Active Regression via Linear-Sample Sparsification

Xue Chen **Eric Price**

UT Austin

Agnostic learning

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.
- Want to find $\widehat{f}$ so $y \approx \widehat{f}(x)$ on fresh samples.

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.
- Want to find $\widehat{f}$ so $y \approx \widehat{f}(x)$ on fresh samples.
- This work: adversarial error measured in ℓ_2.

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.
- Want to find $\widehat{f}$ so $y \approx \widehat{f}(x)$ on fresh samples.
- This work: adversarial error measured in ℓ_2. Guaranteed

$$\mathop{\mathbb{E}}_{x,y}[(y - f(x))^2] \leq \sigma^2$$

and want

$$\mathop{\mathbb{E}}_{x,y}[(y - \widehat{f}(x))^2] \leq C\sigma^2$$

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.
- Want to find $\widehat{f}$ so $y \approx \widehat{f}(x)$ on fresh samples.
- This work: adversarial error measured in ℓ_2. Guaranteed

$$\mathop{\mathbb{E}}_{x,y}[(y - f(x))^2] \leq \sigma^2$$

and want

$$\mathop{\mathbb{E}}_{x,y}[(y - \widehat{f}(x))^2] \leq C\sigma^2$$

or (equivalently, up to constants in C)

$$\mathop{\mathbb{E}}_{x}[(f(x) - \widehat{f}(x))^2] \leq C\sigma^2.$$

Agnostic learning

- See pairs (x, y) sampled from unknown distribution.
- Guaranteed $y \approx f(x)$ for some $f \in \mathbb{F}$.
- Want to find $\widehat{f}$ so $y \approx \widehat{f}(x)$ on fresh samples.
- This work: adversarial error measured in ℓ_2. Guaranteed

$$\mathop{\mathbb{E}}_{x,y}[(y - f(x))^2] \leq \sigma^2$$

and want

$$\mathop{\mathbb{E}}_{x,y}[(y - \widehat{f}(x))^2] \leq C\sigma^2$$

or (equivalently, up to constants in C)

$$\|f - \widehat{f}\|_{\mathcal{D}}^2 := \mathop{\mathbb{E}}_{x}[(f(x) - \widehat{f}(x))^2] \leq C\sigma^2.$$

where $\mathcal{D}$ is the marginal distribution on x.

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions

Agnostic learning of linear spaces

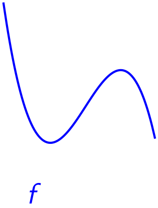
- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d - 1$ polynomials.

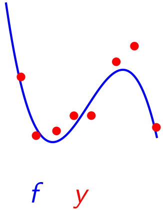
Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



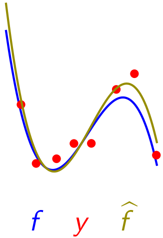
Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d - 1$ polynomials.



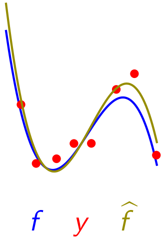
Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



Agnostic learning of linear spaces

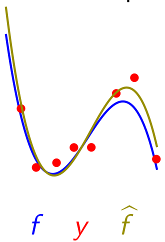
- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.

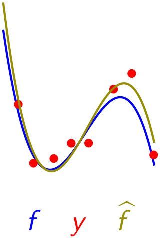


$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$

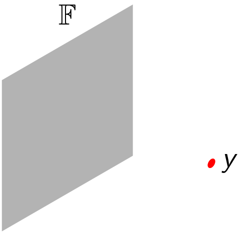
y

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.

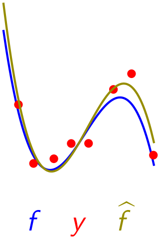


$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$

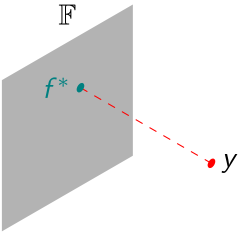


Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.

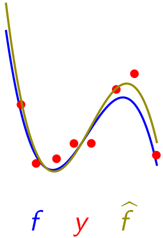


$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$

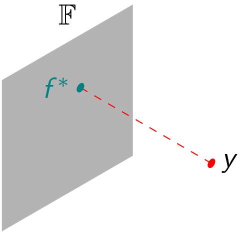


Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



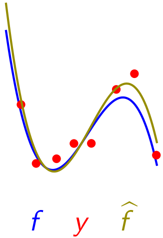
$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$



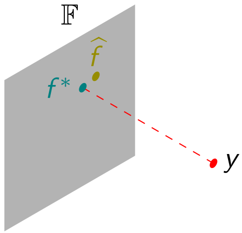
- Ideal: $f^* = \arg\min \|y - f^*\|_{\mathcal{D}}^2$.

