Isotonic Regression

信息科学技术学院　周书予

2021 年 12 月 22 日

Q & A

- Q: 什么是 Isotonic Regression?

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
  - Q: 什么是保序?

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
  - Q: 什么是保序?
  - A: 大家都学过离散/集图了，就不解释了。

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
    - Q: 什么是保序?
    - A: 大家都学过离散/集图了，就不解释了。
    - Q: 什么是回归?

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
  - Q: 什么是保序?
  - A: 大家都学过离散/集图了，就不解释了。
  - Q: 什么是回归?
  - A: 大家都炼过丹了，就不解释了。

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
  - Q: 什么是保序?
  - A: 大家都学过离散/集图了，就不解释了。
  - Q: 什么是回归?
  - A: 大家都炼过丹了，就不解释了。
- Q: 我知道保序回归是什么意思了，但是你是谁?

Q & A

- Q: 什么是 Isotonic Regression?
- A: Isotonic Regression 就是保序回归 (确信)。
- Q: 什么是保序回归?
  - Q: 什么是保序?
  - A: 大家都学过离散/集图了，就不解释了。
  - Q: 什么是回归?
  - A: 大家都炼过丹了，就不解释了。
- Q: 我知道保序回归是什么意思了，但是你是谁?
- A: 这不重要。

问题描述

Definition

给定偏序集 $\mathcal{X}$ 以及 $\mathcal{X}$ 上的函数 $y: \mathcal{X} \to \mathbb{R}, w: \mathcal{X} \to \mathbb{R}^+$。定义一个函数 $z: \mathcal{X} \to \mathbb{R}$ 是保序的，如果 $\forall a, b \in \mathcal{X}, a \preceq b \Rightarrow z_a \leq z_b$。最优化问题

$$L_p(\mathcal{X}, y, w) = \min_z \begin{cases} \sum_{a\in\mathcal{X}} w_a |y_a - z_a|^p, & 1 \leq p < \infty \\ \max_{a\in\mathcal{X}} w_a |y_a - z_a|, & p = \infty \end{cases}, s.t.\ z \text{ 是保序的}$$

被称为 L_p 保序回归问题，简称 L_p 问题。

问题描述

Definition

给定偏序集 $\mathcal{X}$ 以及 $\mathcal{X}$ 上的函数 $y:\mathcal{X}\to\mathbb{R}, w:\mathcal{X}\to\mathbb{R}^+$。定义一个函数 $z:\mathcal{X}\to\mathbb{R}$ 是保序的，如果 $\forall a,b\in\mathcal{X}, a\preceq b\Rightarrow z_a\leq z_b$。最优化问题

$$L_p(\mathcal{X},y,w)=\min_z\begin{cases}\sum_{a\in\mathcal{X}} w_a|y_a-z_a|^p, & 1\leq p<\infty\\ \max_{a\in\mathcal{X}} w_a|y_a-z_a|, & p=\infty\end{cases}, s.t.\ z \text{ 是保序的}$$

被称为 L_p 保序回归问题，简称 L_p 问题。

$p=\infty$ 处的定义是自然的。

可能是热身题

Statement

$\mathcal{X}$ 形如一条链。$p=2$。
或者等价的，给出序列 $\{y_i\}_{i=1}^n, \{w_i\}_{i=1}^n$，需要构造**递增**序列 $\{z_i\}_{i=1}^n$，最小化

$$\sum_{i=1}^{n} w_i(y_i - z_i)^2$$

可能是热身题

Statement

$\mathcal{X}$ 形如一条链。$p=2$。
或者等价的，给出序列 $\{y_i\}_{i=1}^n, \{w_i\}_{i=1}^n$，需要构造**递增**序列 $\{z_i\}_{i=1}^n$，最小化

$$\sum_{i=1}^n w_i(y_i - z_i)^2$$

如果 y_i 递增？

可能是热身题

Lemma

如果存在 $y_i > y_{i+1}$，则最优的 z 一定满足 $z_i = z_{i+1}$。

可能是热身题

Lemma

如果存在 $y_i > y_{i+1}$，则最优的 z 一定满足 $z_i = z_{i+1}$。

一旦出现 $y_i > y_{i+1}$，就可以把 i 和 $i+1$ 合并，得到 $y' = \frac{y_i w_i + y_{i+1} w_{i+1}}{w_i + w_{i+1}}$ 和 $w' = w_i + w_{i+1}$，用 (y', w') 替换 $(y_i, w_i), (y_{i+1}, w_{i+1})$。

