

Learning and Testing Convex Functions

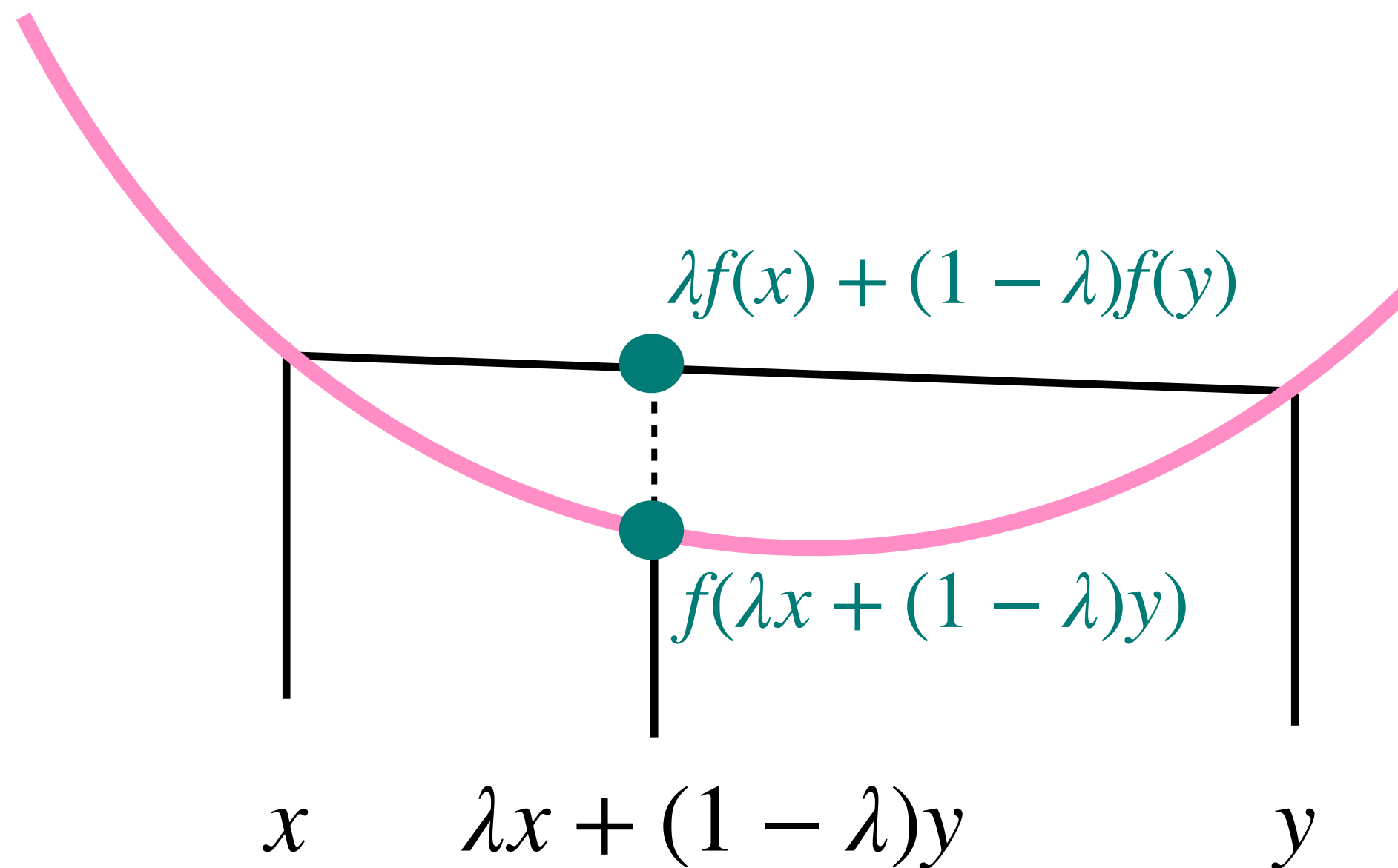
Renato Ferreira Pinto Jr., Cassandra Marcussen,