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$

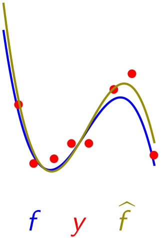


- Ideal: $f^* = \arg\min \|y - f^*\|_{\mathcal{D}}^2$.
- Settle for *empirical risk minimizer* (ERM)

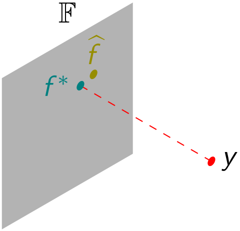
$$\widehat{f} = \arg\min \|y - \widehat{f}\|_S^2 := \frac{1}{m}\sum_{i=1}^{m}(y_i - \widehat{f}(x_i))^2.$$

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$



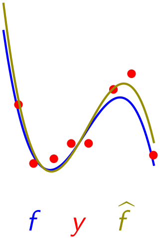
- Ideal: $f^* = \arg\min \|y - f^*\|_{\mathcal{D}}^2$.
- Settle for *empirical risk minimizer* (ERM)

$$\widehat{f} = \arg\min \|y - \widehat{f}\|_S^2 := \frac{1}{m}\sum_{i=1}^{m}(y_i - \widehat{f}(x_i))^2.$$

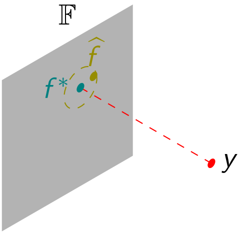
- Idea: with enough samples, empirical norm $\approx$ true norm under $\mathcal{D}$.

Agnostic learning of linear spaces

- Suppose $\mathbb{F}$ is a linear space of functions
  - $f(x) = \alpha^T \phi(x)$ for some $\phi : X \to \mathbb{R}^d$.
  - Example: univariate degree $d-1$ polynomials.



$$\langle f, g \rangle_{\mathcal{D}} := \mathbb{E}_x[f(x)g(x)]$$



- Ideal: $f^* = \arg\min \|y - f^*\|_{\mathcal{D}}^2$.
- Settle for *empirical risk minimizer* (ERM)

$$\widehat{f} = \arg\min \|y - \widehat{f}\|_S^2 := \frac{1}{m}\sum_{i=1}^{m}(y_i - \widehat{f}(x_i))^2.$$

- Idea: with enough samples, empirical norm $\approx$ true norm under $\mathcal{D}$.
  - Will get $\|\widehat{f} - f^*\|_{\mathcal{D}} \le \epsilon \|f^* - y\|_{\mathcal{D}}$.

Agnostic learning of linear spaces: results

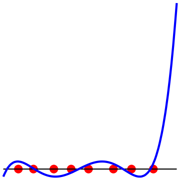
- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.

Agnostic learning of linear spaces: results

- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.

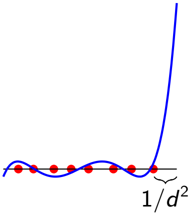
Agnostic learning of linear spaces: results

- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.

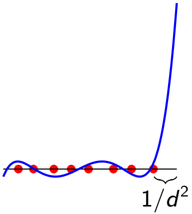


Agnostic learning of linear spaces: results

- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.



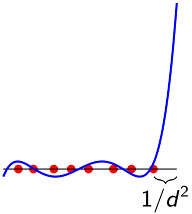
Agnostic learning of linear spaces: results



- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

Agnostic learning of linear spaces: results

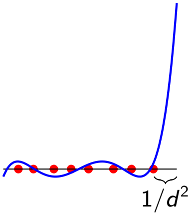


- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

- $O(K \log d + \frac{K}{\epsilon})$ samples suffice for agnostic learning [Cohen-Davenport-Leviatan '13, Hsu-Sabato '14]

Agnostic learning of linear spaces: results

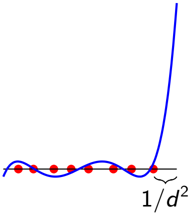


- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

- $O(K \log d + \frac{K}{\epsilon})$ samples suffice for agnostic learning [Cohen-Davenport-Leviatan '13, Hsu-Sabato '14]
  - Mean zero noise: $\|\widehat{f} - f^*\|_{\mathcal{D}}^2 \leq \epsilon \|f^* - y\|_{\mathcal{D}}^2$

