to q in \widehat{G} iff $p \prec_v q$ in G . \widehat{G} is a *violator dag*. While V is well-defined, there may be multiple sets of edges which give the same partial ordering. The construction of \widehat{G} in step 1) will be analyzed later.

Vertices in $V \setminus \widehat{V}$ are not in any violating pair, i.e., on them f is monotonic with respect to all other vertices, and hence in any optimal relabeling their label does not change. Step 2) determines those elements C of \widehat{V} where the label will not be changed. [16, 26] borrow an idea from [22] to determine C . No two elements of C can be a violating pair, i.e., are not comparable in the \prec_v ordering. A set of incomparable elements in a partial order is an antichain, and hence in an optimal relabeling C is a maximum antichain.

Here, rather than emphasizing the antichain's graph properties we look at its subsequent role in constructing an isotonic function. A set $C \subseteq V$ is *f-isotonic* iff f is isotonic on C . Thus a set of vertices is an antichain of maximum size iff it is an *f-isotonic* set of maximum size on the set of vertices that are in violator pairs. From now on we will slightly abuse terminology and say that C is an *f-isotonic* set of maximum size if it is an *f-isotonic* set of maximum size on the set of vertices that are in violator pairs, where C also implicitly contains all vertices not in any violator pair. An isotonic function g is *consistent* with an *f-isotonic* set C iff $g = f$ on C .

Mohring [22] showed how to find C via minimal flows. Figure 2 shows a dag G and one of its violator dags \widehat{G} , which happens to be its transitive reduction. \widehat{G} is transformed into a flow graph \widehat{G}_f as follows: each vertex u of \widehat{G} is replaced by two vertices $u^{\text{in}}, u^{\text{out}}$, where each incoming edge to u becomes an incoming edge to u^{in} with 0 minimum flow, each outgoing edge from u becomes an outgoing edge from u^{out} with minimum flow 0, and there is an edge from u^{in} to u^{out} with minimum flow 1.

Let F be a minimum flow on \widehat{G}_f , i.e., a flow where on each edge is at least the weight on that edge, and the total flow is as small as possible. From F one can determine a maximal cut, and the edges of weight 1 in this cut correspond to C . The size of C is the flow, and it is straightforward to show that C is an antichain.

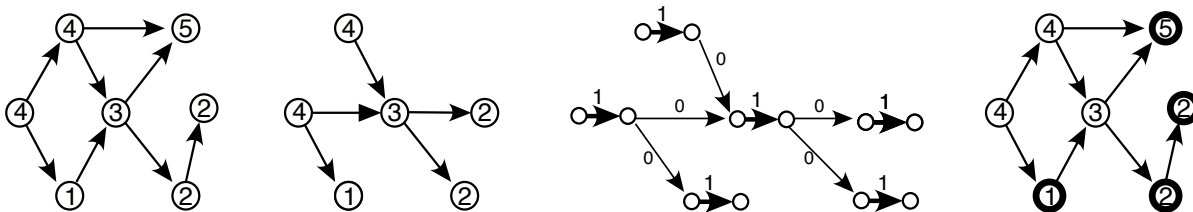


Figure 2: Label function on dag G , one of its violator dags \widehat{G} , flow graph \widehat{G}_f , resulting C

Thus the time of step 3) is the time to find a minimal flow on \widehat{G}_f . The flow is at most $|\widehat{V}|$ and \widehat{G}_f has $\Theta(|\widehat{G}|)$ vertices and $\Theta(|\widehat{E}|)$ edges.

Step 3) determines f' . At each such vertex v there is a “window” of label values $[b(v), t(v)]$ into which the new label must fall. This is the range of values that are monotonically compatible with the values in V' , i.e., the values at vertices where the function will not change. The windows are isotonic in that the bottom values are an isotonic function on V , as are the top values. Any isotonic function within these windows (such as always using the lower bound) can be used as f' , and simple topological sort can be used to finish this step in $\Theta(\widehat{m})$ time. However, one may want to find an f' optimizing secondary objectives, the main contribution of this paper, which adds complexity.

Returning to step 2), there are several ways \widehat{G} can be created. If \prec is given explicitly via dag $G = (V, E)$ then \widehat{G} can be constructed via topological sorting, taking $\Theta(nm)$ time, or via matrix multiplication, taking $\Theta(n^\omega)$ time, where ω is such that matrix multiplication over the integers can be done in $\Theta(n^\omega)$ time. For multidimensional vertices \widehat{G} can be constructed by pairwise comparisons, taking $\Theta(n^2)$ time, and in Section 4 it is shown that this time can be reduced.

While a dag specifies a partial ordering, there may be multiple dags specifying the same partial ordering. For example, a dag with an edge for each pair of vertices in the transitive closure, and one with an edge for each pair in the transitive reduction, specify the same partial order, but the latter may be significantly smaller. Both can be constructed in $\Theta(\min\{\widehat{nm}, \widehat{n}^\omega\})$ time, and therefore [26] uses the transitive reduction for \widehat{G} .

If the data is almost in order and there are few violating pairs then \widehat{G} may be much smaller than G . However, in the worst case even the transitive reduction of \widehat{G} can be quite large. For example, on the linear order on $V = \{1, 2, \dots, n\}$, for n even, suppose f is $\frac{n}{2}+1, \frac{n}{2}+2, \dots, n, 1, 2, \dots, \frac{n}{2}$. Then for $p, q \in V$, $p \prec_v q$ iff $p \in [1, \frac{n}{2}]$ and $q \in [1+\frac{n}{2}, n]$. The transitive closure and reduction are the same, with $\widehat{n} = n$ and $\widehat{m} = n^2/4$. Here $\widehat{m} \gg m$. Further, there can be exponentially maximum f -isotonic sets, as 2, 1, 4, 3, 6, 5, ... shows since one needs to choose 1 from each pair of the form $k, k-1$.

Note that [2] gives an L_1 isotonic regression algorithm that is based on violating pairs and flows but does not require constructing a potentially large violator graph.

3 Secondary Optimality

For general label functions there can be many maximum f -isotonic sets, and for each there may be many possible values for the remaining vertices as long as they satisfy monotonicity. For example, given label values small < medium < large < X-large, for a linearly ordered set with data values large, medium, small, any one of these values provides a maximum f -isotonic set. However, many researchers might prefer using {medium}, with regression values medium, medium, medium. However, even with this choice of maximum f -isotonic set, small, medium, large is also L_0 optimal, as is small, medium, X-large. This raises the questions: which maximum f -isotonic set should be used, and what should the regression

values be for the remaining vertices? This question was also raised in [25].

