Algorithms for Isotonic Regression and Two Regularized Extensions: An Empirical Comparison

Julius Graf Louis Sallé-Tourne

May 24, 2025

Contents

1 Introduction

Isotonic regression consists of finding a monotone function g that best approaches a function for a certain norm $\|\cdot\|$. In the discrete and non-decreasing setting, given observations $((x_i, y_i))_{1\leqslant i\leqslant n}$, where the $(x_i)_{1\leqslant i\leqslant n}$ are non-decreasingly ordered, this reduces to solving the following optimization problem:

$$\min_{g\in\mathcal{M}} \|g-y\|^2 \tag{1}$$

where $\mathcal{M}$ is the subset of $\mathbb{R}^n$ of non-decreasingly ordered vectors. In the whole text, we will assume $(x_i)_{1\leqslant i\leqslant n}$ to be equally spaced for simplicity. Given that $\mathcal{M}$ couples simple geometry with global constraints, isotonic regression is perfect for application of nonlinear-optimization techniques that blend projection operators, first-order methods and barrier approaches.

This project – part of the IEOR 262B class at the University of California at Berkeley – consists in evaluating and comparing different algorithms – PAVA, which we will introduce in section 2, Projected Gradient Descent, Frank-Wolfe and Interior-Point Methods to solve the above problem. Furthermore, we will introduce two extensions to isotonic regressions – ℓ^1-penalized isotonic regression and nearly-isotonic regression – for which we will suggest methods for resolution and implement a numerical analysis. All methods are implemented in `Python` or `R` and assessed on the college-admissions data set of Acharya et al. (2019) [1].

2 Related Work

Isotonic regression is most relevant whenever theory dictates that the response can only rise (or fall) with the predictor. In practice, isotonic regression can be useful to model dose-response and toxicity curves in biostatistics or monotone hazard or cumulative incidence functions in survival analysis. One can also use isotonic regression for ranking scores – as we will do in section 5 – or demand, cost, or item-response functions in economics.

Classical isotonic regression dates back to the seminal monograph of Barlow et al. (1972), who established basic existence, uniqueness, and the $O(n)$ Pool Adjacent Violators Algorithm (PAVA) [2]. Subsequent work clarified its statistical properties – e.g., consistency under a monotone truth (Robertson et al., 1988) – and showed that the projection onto the monotone cone – later defined as $\text{proj}_{\mathcal{M}}$, and as we have said in class – is firmly non-expansive [6]. Since the isotonic regression problem is a convex optimization, rich literature exists on the algorithms used to solve the problem. Projected Gradient Descent (Rosen, 1960), Frank-Wolfe (Frank and Wolfe, 1956), and Interior-Point Methods (Dikin, 1967) are only a few algorithms that can be used to solve the isotonic regression problem [7][4][3].

Several extensions relax the hard monotonicity constraint to obtain models that adapt locally to the signal while retaining global shape information. Tibshirani, Hoefling & Tibshirani (2011) proposed nearly-isotonic regression (NISO) as a softer alternative in which local violations $(g_i - g_{i+1})^+$ are penalized linearly rather than forbidden [8]. Pastukhov (2024) generalizes NISO to a framework combining NISO with the ℓ^1-regularized regression. The pure ℓ^1-regularized regression still imposes the hard monotonicity constraint but promotes sparsity. As part of the numerical simulations, we will investigate how those two extensions – NISO and ℓ^1-regularization – compare to isotonic regression in terms of solution shape, convergence speed and precision.

3 Problem Formulation and Preliminaries

Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ for $n \geqslant 1$. Let $y \in \mathbb{R}^n$. Consider the problem

$$\begin{aligned}(\mathcal{P})\colon \text{ minimize} \quad & \frac{1}{2}\|g-y\|^2 \\ \text{s.t.} \quad & g \in \mathcal{M}\end{aligned}$$

First, problem $(\mathcal{P})$ is a convex optimization problem (see lemma 3.1 below) and its objective function is strictly convex and coercive. Thus, $(\mathcal{P})$ admits a unique solution and we will see in subsection 3.1 how it may be computed. The following lemma holds and its proof only requires small verifications.

Lemma 3.1. *$\mathcal{M}$ is a closed, convex polyhedral cone with non-empty interior* $\mathrm{int}(\mathcal{M})$ *the the subset of* $\mathbb{R}^n$ *of increasingly ordered vectors.*

3.1 PAVA and Projected Gradient Method

Naturally, the optimal candidate one could think of for solving $(\mathcal{P})$ is $\mathrm{proj}_{\mathcal{M}}(y)$ which exists and is unique by class since $\mathcal{M}$ is a closed convex set. PAVA, the Pool Adjacent Violators Algorithm, introduced by Barlow et al. (1972) is a method that allows to compute algorithmically the operator $\mathrm{proj}_{\mathcal{M}}$. Grotzinger and Witzgall (1984) implemented PAVA in $O(n)$, making this algorithm very efficient. From an intuitive point of view, the algorithm works as follows, as explained in [8]. The idea is to begin at the left-most value y_1 and then move towards the right until we find a point y_{i+1} such that $y_i > y_{i+1}$ i.e. not respecting monotonicity. Then, we replace y_i and y_{i+1} by the average of y_i and y_{i+1}. Should this mean be smaller than y_{i-1}, one repeats the averaging process until monotonicity is established (on the left-hand side of index i). Then continue towards the right until index n is reached. More formally, PAVA writes down as follows.