Elchanan Mossel, Shivam Nadimpalli

Convex functions

Convex function $f: \mathbb{R}^n \rightarrow \mathbb{R}$: For every $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$, we have
$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

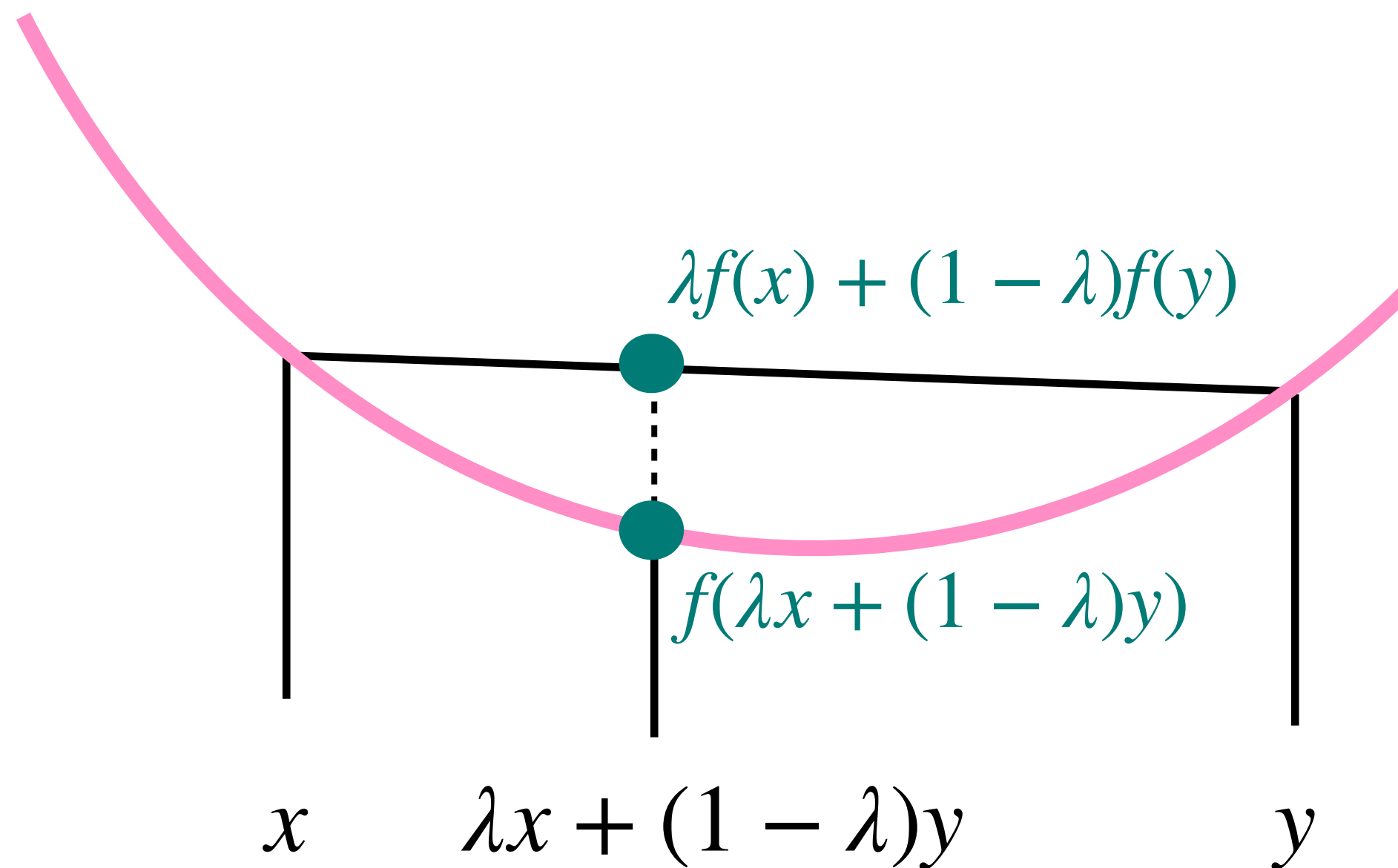
Central to math and CS, studied for centuries, many theoretical and applied uses