When the labels are real numbers one way to choose the best L_0 regression is to minimize L_p error. That is, given $1 \leq p \leq \infty$, for a given label function f choose an isotonic function $g \in \arg \min\{\|f - h\|_p : h \text{ isotonic}, \|f - h\|_0 = \Delta_0(f)\}$. We call this *strong optimality*, denoted $L_{0,p}(f)$. A somewhat weaker optimality is to first choose an f -isotonic set C of maximum size and then minimize the L_p error of all isotonic functions which agree with f on C , i.e., find a $g \in \arg \min\{\|f - h\|_p : h \text{ isotonic}, f = h \text{ on } C\}$. We call this *weak optimality*, denoted $L_{0,p|C}(f)$. Note that for any strongly optimal $g \in L_{0,p}(f)$ there is an f -isotonic set C of maximal size such that $g \in L_{0,p|C}(f)$. For example, for function values 3, 2, 1 on a linear order, for $1 \leq p \leq \infty$, using $C = \{x\}$, $x \in \{1, 2, 3\}$, the isotonic regression function x, x, x is in $L_{0,p|C}(f)$, and 2, 2, 2 is in $L_{0,p}(f)$.

In general finding weakly optimal functions is easier than finding strongly optimal ones. Throughout we examine the values of p most widely utilized, namely $p \in \{1, 2, \infty\}$. More generally one could consider $L_{p,q}$, optimizing the L_q norm among all isotonic regressions that optimize the L_p norm. For $1 < p < \infty$ this is not interesting since there is a unique L_p isotonic regression. For $p = 1$ there may be some variation possible, and for $p = \infty$ there are often many optimal regressions, much like L_0 . However, this general setting will not be considered here.

One may want L_0 as a secondary criterion, not just the primary one, an approach that can be used when the labels merely have an ordering with no metric. For a weak version with a fixed set \mathcal{L} of labels let C_0 be an f -isotonic set of maximal size. Regression values at vertices not in C_0 must be 1 or more labels away from their original value. Among these vertices find an f -isotonic set C_1 of maximum size where all regression values change to adjacent values, and for these vertices pin their values to their trimmed value. This shrinks the windows, and now all vertices that aren't pinned must have regression values 2 or more away from their trimmed value. Find an f -isotonic set C_2 fitting within the windows which maximizes the number of vertices 2 labels away from their trimmed value, etc. An isotonic regression obtained this way is called a *weak $L_{0,0|C}$ regression*. While the size of C_0 is unique, the size of C_1 depends on the choice of C_0 , the size of C_2 depends on C_0 and C_1 , etc.

When the labels are real-valued $L_{0,0|C}$ is not well defined. Instead, for arbitrary real-valued labels let g be an isotonic function where all its values are label values, and let (e_1, e_2, \dots, e_n) be the values $\{|f(v) - g(v)| : v \in V\}$ in sorted order. Viewing this as a sequence of n values, a *strong $L_{0,0}$ isotonic regression* is one which has a sequence lexicographically first among all isotonic functions (this may not be unique). We call an L_0 isotonic regression strong if it is a strong $L_{0,0}$ function.

3.1 Label- vs. Real- Valued Regressions

For $L_{0,p}$ or $L_{0,p|C}$ isotonic regression, $1 \leq p$, in some settings the values of the initial function f must be from a set \mathcal{L} of real-valued labels, while in others they can be arbitrary real numbers. Further, in the former one may want the values of the isotonic regression to be from \mathcal{L} or from arbitrary reals. Since most isotonic regression algorithms are developed for regression values being arbitrary real numbers, or integers, when the values must be from \mathcal{L} the values of an optimal arbitrary real-valued regression need to be converted to labels. Let g be an optimal real-valued isotonic regression for whatever metric is being considered. To construct a g' which is an optimal \mathcal{L} -valued regression at vertex x it suffices to let $g'(x)$ be $g(x)$ rounded to the nearest value of \mathcal{L} , or always rounded down when the nearest larger and nearest smaller values are at the same distance (always rounding up when there is a tie also works). This clearly maintains the isotonic requirement. However, for different values of p different proofs of optimality are needed.

For $p = 1$ one can always have an optimal real-valued regression where all the values are in the set of values of f . Since we are only concerned with the case where f and g' are \mathcal{L} -valued, a $L_{0,1|C}$ or $L_{0,1}$ regression can be chosen to be \mathcal{L} -valued with the same L_1 error of an optimal real-valued regression without

constraints on the regression values. However, it may be that the real-valued regression returned by an L_1 algorithm produces a g where not all values are in \mathcal{L} . For example, if f is 1, 0 then g can be of the form α , α for any $\alpha \in [0, 1]$. If $\mathcal{L} = \{0, 1\}$ then the values of g must be adjusted. In general, if the value x of a level set S is not a label value then anything from $\max\{\ell : \ell \leq x, \ell \in \mathcal{L}\}$ to $\min\{\ell : \ell \geq x, \ell \in \mathcal{L}\}$ is an optimal regression value of S , so for $L_{0,1}$ and $L_{0,1|C}$ setting the regression value of S to be the closest of these, and always rounding down in case of ties, maintains optimality and isotonicity.

For $p = \infty$, throughout we assume that the basic L_∞ isotonic regression for unweighted data is used: for any vertex x , let $\alpha(x) \in \arg \max\{f(u) : u \preceq x\}$ and $\beta(x) \in \arg \min\{f(v) : x \preceq v\}$. Then the real-valued regression value $g(x)$ is $(f(\alpha(x)) + f(\beta(x)))/2$. If a regression value $g'(x) = c$ is used then the error at $\alpha(x)$ is $\geq f(\alpha(x)) - c$, and the error at $\beta(x)$ is $\geq c - f(\beta(x))$, so minimizing the distance from c to x minimizes the L_∞ error imposed by the value of $g'(x)$.

For $p = 2$, for any level set S its optimal regression value is the average of the values in it. If the regression value is x , then for any $\epsilon > 0$, the error of using $x - \epsilon$ as the regression value for S is the same as the error of using $x + \epsilon$, and the error is monotonic in ϵ . Therefore for an \mathcal{L} -valued function the error contributed by S is minimized by setting the regression value to be the closest value of \mathcal{L} , or rounding down if there are two closest values. As before, this also preserves isotonicity.

3.2 Weak $L_{0,\infty|C}$ and Strong $L_{0,\infty}$

Given an f -isotonic set C , for every vertex $v \in V$, v 's C -window is $[w_{C\prec}(v), w_{C\succeq}(v)]$, where $w_{C\prec}(v) = \max\{f(u) : u \in C, u \preceq v\}$ (it is $-\infty$ if v has no predecessor in C) and $w_{C\succeq}(v) = \min\{f(u) : u \in C, u \succeq v\}$ (or $+\infty$ if v has no successor in C), i.e., the window is the range of possible values an isotonic regression can have at v , given that the regression values on C are the values of f . If C is a maximum f -isotonic set and $v \notin C$ then $f(v)$ is not in v 's window, for if it were then v could be added to C to give a yet larger f -isotonic set. Given C , the C -trim of f at v is the closest value to $f(v)$ in v 's C -window, i.e., it is $f(v)$ if $f(v)$ is in v 's C -window, $w_{C\prec}(v)$ if $f(v) < w_{C\prec}(v)$, and $w_{C\succeq}(v)$ if $f(v) > w_{C\succeq}(v)$. The trim error of f due to C , $t_{\text{err}}(f, C)$, is the maximum, over all $x \in V$, of the distance from $f(x)$ to its C -trim. For a vertex v we slightly abuse notation and use $t_{\text{err}}(f, v)$ to mean $t_{\text{err}}(f, \{v\})$. A single vertex is f -isotonic, and

$$t_{\text{err}}(f, v) = \max\{\{\max\{f(v) - f(x) : v \preceq x, f(v) \geq f(x)\}, \max\{f(x) - f(v) : v \succeq x, f(v) \leq f(x)\}\}$$

For an arbitrary f -isotonic set C , $t_{\text{err}}(f, C) = \max\{t_{\text{err}}(f, v) : v \in C\}$.