重复合并直至 y_i 递增。

可能是热身题

Lemma

如果存在 $y_i > y_{i+1}$，则最优的 z 一定满足 $z_i = z_{i+1}$。

一旦出现 $y_i > y_{i+1}$，就可以把 i 和 $i+1$ 合并，得到 $y' = \frac{y_i w_i + y_{i+1} w_{i+1}}{w_i + w_{i+1}}$ 和 $w' = w_i + w_{i+1}$，用 (y', w') 替换 $(y_i, w_i), (y_{i+1}, w_{i+1})$。

重复合并直至 y_i 递增。

实现上可以用单调栈维护合并，复杂度为 $O(n)$。

忘记写标题了

Statement

$\mathcal{X}$ 形如一棵树。$p = 1$。

忘记写标题了

Statement

$\mathcal{X}$ 形如一棵树。$p = 1$。

一种思路是动态规划，记 $dp_{i,j}$ 表示确定 i 子树的取值且 i 节点取值为 j 的最小代价。

忘记写标题了

Statement

$\mathcal{X}$ 形如一棵树。$p = 1$。

一种思路是动态规划，记 $dp_{i,j}$ 表示确定 i 子树的取值且 i 节点取值为 j 的最小代价。

注意到 $dp_{i,j}$ 是关于 j 的线性分段凸函数 (斜率递增)，可以用线段树维护 dp 数组，转移时需要实现线段树合并。

~~不讨论实现细节。~~

整体二分的引入

整体二分是 OI 中常见的用于处理多组二分询问的算法，核心思路是通过整体性的预处理以避免信息的重复计算，或者维护必要的限制条件。

整体二分的引入

整体二分是 OI 中常见的用于处理多组二分询问的算法，核心思路是通过整体性的预处理以避免信息的重复计算，或者维护必要的限制条件。
然后我们引入一些约定。

整体二分的引入

整体二分是 OI 中常见的用于处理多组二分询问的算法，核心思路是通过整体性的预处理以避免信息的重复计算，或者维护必要的限制条件。
然后我们引入一些约定。

Definition

将序列 z 中不大于 a 的元素变成 a，不小于 b 的元素变成 b，称这个过程为 z 向 $S = \{a, b\}$ 取整。

整体二分的引入

整体二分是 OI 中常见的用于处理多组二分询问的算法，核心思路是通过整体性的预处理以避免信息的重复计算，或者维护必要的限制条件。
然后我们引入一些约定。

Definition

将序列 z 中不大于 a 的元素变成 a，不小于 b 的元素变成 b，称这个过程为 z 向 $S=\{a,b\}$ 取整。

Definition

对于一个 L_p 问题，定义 L_p^S 问题为把 z_i 取值限定在 S 内的原 L_p 问题。

整体二分的引入

整体二分是 OI 中常见的用于处理多组二分询问的算法，核心思路是通过整体性的预处理以避免信息的重复计算，或者维护必要的限制条件。
然后我们引入一些约定。

Definition

将序列 z 中不大于 a 的元素变成 a，不小于 b 的元素变成 b，称这个过程为 z 向 $S=\{a,b\}$ 取整。

Definition

对于一个 L_p 问题，定义 L_p^S 问题为把 z_i 取值限定在 S 内的原 L_p 问题。

Definition

$\mathcal{X}$ 的一个子集 $\mathcal{U}$ 的 L_p-mean 定义为 $\min_{k\in\mathbb{R}}\begin{cases}\sum_{a\in\mathcal{U}} w_a|y_a-k|^p, & 1\le p<\infty\\ \max_{a\in\mathcal{U}} w_a|y_a-k|, & p=\infty\end{cases}$。

