Adaptively Partitioning Max-Affine Estimators for Convex Regression

Gábor Balázs
gabalz.gandg.ai

2022

Convex Regression

Defined by a distribution μ,

- $(\boldsymbol{x}, y) \sim \mu$, convex set $\mathcal{X} \subseteq \mathbb{R}^d$, $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathbb{R}$,
- squared loss, regression function $f_*(\boldsymbol{x}) \doteq \mathbb{E}[y|\boldsymbol{x}]$ a.s..,
- covariate $\boldsymbol{x}$, and noise $(y - f_*(\boldsymbol{x}))|\boldsymbol{x}$ are subgaussian,
- i.i.d. training data $\{(\boldsymbol{x}_i, y_i) : i \in [n]\}$ of $n \in \mathbb{N}$ samples,
- f_* is convex and L-Lipschitz on $\mathcal{X}$.

For each μ let the box dimension be the smallest $d_{\mathcal{X}} \in [0, d]$ such that the covering number of the covariates $\mathcal{X}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ satisfies

$$N_{\|\cdot\|}(\mathcal{X}_n, \epsilon) = O(\epsilon^{-d_{\mathcal{X}}}) \text{ a.s.}$$

Adaptive generalization bounds are scaling with $d_{\mathcal{X}}$ instead of d.

Convex Nonparametric Least Squares (CNLS)

Given partition $\mathcal{P}_K = \{\mathcal{C}_k : k \in [K]\}$ over sample indices $[n]$, compute LSE over $\mathcal{F}_n(\mathcal{P}_K) = \{f \mid f \text{ is max-affine and induces } \mathcal{P}_K\}$ with QP:

$$\min_{\substack{\boldsymbol{a}_1,\dots,\boldsymbol{a}_K \in \mathbb{R}^d,\\ b_1,\dots,b_K \in \mathbb{R}}} \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \left| \boldsymbol{a}_k^\top \boldsymbol{x}_i + b_k - y_i \right|^2$$
$$\text{subject to } \boldsymbol{a}_k^\top \boldsymbol{x}_i + b_k \geq \boldsymbol{a}_l^\top \boldsymbol{x}_i + b_l$$
$$\text{and } \ \|\boldsymbol{a}_k\|_\infty \leq L \text{ for all } i \in \mathcal{C}_k \text{ and } k, l \in [K]$$

The max-affine estimate: $f_n(\boldsymbol{x}) \doteq \max_{k \in [K]} \boldsymbol{a}_k^\top \boldsymbol{x} + b_k$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

Choosing trivial partition $\mathcal{P}_n = \{\{i\} : i \in [n]\}$ with $K = n$ computes the CNLS estimator which has a generalization bound as $\left\| f_n^{\text{CNLS}} - f_* \right\|_\mu^2 = O(n^{-2/d})$, but it is expensive and overfits.

Is there another choice with theoretical guarantee?

Another choice: Farthest Point Clustering (FPC)

Adaptive FPC (AFPC) to compute $\mathcal{P}_K$:

- minimize the maximum cell diameter as objective $\Delta(\mathcal{P}_K)$,
- no worse result than twice the optimal $\Delta(\mathcal{P}_K) \leq 2\Delta(\mathcal{P}_K^*)$,
- incremental, stops when $\Delta(\mathcal{P}_K) \approx \sqrt{K/n}$ or $K \approx n^{d/(d+2)}$.

AFPC guarantees: $\Delta(\mathcal{P}_K) = O(K^{-1/d_{\mathcal{X}}})$ and $K = O(n^{d_{\mathcal{X}}/(2+d_{\mathcal{X}})})$.

Problems when $\mathcal{P}_K$ is not the trivial partition:

- hard to bound the distance between $\mathcal{F}_n(\mathcal{P}_K)$ and f_*,
- unclear whether any max-affine linearization of f_* induces $\mathcal{P}_K$.

Idea: consider a bigger piecewise-linear hypothesis class
and penalize if partition $\mathcal{P}_K$ is not induced.

Adaptively Partitioning CNLS (APCNLS)

APCNLS: compute $\mathcal{P}_K$ by AFPC and modify QP:

$$\min_{\substack{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_K \in \mathbb{R}^d, \\ b_1,\ldots,b_K \in \mathbb{R}, \\ V \in \mathbb{R}}} \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \left| \boldsymbol{a}_k^\top \boldsymbol{x}_i + b_k - y_i \right|^2 + \beta V^2$$

$$\text{subject to } \boldsymbol{a}_k^\top \boldsymbol{x}_i + b_k \geq \boldsymbol{a}_l^\top \boldsymbol{x}_i + b_l - V$$

$$\text{and } \|\boldsymbol{a}_k\|_\infty \leq L \text{ for all } i \in \mathcal{C}_k \text{ and } k, l \in [K]$$

The max-affine estimate: $f_n(\boldsymbol{x}) \doteq \max_{k \in [K]} \boldsymbol{a}_k^\top \boldsymbol{x} + b_k$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

Main result: adaptive convergence rate guarantee for $\beta = d \ln(n)$:

$$\left\| f_n^{\text{APCNLS}} - f_* \right\|_\mu^2 = O\left(n^{-2/(2+d_{\mathcal{X}})}\right).$$

Discussion

New convex regression algorithm: APCNLS

Theoretically:

- it has adaptive rate $O(n^{-2/(2+d_{\mathcal{X}})})$ scaling by $d_{\mathcal{X}}$ instead of d,
- in the worst-case guarantee (when $d_{\mathcal{X}} = d$) the bounds of APCNLS $O(n^{-2/(2+d)})$ and CNLS $O(n^{-2/d})$ are comparable,
- APCNLS has a better computational complexity than CNLS.

Empirically, APCNLS provides:

- competitive performance for medium noise and for linear manifolds,
- better performance for high noise and nonlinear manifolds,
- much better computational time in all cases.

Thanks for watching, hope to see you at the poster!
Or visit gabalz.gandg.ai to contact...