Lemma 1 Give a set of real-valued labels, a label function f on a dag $G = (V, E)$, an f -isotonic set $C \subseteq V$, and an L_∞ -optimal isotonic regression g of f , let g' be g trimmed to C . Then g' is isotonic and $\|f - g'\|_\infty = \max\{\|f - g\|_\infty, t_{\text{err}}(f, C)\}$.

Proof: The C -windows are isotonic in that their lower bounds and upper bounds are isotonic functions. Trimming an isotonic function to isotonic windows always results in an isotonic function.

To show the upper bound on $\|f - g'\|_\infty$, for any $x \in V$ suppose $g(x) \geq f(x)$ and let $y \in C$, $x \preceq y$, be such that $w_{C\succeq}(x) = f(y)$. If $g(x) \geq w_{C\succeq}(x)$ then $g'(x) = w_{C\succeq}(x) \leq t_{\text{err}}(f, y)$; if $g(x) \leq w_{C\prec}(x)$ then $g'(x)$ is $g(x)$ trimmed up to $w_{C\prec}(x) \leq f(x)$ and $|g'(x) - f(x)| < |g(x) - f(x)|$; and if $g(x) \in [w_{C\prec}(x), w_{C\succeq}(x)]$ then $g'(x) = g(x)$. Thus in all cases $|g'(x) - f(x)| \leq \max\{|g(x) - f(x)|, t_{\text{err}}(f, y)\}$, which shows that $\|f - g'\|_\infty \leq \max\{\|f - g\|_\infty, t_{\text{err}}(f, C)\}$. Similar results hold if $g(x) \leq f(x)$, and since both terms in the max are also lower bounds the equality is proven. \square

Proposition 2 Given a set of real-valued labels, a label function f on a dag $G = (V, E)$, and an f -isotonic set $C \subseteq V$, an $L_{0,\infty|C}$ isotonic regression of f can be obtained in $\Theta(m)$ time.

Proof: The standard simple isotonic regression g which minimizes $\|f - g\|_\infty$, namely $g(x) = (\max\{f(y) : y \preceq x\} + \min\{f(y) : y \succeq x\})/2$, can be computed via topological sort, taking $\Theta(m)$ time. Using this specific g minimizes the time and the lemma shows it minimizes the error since $t_{\text{err}}(f, C)$ is independent of the isotonic function used. \square

Theorem 3 *Given a set of real-valued labels, a label function f on a dag $G = (V, E)$, and a violator graph $\widehat{G} = (\widehat{V}, \widehat{E})$, an $L_{0,\infty}$ isotonic regression of f can be obtained in $\Theta(\mathcal{T}(\widehat{n}, \widehat{m}) \log n)$ time.*

Proof: The proof of the theorem shows that to find a $L_{0,\infty}$ isotonic regression g of f we merely need to find an f -isotonic set C of size $s = n - \Delta_\infty(f)$ that minimizes trim-err among all such sets. We can determine s by just running the algorithm to find an optimal relabeling function of f .

To find C , there is an f -isotonic set of size s with $t_{\text{err}} \leq t$ iff the minimum satisfying flow is s on an adjusted violator graph where for every v where $r(v) > t$, the required flow from v^{in} to v^{out} is set to 0. A simple binary search on the trim-err values of the vertices can be used to find the smallest possible t . The regression function used in the lemma can be found in $\Theta(m)$ time, as can $w_{C \prec}(\cdot)$ and $w_{C \succ}(\cdot)$, so the time is dominated by the time to find the violator graph and the logarithmic number of iterations to determine a maximal f -isotonic set. \square

3.3 Weak $L_{0,p|C}$ Optimality, $p \in \{1, 2\}$

For $L_{0,1|C}$, finding an L_1 optimal regression and trimming may not be optimal. On a linear order, suppose the values are $0^*, 3, 1^*, -1, -2, -3, -4, 2^*$, where $*$ indicates the unique maximum isotonic set C . The unique L_1 optimal isotonic regression has values $-1, -1, -1, -1, -1, -1, 2$, which, when trimmed, is $0, 0, 1, 1, 1, 1, 2$, with L_1 error 13. However, the unique $L_{0,1|C}$ isotonic regression is $0, 1, 1, 1, 1, 1, 2$, with error 12.

A different approach is to trim the results then find the optimal regression of the the trimmed values. Simple examples shows this does not work for $L_{0,\infty|C}$, but it does for $L_{0,1|C}$.

Proposition 4 *Given a set of real-valued labels, a label function f on a dag $G = (V, E)$, and an f -isotonic set $C \subseteq V$, an $L_{0,1|C}$ isotonic regression can be obtained by trimming f to C and finding an L_1 isotonic regression of the trimmed values.*

Proof: Any $L_{0,1|C}$ isotonic regression g must have its values in $[w_{C \prec}(v), w_{C \succ}(v)]$ for all v , and hence if $f(v) < w_{C \prec}(v)$ then the error at v is $(w_{C \prec}(v) - f(v)) + (g(v) - w_{C \prec}(v))$. Similarly if $f(v) > w_{C \succ}(v)$ then the error at v is the distance to the trimmed value, $w_{C \succ}(v)$, plus the distance from $w_{C \succ}(v)$ to $g(v)$. Therefore a function minimizing the sum of the distances from the trimmed values to their regression values is also a function minimizing the L_1 error of an isotonic regression consistent with C , and vice versa. \square

Trimming is just a topological sort operation on the original graph, so can be completed in $\Theta(m)$ time. Since C is given, the total time is determined by the time to find the L_1 isotonic regression on the original graph. The fastest known algorithms depend on the graph [31]. The fastest for d -dimensional grids, or points in arbitrary position in d -dimensional space, is discussed in Section 4. The fastest for general dags takes $\Theta(nm + n^2 \log n)$ time [2] time or $\tilde{O}(\min\{m^{1+o(1)}, m + n^{1.5}\})$ time with high probability [8, 12].

