# Lec 13: Shape-constrained Regression & Course Recap

Yanjun Han

Dec. 12, 2023

Regression: given $(x_1, y_1), \cdots, (x_n, y_n)$, estimate $f(x) = \mathbb{E}[Y \mid X = x]$

Previous lectures: smoothness assumption on $f$ ($\|f^{(k)}\|_\infty \le L$ or $\|f^{(k)}\|_2 \le L$);

several estimators (Nadaraya-Watson, local poly, splines, Fourier, Wavelet, etc.);

approximation theory plays a key role.

This lecture: shape constraint on $f$ (monotone, convex, $\cdots$)

Isotonic regression: $f(x_1) \le f(x_2)$ as long as $x_1 \le x_2$ ($f$ increasing)

W.l.o.g. assume that $x_1 < x_2 < \cdots < x_n$, and

$y_i \sim N(x_i, \sigma^2)$, $i = 1, 2, \cdots, n$.

Motivation from MLE: instead of estimating the entire function $f$, let's estimate $(f(x_1), \cdots, f(x_n))$ first

Q: given estimates of $(\hat{\theta}_1, \cdots, \hat{\theta}_n)$ of $(f(x_1), \cdots, f(x_n))$, when do they give rise to a monotone function $\hat{f}$?

A: very easy — just need $\hat{\theta}_1 \le \hat{\theta}_2 \le \cdots \le \hat{\theta}_n$!

(use piecewise constant/linear function to find $\hat{f}$)

( Similar idea to splines: in smoothing spline, one also hypothetically:

1. fix the estimates $(\hat{\theta}_1, \cdots, \hat{\theta}_n)$ for $(f(x_1), \cdots, f(x_n))$;
2. construct the most smooth function $\hat{f}$ with $\hat{f}(x_i) = \hat{\theta}_i$, $i = 1, 2, \cdots, n$

   — $\hat{f}$ turns out to be a spline!
3. find $(\hat{\theta}_1, \cdots, \hat{\theta}_n)$ to minimize $\frac{1}{n} \sum_{i=1}^{n} (\hat{\theta}_i - y_i)^2 + \lambda \cdot R(\hat{f})$. )

## Resulting estimator :

$(\hat{\theta}_1, \cdots, \hat{\theta}_n)$ is the solution to the following program:

$$\min \quad \sum_{i=1}^{n} (y_i - \hat{\theta}_i)^2$$

$$\text{s.t.} \quad \hat{\theta}_1 \le \hat{\theta}_2 \le \cdots \le \hat{\theta}_n .$$

**Computation** : a convex program with $n$ variables & $(n-1)$ constraints

$\longrightarrow$ interior point method solves it in time $\tilde{O}(n^{\omega + \frac{1}{2}})$, where $\omega \le 2.373$ is the matrix multiplication exponent

The $O(n^2)$ exact algorithm used in many solvers : PAVA !

## Pool Adjacent Violators Algorithm (PAVA) :

Overall idea: split $\{1, 2, \cdots, n\}$ into several consecutive blocks $B_1, \cdots, B_m$, and inside block $B_j$, use the sample average

$$\hat{\theta}_i = \frac{1}{|B_j|} \sum_{k \in B_j} y_k \quad \text{for all} \quad i \in B_j .$$

(call this common value $v_j$ afterwards)

1. Initialization: set $m = n$, and $B_j = \{j\}$ for all $j = 1, 2, \cdots, m$

   (consequently $v_j = y_j$)