Agnostic learning of linear spaces: results

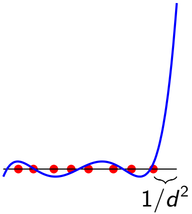


- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

- $O(K \log d + \frac{K}{\epsilon})$ samples suffice for agnostic learning [Cohen-Davenport-Leviatan '13, Hsu-Sabato '14]
  - Mean zero noise: $\|\widehat{f} - f^*\|_{\mathcal{D}}^2 \leq \epsilon \|f^* - y\|_{\mathcal{D}}^2$
  - Generic noise: $\|\widehat{f} - f\|_{\mathcal{D}}^2 \leq (1+\epsilon)\|f - y\|_{\mathcal{D}}^2$

Agnostic learning of linear spaces: results

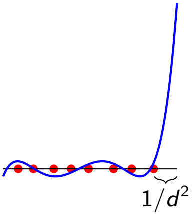


- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

- $O(K \log d + \frac{K}{\epsilon})$ samples suffice for agnostic learning [Cohen-Davenport-Leviatan '13, Hsu-Sabato '14]
  - Mean zero noise: $\|\widehat{f} - f^*\|_{\mathcal{D}}^2 \leq \epsilon \|f^* - y\|_{\mathcal{D}}^2$
  - Generic noise: $\|\widehat{f} - f\|_{\mathcal{D}}^2 \leq (1+\epsilon)\|f - y\|_{\mathcal{D}}^2$
- Also necessary (coupon collector)

Agnostic learning of linear spaces: results



- Degree 5 polynomial, $\sigma = 1$, $x \in [-1, 1]$.
- (Matrix) Chernoff bound depends on

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

- $O(K \log d + \frac{K}{\epsilon})$ samples suffice for agnostic learning [Cohen-Davenport-Leviatan '13, Hsu-Sabato '14]
  - Mean zero noise: $\|\widehat{f} - f^*\|_{\mathcal{D}}^2 \le \epsilon \|f^* - y\|_{\mathcal{D}}^2$
  - Generic noise: $\|\widehat{f} - f\|_{\mathcal{D}}^2 \le (1+\epsilon)\|f - y\|_{\mathcal{D}}^2$
- Also necessary (coupon collector)
- **How can we avoid the dependence on K?**

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathbb{E}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathbb{E}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y | X = x_i)$.

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathbb{E}_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y|X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y | X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).
- Active learning model:

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

  with

$$\kappa := \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  For linear spaces of functions, $\kappa = d$.
- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y|X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).
- Active learning model:
  - Receive $x_1, \ldots, x_m \sim \mathcal{D}$

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathbb{E}_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y|X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).
- Active learning model:
  - Receive $x_1, \ldots, x_m \sim \mathcal{D}$
  - Pick $S \subset [m]$ of size s

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathbb{E}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

For linear spaces of functions, $\kappa = d$.

- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y | X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).
- Active learning model:
  - Receive $x_1, \ldots, x_m \sim \mathcal{D}$
  - Pick $S \subset [m]$ of size s
  - See y_i for $i \in S$.

Our result: avoid K with more powerful access patterns

- With more powerful access models, can replace

$$K := \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

with

$$\kappa := \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

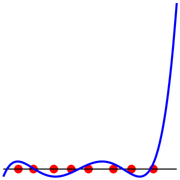
For linear spaces of functions, $\kappa = d$.
- Query model:
  - Can pick x_i of our choice, see $y_i \sim (Y|X = x_i)$.
  - Know $\mathcal{D}$ (which just defines $\|f - \widehat{f}\|_{\mathcal{D}}$).
- Active learning model:
  - Receive $x_1, \ldots, x_m \sim \mathcal{D}$
  - Pick $S \subset [m]$ of size s
  - See y_i for $i \in S$.
- Some results for non-linear spaces.

Query model: basic approach

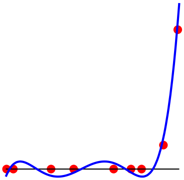
- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.

Query model: basic approach

- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.