Algorithm 1 Pool Adjacent Violators Algorithm for $(\mathcal{P})$

Initialize blocks via $g_i \leftarrow y_i$, $w_i \leftarrow 1$ for $i \in [\![1, n]\!]$, set $k \leftarrow 1$ and $m \leftarrow n$
while $k < m$ **do**
  if $g_k \leqslant g_{k+1}$ **then**
    $k \leftarrow k + 1$
  else
    Weight $g_k \leftarrow (w_k g_k + w_{k+1} g_{k+1})/(w_k + w_{k+1})$ and set $w_k \leftarrow w_k + w_{k+1}$
    Delete block $k + 1$ and $m \leftarrow m - 1$
    $k \leftarrow \max\{1,\, k - 1\}$
  end if
end while
Replicate each pooled value across its block to obtain the isotonic $g^* = \mathrm{proj}_{\mathcal{M}}(y)$.

The Projected Gradient Method (PGD) we have covered in class uses the projection onto $\mathcal{M}$ at each iteration. Naturally, this algorithm is less efficient than the benchmark PAVA. Since $(\mathcal{P})$ is already a projection problem, the Projected Gradient Method indeed doesn't make much sense for that. Since the goal of this report is to compare the performance of different iterative methods covered in class on $(\mathcal{P})$, we still included it.

Algorithm 2 Projected Gradient Method for $(\mathcal{P})$

Initialize $x^0 \in \mathcal{M}$, relaxation schedule $(\tau_n)_{n\in\mathbb{N}} \in]0, 1]^{\mathbb{N}}$ and stepsizes $(\gamma_n)_{n\in\mathbb{N}} \in]0, +\infty[^{\mathbb{N}}$
for $k = 0, \ldots, K - 1$ **do**
  $g^{k+1} \leftarrow g^k + \tau_k(\mathrm{proj}_{\mathcal{M}}(g^k - \gamma_k(g^k - y)) - g^k)$
end for

3.2 Interior-Point Methods

Next, we turn towards interior-point methods. Let us first introduce the notion of barrier functions, already visited in class.

Definition 3.1. *Let S be a closed, convex subset of $\mathbb{R}^n$ with non-empty interior. A continuous function $\varphi\colon \operatorname{int}(S) \to \mathbb{R}$ is called a barrier function if $\varphi(x) \to +\infty$ whenever $x \to \partial S$.*

The idea of interior point methods is to introduce a barrier parameter $\theta > 0$ and a barrier function φ and, instead of solving $(\mathcal{P})$ directly, solve

$$\begin{aligned} \mathcal{P}(\theta)\colon \text{ minimize} \quad & \|g - y\|^2 + \theta\varphi(g) \\ \text{s.t.} \quad & g \in \operatorname{int}(\mathcal{M}) \end{aligned}$$

where the barrier parameter $\theta > 0$ trades off between two behaviors. When θ is small, the solution to $\mathcal{P}(\theta)$ is close to a solution of $(\mathcal{P})$. When θ is larger, $\mathcal{P}(\theta)$ behaves more like a truly unconstrained problem. Hence, the Interior Point Method (IPM) consists in iteratively reducing $\theta \to 0$, while at each iteration moving towards an optimal direction for $\mathcal{P}(\theta)$, which will eventually allow to land at an optimal solution to $(\mathcal{P})$.

To build the IPM more efficiently, we will formulate $(\mathcal{P})$ in standard quadratic programming form before introduction the barrier penalty. We can write the $n-1$ monotonicity constraints compactly in the form $Dg \geqslant 0$, with

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \in \mathcal{M}_{n-1,n}(\mathbb{R}).$$

We now want to write $(\mathcal{P})$ in standard quadratic programming. Defining

$$\begin{aligned} Q &= \begin{bmatrix} I_n & -I_n & 0 \\ -I_n & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathcal{M}_{3n-1}(\mathbb{R}) \\ A &= \begin{bmatrix} D & -D & -I_{n-1} \end{bmatrix} \in \mathcal{M}_{n-1,3n-1}(\mathbb{R}) \\ c &= \begin{bmatrix} -y^\top & y^\top & 0^\top \end{bmatrix}^\top \in \mathbb{R}^{3n-1} \end{aligned}$$

we obtain, via the lemma below, the standard quadratic programming formulation for $(\mathcal{P})$

Lemma 3.2. $Q \succcurlyeq 0$.

Proof. For all $z \in \mathbb{R}^{3n-1}$, let us write $z = (z_1, z_2, z_3)$ with $z_1, z_2 \in \mathbb{R}^n$ and $z_3 \in \mathbb{R}^{n-1}$. We have

$$z^\top Q z = \begin{bmatrix} z_1^\top & z_2^\top & z_3^\top \end{bmatrix} \begin{bmatrix} I_n & -I_n & 0 \\ -I_n & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = z_1^\top z_1 - 2 z_1^\top z_2 + z_2^\top z_2 = \|z_1 - z_2\|_2^2 \geqslant 0.$$

□

We now can write problem $(\mathcal{P})$ as

$$\begin{aligned} (\mathcal{P})\colon \text{ minimize} \quad & \frac{1}{2} x^\top Q x + c^\top x \\ \text{s.t.} \quad & \begin{cases} Ax = 0 \\ x \geqslant 0 \end{cases} \end{aligned}$$