2. Iteration: if $\exists$ adjacent blocks with $v_j > v_{j+1}$, (adjacent violators)

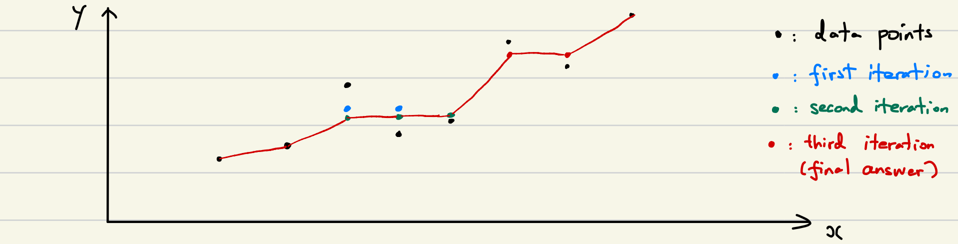
   pick an arbitrary pair, (leftmost, rightmost, random, $\cdots$)

   merge these two blocks. (and update $(v_j, v_{j+1})$ to a single $v$)

   Go back to step 2.

3. Stopping criterion: if $v_j \le v_{j+1}$ for all $j = 1, 2, \cdots, m-1$, then output the resulting $(\hat{\theta}_1, \cdots, \hat{\theta}_n)$.

## An illustration of PAVA:



## Correctness of PAVA.

Karush-Kuhn-Tucker (KKT) condition:

For convex $f$ and $g_1, \cdots, g_m$.

$x^*$ is the solution to $\begin{cases} \min & f(x) \\ \text{s.t.} & g_i(x) \le 0, \quad i=1,2,\cdots,m \end{cases}$

$\iff$ $\exists\ (\lambda_1^*, \cdots, \lambda_m^*)$ such that the following holds:

(Stationarity) $\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g(x_i^*) = 0$

(primal feasibility) $g_i(x^*) \le 0, \quad i=1,\cdots,m$

(dual feasibility) $\lambda_i^* \ge 0, \quad i=1,\cdots,m$

(complementary slackness) $\lambda_i^* g(x_i^*) = 0, \quad i=1,\cdots,m$

Application to PAVA: need to find $(\hat\theta_1,\cdots,\hat\theta_n, \lambda_1, \cdots, \lambda_{n-1})$ s.t.

1. $y_i - \hat\theta_i = \lambda_i - \lambda_{i-1}, \quad \forall i=1,\cdots,n$ $(\lambda_0 \triangleq 0, \ \lambda_n \triangleq 0)$
2. $\hat\theta_i \le \hat\theta_{i+1}, \quad \forall\ i=1,\cdots,n-1$
3. $\lambda_i \ge 0, \quad \forall\ i=1,\cdots,n-1$
4. $\lambda_i(\hat\theta_i - \hat\theta_{i+1}) = 0, \quad \forall i=1,\cdots,n-1.$

High-level idea: PAVA maintains 1. 3. 4. and tries to arrive at 2.

Formal Pf. Initialization: $\hat{\theta}_i = y_i$, $\lambda_i \equiv 0$ (1.3.4 hold)

Iteration: suppose we merge $B_j$ & $B_{j+1}$:

$$i_1-1 \quad \underbrace{i_1 \quad \cdots \quad i_2-1}_{B_j} \quad \underbrace{i_2 \quad \cdots \quad i_3-1}_{B_{j+1}} \quad i_3$$

Values of $(\hat{\theta}, \lambda)$ before merging:

$$\begin{cases} \hat{\theta}_i = v_j, \quad i_1 \le i < i_2; \qquad \hat{\theta}_i = v_{j+1}, \quad i_2 \le i < i_3. \\ \lambda_{i_1-1} = \lambda_{i_2-1} = \lambda_{i_3-1} = 0 \quad \text{(complementary slackness)} \\ \lambda_i - \lambda_{i-1} = y_i - v_j, \quad i_1 \le i < i_2 \\ \lambda_i - \lambda_{i-1} = y_i - v_{j+1}, \quad i_2 \le i < i_3 \qquad \text{(stationarity)} \\ \lambda_i \geq 0 \qquad \text{(dual feasibility)} \end{cases}$$