$p = 1$

Lemma

L_1 问题一定存在一组最优解 z，满足 $z_i \in \{y_1, y_2, \cdots, y_n\}$。

$p = 1$

Lemma

L_1 问题一定存在一组最优解 z，满足 $z_i \in \{y_1, y_2, \cdots, y_n\}$。

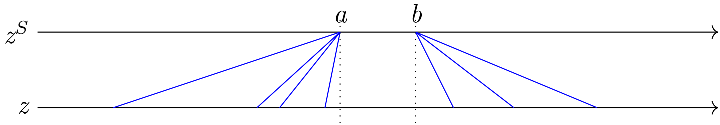
Theorem

在一个 L_1 问题中，若所有 y_i 都不在 (a, b) 中，记 z^S 表示 L_1^S 问题的一组最优解，那么一定存在一组 L_1 满足 $z_i \notin (a, b)$ 的最优解 z，使得 z 向 S 取整可以得到 z^S。

$p = 1$

Theorem

在一个 L_1 问题中，若所有 y_i 都不在 (a, b) 中，记 z^S 表示 L_1^S 问题的一组最优解，那么一定存在一组 L_1 满足 $z_i \notin (a, b)$ 的最优解 z，使得 z 向 S 取整可以得到 z^S。

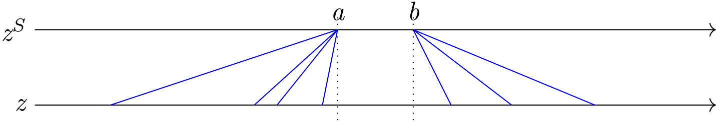


证明？

$p = 1$

Theorem

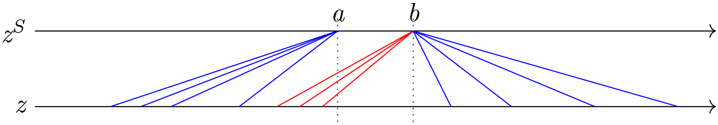
在一个 L_1 问题中，若所有 y_i 都不在 (a, b) 中，记 z^S 表示 L_1^S 问题的一组最优解，那么一定存在一组 L_1 满足 $z_i \notin (a, b)$ 的最优解 z，使得 z 向 S 取整可以得到 z^S。



证明？反证，先陈述否命题：任意一组 L_1 满足 $z_i \notin (a, b)$ 的最优解 z，都存在 j 使 $z_j \leq a, z_j^S = b$，或者 $z_j \geq b, z_j^S = a$。

注意到两种情况是不可能同时出现的，所以不失一般性地只考虑出现前者。

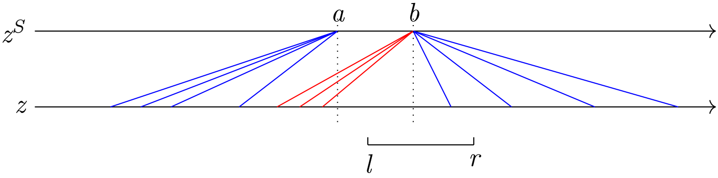
大讨论



把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

[1] 可能退化成一个点。

大讨论

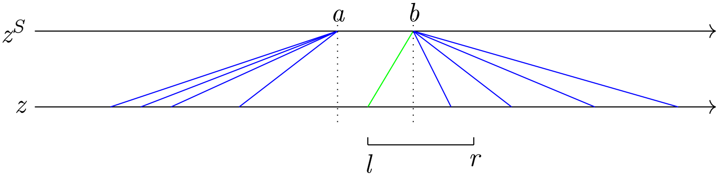


把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 1: $a \leq l$.

[1]可能退化成一个点。

大讨论

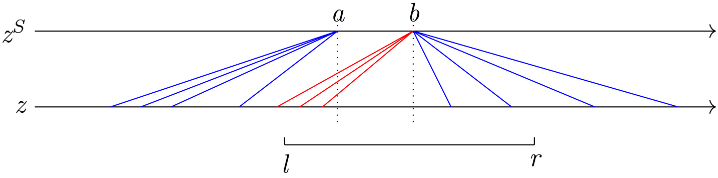


把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 1: $a \le l$. 此时把 U 中元素的 z_i 都改成 $\min\{l, b\}$，就能使结果严格变优，同时没有破坏原有偏序结构，于是 z 的最优性假设就寄了。

[1] 可能退化成一个点。

大讨论

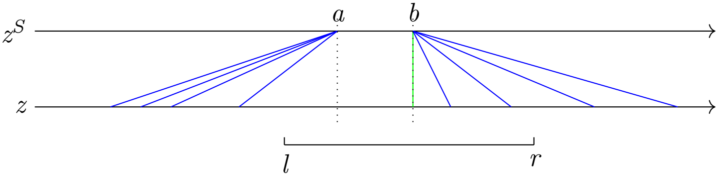


把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 2: $l \le a, b \le r$.