Convex functions

Convex function $f: \mathbb{R}^n \rightarrow \mathbb{R}$: For every $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$, we have
$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Central to math and CS, studied for centuries, many theoretical and applied uses



Fundamental questions:

How many samples to learn a convex function?

How many samples to test if $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is essentially convex?

Setting up the problem

Distance metric — standard Gaussian space:

- $\gamma = \mathcal{N}(0, I_n)$ is standard n -dimensional Gaussian distribution
- $L^2(\gamma) = \ell_2$ distance:

$$\|f - g\|_{L^2(\gamma)} = \left(\mathbb{E}_{z \sim N(0, I_n)} \left[(f(z) - g(z))^2 \right] \right)^{1/2} = \left(\int_{\mathbb{R}^n} (f(z) - g(z))^2 d\gamma(z) \right)^{1/2}$$

Setting up the problem

Distance metric — standard Gaussian space:

- $\gamma = \mathcal{N}(0, I_n)$ is standard n -dimensional Gaussian distribution
- $L^2(\gamma) = \ell_2$ distance:

$$\|f - g\|_{L^2(\gamma)} = \left(\mathbb{E}_{z \sim \mathcal{N}(0, I_n)} \left[(f(z) - g(z))^2 \right] \right)^{1/2} = \left(\int_{\mathbb{R}^n} (f(z) - g(z))^2 d\gamma(z) \right)^{1/2}$$

- **Distance to nearest convex L -Lipschitz function:**

$$d_{\text{conv}}^L(f) := \inf_{\substack{g : \mathbb{R}^n \rightarrow \mathbb{R} \\ g \text{ convex, } L\text{-Lipschitz}}} \|f - g\|_{L^2(\gamma)}$$

Setting up the problem

Distance metric — standard Gaussian space:

- $\gamma = \mathcal{N}(0, I_n)$ is standard n -dimensional Gaussian distribution
- $L^2(\gamma) = \ell_2$ distance:

$$\|f - g\|_{L^2(\gamma)} = \left(\mathbb{E}_{z \sim \mathcal{N}(0, I_n)} \left[(f(z) - g(z))^2 \right] \right)^{1/2} = \left(\int_{\mathbb{R}^n} (f(z) - g(z))^2 d\gamma(z) \right)^{1/2}$$

- **Distance to nearest convex L -Lipschitz function:**

$$d_{\text{conv}}^L(f) := \inf_{\substack{g : \mathbb{R}^n \rightarrow \mathbb{R} \\ g \text{ convex, } L\text{-Lipschitz}}} \|f - g\|_{L^2(\gamma)}$$