For $L_{0,2|C}$ simple examples show that neither finding an optimal regression and then trimming it (as for $L_{0,\infty|C}$) nor trimming and then finding an optimal regression (as for $L_{0,1|C}$) always gives an optimal answer. Instead we borrow a technique appearing in [25]. Their technique applies more generally, finding $L_{0,2}$, though only when regression values are restricted to the labels. Further, the time of their approach replaces n with $n\ell$, and it is not clear how to utilize modern flow algorithms.

$\widehat{G}_f = (\widehat{V}_f, \widehat{E}_f)$ {the flow graph of a violator dag $\widehat{G} = (\widehat{V}, \widehat{E})$ for G }
 $W = \emptyset$ {the initial set of vertices v where their regression value $f'(v)$ has been determined}
 for $d = 0, \ell - 1$
 g = trim of f , trimmed using the values of f' on W
 V' = vertices of V where g and f differ by exactly d steps
 G' = flow graph constructed from \widehat{G}_f , where all vertices in $V - V'$ are collapsed
 C = an f -isotonic set of maximum size determined by G'
 for all $v \in C$, $f'(v) = g(v)$
 $W = W \cup C$

Figure 3: Determining an f' which is an $L_{0,0|C}$ regression of f on G

Proposition 5 *Given a set of real-valued labels where all labels are integers in $[1, U]$, a label function f on a dag $G = (V, E)$, and an f -isotonic set $C \subseteq V$, the $L_{0,2|C}$ isotonic regression can be found in time linear in the time to do an integer-valued L_2 isotonic regression of a weighted function on G when all values and weights are integers in $[0, (nU)^3]$.*

Proof: For sets S_1, S_2 of size $\leq n$, with unit weights and where all entries are integers in $[1, U]$, their means (their L_2 regressions) are either the same or differ by less than $\alpha = 1/(nU)^2$. A weight W will be assigned to each vertex in C , and the unit weight retained at all other vertices, so that any set containing a vertex v of C will have a mean x less than $\alpha/2$ from $f(v)$. For arbitrary additional vertices $x - f(v)$ is no larger than $(Wf(v) + nU)/W - f(v) = nU/W$, so choosing $W = 2nU/\alpha$ suffices.

Multiplying all values by $\lceil 1/\alpha \rceil$ guarantees that the integer L_2 regression of the scaled values differ by more than $1/\alpha$ if the sets do not intersect C , and that any level set containing a member of C will have a regression value which rounds to $f(v)$. Scaling the values back by dividing by $\lceil \alpha \rceil$ gives the solution to the original problem. Note that one can then obtain the exact regression values by computing the mean of each level set. \square .

Here too the fastest known algorithms depend on the dag, where the fastest algorithm for L_2 isotonic regression for general dags currently known is $\Theta(nm \log(m/n))$ [18] for exact regression on arbitrary real-valued functions, or $\min\{\tilde{\Theta}(n^\omega), \Theta(m^{1.5})\}$ with high probability [20] for integer-valued functions, where the time includes a factor dependent on $\log U$. See [31] for updates on the fastest algorithms for various dags.

3.4 Weak $L_{0,0|C}$ Optimality

Determining $L_{0,0|C}$ appears to require a different approach, given in Figure 3. The time analysis is straightforward.

Theorem 6 *Given a label function f on a graph G , and a violator dag $\widehat{G} = (\widehat{V}, \widehat{E})$, the algorithm in Figure 3 produces an $L_{0,0|C}$ regression of f in $\Theta(\mathcal{T}(\widehat{n}, \widehat{m})\ell)$ time.*

Previous authors used $\mathcal{T} = \Theta(n^3)$. Determining the best flow algorithm to use depends on the relationship of \widehat{n} to \widehat{m} , see Section 2, but in all cases can be a significant improvement on this. In some cases this is unknown a priori, while in others it is and can be exploited. See Section 4 for such a case.

4 Multidimensional Orderings

Given points $x = (x_1, \dots, x_d)$, $y = (y_1, \dots, y_d)$, $x \neq y$, in a d -dimensional space, y dominates x iff $x_i \leq y_i$ for $1 \leq i \leq d$. Domination is also known as multidimensional ordering, component-wise order, or product

ordering. There is no requirement that the dimensions are the same, merely that each is linearly ordered. This is an extremely important class of dags since there are a vast number of papers in many different applications which use such orderings, thus algorithms to handle large datasets with such orderings are quite important.

The edges corresponding to multidimensional ordering are implied, not explicit. Given a set V of n d -dimensional points, simple pairwise comparisons could be used to create an explicit dag, but this would require $\Theta(n^2)$ time and may generate a dag with $\Theta(n^2)$ edges (the time depends on d since comparing points can take time linear in d , but this small aspect is usually ignored, or reflected by a statement such as “where the implied constants depend on d ”). The violator graph could be constructed the same way, and would have the same problems. However, more concise graphs are possible by embedding V into a larger dag $\check{G} = (\check{V}, \check{E})$, where $V \subset \check{V}$, that preserves the ordering on V . I.e., if $x, y \in V$ then $x \prec y$ iff there is a path in \check{G} from x to y . The vertices in $\check{V} \setminus V$ are sometimes known as *Steiner vertices*. An explicit construction of this appears in [30], where it is shown how to embed into a dag with $\Theta(n \log^d n)$ vertices and edges, with the construction taking time proportional to this. In [30] the resulting graph is called a *rendezvous graph* since for any $x \prec y$ there is a unique Steiner vertex s , their rendezvous, such that there is an edge from y to s and one from s to x . That paper also shows how to construct a compressed rendezvous graph, where one dimension is treated differently, that has $\Theta(n \log^{d-1} n)$ vertices and edges, where again the construction takes time proportional to its size. Both of these increase the number of vertices compared to the simplest dag representation, but this is more than compensated for by a significant reduction in the worst-case number of edges.

An interesting aspect of this is that the violator graph can be similarly constructed. Given a label function f , create the violator graph by using a $(d + 1)$ -dimensional ordering on V by adding an extra dimension with value $f(v)$ at each vertex v , where the ordering is reversed at this additional dimension, i.e., $(x, f(x)) \prec (y, f(y))$ iff $y \prec x$ and $f(x) > f(y)$. Using the compressed rendezvous construction mentioned above constructs a violator graph in $\Theta(n \log^d n)$ time, having $\Theta(n \log^d n)$ vertices and edges. This is an unusual setting in that both the original graph and the violator graph are nearly linear in size and can be constructed in nearly linear time.

