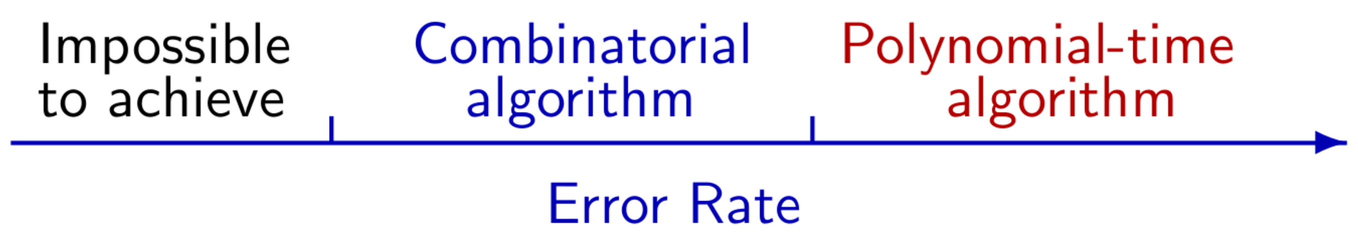
Lower Bounds on the Performance of Polynomial-time Algorithms for Sparse Linear Regression

Yuchen Zhang Martin J. Wainwright Michael I. Jordan

University of California, Berkeley

Statistical optimality vs. Computation tractability

Active research differentiating "optimal rate" and "computable optimal rate" in statistical estimation and machine learning:



- Sparse PCA detection (Berthet and Rigollet, 2013).
- Submatrix detection (Ma and Wu, 2013).
- Learning halfspaces (Daniely, Linial & Shalev-Shwartz, 2013)
- Sparse linear regression (our work)

Sparse Linear Regression: A Classical Problem

Observe a design matrix $X \in \mathbb{R}^{n\times d}$ and a response vector $y = X\theta^* + w$ such that

- $w \sim N(0, \sigma^2 I_{d\times d})$.
- $\theta^* \in \mathbb{R}^d$ is k-sparse, $k \ll d$.

Goal: find a k-sparse estimator $\widehat{\theta}$ of θ^* such that the prediction loss $\mathbb{E}[\|X(\widehat{\theta} - \theta^*)\|_2^2]$ is small.

$$\underbrace{\begin{bmatrix} y \end{bmatrix}}_{\mathbb{R}^n} = \underbrace{\begin{bmatrix} \quad X \quad \end{bmatrix}}_{\mathbb{R}^{n\times d}} \times \underbrace{\begin{bmatrix} \theta^* \end{bmatrix}}_{\mathbb{R}^d} + \underbrace{\begin{bmatrix} w \end{bmatrix}}_{\mathbb{R}^n}$$

Application: signal processing, financial data analysis, bioinformatics, imaging technology, etc.

Algorithms and Upper Bounds

Combinatorial Algorithm (NP-hard): ℓ_0-based estimator $\widehat{\theta}_{\ell_0} := \arg\min_{\theta\in\mathbb{B}_0(k)} \|X\theta - y\|_2$:

$$\sup_{\theta^*\in\mathbb{B}_0(k)} \mathbb{E}\Big[\frac{1}{n}\|X(\widehat{\theta}_{\ell_0} - \theta^*)\|_2^2\Big] \lesssim \frac{\sigma^2 k \log(d)}{n}$$

Poly-time Algorithm: run Lasso, then truncate it to be k-sparse.

$$\sup_{\theta^*\in\mathbb{B}_0(k)} \mathbb{E}\Big[\frac{1}{n}\|X(\widehat{\theta}_{\mathrm{TL}} - \theta^*)\|_2^2\Big] \lesssim \frac{\sigma^2 k \log(d)}{\gamma^2 n}.$$

where $\gamma < 1$ is the Restricted Eigenvalue of matrix X.

Key Observation: A $1/\gamma^2$ performance gap between the combinatorial estimator and the (known) poly-time estimator. Is there a better poly-time estimator?

Main Theoretical Result

Take-home Message

- There is a fundamental performance gap between poly-time algorithms and exponential-time algorithms for sparse linear regression.
- Gap is characterized by the restricted eigenvalue.

Theorem Assume $\mathbf{NP} \not\subset \mathbf{P/poly}$. For any $\gamma > 0$ and any (d, n, k) relation, there is an $X \in \mathbb{R}^{n\times d}$ with restricted eigenvalue γ, such that any k-sparse poly-time estimator $\widehat{\theta}_{\mathrm{poly}}$ satisfies:

$$\sup_{\theta^*\in\mathbb{B}_0(k)} \mathbb{E}\Big[\frac{1}{n}\|X(\widehat{\theta}_{\mathrm{poly}} - \theta^*)\|_2^2\Big] \gtrsim \frac{\sigma^2 k \log d}{\gamma^2 n}.$$

Proof Sketch

We prove the hardness of estimating sparse vector θ^* by a chain of reduction arguments.

Step 1: Construct a hard problem P1

- There is a matrix M, such that given Mu^* for some sparse vector u^*, recovering u^* is computationally hard.
- Prove the hardness of **P1** by reducing from the exact 3-set covering problem (NP-hard).

Step 2: Reduce P1 to an auxiliary problem P2′

- Given (M,Mu^*), construct matrix

$$X := \begin{bmatrix} M \\ \gamma G \end{bmatrix} \quad G \text{ is random Gaussian matrix}$$

$$y' := \begin{bmatrix} M \\ 0 \end{bmatrix} \frac{u^*}{\gamma} + w \quad w \text{ is random Gaussian vector}$$

  The goal of problem **P2′** is to recover $\theta^* := u^*/\gamma$.
- **P2′** is computationally hard since **P1** is hard.

Step 3: Reduce P2′ to the regression problem P2

- For regression problem **P2**, given

$$X := \begin{bmatrix} M \\ \gamma G \end{bmatrix} \quad G \text{ is random Gaussian matrix}$$

$$y := X\theta^* + w \quad w \text{ is random Gaussian vector}$$

  The goal is to recover θ^*.
- Observe that $\|y - y'\|_2 = \|Gu^*\|_2$. Choosing reasonably small u^*, then the hardness of **P2′** implies the hardness of **P2**.