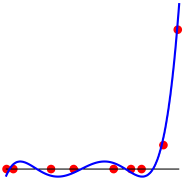
Query model: basic approach



- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.
- Improved by biasing samples towards high-variance points.

$$\mathcal{D}'(x) = \quad \mathcal{D}(x) \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} \quad f(x)^2$$

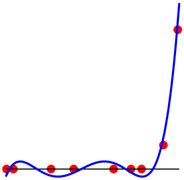
Query model: basic approach



- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.
- Improved by biasing samples towards high-variance points.

$$\mathcal{D}'(x) = \frac{1}{\kappa}\mathcal{D}(x) \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

Query model: basic approach



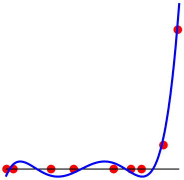
- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.
- Improved by biasing samples towards high-variance points.

$$\mathcal{D}'(x) = \frac{1}{\kappa}\mathcal{D}(x) \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Estimate norm via

$$\|f\|_{S,\mathcal{D}'}^2 := \frac{1}{m}\sum_{i=1}^{m} \frac{\mathcal{D}(x_i)}{\mathcal{D}'(x_i)} f(x_i)^2$$

Query model: basic approach



- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.
- Improved by biasing samples towards high-variance points.

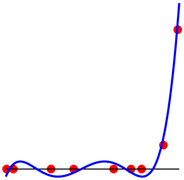
$$\mathcal{D}'(x) = \frac{1}{\kappa}\mathcal{D}(x) \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Estimate norm via

$$\|f\|_{S,\mathcal{D}'}^2 := \frac{1}{m}\sum_{i=1}^{m} \frac{\mathcal{D}(x_i)}{\mathcal{D}'(x_i)} f(x_i)^2$$

- Still equals $\|f\|_{\mathcal{D}}^2$ in expectation, but now max contribution is κ.

Query model: basic approach



- ERM needs empirical norm to $\|f\|_S$ to approximate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- This takes $O(K \log d)$ samples from $\mathcal{D}$.
- Improved by biasing samples towards high-variance points.

$$\mathcal{D}'(x) = \frac{1}{\kappa}\mathcal{D}(x) \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Estimate norm via

$$\|f\|^2_{S,\mathcal{D}'} := \frac{1}{m}\sum_{i=1}^{m} \frac{\mathcal{D}(x_i)}{\mathcal{D}'(x_i)} f(x_i)^2$$

- Still equals $\|f\|^2_{\mathcal{D}}$ in expectation, but now max contribution is κ.
  - This gives $O(\kappa \log d)$ sample complexity by Matrix Chernoff.

Bounding κ for linear function spaces

$$\kappa = \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Express $f \in \mathbb{F}$ via an orthonormal basis:

$$f(x) = \sum_j \alpha_j \phi_j(x).$$

Bounding κ for linear function spaces

$$\kappa = \mathbb{E}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Express $f \in \mathbb{F}$ via an orthonormal basis:

$$f(x) = \sum_{j} \alpha_j \phi_j(x).$$

- Then

$$\sup_{\|f\|_{\mathcal{D}}=1} f(x)^2 = \sup_{\|\alpha\|_2=1} \langle \alpha, \{\phi_j(x)\}_{j=1}^{d} \rangle^2$$

Bounding κ for linear function spaces

$$\kappa = \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Express $f \in \mathbb{F}$ via an orthonormal basis:

$$f(x) = \sum_{j} \alpha_j \phi_j(x).$$

- Then

$$\sup_{\|f\|_{\mathcal{D}}=1} f(x)^2 = \sup_{\|\alpha\|_2=1} \langle \alpha, \{\phi_j(x)\}_{j=1}^{d} \rangle^2 = \sum_{j=1}^{d} \phi_j(x)^2.$$

Bounding κ for linear function spaces

$$\kappa = \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2$$

- Express $f \in \mathbb{F}$ via an orthonormal basis:

$$f(x) = \sum_{j} \alpha_j \phi_j(x).$$

- Then

$$\sup_{\|f\|_{\mathcal{D}}=1} f(x)^2 = \sup_{\|\alpha\|_2=1} \langle \alpha, \{\phi_j(x)\}_{j=1}^{d} \rangle^2 = \sum_{j=1}^{d} \phi_j(x)^2.$$

- Hence

$$\kappa = \sum_{j=1}^{d} \mathop{\mathbb{E}}_{x} \phi_j(x)^2 = d.$$

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?
  - Not with independent sampling (coupon collector).

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?
  - Not with independent sampling (coupon collector).
  - Analogous to Batson-Spielman-Srivastava linear size sparsification.

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?
  - Not with independent sampling (coupon collector).
  - Analogous to Batson-Spielman-Srivastava linear size sparsification.
  - **Yes** – using Lee-Sun sparsification.