Indeed, since the sign of g is not imposed, we decompose g into g_+ and g_-, and add a slack variable $z \in \mathbb{R}^{n-1}$ associated to the $n-1$ monotonicity constraints, in order to define a new

variable $x = \begin{bmatrix} g_+^\top & g_-^\top & z^\top \end{bmatrix}^\top \in \mathbb{R}^{3n-1}$. Now let $\theta > 0$. We will use the logarithmic barrier function φ defined on the interior of $\ker(A) \cap \mathbb{R}_+^{3n-1}$ by

$$\begin{aligned} \varphi\colon \ker(A) \cap (\mathbb{R}_+^*)^{3n-1} &\to \mathbb{R} \\ x &\mapsto -\sum_{j=1}^{3n-1} \ln(x_j) \end{aligned}$$

Instead of solving $(\mathcal{P})$, we will thus solve

$$\begin{aligned} \mathcal{P}(\theta)\colon \text{ minimize } & \frac{1}{2} x^\top Q x + c^\top x - \theta \sum_{j=1}^{3n-1} \ln(x_j) \\ \text{s.t. } & Ax = 0 \end{aligned}$$

In fact, the constraint $x > 0$ is implicit for the domain of the barrier function. We now write the KKT conditions for $\mathcal{P}(\theta)$ at $\bar{x} > 0$. Since $\mathcal{P}(\theta)$ is a convex optimization problem and a Slater point to $\mathcal{P}(\theta)$ exists, the KKT conditions are necessary and sufficient to characterize local optimality. We thus want to move into a direction satisfying the KKT conditions. Let $\bar{X} = \operatorname{diag}((\bar{x}_j)_{1 \leqslant j \leqslant n})$. In what follows we denote $e = (1, \dots, 1)$ and $e_j = (\delta_{i,j})_{1 \leqslant i \leqslant n}$ for $\delta_{i,j}$ the Kronecker symbol and $j \in [\![1, n]\!]$. We have

$$(\mathbf{K}) \begin{cases} Q\bar{x} + c - \theta \bar{X}^{-1} e + A^\top \nu = 0 \\ A\bar{x} = 0 \end{cases}$$

for some $\nu \in \mathbb{R}^{n-1}$. Letting $s \in \mathbb{R}^{3n-1}$ such that $\bar{x}_j s_j = \theta$ for all $j \in [\![1, 3n-1]\!]$, we can rewrite $(\mathbf{K})$ as

$$(\mathbf{L}) \begin{cases} Q\bar{x} + c - s + A^\top \nu = 0 \\ A\bar{x} = 0 \\ \bar{X} S e - \theta e = 0 \end{cases}$$

System $(\mathbf{L})$ allows to compute the Newton direction $\delta = (\delta_{\bar{x}}, \delta_\nu, \delta_s)$ in which to move at each iteration. Precisely, $(\mathbf{L})$ writes as $\Psi(\bar{x}, \nu, s) = 0$ for $\Psi\colon \mathbb{R}^{7n-3} \to \mathbb{R}^{7n-3}$ trivially defined through $(\mathbf{L})$. Thus, the Newton direction satisfies $\nabla\Psi(\bar{x}, \nu, s)\delta = \Psi(\bar{x}, \nu, s)$, i.e. δ solves

$$\begin{bmatrix} Q & A^\top & -I_{3n-1} \\ A & 0 & 0 \\ S & 0 & \bar{X} \end{bmatrix} \begin{bmatrix} \delta_{\bar{x}} \\ \delta_\nu \\ \delta_s \end{bmatrix} = \begin{bmatrix} Q\bar{x} + c - s + A^\top \nu \\ A\bar{x} \\ \bar{X} S e - \theta e \end{bmatrix}.$$

This leads to the following algorithm, given by [5].

Algorithm 3 Interior Point Method for $(\mathcal{P})$

1: Initialize stepsize (x^0, ν^0, s^0), $k = 0$, $\theta^0 = \langle x^0, s^0 \rangle / (3n-1)$ and $\alpha \in]0, 1[$
2: **while** optimality is not reached or $k \leqslant K - 1$ **do**
3: $\quad (\bar{x}, \bar{\nu}, \bar{s}) \leftarrow (x^k, \nu^k, s^k)$ and $\tilde{\theta} \leftarrow \theta^k$
4: $\quad$ Shrink θ as $\bar{\theta} \leftarrow \alpha\tilde{\theta}$
5: $\quad$ Find $\delta \in \mathbb{R}^{7n-3}$ such that $\nabla\Psi(\bar{x}, \bar{\nu}, \bar{s})\delta = \Psi(\bar{x}, \bar{\nu}, \bar{s})$ by solving

$$\begin{bmatrix} Q & A^\top & -I_{3n-1} \\ A & 0 & 0 \\ \bar{S} & 0 & \bar{X} \end{bmatrix} \begin{bmatrix} \delta_{\bar{x}} \\ \delta_{\bar{\nu}} \\ \delta_{\bar{s}} \end{bmatrix} = \begin{bmatrix} Q\bar{x} + c - \bar{s} + A^\top \bar{\nu} \\ A\bar{x} \\ \bar{X}\bar{S}e - \bar{\theta}e \end{bmatrix}$$