[1] 可能退化成一个点。

大讨论

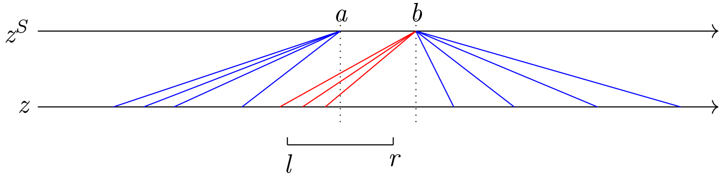


把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 2: $l \leq a, b \leq r$. 此时把 U 中元素的 z_i 都改成 b，结果一定不会变劣，而又在没有破坏原有偏序结构的情况下构造出了“好”的情况，于是“不存在”的假设就寄了。

[1] 可能退化成一个点。

大讨论

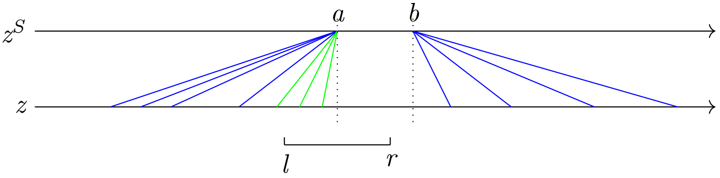


把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 3: $l \le a, r < b$.

[1] 可能退化成一个点。

大讨论



把“不好”的集合记作 U，考虑 U 的 L_1 均值的取值区间[1] $[l, r]$。

Case 3: $l \le a, r < b$. 此时把 U 中元素的 z_i^S 都改成 a，结果严格变优，偏序结构也没有破坏，于是 z^S 的最优性假设就寄了。

[1]可能退化成一个点。

总之就是非常对，并且扩展到任意 $p < \infty$ 都是对的

前面我们说过了，对于 L_1 问题，一定存在一组最优解 z，其中每个元素的取值都是某个出现过的 y_i 值。

于是 z 的取值集合大小不超过 n，只需要在这个集合上整体二分就可以了。

总之就是非常对，并且扩展到任意 $p < \infty$ 都是对的

前面我们说过了，对于 L_1 问题，一定存在一组最优解 z，其中每个元素的取值都是某个出现过的 y_i 值。

于是 z 的取值集合大小不超过 n，只需要在这个集合上整体二分就可以了。

这样的性质在 $p > 1$ 时不再满足。

总之就是非常对，并且扩展到任意 $p < \infty$ 都是对的

前面我们说过了，对于 L_1 问题，一定存在一组最优解 z，其中每个元素的取值都是某个出现过的 y_i 值。

于是 z 的取值集合大小不超过 n，只需要在这个集合上整体二分就可以了。

这样的性质在 $p > 1$ 时不再满足。不过我们有另一个结论。

Lemma

$p > 1$ 时，任意集合的 L_p-mean 是唯一的。

总之就是非常对，并且扩展到任意 $p < \infty$ 都是对的

前面我们说过了，对于 L_1 问题，一定存在一组最优解 z，其中每个元素的取值都是某个出现过的 y_i 值。

于是 z 的取值集合大小不超过 n，只需要在这个集合上整体二分就可以了。

这样的性质在 $p > 1$ 时不再满足。不过我们有另一个结论。

Lemma

$p > 1$ 时，任意集合的 L_p-mean 是唯一的。

由于 X 有限，其所有子集的 L_p-mean 取值范围也有限，故对于任意 $a \in \mathbb{R}$，总能找到 $\varepsilon > 0$ 使得 $(a, a + \varepsilon)$ 内没有最优解里的元素。