Access model — i.i.d. labeled samples $(x, f(x))$, $x \sim \mathcal{N}(0, I_n)$

Setting up the problem

Gaussian space &
 ℓ_2 distance

Real-valued functions in high
dimensions

→ “ ℓ_2 metric” natural

Prior work: discrete domains +
Hamming distance

not suitable for real-valued
“Gaussian space” is natural

Setting up the problem

Gaussian space &
 ℓ_2 distance

Real-valued functions in high dimensions

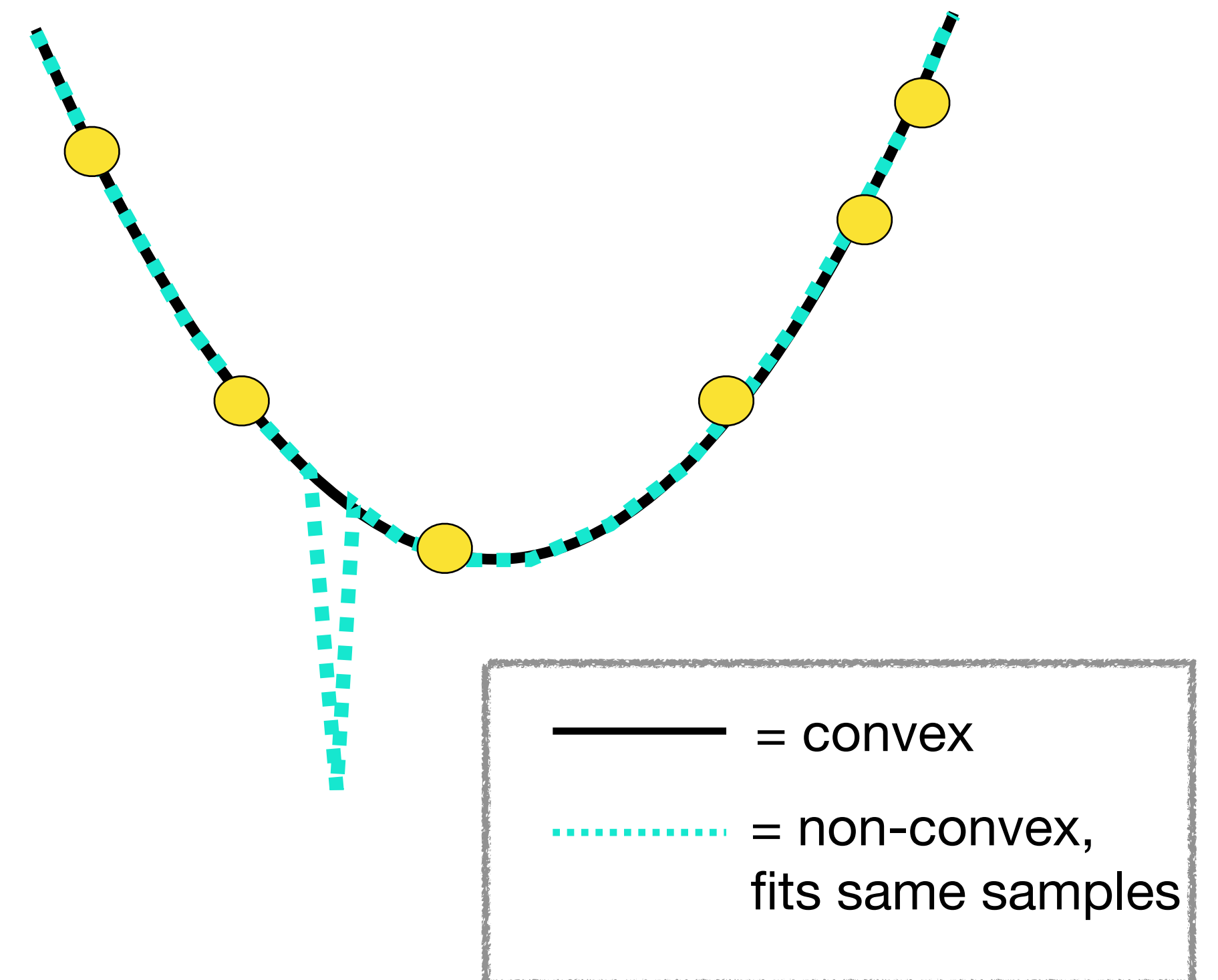
→ “ ℓ_2 metric” natural

Prior work: discrete domains +
Hamming distance

not suitable for real-valued
“Gaussian space” is natural

Lipschitz property

Without **regularity**, convexity cannot be inferred from finitely many samples, even in 1D.