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?
  - Not with independent sampling (coupon collector).
  - Analogous to Batson-Spielman-Srivastava linear size sparsification.
  - **Yes** – using Lee-Sun sparsification.
- Mean zero noise: $\mathbb{E}[(\widehat{f}(x) - f(x))^2] \leq \epsilon \, \mathbb{E}[(y - f(x))^2]$.

Query model: so far

- Upsampling x proportional to $\sup_f f(x)^2$ gets $O(d \log d)$ sample complexity.
  - Essentially the same as leverage score sampling.
  - Also analogous to Spielman-Srivastava graph sparsification
- Can we bring this down to $O(d)$?
  - Not with independent sampling (coupon collector).
  - Analogous to Batson-Spielman-Srivastava linear size sparsification.
  - **Yes** – using Lee-Sun sparsification.
- Mean zero noise: $\mathbb{E}[(\widehat{f}(x) - f(x))^2] \leq \epsilon \, \mathbb{E}[(y - f(x))^2]$.
- Generic noise: $\mathbb{E}[(\widehat{f}(x) - f(x))^2] \leq (1+\epsilon) \, \mathbb{E}[(y - f(x))^2]$.

Active learning

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty$

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.
  - Hence $m = \Theta(K \log d + \frac{K}{\epsilon})$ optimal.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.
  - Hence $m = \Theta(K \log d + \frac{K}{\epsilon})$ optimal.
- Our result: both at the same time.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.
  - Hence $m = \Theta(K \log d + \frac{K}{\epsilon})$ optimal.
- Our result: both at the same time.
  - In this talk: mostly $s = O(d \log d)$ version.

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \ldots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.
  - Hence $m = \Theta(K \log d + \frac{K}{\epsilon})$ optimal.
- Our result: both at the same time.
  - In this talk: mostly $s = O(d \log d)$ version.
  - Prior work: $s = O((d \log d)^{5/4})$ [Sabato-Munos '14],

Active learning

- Query model supposes we know $\mathcal{D}$ and can query any point.
- Active learning:
  - Get $x_1, \dots, x_m \sim \mathcal{D}$.
  - Pick $S \subseteq [m]$ of size s.
  - Learn y_i for $i \in S$.
- Minimize s:
  - $m \to \infty \implies$ learn $\mathcal{D}$ and query any point $\implies$ query model.
  - Hence $s = \Theta(d)$ optimal.
- Minimize m:
  - Label every point $\implies$ agnostic learning.
  - Hence $m = \Theta(K \log d + \frac{K}{\epsilon})$ optimal.
- Our result: both at the same time.
  - In this talk: mostly $s = O(d \log d)$ version.
  - Prior work: $s = O((d \log d)^{5/4})$ [Sabato-Munos '14], $s = O(d \log d)$ via "volume sampling" [Derezinski-Warmuth-Hsu '18].

Active learning

- Warmup: suppose we know $\mathcal{D}$.

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] \propto \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

- Just needs $s = O(d \log d)$.

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

- Just needs $s = O(d \log d)$.
- Chance each sample gets labeled is

$$\mathop{\mathbb{E}}_{x}[\Pr[\text{Label } x_i]] = \frac{\kappa}{K}$$

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

- Just needs $s = O(d \log d)$.
- Chance each sample gets labeled is

$$\mathop{\mathbb{E}}_{x}[\Pr[\text{Label } x_i]] = \frac{\kappa}{K} = \frac{d}{K}.$$

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

- Just needs $s = O(d \log d)$.
- Chance each sample gets labeled is

$$\mathop{\mathbb{E}}_{x}[\Pr[\text{Label } x_i]] = \frac{\kappa}{K} = \frac{d}{K}.$$

- Gives $m = O(K \log d)$ unlabeled samples, $s = O(d \log d)$ labeled samples.

Active learning

- Warmup: suppose we know $\mathcal{D}$.
- Can simulate the query algorithm via rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

- Just needs $s = O(d \log d)$.
- Chance each sample gets labeled is

$$\mathbb{E}_x[\Pr[\text{Label } x_i]] = \frac{\kappa}{K} = \frac{d}{K}.$$

- Gives $m = O(K \log d)$ unlabeled samples, $s = O(d \log d)$ labeled samples.

Active learning

without knowing $\mathcal{D}$

Active learning

without knowing $\mathcal{D}$

- Want to perform rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

but don't know $\mathcal{D}$.