6: $\quad$ Update $(x^{k+1}, \nu^{k+1}, s^{k+1}) \leftarrow (\bar{x}, \bar{\nu}, \bar{s}) + (\delta_{\bar{x}}, \delta_{\bar{\nu}}, \delta_{\bar{s}})$, $\theta^{k+1} \leftarrow \bar{\theta}$ and $k \leftarrow k + 1$
7: **end while**
8: Recover g^* as the first $2n$ components of x^*

3.3 Frank-Wolfe Method

For the Frank-Wolfe method, one would have to require $\mathcal{M}$ to be compact, which does not hold because $\mathcal{M}$ is *a priori* not compact. As we have seen, $\text{proj}_{\mathcal{M}}(y)$ can be computed via PAVA. Therefore, looking much further than $\text{proj}_{\mathcal{M}}(y)$ will not be useful to the resolution of $(\mathcal{P})$, thus giving us, informally, the existence of a bound beyond which the objective function won't be "small" anymore. This leads to the following assumption.

Assumption 1. *There exists $B > 0$ such that $(\mathcal{P})$ admits the same minimizers as the truncated problem*

$$(\mathcal{P}_B)\colon \text{ minimize } \frac{1}{2}\|g-y\|^2 \quad s.t. \quad g \in \mathcal{M} \cap [-B, B]^n.$$

This assumption will almost always be satisfied. For instance, in the numerical application, $y \in [0,1]^n$. The Frank-Wolfe method for $(\mathcal{P})$ thus applies and writes in the following way:

Algorithm 4 Frank-Wolfe Method for $(\mathcal{P})$

1: Initialize $g^0 \in \mathcal{M}$ and $k \leftarrow 0$. Let $f(g) = \frac{1}{2}\|g-y\|^2$ for all $g \in \mathcal{M}$.
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Compute $\nabla f(g^k) = g^k - y$
4: Solve linear optimization problem $\tilde{g}^k \leftarrow \arg\min_{g\in\mathcal{M}}\{f(g^k) + \nabla f(g^k)^\top (g - g^k)\}$
5: Perform line-search $\bar{\alpha}^k \leftarrow \arg\min_{\alpha\in[0,1]} f(g^k + \alpha(\tilde{g}^k - g^k))$
6: Update $g^{k+1} \leftarrow g^k + \bar{\alpha}^k(\tilde{g}^k - g^k)$
7: **end for**

Instead of the box constraint in $(\mathcal{P}_B)$, one might want to consider a regularization constraint like $\|g\|_p \leqslant \delta$ for $p = 1$ or $p = 2$. This will be part of future work.

4 Two Relaxations of $(\mathcal{P})$

4.1 ℓ^1-Regularized Isotonic Regression

While isotonic regression enforces strict monotonicity, real-world datasets often include features whose relationship with the outcome is weak, noisy, or only approximately monotonic. Instead of manually selecting features to include in a monotonic model, we introduce ℓ^1-regularization to automatically promote sparsity – allowing the model to retain only those features that contribute meaningfully to a monotonic trend. ℓ^1-regularization thus induces sparsity and selects the most relevant features while maintaining strict monotonicity. The associated problem writes:

$$(\mathcal{P}^1_\lambda)\colon \text{ minimize } \frac{1}{2}\|g-y\|^2 + \lambda\|Dg\|_1 \quad \text{s.t.} \quad g \in \mathcal{M}$$

where $D\colon \mathbb{R}^n \to \mathbb{R}^{n-1}$ is the difference operator. Since $g \in \mathcal{M}$ is equivalent to $Dg \geqslant 0$, we can write $\|Dg\|_1 = e^\top Dg$, such that we have:

$$(\mathcal{P}^1_\lambda)\colon \text{ minimize } \frac{1}{2}x^\top Qx + c_\lambda^\top x \quad \text{s.t.} \quad \begin{cases} Ax = 0 \\ x \geqslant 0 \end{cases}$$

for $c_\lambda = \begin{bmatrix} -y^\top & y^\top & \lambda e^\top \end{bmatrix}^\top \in \mathbb{R}^{3n-1}$ (note that e here is in $\mathbb{R}^{n-1}$). Using the work done in Subsection 3.2, we obtain the following algorithm to solve $(\mathcal{P}^1_\lambda)$, in which Ψ_λ is defined as Ψ, by replacing c by c_λ naturally (KKT conditions still work because the objective remains convex).

Algorithm 5 Interior Point Method for $(\mathcal{P}^1_\lambda)$

1: Initialize stepsize (x^0, ν^0, s^0), $k = 0$, $\theta^0 = \langle x^0, s^0 \rangle / (3n-1)$ and $\alpha \in]0, 1[$
2: **while** optimality is not reached or $k \leqslant K - 1$ **do**
3: $(\bar{x}, \bar{\nu}, \bar{s}) \leftarrow (x^k, \nu^k, s^k)$ and $\tilde{\theta} \leftarrow \theta^k$
4: Shrink θ as $\bar{\theta} \leftarrow \alpha\tilde{\theta}$.
5: Find $\delta \in \mathbb{R}^{7n-3}$ such that $\nabla\Psi_\lambda(\bar{x}, \bar{\nu}, \bar{s})\delta = \Psi_\lambda(\bar{x}, \bar{\nu}, \bar{s})$ by solving

$$\begin{bmatrix} Q & A^\top & -I_{3n-1} \\ A & 0 & 0 \\ \bar{S} & 0 & \bar{X} \end{bmatrix} \begin{bmatrix} \delta_{\bar{x}} \\ \delta_{\bar{\nu}} \\ \delta_{\bar{s}} \end{bmatrix} = \begin{bmatrix} Q\bar{x} + c_\lambda - \bar{s} + A^\top\bar{\nu} \\ A\bar{x} \\ \bar{X}\bar{S}e - \bar{\theta}e \end{bmatrix}.$$