Using this gives:

Theorem 7 *For any dimension $d \geq 2$, given a real-valued function f on n d -dimensional points, an L_0 isotonic regression can be found in $\Theta(\mathcal{T}(n \log^d n, n \log^d n))$ time, and an $L_{0,\infty}$ isotonic regression can be found in time a factor of $\log n$ slower, where the implied constants depend upon d .*

Proof: As noted above, a violator dag \widehat{G} can be constructed in $\Theta(n \log^d n)$ time, with $\Theta(n \log^d n)$ edges and nodes. The standard conversion of \widehat{G} into a flow graph is used, except that the Steiner vertices are not expanded into a pair and all edges into and out of them have 0 minimal flow. Using this, a maximum f -isotonic set C can be found in the time claimed. This gives the result for L_0 . The result for $L_{0,\infty}$ follows from Theorem 3. \square

Corollary 8 *Given a real-valued function f on n d -dimensional points, and given an f -isotonic set C of maximum size, an $L_{0,1|C}$ isotonic regression can be found in*

- a) $\Theta(n \log n)$ time if $d = 1$,
- b) $\Theta(n \log n)$ time $d = 2$ and the points form a grid,
- c) $\Theta(n \log^2 n)$ time if $d = 2$ and the points are in arbitrary position, and
- d) $\Theta(n^{1.5} \log^{d+1})$ time if $d \geq 3$, where the implied constants depend on d .

Proof: This follows from Proposition 4, which states that, given C , one merely needs to trim and then find an optimal L_1 regression. The fastest known times for the L_1 regression appear in [1] for a), [29] for b), and [30] for c) and d). \square .

Corollary 9 Given a real-valued function f on n d -dimensional points, and given an f -isotonic set C of maximum size, an $L_{0,\infty|C}$ isotonic regression can be found in

- a) $\Theta(n)$ time if the points form a grid, and
- b) $\Theta(n \log^{d-1} n)$ time if the points are in arbitrary position,

where for both cases the implied constants depend on d .

Proof: This follows from Theorem 1, which states that one merely needs to find an L_∞ isotonic regression and trim it, coupled with the fact that the basic L_∞ isotonic regression can be determined in time linear in the number of edges. \square

The following appears in [32]:

Corollary 10 Given a function f on n d -dimensional points, where all values are integers in the range $[0, U]$, and given an f -isotonic set C of maximum size, an $L_{0,2|C}$ isotonic regression, where all values must be integers, can be found in:

- a) $\Theta(n)$ time if $d = 1$,
- b) $\Theta(n \log U)$ time if $d = 2$ and the points form a grid,
- c) $\Theta(n \log n \log U)$ time if $d = 2$ and the points are in arbitrary position,
- d) $o(n^{1.5n})$ time if $d \geq 3$, where the implied constants depend on d .

\square

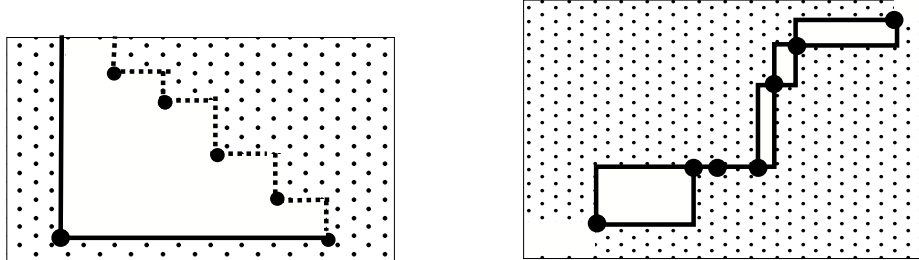
5 $L_{0,p}$ Isotonic Regression on Linear Orders

For linear orders we can efficiently determine $L_{0,p}$ via a left-right scan, rather than constructing a violator graph, but cannot use the PAV (pool adjacent violators) algorithm since it does not hold for L_0 error. E.g., if $\mathcal{L} = \{0, 1, 2\}$ and the data values are 2, 2, 2, 0, 0, 1, 1, then after processing the first 5 values the unique optimal L_0 regression is the single level set 2, 2, 2, 2, 2. When the 6th is processed the result could be either 2, 2, 2, 2, 2, 2 or 0, 0, 0, 0, 0, 1, and when the last is processed the unique answer is 0, 0, 0, 0, 0, 1, 1. Thus whichever choice is used at the end of the 6th, then either 5 to 6, or 6 to 7, results in merging adjacent violators but getting 2 level sets, i.e., they are not pooled.

To determine $L_{0,p}$ on linear orders we instead use an approach based on the standard algorithm for maximal nondecreasing subsequences. At each vertex i we determine a pair (c, s) , where c is the size of the largest f -isotonic set from 1 to i , and s is the smallest error among all such sets. If the previous best value at i was (c_1, s_1) and now it is determined it can be reached in (c_2, s) , then the best value at i is given by the maxmin operation, where

$$\text{maxmin}\{(c_1, s_1), (c_2, s_2)\} = \begin{cases} (c_1, s_1) & \text{if } c_1 > c_2 \\ (c_2, s_2) & \text{if } c_1 < c_2 \\ (c_1, \min\{s_1, s_2\}) & \text{if } c_1 = c_2 \end{cases}$$

We will prove



There are no data points in the blank areas (including the lower left and upper right on the right figure) or on the boldface lines (other than vertices of C or possible successors). For both figures there might be data points in the background, or for the left figure perhaps on the dotted lines (points there aren't successors).

Figure 4: Left: A point (lower left) and all of its possible successors. Right: C for some data set.

Theorem 11 *Given a real-valued label function f on a linear ordering of length n , where there are ℓ different initial label values and regression values can be arbitrary real numbers, an $L_{0,p}$ isotonic regression can be found in $\Theta(n\ell)$ time for $p \in \{1, 2\}$ and $\Theta(n \log \ell)$ time for $p = \infty$. Further, the same times can be achieved if all initial and regression values must come from a real-valued label set \mathcal{L} of size ℓ .*

and

Theorem 12 *Given a label function f on a linear ordering of length n , where the initial and regression values are in a label set \mathcal{L} of size ℓ , a strong L_0 isotonic regression can be found in $\Theta(n\ell^2)$ time.*

Since the ordering is linear we assume the vertices are $1, \dots, n$.

5.1 $p = 1$

For $L_{0,1}$ there is a quite simple algorithm based on the fact that the optimal value of a level set can be taken to be a median value of the set, which implies that it can be chosen to be one of the values of f . We can then assume that we use a label set $\mathcal{L} = \{\lambda_1, \dots, \lambda_\ell\}$ which is these values. For a (c, s) -valued array $A(1:n, 1:\ell)$, let $A(i, j)$ be the optimal $L_{0,1}$ regression of f on $[1, i]$ among those with value λ_j at i . Then

$$A(i, j) = \max \min \{A(i-1, j) : j \leq i\} + \begin{cases} (0, |f(i) - \lambda_j|) & \text{if } f(i) \neq \lambda_j \\ (1, 0) & \text{if } f(i) = \lambda_j \end{cases}$$

This can easily be computed in $\Theta(n\ell)$ time. An $L_{0,1}$ isotonic regression is one with an optimal $\max \min$ value in $A(n, \cdot)$ and it is straightforward to determine the regression.

