Isotonic Regression and Flow

Yu Cong

April 20, 2023

Overview

1. **isotonic regression**
2. **dp**
3. **flow**

isotonic regression

L1 error:

$$\min \sum_{i=1}^{n} w_i |x_i - a_i|$$

L2 error:

$$\min \sum_{i=1}^{n} w_i (x_i - a_i)^2$$

L? error...

$$\min \sum_{i=1}^{n} w_i e_i(x_i)$$

$$s.t. \quad x_1 \leq x_2 \leq ... \leq x_n$$
$$w_i \geq 0, a \in \mathbb{R}^n$$

example

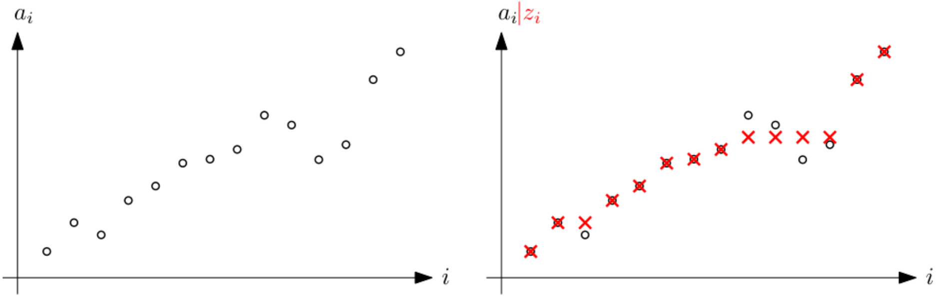


Figure 1 The original data sequence $(a_1, \ldots, a_n)$ is shown on the left. The crosses on the right form a monotone approximation $(z_1, \ldots, z_n)$. It contains two runs that are longer than a single element.

dynamic programming approach

There exists $O(n \log n)$ dp algorithm for L1 error isotonic regression.
`http://drops.dagstuhl.de/opus/volltexte/2018/10027/`
Consider the subproblems:

$$f_k(x) = \min\{\sum_{i=1}^{k} w_i |x_i - a_i| : x_1 \leq x_2 \leq ... \leq x_k = x\}$$

for $k = 1, 2, ..., n$ and $x \in \mathbb{R}$.
This approach differs somewhat from standard use of dynamic programming technique, as the intermediate objects of our dynamic programming recursion are real functions, and thus infinite objects.

We get the following straightforward dynamic programming recursion:

$$f_k(x) = \min\{f_{k-1}(y) : y \leq x\} + w_k|x - a_k| \quad (k = 1, 2, ..., n; x \in \mathbb{R})$$
$$f_0(x) = 0 \quad (x \in R)$$

How fast can we finish each transition?

Lemma

Each $f_k(x)$ is a piecewise linear convex function;
Breakpoints of f_k are located at a subset of the points $a_1, a_2, ..., a_k$;
The leftmost piece has slope $-\sum_{i=1}^{k} w_i$. The rightmost piece has slope w_k.

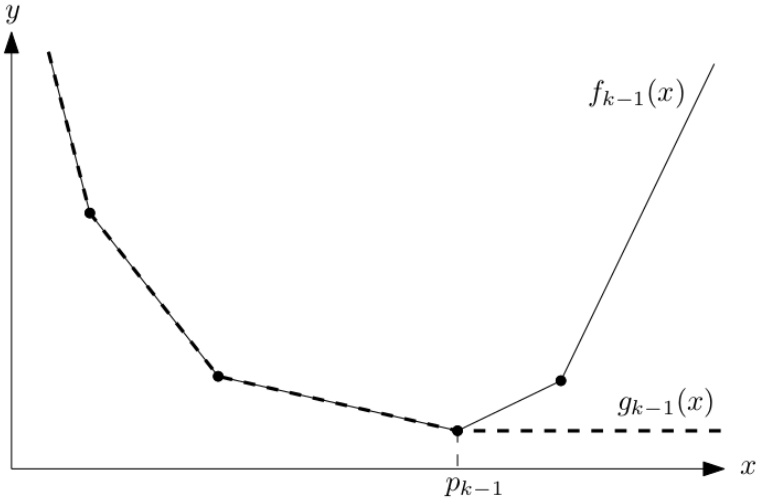


Figure 2 Constructing g_{k-1} from f_{k-1}.

part of the proof...