6: Update $(x^{k+1}, \nu^{k+1}, s^{k+1}) \leftarrow (\bar{x}, \bar{\nu}, \bar{s}) + (\delta_{\bar{x}}, \delta_{\bar{\nu}}, \delta_{\bar{s}})$, $\theta^{k+1} \leftarrow \bar{\theta}$ and $k \leftarrow k + 1$
7: **end while**
8: Recover g^* as the first $2n$ components of x^*

4.2 Nearly-Isotonic Regression

Nearly-isotonic regression allows controlled monotonicity violations to improve the fit (as discussed in the work of Tibshirani et al. [8]). The associated problem writes:

$$(\mathcal{P}^+_\lambda): \text{ minimize } \frac{1}{2}\|g - y\|^2 + \lambda \sum_{i=1}^{n-1} (g_i - g_{i+1})_+$$
$$\text{s.t. } g \in \mathbb{R}^n$$

As the following table sums up, we cannot apply the IPM or PGD to $(\mathcal{P}^+_\lambda)$ because of a lack of smoothness (while $(\mathcal{P}^1_\lambda)$ is not smooth too, the definition of $\mathcal{M}$ makes it smooth), nor the Frank-Wolfe method because it requires a compact constraint set, which $\mathbb{R}^n$ fails to satisfy.

Feature	$(\mathcal{P})$	$(\mathcal{P}^1_\lambda)$	$(\mathcal{P}^+_\lambda)$
Feasible set	Polyhedral cone	Polyhedral cone	$\mathbb{R}^n$
Smooth?	Yes	No	No
Strongly convex?	Yes	Yes	Yes
Projection needed?	Onto $\mathcal{M}$	Onto $\mathcal{M}$	No
Typical solution shape	Non-decreasing staircase	Sparser staircase	May decrease locally

Table 1: Key properties of the three formulations

Investigating further methods to solve $(\mathcal{P}^+_\lambda)$ can be explored in the future. We remind here that Tibshirani et al. (2011) already presented a modified PAVA, which allows to solve $(\mathcal{P}^+_\lambda)$ in $O(n \log n)$. Due to a lack of time, and for the sake of clarity of the other notions presented in this report, we decided not to include a theoretical analysis of the modified PAVA and other algorithms in this report. We did, still, include a numerical analysis of $(\mathcal{P}^+_\lambda)$, using the ECOS solver to solve the problem.

5 Numerical Simulations

We work with the College Admissions dataset provided by Acharya et al. (2019), which contains applicants' GRE scores and their probability of admission [1]. Figure 1 shows the raw relationship between GRE score and chance of admit, motivating our assumption of a non-decreasing.

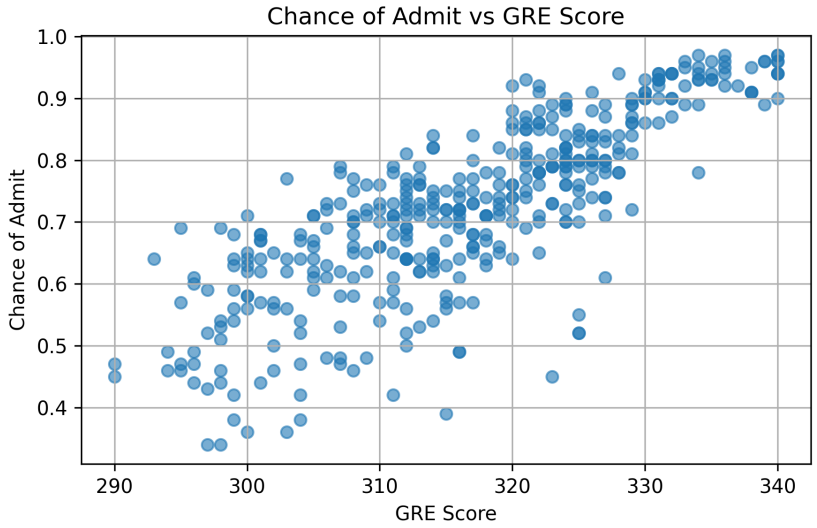


Figure 1: Scatter of Chance of Admit vs. GRE Score.

We implement PAVA, PGD and Frank-Wolfe ourselves in Python and compare them to an IPM via CVXPY. For $(\mathcal{P})$ we call the OSQP solver (an operator-splitting quadratic programming method). For the ℓ^1-regularized problem $(\mathcal{P}_\lambda^1)$ and nearly-isotonic problem $(\mathcal{P}_\lambda^+)$, we call the ECOS solver, which implements a primal-dual interior-point algorithm aligning with algorithm 5: it uses log-barrier terms to enforce inequality constraints and each Newton step solves a linear KKT system. The source code employed for the numerical simulations is publicly accessible at `https://github.com/louis-salletourne/isotonic-regression`.