Our results

Learning

Testing

Our results

Learning

Upper bound: Agnostic proper learner for Lipschitz convex functions, $n^{O(L^2/\varepsilon^2)}$ samples

Lower bounds: $n^{\text{poly}(L/\varepsilon)}$ CSQ queries, $\Omega(n)$ information-theoretic

Testing

Our results

Learning

Upper bound: Agnostic proper learner for Lipschitz convex functions, $n^{O(L^2/\varepsilon^2)}$ samples

Lower bounds: $n^{\text{poly}(L/\varepsilon)}$ CSQ queries, $\Omega(n)$ information-theoretic

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

Our results

Learning

Upper bound: **Agnostic proper** learner for Lipschitz convex functions, $n^{O(L^2/\varepsilon^2)}$ samples

Lower bounds: $n^{\text{poly}(L/\varepsilon)}$ **CSQ queries**, $\Omega(n)$ information-theoretic

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

Our results

Learning

Upper bound: Agnostic proper learner for Lipschitz convex functions, $n^{O(L^2/\varepsilon^2)}$ samples

Lower bounds: $n^{\text{poly}(L/\varepsilon)}$ CSQ queries, $\Omega(n)$ information-theoretic

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

Prior work

Us: Sample complexity, as a function of n , for real-valued Lipschitz functions under $L^2(\gamma)$

Prior work

Us: Sample complexity, as a function of n , for real-valued Lipschitz functions under $L^2(\gamma)$

Statistics: optimal rates for convex regression where dimension is fixed and number of samples is very large

Property testing: convexity testing under discrete or constrained settings

ℓ_p **testing:** real-valued setting but assume bounded range and full query access

Let's begin with **learning**
convex functions

Learning-theory framework

Agnostic

Proper

Improper

Learning-theory framework

Agnostic

No promise on f .

Compete with the best convex
 L -Lipschitz approximation:

$$||f - g|| \leq d_{\text{conv}}^L(f) + \varepsilon$$

Proper

Improper

Learning-theory framework

Agnostic

No promise on f .

Compete with the best convex L -Lipschitz approximation:

$$||f - g|| \leq d_{\text{conv}}^L(f) + \varepsilon$$

Proper

Return hypothesis g in the target class:
convex and L -Lipschitz

Improper

Do not need to return hypothesis g in the
target class: convex and L -Lipschitz

Agnostic proper learner

Algorithm:

0. Draw samples $x \sim \mathcal{N}(0, I_n)$, receive $f(x)$.
1. **Low-degree approximation:** Learn the low-degree Hermite expansion (an improper learner)
2. **Convex regression on low-degree approximation:** Project onto convex, L -Lipschitz functions (makes it proper)

Agnostic proper learner

Algorithm:

0. Draw samples $x \sim \mathcal{N}(0, I_n)$, receive $f(x)$.
1. **Low-degree approximation:** Learn the low-degree Hermite expansion (an improper learner)
Use samples from f
2. **Convex regression on low-degree approximation:** Project onto convex, L -Lipschitz functions (makes it proper)
NO samples from f ; use “fake samples” from low-degree approximation
Exponential cost in runtime only

Step 1: Low-degree method

1. **Low-degree approximation:** Learn low-degree Hermite expansion of f

Step 1: Low-degree method

1. **Low-degree approximation:** Learn low-degree Hermite expansion of f

Lemma: L -Lipschitz f concentrates its ℓ_2 -mass on Hermite degree $\leq d = \lfloor L^2/\varepsilon^2 \rfloor^*$

Less technical: L -Lipschitz f is close to degree $\leq d = \lfloor L^2/\varepsilon^2 \rfloor$.

*Not specific to convexity — a property of Lipschitz functions!

Step 1: Low-degree method

1. Low-degree approximation: Learn low-degree Hermite expansion of f

Lemma: L -Lipschitz f concentrates its ℓ_2 -mass on Hermite degree $\leq d = \lfloor L^2/\varepsilon^2 \rfloor^*$

Less technical: L -Lipschitz f is close to degree $\leq d = \lfloor L^2/\varepsilon^2 \rfloor$.