Prove by induction on k. Define

$$g_{k-1}(x) = \min\{f_{k-1}(y) : y \leq x\}$$

Assume by induction that all properties of the lemma hold for f_{k-1}. The function f_{k-1} is first monotonically decreasing to a minimum and then monotonically increasing. We denote by p_{k-1} the (not necessarily unique) position where the minimum occurs. If $x \leq p_{k-1}$ then $y = x$ is the optimum choice, and $g_{k-1}(x) = f_{k-1}(x)$. If $x \geq p_{k-1}$ then the optimum choice is $y = p_{k-1}$, and $g_{k-1}(x) = f_{k-1}(p_{k-1})$.

...

□

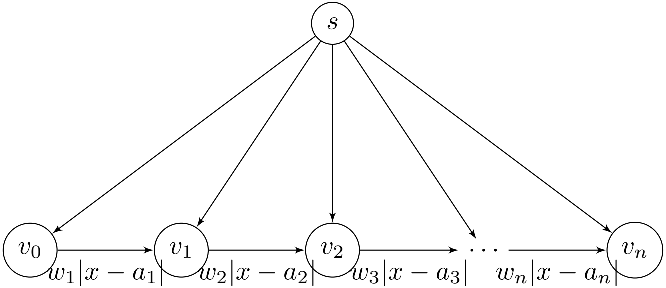
$O(\log n)$ time transition

- compute g_{k-1}. We can repeatedly remove the last line segment until we find p_{t-1}. The number of line segments in f_k is $O(n)$ so there are at most n removal;
- add $w_k|x - a_k|$. We have to add $w_k(x - a_i)$ to g_{k-1} for $x \leq a_k$ and add $w_k(a_i - x)$ to g_{k-1} for $x > a_k$.

We store function f_k in slope difference form. For n line segments $\{y = c_1x + d_1 : (x \in [l_1, r_1]), ..., y = c_nx + d_n : (x \in [l_n, r_n])\}$, the slope form is $\{c_1, c_2, ..., c_n\}$. The slope difference form is $\{c_1, c_2 - c_1, ..., c_n - c_{n-1}\}$.
If we use a priority queue to store intervals ordered by the their positions, each transition can be done in amortized $O(\log n)$.

reduce to min-cost flow

https://chaoxuprime.com/posts/2015-01-27-bounded-regression-on-data-streams.html



The min-cost flow from s to v_n is the answer to L1 isotonic regression.

series parallel graph

https://en.wikipedia.org/wiki/Series-parallel_graph

Definition

A graph is an SP-graph, if it may be turned into K2 by a sequence of the following operations:

- Replacement of a pair of parallel edges with a single edge that connects their common endpoints;
- Replacement of a pair of edges incident to a vertex of degree 2 other than s or t with a single edge.

decomposition tree

https://jeffe.cs.illinois.edu/teaching/algorithms/book/10-maxflow.pdf
exercise 19.

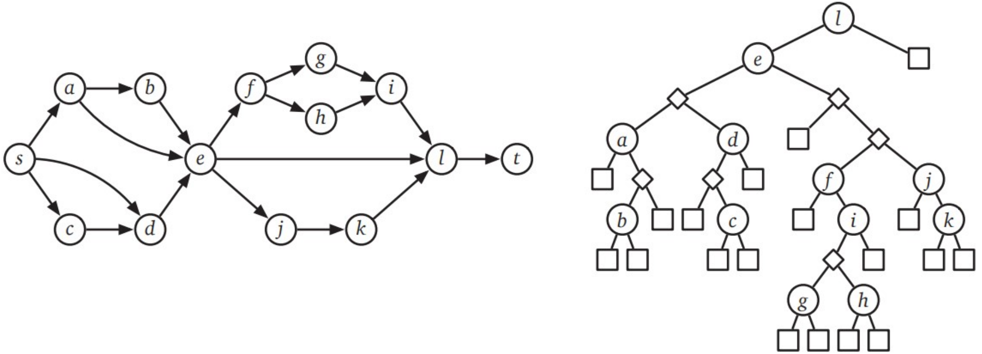


Figure 10.13. A series-parallel graph and a corresponding decomposition tree. Squares in the decomposition tree are leaves; diamonds are parallel nodes.

min-cost flow on SP-graph

The decomposition tree of a SP-graph can be constructed in linear time.
Every diamond node or circle node in the decomposition tree in Fig 10.13 represents a two terminal sub SP-graph.