5.2 $p = 2$

For $L_{0,p}$, $p > 1$, if the isotonic regression is restricted to values in \mathcal{L} the same approach as for $p = 1$ can be used, but when regression values can arbitrary real numbers a more complex approach is needed. Suppose an $L_{0,p}$ isotonic regression f' is equal to f on an f -isotonic set $C = \{i_0 < \dots < i_k\}$ of maximum size, for some $k \geq 0$. For $0 \leq j < k$ consider the closed rectangle with lower left coordinate $(i_j, f(i_j))$ and upper right coordinate $(i_{j+1}, f(i_{j+1}))$. Except at these corners there cannot be any point $(x, f(x))$ in the rectangle for if there were then C would not have maximum size since the point could be added to it. Thus if we have

```

array T(0 : n+1) of (c, s) pairs {also keep track of predecessor giving optimal value}
f(0) = -∞
f(n+1) = ∞
for j = 0, n+1
    T(j) = (0, 0)
for i = 0, n
    successorvalue = ∞, initial level set value f(i)
    for j = i+1, n
        if ((f(j) < f(i)) ∨ (f(j) ≥ successorvalue)) then {j is not a successor}
            add level set for j
            merge level sets on [i, j] trimmed to [f(i), successorvalue]
        else {j is a successor}
            T(j) = maxmin{T(j), T(i) + (1, error of level sets on [i, j-1] trimmed to [f(i), f(j)])}
            if (f(i) = f(j)) then exit for-loop
            successorvalue = f(j)

```

Figure 5: Determining an $L_{0,p}$ Isotonic Regression on a Linear Order, $p < \infty$

an optimal f -isotonic regression from 0 to i , and at i the regression value is $f(i)$, then for the next index j , in increasing order, where the regression value is the same as $f(j)$ we need only consider those j where $f(j) > f(i)$, and $f(j) < f(k)$ for all $i+1 \leq k < j$. To simplify notation, at a vertex i the *potential successors* is the unique sequence of vertices $i < m_1 < \dots < m_k$ of maximum length where $f(m_1) > f(i)$ and $f(m_1) > f(m_2) > \dots > f(m_k)$. See Figure 4. When we refer to the L_p error of a level set on $[i, j]$ we actually mean the sum of the p^{th} power of the pointwise regression errors using a constant function which minimizes the error among trimmed regression values permitted (if no trimming is used then it would be the L_p mean). This observation immediately gives the algorithm in Figure 5.

To use this algorithm to prove Theorem 11 for $p = 2$ we need to be able to efficiently merge the level sets and determine the optimal error subject to the trimming requirement. As in the standard prefix approach to isotonic regression, if a level set has a value b and its predecessor has value a , where $a \geq b$, they are combined and the value of the result is calculated (along with ancillary numbers such as the sum of the f values). Before any calculations of L_2 errors we first compute $S_1(j) = \sum_{k=1}^j f(k)$ and $S_2(j) = \sum_{k=1}^j f(k)^2$ for $1 \leq j \leq n$, and let $S_1(0) = S_2(0) = 0$. In the prefix approach for L_2 isotonic regression each level set's regression value is just the average of its f values. If this is c , and the level set is trimmed to $[a, b]$, then c is used as the level set's regression value if $c \in [a, b]$, while otherwise it is a if $c < a$ and b if $c > b$. For a level set on $[i, j]$, the square of the L_2 error of using d as the regression value is

$$S_2(j) - S_2(i-1) - 2d(S_1(j) - S_1(i-1)) + d^2(j - i + 1)$$

so calculating a trimmed level set's error, even when the mean is not used as the regression value, takes constant time.

This shows that for any i , the total time to find the maxmin values for all successors is linear in the time to find the last successor, or to n if there is no later j value where $f(i) = f(j)$. Thus for any fixed λ , the worst-case time to determine successor values for all i where $f(i) = \lambda$ is $\Theta(n)$ and the total time is $\Theta(n\ell)$.

5.3 $p = \infty$

There is a very simple $\Theta(n \log \ell)$ time algorithm for $p = \infty$. First determine the trim error of every point, taking $\Theta(n)$ time. Then do the standard algorithm for finding a longest nondecreasing sequence, inserting points into a balanced tree, where each node stores the length of the longest nondecreasing decreasing sequence reaching that point, and the minimum maximum trim error of such a sequence. When a new point is inserted it adds 1 to the length of its predecessor, and computes the max of its trim error and the predecessor's. A successor point is eliminated iff it has length less than the new point, or the same length and equal or greater trim error.

This completes the proof of Theorem 11. \square

5.4 Strong L_0

An algorithm for finding a strong L_0 isotonic regression can be based on that for $L_{0,1}$. Given an isotonic function g on $[1, m]$ let $v = (v_0, v_1, \dots, v_{\ell-1})$ be a vector where v_i is the number of entries of g which are i labels away from the values of f on $[1, m]$. Recall that a strong L_0 isotonic regression of f on $[1, n]$ maximizes v_0 , and among the functions that maximize v_0 maximizes v_1 , etc. Let $A(i, j)$ be a vector-valued array, $1 \leq i \leq n$, $1 \leq j \leq n$, where $A(i, j)$ is a vector corresponding to a strongest isotonic regression of f on $[1, i]$, among those that are λ_j at i . For vectors u, v of length n let $v_{\max}(u, v) = u$ if there is an i such that $u(k) = v(k)$ for $k < i$ and $u(i) > v(i)$, and v otherwise. Then

$$A(i, j) = v_{\max}\{A(i-1, j) : j \leq i\} + (v_0, \dots, v_{|f(i)-\lambda_j|} + 1, \dots, v_{\ell-1})$$

A strong L_0 isotonic regression optimizes $A(n, \cdot)$ and can be computed in $\Theta(n\ell^2)$ time. This completes the proof of Theorem 12. \square

6 L_p Isotonic Regression with Hamming Distance Penalty

In some estimation problems the objective is similar to $L_{0,p}$, namely, given a function f and constant $\alpha > 0$, find an isotonic function g that minimizes $\|f-g\|_p + \alpha\|f-g\|_0$, i.e, an L_p isotonic regression with an L_0 , or Hamming distance, penalty function. It is a form of regularized regression. Note that for α sufficiently large (relative to f) this is the same as $L_{0,p}$. For $p \in \{1, 2\}$ algorithms are given for the case where the dag is linear, while for $p = \infty$ an algorithm is given for arbitrary dags.

6.1 $p \in \{1, 2\}$

Here we use the ideas in Section 5, but for each possible regression value we no longer keep track of the number of vertices where the regression had the same value as the original function, but rather just the minimum total cost of reaching that value.

Proposition 13 *Given a real-valued function f on a linear ordering of length n and a constant $\alpha > 0$, a real-valued isotonic function g which minimizes $\|f-g\|_p + \alpha\|f-g\|_0$ among all isotonic functions can be found in $\Theta(n^2)$ time for $p \in \{1, 2\}$.*

For this problem one can view the set of labels \mathcal{L} to be the values of f , and thus there may be n labels. For $p = 1$ the algorithm for $L_{0,1}$ in Theorem 11 can be applied here by merely changing the recurrence to

$$A(i, j) = \min\{A(i-1, j) : j \leq i\} + \alpha|f(i) - \lambda_j| \cdot (f(i) \neq \lambda_j)$$

Thus the time for an L_1 penalty can be expressed as $\Theta(n\ell)$, where ℓ is the number of different initial values.