Active learning

without knowing $\mathcal{D}$

- Want to perform rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

  but don't know $\mathcal{D}$.
- Just need to estimate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.

Active learning

without knowing $\mathcal{D}$

- Want to perform rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

  but don't know $\mathcal{D}$.
- Just need to estimate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- Matrix Chernoff gets this with $m = O(K \log d)$ unlabeled samples.

Active learning

without knowing $\mathcal{D}$

- Want to perform rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

  but don't know $\mathcal{D}$.
- Just need to estimate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- Matrix Chernoff gets this with $m = O(K \log d)$ unlabeled samples.
- Gives $m = O(K \log d)$ unlabeled samples, $s = O(d \log d)$ labeled samples.

Active learning

without knowing $\mathcal{D}$

- Want to perform rejection sampling:

$$\Pr[\text{Label } x_i] = \frac{1}{K} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x_i)^2.$$

  but don't know $\mathcal{D}$.
- Just need to estimate $\|f\|_{\mathcal{D}}$ for all $f \in \mathbb{F}$.
- Matrix Chernoff gets this with $m = O(K \log d)$ unlabeled samples.
- Gives $m = O(K \log d)$ unlabeled samples, $s = O(d \log d)$ labeled samples.
- Can improve to $m = O(K \log d)$, $s = O(d)$.

Getting to $s = O(d)$

Based on Lee-Sun '15

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:
  - Noise variance bounded for every sample:

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

$$\mathop{\mathbb{E}}_{x \sim D} f(x)^2 \approx \sum_{i=1}^{s} \frac{1}{s} \frac{D(x_i)}{\mathcal{D}_i(x_i)} f(x_i)^2$$

  - Noise variance bounded for every sample:

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

$$\mathop{\mathbb{E}}_{x \sim D} f(x)^2 \approx \sum_{i=1}^{s} \frac{1}{s} \frac{D(x_i)}{\mathcal{D}_i(x_i)} f(x_i)^2$$

  - Noise variance bounded for every sample:

$$\frac{1}{s} K_{\mathcal{D}_i} \leq \epsilon$$

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

$$\mathop{\mathbb{E}}_{x \sim D} f(x)^2 \approx \sum_{i=1}^{s} \frac{1}{s} \frac{D(x_i)}{\mathcal{D}_i(x_i)} f(x_i)^2$$

  - Noise variance bounded for every sample:

$$\sup_{f,x} \frac{f(x)^2}{\mathbb{E}_{x' \sim \mathcal{D}_i} f(x')^2} =: \frac{1}{s} K_{\mathcal{D}_i} \leq \epsilon$$

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

$$\mathop{\mathbb{E}}_{x \sim D} f(x)^2 \approx \sum_{i=1}^{s} \alpha_i \frac{D(x_i)}{\mathcal{D}_i(x_i)} f(x_i)^2$$

  - Noise variance bounded for every sample:

$$\sup_{f,x} \frac{f(x)^2}{\mathbb{E}_{x' \sim \mathcal{D}_i} f(x')^2} =: \alpha_i K_{\mathcal{D}_i} \leq \epsilon$$

Getting to $s = O(d)$

Based on Lee-Sun '15

- $O(d \log d)$ comes from coupon collector.
- Change to non-independent sampling:
  - $x_i \sim \mathcal{D}_i$ where $\mathcal{D}_i$ depends on $x_1, \ldots, x_{i-1}$.
  - $D_1 = \mathcal{D}'$, D_2 avoids points near x_1, etc.
- Need two properties:
  - Norms preserved for all functions in class:

$$\mathop{\mathbb{E}}_{x \sim D} f(x)^2 \approx \sum_{i=1}^{s} \alpha_i \frac{D(x_i)}{\mathcal{D}_i(x_i)} f(x_i)^2$$

  - Noise variance bounded for every sample:

$$\sup_{f,x} \frac{f(x)^2}{\mathbb{E}_{x' \sim \mathcal{D}_i} f(x')^2} =: \alpha_i K_{\mathcal{D}_i} \leq \epsilon$$

- Both properties achievable with Lee-Sun sparsification.