5.1 Benchmark on $(\mathcal{P})$

Table 2 collects final objective, relative error to PAVA, iteration counts and runtimes. PAVA itself and our Frank-Wolfe and PGD implementations all recover the exact optimum in under half a millisecond. IPM via OSQP delivers the same solution to machine precision (1.6×10^{-15}) but requires several thousand Newton iterations and ~ 50 ms.

Algorithm	Objective	Relative Error	# Iterations	Runtime (ms)
PAVA	1.2627	0	1	1.3249
PGD	1.2627	0	2	0.3934
FW	1.2627	0	1	0.1763
IPM	1.2627	1.6×10^{-15}	3850	49.9039

Table 2: Comparison of PAVA, Projected Gradient (PGD), Frank-Wolfe (FW) and Interior-Point (IPM) on $(\mathcal{P})$.

Figure 2 confirms all four fits coincide exactly. Figure 3 shows the PGD fit and residuals. The residuals exhibit no obvious pattern or autocorrelation, suggesting that the assumption of independent errors is reasonable.

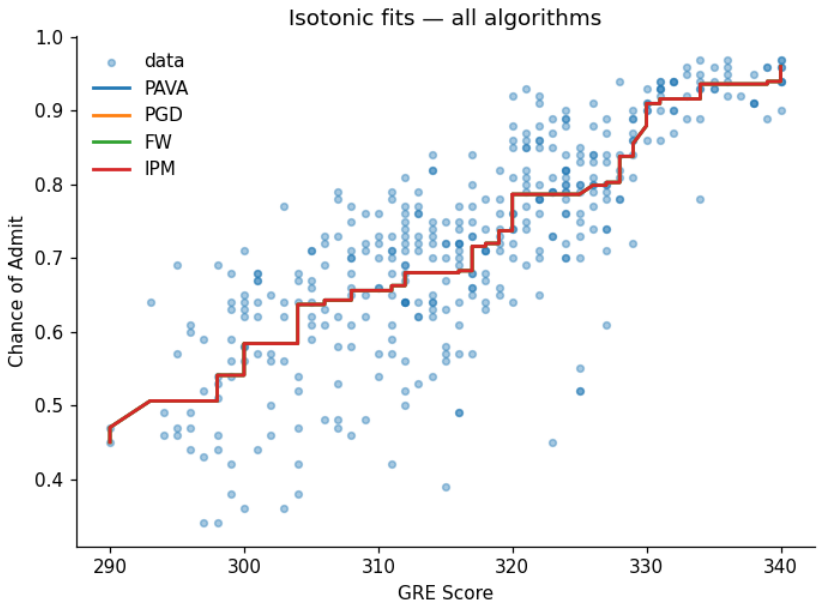
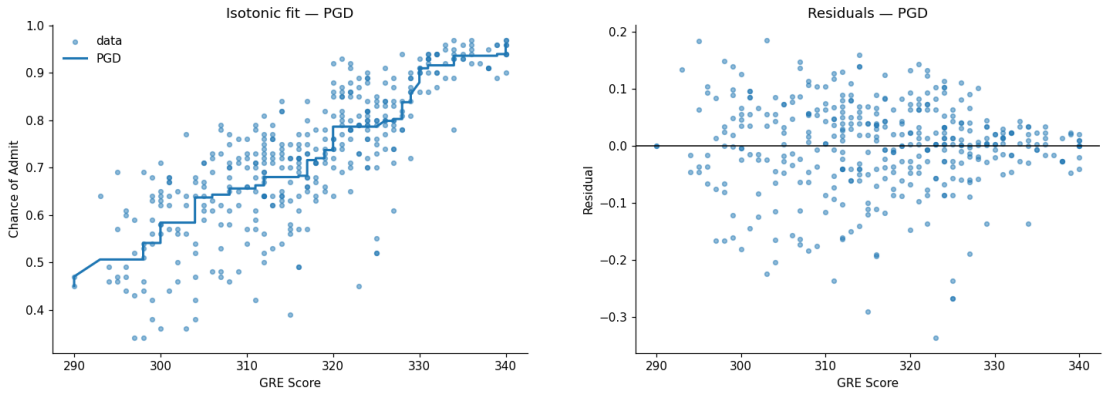


Figure 2: Isotonic fits for PAVA, PGD, FW and IPM on $(\mathcal{P})$.



PGD fit PGD residuals

Figure 3: PGD Regression and Residual Plots

5.2 IPM on $(\mathcal{P}_\lambda^1)$

We now solve problem $(\mathcal{P}_\lambda^1)$ via IPM (ECOS) for $\lambda \in \{0.01, 0.1, 1, 10, 100\}$. Table 3 reports final objective, RMSE (vs. raw y), relative error to PAVA, runtimes and IPM iterations.

λ	Objective	RMSE	Relative Error	Runtime (ms)	# Iterations
0.01	1.2677	0.0795	7.62×10^{-4}	15.7628	19
0.1	1.3091	0.0796	5.49×10^{-3}	9.5379	20
1.0	1.6648	0.0806	1.85×10^{-2}	9.3300	19
10.0	3.5569	0.1118	1.07×10^{-1}	10.0069	21
100.0	4.0573	0.1424	1.61×10^{-1}	6.9380	12

Table 3: IPM on ℓ^1-regularized isotonic regression $(\mathcal{P}_\lambda^1)$.