How to use this: Learn the $\approx n^d$ coefficients by regression
Now we have access to some explicit $\tilde{f}$

*Not specific to convexity — a property of Lipschitz functions!

Step 2: Convex regression

2. Convex regression on low-degree approximation: Project onto convex, L -Lipschitz functions (makes it proper)

Step 2: Convex regression

2. Convex regression on low-degree approximation: Project onto convex, L -Lipschitz functions (makes it proper)

Let $\tilde{f}$ be the low-degree polynomial.

Fit $\tilde{f}$ by a convex, L -Lipschitz function *via “samples” from $\tilde{f}$*

Quadratically-constrained convex program on fine grid over box $B_\infty(r)$.

Step 2: Convex regression

2. Convex regression on low-degree approximation: Project onto convex, L -Lipschitz functions (makes it proper)

Let $\tilde{f}$ be the low-degree polynomial.

Fit $\tilde{f}$ by a convex, L -Lipschitz function *via “samples” from $\tilde{f}$*

Quadratically-constrained convex program on fine grid over box $B_\infty(r)$.

Key point: Cost is runtime, not samples!

Can we get better bounds?

Can we get better bounds?

Answer: Only with a substantially different approach....

Low-degree approximation:

Fundamentally is “Correlational Statistical Queries” (CSQ):

Estimated Fourier coefficients $\hat{f}(\alpha) = \langle f, h_\alpha \rangle$

Correlational statistical queries: Return approximately $\langle f, g \rangle$ for different g .

Can we get better bounds?

Answer: Only with a substantially different approach....

Low-degree approximation:

Fundamentally is “Correlational Statistical Queries” (CSQ):

Estimated Fourier coefficients $\hat{f}(\alpha) = \langle f, h_\alpha \rangle$

Correlational statistical queries: Return approximately $\langle f, g \rangle$ for different g .

We prove CSQ LB of $n^{\text{poly}(L/\varepsilon)}$.

CSQ lower bound

Standard method:

Agnostic convex learner $\implies$ weak learner for $\mathcal{F}$ $\implies$ many CSQ queries

Weak learning: only need learner to reach correlation $\langle f, g \rangle \geq 1/\text{poly}(n)$

CSQ lower bound

Standard method:

Agnostic convex learner $\implies$ weak learner for $\mathcal{F}$ $\implies$ many CSQ queries

Weak learning: only need learner to reach correlation $\langle f, g \rangle \geq 1/\text{poly}(n)$

Approach:

1. Build family $\mathcal{F}$ of Lipschitz functions with low pairwise correlation
 - *Use the argument of [Diakonikolas, Kane, Zarifis 20] for CSQ lower bounds on ReLUs.*
 - *We make the functions Lipschitz via noise operator.*
2. Each $f \in \mathcal{F}$ correlated with a convex function: a projected ReLU $x \rightarrow \text{ReLU}(\langle x, u \rangle)$

CSQ lower bound

Standard method:

Agnostic convex learner $\implies$ weak learner for $\mathcal{F} \implies$ many CSQ queries

Weak learning: only need learner to reach correlation $\langle f, g \rangle \geq 1/\text{poly}(n)$

Approach:

1. Build family $\mathcal{F}$ of Lipschitz functions with low pairwise correlation
 - *Use the argument of [Diakonikolas, Kane, Zarifis 20] for CSQ lower bounds on ReLUs.*
 - *We make the functions Lipschitz via noise operator.*
2. Each $f \in \mathcal{F}$ correlated with a convex function: a projected ReLU $x \rightarrow \text{ReLU}(\langle x, u \rangle)$

Near orthogonality + correlation with convex function $\implies$ CSQ lower bound (standard route)

Let's move to **testing** convex
functions....

Our results

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

Our results

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

What does two-sided mean?