Updates of $(\hat{\theta}', \lambda')$ after merging:

$$\begin{cases} \hat{\theta}'_i = v \triangleq \frac{1}{i_3 - i_1} \sum_{k=i_1}^{i_3-1} y_k, \quad i_1 \le i < i_3 \\ \lambda'_i = \sum_{k=i_1}^{i} (y_k - v), \quad i_1 \le i < i_3 \end{cases}$$

Verification of properties 1.3.4:

1. stationarity: for $i_1 \le i < i_3$,

$$\lambda'_i - \lambda'_{i-1} = y_i - v = y_i - \hat{\theta}'_i$$

4. complementary slackness:

$$\lambda'_{i_1-1} = \lambda_{i_1-1} = 0$$

$$\lambda_i(\hat{\theta}_{i+1} - \hat{\theta}_i) = \lambda_i(v-v) = 0, \quad i_1 \le i < i_3 - 1$$

$$\lambda'_{i_3-1} = \sum_{k=i_1}^{i_3-1} (y_k - v) = 0 \quad \text{by defn. of } v$$

3. dual feasibility:

we only merge blocks when $v_j \geq v_{j+1} \Rightarrow v_j \geq v \geq v_{j+1}$

therefore:

$$i_1 \le i < i_2: \quad \lambda'_i = \sum_{k=i_1}^{i} (y_k - v) \geq \sum_{k=i_1}^{i} (y_k - v_1) = \lambda_i \geq 0;$$

$$i_2 \le i < i_3: \quad \lambda'_i = \sum_{k=i_1}^{i} (y_k - v) = -\sum_{k=i+1}^{i_3-1} (y_k - v)$$

$$\geq -\sum_{k=i+1}^{i_3-1} (y_k - v_2) = \lambda_i \geq 0.$$

PAVA stops in $\le n-1$ iterations $\Rightarrow$ 2 holds in the end, so PAVA is correct!

Statistical property (pf omitted)

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(f(x_i)-\hat{\theta}_i\right)^2\right]=O\left(n^{-2/3}\right).$$

Convex regression : $f(x)=\mathbb{E}[Y \mid X=x]$ is convex

Estimator in 1-D: $(\hat{\theta}_1, \cdots, \hat{\theta}_n)$ is the solution to

$$\min \sum_{i=1}^{n}(y_i-\hat{\theta}_i)^2$$

$$\text{s.t. } \frac{\hat{\theta}_i-\hat{\theta}_{i-1}}{x_i-x_{i-1}} \le \frac{\hat{\theta}_{i+1}-\hat{\theta}_i}{x_{i+1}-x_i}, \quad i=2,\cdots,n-1$$

(increasing derivative)

Statistical property:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_i-f(x_i)\right)^2\right]=O\left(n^{-4/5}\right).$$

High-dimension: interesting phenomena could happen

(see, e.g., "optimality of maximum likelihood for log-concave density estimation & bounded convex regression" by Gil Kur et al., 2019 )

# Course Recap.

1. Parametric models : find the right model & apply MLE

   1.1 make MLE computationally efficient :

   generalized linear model , exponential family

   ( estimation , confidence interval ( bootstrap ) , testing , etc. )

   1.2 adapt MLE to complicated scenarios :

   empirical likelihood , partial likelihood , EM algorithm

   1.3 MLE fails sometimes : empirical Bayes

2. Semiparametric models : deal with nuisance

   2.1 Full MLE : profile MLE ( ( Cox model ) )

   2.2 Take nuisance as given : orthogonality

   score , efficient score , estimating function / equation , Neyman orthogonality

   2.3 Example : causal inference

3. Nonparametric models : explicit bias - variance tradeoff

   3.1 locality : kernel ( Nadaraya - Watson , KDE , ··· )

   3.2 function approximation :

   time domain ( polynomials , splines , ··· )

   transformed domain ( Fourier , wavelets , ··· )

   linear vs. nonlinear ( WLS , Ridge regression , projection , thresholding )

   3.3 MLE : isotonic / convex regression