Nonlinear spaces

Nonlinear spaces

- Consider functions with sparse Fourier representations:

$$f(x) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x}.$$

Nonlinear spaces

- Consider functions with sparse Fourier representations:

$$f(x) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x}.$$

- Can pick sample points $x \in [0, 1]$, want to minimize

$$\mathop{\mathbb{E}}_{x \in [0,1]} (\widehat{f}(x) - f(x))^2.$$

Nonlinear spaces

- Consider functions with sparse Fourier representations:

$$f(x) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x}.$$

- Can pick sample points $x \in [0, 1]$, want to minimize

$$\mathop{\mathbb{E}}_{x \in [0,1]} (\widehat{f}(x) - f(x))^2.$$

- For noise tolerance, need empirical norm $\approx$ actual norm.

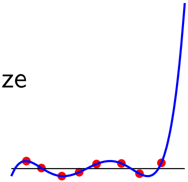
Nonlinear spaces

- Consider functions with sparse Fourier representations:

$$f(x) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x}.$$

- Can pick sample points $x \in [0, 1]$, want to minimize

$$\mathop{\mathbb{E}}_{x \in [0,1]} (\widehat{f}(x) - f(x))^2.$$



- For noise tolerance, need empirical norm $\approx$ actual norm.

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - Unknown exactly what this is for Fourier-sparse signals.

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - Unknown exactly what this is for Fourier-sparse signals.
  - $d^2 \leq K \lesssim d^4 \log^3 d$. [Chen-Kane-Price-Song '16]

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - Unknown exactly what this is for Fourier-sparse signals.
  - $d^2 \leq K \lesssim d^4 \log^3 d$. [Chen-Kane-Price-Song '16]

- Biasing the samples lets us reduce this to

$$\kappa = \mathbb{E}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - Unknown exactly what this is for Fourier-sparse signals.
  - $d^2 \le K \lesssim d^4 \log^3 d$. [Chen-Kane-Price-Song '16]

- Biasing the samples lets us reduce this to

$$\kappa = \mathop{\mathbb{E}}_x \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - $d \le \kappa \lesssim d \log^2 d$.

Estimating the norm in nonlinear spaces

- Uniform sampling depends on

$$K = \sup_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - Unknown exactly what this is for Fourier-sparse signals.
  - $d^2 \le K \lesssim d^4 \log^3 d$. [Chen-Kane-Price-Song '16]

- Biasing the samples lets us reduce this to

$$\kappa = \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} f(x)^2.$$

  - $d \le \kappa \lesssim d \log^2 d$.

- Analogous to distinction between Markov Brothers' inequality and Bernstein's inequality for polynomials.

Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.

Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i)$$

Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta})$$

Proof sketch: κ for Fourier-sparse functions

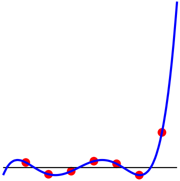
- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$

Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

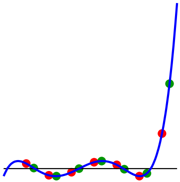
$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$



Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

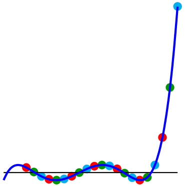
$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$



Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$



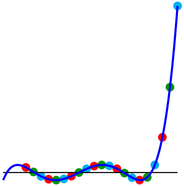
Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$

- In particular, for $i^* = \arg\max |\beta_i|$,

$$|f(x + i^* \Delta)| \leq \sum_{i \in [d] \setminus i^*} |f(x + i\Delta)|$$



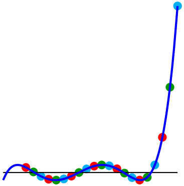
Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$

- In particular, for $i^* = \arg\max |\beta_i|$,

$$|f(x + i^* \Delta)| \leq \sum_{i \in [d] \setminus i^*} |f(x + i\Delta)|$$



- Hence

$$|f(x)| \leq \sum_{\substack{i=-d \\ i \neq 0}}^{d} |f(x + i\Delta)|$$

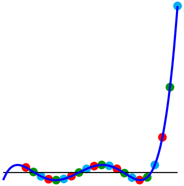
Proof sketch: κ for Fourier-sparse functions

- For any $\Delta > 0$, consider the degree-d polynomial $p(z) = \sum_{i=1}^{d} \beta_i z^i$ with roots at $e^{2\pi \mathbf{i} f_j \Delta}$ for all j.
- For any x,

$$\sum_{i=1}^{d} \beta_i f(x + \Delta i) = \sum_{j=1}^{d} v_j e^{2\pi \mathbf{i} f_j x} p(e^{2\pi \mathbf{i} f_j \Delta}) = 0.$$