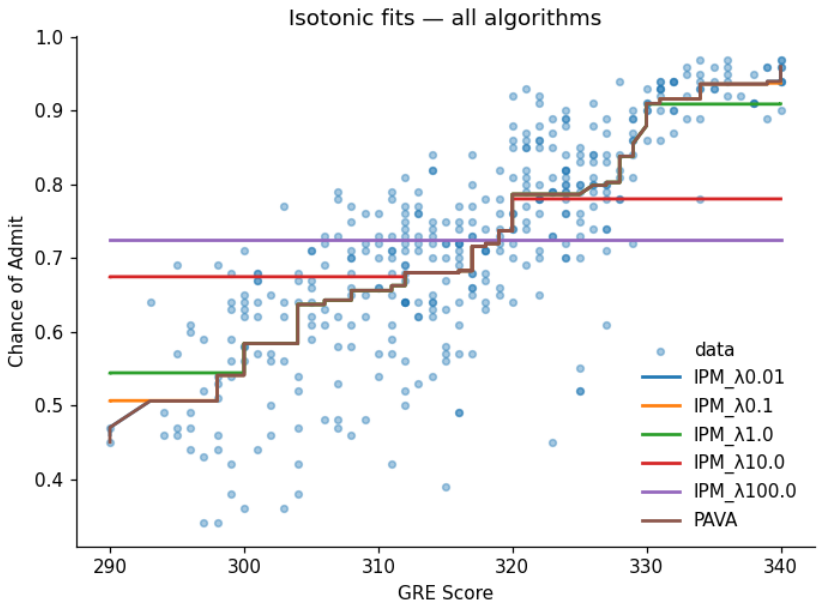
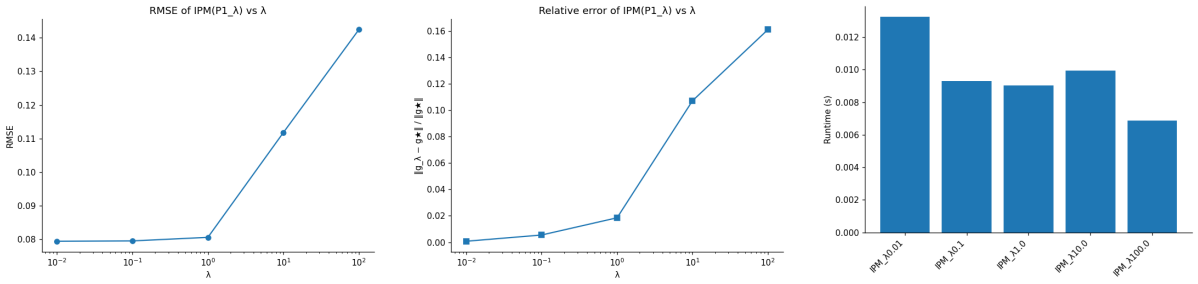


Figure 4: Isotonic fits for IPM($\mathcal{P}_\lambda^1$) across all λ.



(a) RMSE vs. λ (log-scale).

(b) Relative error vs. λ (log-scale).

(c) Runtime vs. λ.

Figure 5: IPM($\mathcal{P}_\lambda^1$) performance as a function of λ.

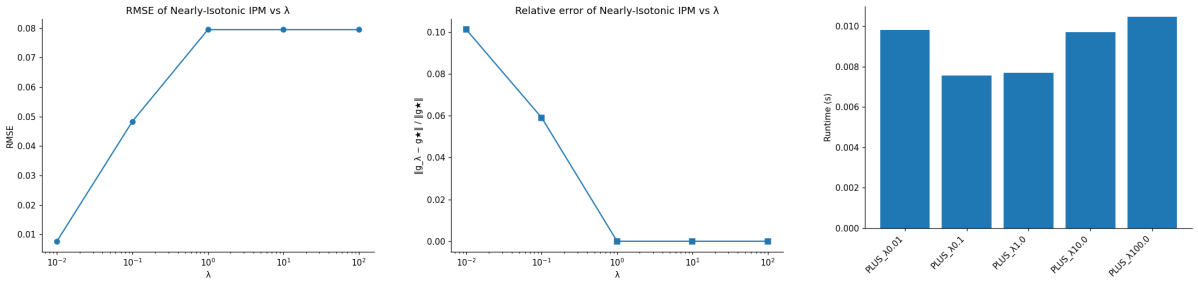
As λ increases, the staircase becomes progressively coarser (Figure 4), and RMSE rises from 0.0795 at $\lambda = 0.01$ to 0.1424 at $\lambda = 100$, reflecting increased under-fitting. The relative error to the PAVA solution likewise grows from 7.6×10^{-4} to 1.6×10^{-1}. Despite these changes in fit, the IPM solve time remains under 16 ms per problem and iteration counts stay near 20, showing that moderate ℓ^1 regularization sparsifies the fit with negligible extra computational cost.

5.3 IPM on ($\mathcal{P}_\lambda^+$)

We now move on to solve ($\mathcal{P}_\lambda^+$) numerically using the ECOS solver. Table 4 shows that for small $\lambda = 0.01$, the nearly-isotonic model captures noise (RMSE at 0.0077) at the expense of monotonicity (relative error at 0.1013). As λ grows, the monotonicity constraint is enforced more strongly, overfitting is reduced, and the solution converges to the strict isotonic fit (by $\lambda = 1$ it is numerically identical).