总之就是非常对，并且扩展到任意 $p < \infty$ 都是对的

前面我们说过了，对于 L_1 问题，一定存在一组最优解 z，其中每个元素的取值都是某个出现过的 y_i 值。

于是 z 的取值集合大小不超过 n，只需要在这个集合上整体二分就可以了。

这样的性质在 $p > 1$ 时不再满足。不过我们有另一个结论。

Lemma

$p > 1$ 时，任意集合的 L_p-mean 是唯一的。

由于 X 有限，其所有子集的 L_p-mean 取值范围也有限，故对于任意 $a \in \mathbb{R}$，总能找到 $\varepsilon > 0$ 使得 $(a, a + \varepsilon)$ 内没有最优解里的元素。

此时欲做 $L_p^{\{a,a+\varepsilon\}}$ 问题，相比于比较 z_i 取 a 与取 $a + \varepsilon$ 的差值 (太小了！)，可以考虑求代价函数在 a 处的导数作为替代。

算法模板

以 $p = 1$ 为例。

```
function SOLVE(𝒳, Y)
    if |Y| = 1 or 𝒳 = ∅ then
        ...this case is trivial
    else
        mid ← ⌈|Y|/2⌉
        a, b ← the mid-th and (mid + 1)-th smallest element in Y
        solve L_1^S problem for S = {a, b} and obtain z^S
        partition 𝒳 into 𝒳_a, 𝒳_b via z^S
        SOLVE(𝒳_a, first mid elements in Y)
        SOLVE(𝒳_b, last (|Y| − mid) elements in Y)
    end if
end function
```

某校内胡策题

Statement

给定一张 n 个点 m 条边的无向连通图 $G=(V,E)$ 和边集 $E_1,E_2\subseteq E$，每条边有初始权值 d_i，定义一次操作为把一条边的权值加 1 或减 1 ，求至少需要多少次操作可以使

- 边集 E_1 是整张图的一棵最小生成树；
- 边集 E_2 是整张图的一棵最大生成树。

拟阵？

不加证明地引用 [3] 中的一个结论

Theorem(强基交换定理)

对于拟阵 M，假设存在两个不同的基 A, B，那么对于任意一个元素 $x \in A \setminus B$，都存在一个元素 $y \in B \setminus A$，满足 $A \setminus \{x\} \cup \{y\}$ 和 $B \setminus \{y\} \cup \{x\}$ 都是拟阵的基。

拟阵？

不加证明地引用 [3] 中的一个结论

Theorem(强基交换定理)

对于拟阵 M，假设存在两个不同的基 A, B，那么对于任意一个元素 $x \in A \setminus B$，都存在一个元素 $y \in B \setminus A$，满足 $A \setminus \{x\} \cup \{y\}$ 和 $B \setminus \{y\} \cup \{x\}$ 都是拟阵的基。

同样不加证明地声称最小/大生成树是拟阵，并且指出题目中要求的条件成立当且仅当

- $\forall e' \in E \setminus E_1$，若其可以替换 E_1 上的一条边 e，则 $w_{e'} \geq w_e$；E_2 同理。

拟阵？

不加证明地引用 [3] 中的一个结论

Theorem(强基交换定理)

对于拟阵 M，假设存在两个不同的基 A, B，那么对于任意一个元素 $x \in A \setminus B$，都存在一个元素 $y \in B \setminus A$，满足 $A \setminus \{x\} \cup \{y\}$ 和 $B \setminus \{y\} \cup \{x\}$ 都是拟阵的基。

同样不加证明地声称最小/大生成树是拟阵，并且指出题目中要求的条件成立当且仅当

- $\forall e' \in E \setminus E_1$，若其可以替换 E_1 上的一条边 e，则 $w_{e'} \geq w_e$；E_2 同理。

这样就把原问题转化成了保序回归问题，套用前面说过的方法就可以解决了。

是不是有什么忘了讲了

我们好像还没有说 L_p^S 问题咋做...

$L_p \to L_p^S$ 的规约把问题从回归转化成了 2 分类，但此时仍需要考虑偏序关系。

是不是有什么忘了讲了

我们好像还没有说 L_p^S 问题咋做...