For $p = 2$ the algorithm is the same as that in Theorem 11 for $L_{0,2}$, except that here at each potential successor one merely keeps track of the minimum total cost of getting there.

6.2 $p = \infty$

We give a result which is more general than linear orders:

Theorem 14 *Given a real-valued function f on a dag $G(V, E)$ of n vertices, let $k = n - \Delta_0(f)$. Then given a constant $\alpha > 0$, a real-valued isotonic function g which minimizes $\|f-g\|_\infty + \alpha\|f-g\|_0$ among all isotonic functions can be found in*

- a) $\Theta(nk \log(n/k))$ time if G is a linear order
- b) $\Theta(k \log(n/k) \mathcal{T}(\hat{n}, \hat{m}))$ time for an arbitrary dag if a violator graph \hat{G} is given.

We assume $k > 0$. One can first check if the data is isotonic, taking $\Theta(m)$ time, and if it is then $g = f$, the total error is 0, and $k = 0$. If the data isn't isotonic then $k > 0$.

If α is sufficiently large then an isotonic function g is in $L_{0,\infty}(f)$ iff it minimizes $\|f-g\|_\infty + \alpha\|f-g\|_0$. Theorem 3 shows an $L_{0,\infty}$ regression can be found by first determining $k = n - \Delta_0(f)$ and then finding the smallest t_{err} t such that there is an f -isotonic set C of size k among the vertices v with $t_{\text{err}}(v) \leq t$. Then f trimmed to C is a $L_{0,\infty}$ isotonic regression. Further, t can be found by a binary search among the t_{err} values, where at each step one uses a flow algorithm to find a largest f -isotonic set among the vertices with $t_{\text{err}} \leq t$.

The same approach can be used to find an isotonic function g minimizing $\|f-g\|_\infty + \alpha\|f-g\|_0$. For an isotonic function g let $C(g)$ be the set of vertices where g and f have the same value. Then $\|f-g\|_0 = i$ for some $1 \leq i \leq n$ and g must have the property that $t_{\text{err}}(C(g))$ is the minimum t_{err} over all isotonic functions h such that $\|f-h\|_0 = i$ (g may not be unique).

This gives a simple way to minimize $\|f-g\|_\infty + \alpha\|f-g\|_0$: for each $1 \leq i \leq n$ let $g(i)$ be an isotonic function with minimal t_{err} among those functions g' where $|C(g')| = n - \|f-g'\|_0$. Then a function minimizing $\min\{\|f-g(i)\|_\infty + \alpha\|f-g(i)\|_0 : 1 \leq i \leq n\}$ minimizes $\|f-h\|_\infty + \alpha\|f-h\|_0$ among all isotonic functions h .

The primary difference between Theorem 14 a) and b) is the technique used to determine the minimum t_{err} needed to achieve $|C(g(i))| = i$. For both we first determine $\Delta_0(f)$ via the relevant regression algorithm. We also sort points by their t_{err} value.

Proof of a): For the linear order find an optimal L_∞ regression h . Then start with an empty linear order and add points one at a time, keeping them ordered by their order in V . After each insertion we determine the size of the longest increasing sequence, and when this value changes from $i-1$ to i we record the t_{err} where this occurred. After all points have been added we determine the optimal i value, create a linear order of all points with $t_{\text{err}} \leq$ the t_{err} needed to obtain this, find a maximum length nondecreasing subsequence C and trim h to this. Note that we could have started by immediately inserting all points v where $t_{\text{err}}(v) \leq \|f-e\|_\infty$.

Sorting the vertices by their t_{err} and inserting them one at a time and determining the longest increasing subsequence is called the *dynamic longest increasing subsequence* problem and has been studied by several authors [15, 17, 19] in the more difficult setting where there can be deletions as well as insertions. When there are only insertions a $\Theta(nk \log(n/k))$ algorithm appears in [11]. A randomized algorithm in [19] for the fully dynamic problem takes $\tilde{\Theta}(n^{5/3})$ time but has a very small chance of producing an incorrect answer (note that they call this exact though it is not always correct). Here we only do insertions and don't need to produce an

LIS every step, just determine its length, so it is likely that an exact algorithm taking $o(n^2)$ time exists. This would immediately improve the time for a).

Proof of b): Insert the points into the violator graph where for any points not currently inserted we set their minimum flow requirement to 0. We determine minimum flow in this adjusted graph, which plays the same role as finding the maximum nondecreasing sequence for the linear order.

The difference is that we don't insert all of the points in sequential order by their t_{err} value. That would result in an algorithm taking $\Theta(n\mathcal{T}(\hat{n}, \hat{m}))$ time. We reduce this slightly, replacing n with $k \log(n/k)$, by using the fact that we are searching for k values and these values are monotonic in the values of t_{err} . This is equivalent to searching for a set S of k unknown values in an ordered set of n values, where at each probe one can determine if there are any smaller elements of S . To find them all there are essentially k binary searches in contiguous regions with unknown boundaries. Since the log function is concave this is maximized when the regions are all the same size, so the worst-case number of probes is $\Theta(k \log(n/k))$.

This finishes the proof of b), which finishes the proof of the theorem. \square

7 Final Remarks

Datasets are quickly becoming vastly larger and more complex, often with order-constrained relationships between independent and dependent variables, but the relationships are difficult to evaluate due to noise. Because of this, nonparametric regression, particularly isotonic regression, is becoming increasingly important. Various criteria may be used for deciding how to enforce isotonic (monotonic) assumptions, and for linearly ordered labels one criterion is to minimize the number of labels that must be changed. When optimizing this L_0 distance (aka Hamming distance), in general there will be many L_0 isotonic regressions and choosing among them can be useful but difficult. We choose ones which optimize secondary criteria based on other L_p metrics, an approach also used in [25]. We gave algorithms for computing them, and many authors have already shown the utility of isotonic regression for L_0 and other metrics [3, 7, 9, 15, 16, 17, 19, 24, 25, 26, 33].

The algorithms utilize maximum flow algorithms, and for $L_{0,1|C}$ and $L_{0,2|C}$ also utilize algorithms for L_1 and L_2 isotonic regression. Since there continues to be advances in algorithms for these problems, and we just use them via subroutine calls, we have given algorithms stated in general terms, using $\mathcal{T}(n, m)$ to represent the fastest maximum flow algorithm relevant to the problem, and general references to the currently fastest L_1 and L_2 algorithms. The wikipedia page [34] is usually up-to-date on the fastest flow algorithms, and [31] is usually up-to-date on the fastest L_p isotonic algorithms for various dags and $p \in \{0, 1, 2, \infty\}$.