λ	Objective	RMSE	Relative Error	Runtime (ms)	# Iterations
0.01	0.1583	0.0077	0.10124636	9.7971	15
0.1	0.8864	0.0483	0.05909950	7.5581	17
1.0	1.2627	0.0795	0.0000010353	7.6811	17
10.0	1.2627	0.0795	0.0000006649	9.7108	22
100.0	1.2627	0.0795	0.0000002604	10.4671	24

Table 4: IPM on nearly-isotonic regression ($\mathcal{P}_\lambda^+$).



(a) RMSE vs. λ (log-scale). (b) Relative error vs. λ (log-scale). (c) Runtime vs. λ.

Figure 6: IPM($\mathcal{P}_\lambda^+$) performance vs. λ.

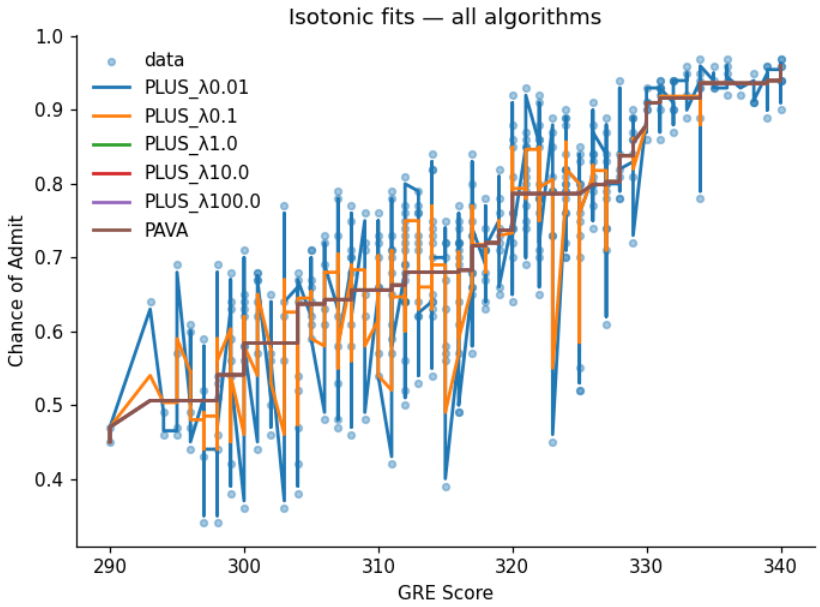


Figure 7: Nearly-isotonic fits for IPM($\mathcal{P}_\lambda^+$) across λ.

6 Conclusion

In this report, we showed that when the goal is strict isotonic regression on a univariate sequence, the classical Pool Adjacent Violators Algorithm remains the gold standard: it attains the exact optimum in $O(n)$ time and, on the GRE-admission benchmark, converges in ≈ 1 ms. Projected Gradient Descent and Frank-Wolfe achieve the same solution almost as quickly. Interior-Point Methods are one to two orders of magnitude slower, but they become interesting once additional

convex penalties are introduced. With an ℓ^1 difference penalty, IPM produces progressively sparser "staircases" reducing model complexity with only a marginal rise in RMSE for $\lambda \leqslant 1$. The nearly-isotonic penalty trades exact monotonicity for a tighter fit at small λ and smoothly interpolates back to the strict solution as λ grows.

Since we have only covered nearly-isotonic regression numerically here, future work should focus on evaluating methods to solve nearly-isotonic regression – such as the modified PAVA introduced by Tibshirani et al. (2011) [8] – both theoretically and numerically to get a deeper understanding. Furthermore, we did not care too much about theoretical properties when applying the ECOS solver to $(\mathcal{P}_\lambda^+)$. Thus, work on further enhancing the modified PAVA or reformulating $(\mathcal{P}_\lambda^1)$ and $(\mathcal{P}_\lambda^+)$ for immediate application of classical methods – like Frank-Wolfe or Interior-Point Methods – could be a great avenue for further research. In particular, one could start by considering a regularization constraint like $\|g\|_p \leqslant \delta$ for $p = 1$ or $p = 2$ instead of the box constraint $g \in [-B, B]^n$.

References

[1] Mohan S Acharya, Asfia Armaan, and Aneeta S Antony. A Comparison of Regression Models for Prediction of Graduate Admissions. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5, 2019.

[2] Richard E Barlow and Hugh D Brunk. The Isotonic Regression Problem and its Dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972.

[3] Iliya Iosiphovich Dikin. Iterative Solution of Problems of Linear and Quadratic Programming. In *Soviet Math. Dokl.*, volume 8, pages 674–675, 1967.

[4] Marguerite Frank, Philip Wolfe, et al. An Algorithm for Quadratic Programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[5] Jacek Gondzio. Interior Point Methods 25 Years Later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[6] Tim Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference*. John Wiley & Sons, New York, 1988.

[7] Jo Bo Rosen. The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints. *Journal of the society for industrial and applied mathematics*, 8(1):181–217, 1960.

[8] Ryan J Tibshirani, Holger Hoefling, and Robert Tibshirani. Nearly-Isotonic Regression. *Technometrics*, 53(1):54–61, 2011.