- In particular, for $i^* = \arg\max |\beta_i|$,

$$|f(x + i^* \Delta)| \leq \sum_{i \in [d] \setminus i^*} |f(x + i\Delta)|$$



- Hence (with a little more care)

$$|f(x)|^2 \leq 3 \sum_{\substack{i=-2d \\ i \neq 0}}^{2d} |f(x + i\Delta)|^2$$

Proof sketch: κ for Fourier-sparse functions

Lemma

If f is d-Fourier-sparse, then for all x and Δ we have

$$|f(x)|^2 \leq 3 \sum_{\substack{i=-2d \\ i \neq 0}}^{2d} |f(x + i\Delta)|^2$$

Proof sketch: κ for Fourier-sparse functions

Lemma

If f is d-Fourier-sparse, then for all x and Δ we have

$$|f(x)|^2 \leq 3 \sum_{\substack{i=-2d \\ i \neq 0}}^{2d} |f(x + i\Delta)|^2$$

- Suppose $\mathcal{D}$ is uniform on $[-1, 1]$. Then for all $x \in [-1, 1]$,

$$|f(x)|^2 \lesssim \frac{d \log d}{1 - |x|} \mathop{\mathbb{E}}_{x'} f(x')^2.$$

by integrating Δ from 0 to $1 - |x|$.

Proof sketch: κ for Fourier-sparse functions

Lemma

If f is d-Fourier-sparse, then for all x and Δ we have

$$|f(x)|^2 \leq 3 \sum_{\substack{i=-2d \\ i \neq 0}}^{2d} |f(x + i\Delta)|^2$$

- Suppose $\mathcal{D}$ is uniform on $[-1, 1]$. Then for all $x \in [-1, 1]$,

$$|f(x)|^2 \lesssim \frac{d \log d}{1 - |x|} \mathop{\mathbb{E}}_{x'} f(x')^2.$$

  by integrating Δ from 0 to $1 - |x|$.
- Hence

$$\kappa = \mathop{\mathbb{E}}_{x} \sup_{\substack{f \in \mathbb{F} \\ \|f\|_{\mathcal{D}}=1}} |f(x)|^2 \lesssim d \log^2 d.$$

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.
- Linear spaces: $O(\kappa \log d)$ for *every* $f \in \mathbb{F}$ by matrix Chernoff.

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.
- Linear spaces: $O(\kappa \log d)$ for *every* $f \in \mathbb{F}$ by matrix Chernoff.
- Sparse Fourier: need to union bound over a net

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.
- Linear spaces: $O(\kappa \log d)$ for *every* $f \in \mathbb{F}$ by matrix Chernoff.
- Sparse Fourier: need to union bound over a net
  - Known net size is $2^{\tilde{O}(d^3)}$.

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.
- Linear spaces: $O(\kappa \log d)$ for *every* $f \in \mathbb{F}$ by matrix Chernoff.
- Sparse Fourier: need to union bound over a net
  - Known net size is $2^{\tilde{O}(d^3)}$.
  - Gives $\tilde{O}(d^3\kappa) = \tilde{O}(d^4)$ queries/labeled samples.

Query/active learning in nonlinear spaces

- Biased sampling means $\|f\|_S \approx \|f\|_{\mathcal{D}}$ in $O(\kappa)$ samples for any *single* $f \in \mathbb{F}$.
- Linear spaces: $O(\kappa \log d)$ for *every* $f \in \mathbb{F}$ by matrix Chernoff.
- Sparse Fourier: need to union bound over a net
  - Known net size is $2^{\tilde{O}(d^3)}$.
  - Gives $\tilde{O}(d^3 \kappa) = \tilde{O}(d^4)$ queries/labeled samples.
  - Gives $\tilde{O}(d^3 K) = \tilde{O}(d^7)$ unlabeled samples.

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?
  - Logistic regression?

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?
  - Logistic regression?
- Better theory for active learning ?

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?
  - Logistic regression?
- Better theory for active learning ?
  - Choose sample points sequentially.

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?
  - Logistic regression?
- Better theory for active learning ?
  - Choose sample points sequentially.
  - Dynamically changing functions.

Conclusions and open questions

- Active learning can be optimal in both criteria simultaneously
  - $O(K \log d + \frac{K}{\epsilon})$ unlabeled examples.
  - $O(d/\epsilon)$ labeled examples
- Gets some improvement for Fourier-sparse signals.
  - Tight results via chaining and/or better net?
- Can we go beyond ℓ_2 and linear spaces?
  - Logistic regression?
- Better theory for active learning ?
  - Choose sample points sequentially.
  - Dynamically changing functions.

Thank You