Define the cost function $cost(x)$ of a two terminal network to be the minimum cost when the total flow from one terminal to another is x. We can compute the cost function for every circle or diamond node during a traverse of the decomposition tree.

Consider linear cost function first. $cost(x)$ is a linear function for every edge(the smallest SP-graph). Suppose we find a node and the cost functions of its two children have been computed. There two cases:

- series node. $cost(x)$ is the sum of cost functions of its children.
- parallel node. $cost(x)$ is the infimal convolution of two children's cost functions.

Infimal convolution:

$$(f \Box g)(x) = \inf_y f(x - y) + g(y)$$

for piecewise linear functions inf can be replaced with min.

The sum and infimal convolution of tow linear functions are also linear functions.
However, domains of functions are limited in our case. The capacity of an edge should be positive, so cost functions are $cost(x) : \mathbb{R}^+ \to \mathbb{R}$.
Now we know that the cost function is piecewise linear and convex. We can store breakpoints and corresponding slopes and compute the sum and the infimal convolution in linear time.
How to compute infimal convolution?

$$\begin{aligned} \max f(x_1) + g(x_2) \\ s.t. \quad x_1 + x_2 = \xi \\ -x_1 \leq 0, -x_2 \leq 0 \end{aligned}$$

From KKT conditions we see: $\binom{f'(x_1)}{g'(x_2)} + \delta\binom{1}{1} + \mu_1\binom{-1}{0} + \mu_2\binom{0}{-1} = \mathbf{0}$. If x_1 and x_2 are both positive, we have $\mu_1 = \mu_2 = 0$(complementary slackness). Therefore $f'(x_1) = g'(x_2) = -\delta$.

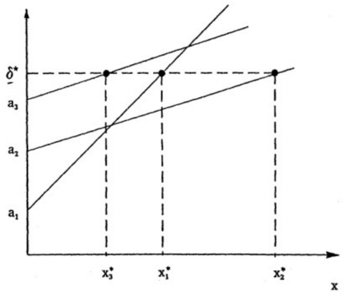


Figure: infimal convolution of 3 quadratic functions

This property holds for any number of differentiable objective functions. Now we see that computing the infimal convolution is related to the intersections...... Actually computing infimal convolution of two piecewise linear convex function is much easier than this.

We use a list to represent the piecewise linear convex function.
$\{(b_1, s_1), (b_2, s_2), ..., (b_n, s_n)\}$.
(b_i, s_i) means that the ith breakpoint is at $x = b_i$, and the slope of the ith interval is s_i.
A straightforward method to compute the infimal convolution is just merge two lists.

`https://www.sciencedirect.com/science/article/abs/pii/S0196677483710485` gives a $O(m \log m)$ algorithm for min-cost flow on SP-graph.(use finger tree to accelerate the sum and the infimal convolution operations)

questions

There are lots of algorithms for isotonic regression.(dp, pool adjacent violators, quadratic programming, computational geometry...) Does these algorithms still work when the monotone constraint is replaced with $l_i \leq x_i - x_{i-1} \leq u_i$?

How to solve this problem in $O(n \log n)$? `https://www.spoj.com/problems/CCROSSX/`
We can get a dp algorithm from min-cost flow on SP-graph.

How fast can we solve min-cost flow on SP-graphs under quadratic cost functions? The hardest part is the infimal convolution. Infimal convolution of n convex quadratic functions can be computed in $O(n \log n)$.
`https://doi.org/10.1016/j.cam.2011.04.011`