Finally, we introduced strong L_0 isotonic regression (Section 3), aka $L_{0,0}$, which appears to be a natural choice among L_0 isotonic regressions with arbitrary labels, either numeric or nonnumeric. We gave an efficient algorithm for determining it on linear orders, but not for general dags nor even multidimensional ones. It is very similar to the *strict L_∞ isotonic regression* in [28]. There a $\Theta(\min\{nm, n^\omega\} + n^2 \log n)$ time algorithm was given for determining it, and one taking $\Theta(nm)$ expected time appears in [20]. Given an isotonic regression g of f , let (e_1, e_2, \dots, e_n) be the values $\{|g(v) - f(v)| : v \in V\}$ sorted in decreasing order. A strict L_∞ isotonic regression is one which is first in the lexical ordering of all sequences corresponding to isotonic regressions. It may not be unique. Recall that strong L_0 isotonic regression corresponds to a lexically first vector when the e_i are sorted in increasing order. Strong L_0 maximizes the number of small errors while strict L_∞ minimizes the number of large ones.

References

- [1] 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.

- [2] Angelov, S; Harb, B; Kannan, S; and Wang, L-S (2006), “Weighted isotonic regression under the ℓ_1 norm”, *Symp. on Discrete Algorithms (SODA)*, pp. 783–791.
- [3] De Baets, B (2019), “Monotonicity, a deep property in data science”, SFC2019.
- [4] Belovs, A (2018), “Adaptive lower bound for testing monotonicity on the line”, arXiv:1801:08709.
- [5] Berman, P; Raskhodnikova, S; and Yaroslavtsev, G (2014), “ L_p testing”, *STOC '14*, pp. 164–173.
- [6] Black, H; Chakrabarty, D; and Seshadhri, C (2020), “Domain reduction for monotonicity testing: a $o(d)$ tester for boolean functions in d -dimensions”, *SODA '20*, pp. 1975–1994.
- [7] Brabant, Q; Couceiro, M; Dubois, D; Prade, H; and Rico, Q (2020), “Learning rule sets and Sugeno integrals for monotonic classification problems”, *Fuzzy Sets and Systems*, 401, pp 4–37.
- [8] van den Brend, J; Lee, YT; Liu, YP; Saranurak, T; Sidford, A; Song, Z; and Wang, D (2021), “Minimum cost flows, MDPs, and L_1 regression in nearly linear time for dense instances”, arXiv:2001.005719.
- [9] Cano, J-R; Gutierrez, PA; Krawcsyk, B; Wozniak, M; and García, S (2018), “Monotonic classification: an overview on algorithms, performance measures and data sets”, arXiv:1811.07155
- [10] Chakrabarty, D and Seshadhri, C (2013), “Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids”, *Proc. 45th ACM STOC*, p 419–428.
- [11] Chen, A; Chu, T; and Pinsker, N (2013), “The dynamic longest increasing subsequence problem”, arXiv:1309.7724v4
- [12] Chen, L; Kyng, R; Liu, YP; Peng, R; Probst Gutenberg, M; and Sachdeva, S; (2022), “Maximum flow and minimum cost flow in almost-linear time (Preliminary Version)”, arXiv:2203.00671
- [13] Chen, X.; De, A; Servedio, RA; and Tan, L-Y (2015), “Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries”, *Proc. 47th ACM STOC*, p 519–528.
- [14] Duivesteyn, W; and Feelders, A (2008), “Nearest neighbour classification with monotonicity constraints”, *ECML PKDD 2008*, pp. 301–316.
- [15] Fattal, S and Ron, D (2010), “Approximating the distance to monotonicity in high dimensions”, *ACM Trans. Algorithms* 6, pp. 1–37.
- [16] Feelders, A; Velikova, M; and Daniels, H (2006), “Two polynomial algorithms for relabeling non-monotone data”, Tech. Report UU-CS-2006-046, Dept. Info. Com. Sci., Utrecht Univ.
- [17] Fischer, E; Lehman, E; Newman, I; Raskhodnikova, S; Rubinfeld, R.; and Samorodnitsky, A (2002), “Monotonicity testing over general poset domains”, *STOC'02*.
- [18] Hochbaum, DS; and Queyranne, M (2003), “Minimizing a convex cost closure set”, *SIAM J. Discrete Math* 16, pp. 192–207.
- [19] Kociumaka, T; and Seddighin, S (2021), “Improved dynamic algorithms for longest increasing subsequence”, arXiv:2011.10874v2
- [20] Kyng, R; Rao, A; and Sachdeva, S (2015), “Fast, provable algorithms for isotonic regression in all L_p -norms”, NIPS.

- [21] 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_∞ -norm and the Hamming distance”, *J. Comp. and Sys. Sci.* 75, pp. 451–464.
- [22] Möhring, R (1985), “Algorithmic aspects of comparability graphs and interval graphs”, *Proc. NATO Adv. Study Inst. on Graphs and Order*, Rival, I, ed., 41–101.
- [23] Pijls, W and Porharst, R (2013), “Another note on Dilworth’s decomposition theorem”, *J. Discrete Mathematics*.
- [24] Pijls, W and Potharst, R (2014), “Repairing non-monotone ordinal data sets by changing class labels”, *Econometric Inst. Report EI 2014–29*.
- [25] Rademaker, M; De Baets, B; and De Meyer, H (2009), “Loss optimal monotone relabeling of noisy multi-criteria data sets”, *Information Sciences* 179 (2009), pp. 4089–4097.
- [26] Rademaker, M; De Baets, B; and De Meyer, H (2012), “Optimal monotone relabeling of partially non-monotone ordinal data”, *Optimization Methods and Soft.* 27, 17–31.
- [27] Saks, M and Seshadhri, C (2010), “Local monotonicity reconstruction”, *SIAM J. Computing* 39, 2897–2926.
- [28] Stout, QF (2012), “Strict L_∞ isotonic regression”, *J. Optimization Theory and Applications* 152, pp. 121–135.
- [29] Stout, QF (2013), “Isotonic regression via partitioning”, *Algorithmica* 66, pp. 93–112.
- [30] Stout, QF (2015), “Isotonic regression for multiple independent variables”, *Algorithmica* 71, pp. 450–470.
- [31] Stout, QF, “Fastest known isotonic regression algorithms”, web.eecs.umich.edu/~qstout/IsoRegAlg.pdf
- [32] Stout, QF (2021), “ L_p isotonic regression algorithms using an L_0 approach”, [arXiv:2107.00251v2](https://arxiv.org/abs/2107.00251)
- [33] Verbeke, W; Martens, D; and Baesens, B (2017), “RULEM: A novel heuristic rule learning approach for ordinal classification with monotonicity constraints”, *Applied Soft Computing* 60 (2017), pp. 858–873
- [34] https://en.wikipedia.org/wiki/Maximum_flow_problem