Attribute-Efficient Learning of Monomials over Highly-Correlated Variables

Alexandr Andoni, Rishabh Dudeja, Daniel Hsu, **Kiran Vodrahalli**

Columbia University

Problem Statement

Model: Observe n features-response pairs $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \mathbb{R}$ drawn i.i.d. from the following model:

$$x_i \sim \mathcal{N}(0, \Phi), \quad y_i = f(x_i), \quad f(x) = \Pi_{j \in S} x_j^{\beta_j}.$$

Feature selection problem: Assume f depends on k out of the p features

Efficiency requirement: *attribute-efficient* algorithms require $n = \mathrm{poly}(\log(p), k)$ samples and $\mathrm{poly}(n, p, k)$ run-time.

Prior work: Attribute-efficient learning of polynomials

Boolean domain

- Learning sparse parities is a hard problem!
- Parity ⇔ monomial over $\{-1, +1\}^p$
- Many papers: [Helmbold et. al. '92, Blum '98, Klivans & Servedio '06, Kalai et. al. '09, Kocaoglu et. al '14, ...]
- Most results:
  - Assume product distribution (often uniform)
  - Runtime ~ $\text{dimension}^{c\,*\,\text{sparsity}}$, c < 1
  - NOT attribute-efficient

Takeaway: Boolean setting well-studied and difficult!

Real domain

- Sparse linear regression: attribute-efficient
  - RIP, REC, NSP assumptions on data [Candes '04, Donoho '04, Bickel '09, ...]
- General polynomials (NOT attribute-efficient)
- Sparse polynomials [Andoni et. al. '14]
  - product distribution
  - Gaussian or uniform data
  - Runtime & sample complexity: poly(dimension, 2^{degree}, sparsity)
    - Compare to naive $\text{dimension}^{\text{degree}}$

Takeaway: Most work linear, rest assumes product distribution.

This work: Non-product distributions for monomials

- One weird trick: Take the log of features and responses, run Lasso!
  - ⇒ **Attribute-efficient algorithm!**
- Learns k-sparse monomials
- Gaussian data
- Variance 1, covariance at most 1 - ε
  - **Arbitrarily high correlation between features!**
- Runtime: poly(samples, dimension, sparsity)
- Sample complexity: ~ $\dfrac{k^2 \log(2k)}{\epsilon} \cdot \log^2\left(\dfrac{2p}{\delta}\right)$

Binary Data Setting (reference for details)

- Boolean features (Valiant '84, Littlestone '88, Helmbold et. al. '92, Klivans et. al. '06, Valiant '15):
  - Conjunctions over $\{0, 1\}^p$ are learnable efficiently
  - Monomials over $\{+1, -1\}^p$ are parity functions and are PAC learnable
  - k-sparse parities: Sample efficient ($\text{poly}(\log(p), k)$), computationally inefficient ($O(p^k)$)
    - Runtime improvement over naive case: $O(p^{k/2})$
    - Improper learner: $O(p^{1-1/k})$ samples, $O(p^4)$ runtime
  - Attribute-inefficient noisy parity: $O(p^{0.8k}\,\text{poly}\left(1/(1-2\eta)\right)$ time for data under uniform dist.
    - η is noise parameter

- Average case analysis for learning parity (Kalai et. al. '09, Kocauglu et. al. '14):
  - Learn DNF/ functions defined on $\{+1, -1\}^p$
  - Can learn over adversarial + perturbed product distribution
  - Can learn in smoothed analysis settings (adversarial + perturbed function)