$L_p \to L_p^S$ 的规约把问题从回归转化成了 2 分类，但此时仍需要考虑偏序关系。

(摆烂讲法)$a \preceq b$ 视作 $a \to b$ 的一条有向边，此时需要在建出的图中找出一个闭合子图并让其取 S 中较大的值，同时还要求最小代价，可以使用网络流解决。

你已经完全掌握这个算法了，来看一道省选题吧

「联合省选 2020 A」魔法商店

有 n 件标有「价格」和「魅力值」的商品。
定义一个商品集合是「好」的，如果其任意子集「魅力值」异或和不为 0，同时集合大小是满足前者时最大的。
可以以 $(v-v')^2$ 的代价把一件商品的「价格」从 v 修改成 v'。**只能改成整数**。
要求最小化总代价，使最终给定集合 A 是「价格」和最小的「好」的集合，集合 B 是「价格」和最大的「好」的集合。

你已经完全掌握这个算法了，来看一道省选题吧

「联合省选 2020 A」魔法商店

有 n 件标有「价格」和「魅力值」的商品。
定义一个商品集合是「好」的，如果其任意子集「魅力值」异或和不为 0，同时集合大小是满足前者时最大的。
可以以 $(v-v')^2$ 的代价把一件商品的「价格」从 v 修改成 v'。**只能改成整数。**
要求最小化总代价，使最终给定集合 A 是「价格」和最小的「好」的集合，集合 B 是「价格」和最大的「好」的集合。

$\mathbb{F}_2$ 上线性空间也是拟阵...

周歪歪的序列

Statement

有一个长度为 N 的序列以及 M 个形如“第 k 个数是区间 $[l, r]$ 内最小/大的数”限制。
你需要尽量少地修改序列，使得序列满足限制。“尽量少”指的是最小化所有元素的变化量之和。

周歪歪的序列

Statement

有一个长度为 N 的序列以及 M 个形如“第 k 个数是区间 $[l,r]$ 内最小/大的数”限制。
你需要尽量少地修改序列，使得序列满足限制。“尽量少”指的是最小化所有元素的变化量之和。

有没有比网络流更快的实现 2 分类的算法呢？

周歪歪的序列

Statement

有一个长度为 N 的序列以及 M 个形如“第 k 个数是区间 $[l, r]$ 内最小/大的数”限制。
你需要尽量少地修改序列，使得序列满足限制。“尽量少”指的是最小化所有元素的变化量之和。

有没有比网络流更快的实现 2 分类的算法呢？

动态规划，$dp_{i,0/1}$ 表示前 i 个元素已经完成分类时，且第 i 个元素分在第 $0/1$ 类的最小代价。

转移使用线段树优化。

具体实现细节可以询问出题人（雾

$p = \infty$

$p = \infty$ 时，前述做法不再适用了。

$p = \infty$

$p = \infty$ 时，前述做法不再适用了。

但实际上 $p = \infty$ 的问题比 $p < \infty$ 要更 (自然语言意义上的) 简单。考虑

$$\text{minimize} \ \max_{a \in \mathcal{X}} w_a |y_a - z_a|$$

$p = \infty$

$p = \infty$ 时，前述做法不再适用了。

但实际上 $p = \infty$ 的问题比 $p < \infty$ 要更 (自然语言意义上的) 简单。考虑

$$\text{minimize} \max_{a \in \mathcal{X}} w_a |y_a - z_a|$$

“最小化最大值” 可以尝试用二分来解决。

$p = \infty$

$p = \infty$ 时，前述做法不再适用了。

但实际上 $p = \infty$ 的问题比 $p < \infty$ 要更 (自然语言意义上的) 简单。考虑

$$\text{minimize} \max_{a \in \mathcal{X}} w_a |y_a - z_a|$$

“最小化最大值”可以尝试用二分来解决。

二分结果，从而为每个 $a \in \mathcal{X}$ 确定可行的取值范围。

按照 $\mathcal{X}$ 的拓扑序依次确定每个 z_a 的取值。贪心是正确的。

Reference

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

高睿泉.
浅谈保序回归问题.
IOI2018 中国国家候选队论文集, pages 23–33, 2018.

杨乾澜.
浅谈拟阵的一些拓展及其应用.
IOI2018 中国国家候选队论文集, pages 143–163, 2018.