Can accept functions that are not L -Lipschitz convex, and reject functions that are L -Lipschitz convex.

What does tolerant mean?

Accept when CLOSE to L -Lipschitz convex whp,
Reject when FAR from L -Lipschitz convex whp.

Testing-via-learning framework

Goldreich, Goldwasser, Ron (1998)

Agnostic proper learner with q samples



Two-sided tolerant tester with q samples

Testing-via-learning framework

Goldreich, Goldwasser, Ron (1998)

Agnostic proper learner with q samples



Two-sided tolerant tester with q samples

The reduction:

1. Learn the best convex L -Lipschitz g for f
2. Estimate the true distance $\|f - g\|_{L^2(\gamma)}$ with $\text{poly}(L/\varepsilon)$ samples
3. Accept iff the estimate is small

Testing-via-learning framework

Goldreich, Goldwasser, Ron (1998)

Agnostic proper learner with q samples



Two-sided tolerant tester with q samples

$$q = n^{O(L^2/\epsilon^2)}$$

The reduction:

1. Learn the best convex L -Lipschitz g for f
2. Estimate the true distance $\|f - g\|_{L^2(\gamma)}$ with $\text{poly}(L/\epsilon)$ samples
3. Accept iff the estimate is small

Our results

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ samples

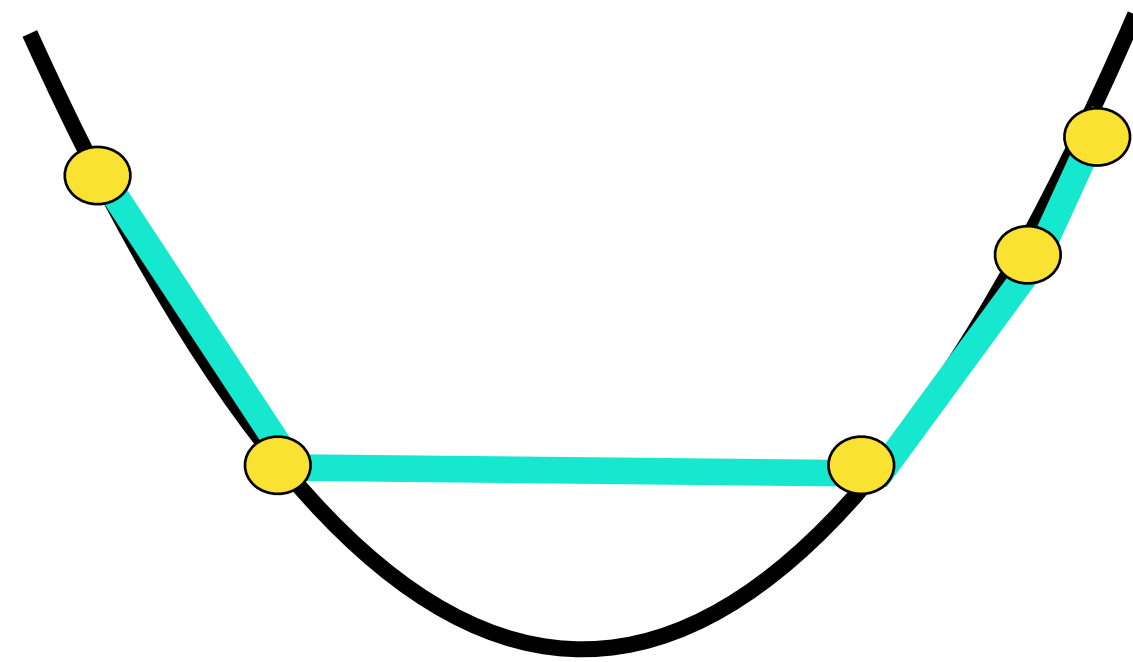
One-sided (never reject a convex function): $O(\sqrt{n}L/\varepsilon)^n$ samples

What does one-sided mean?

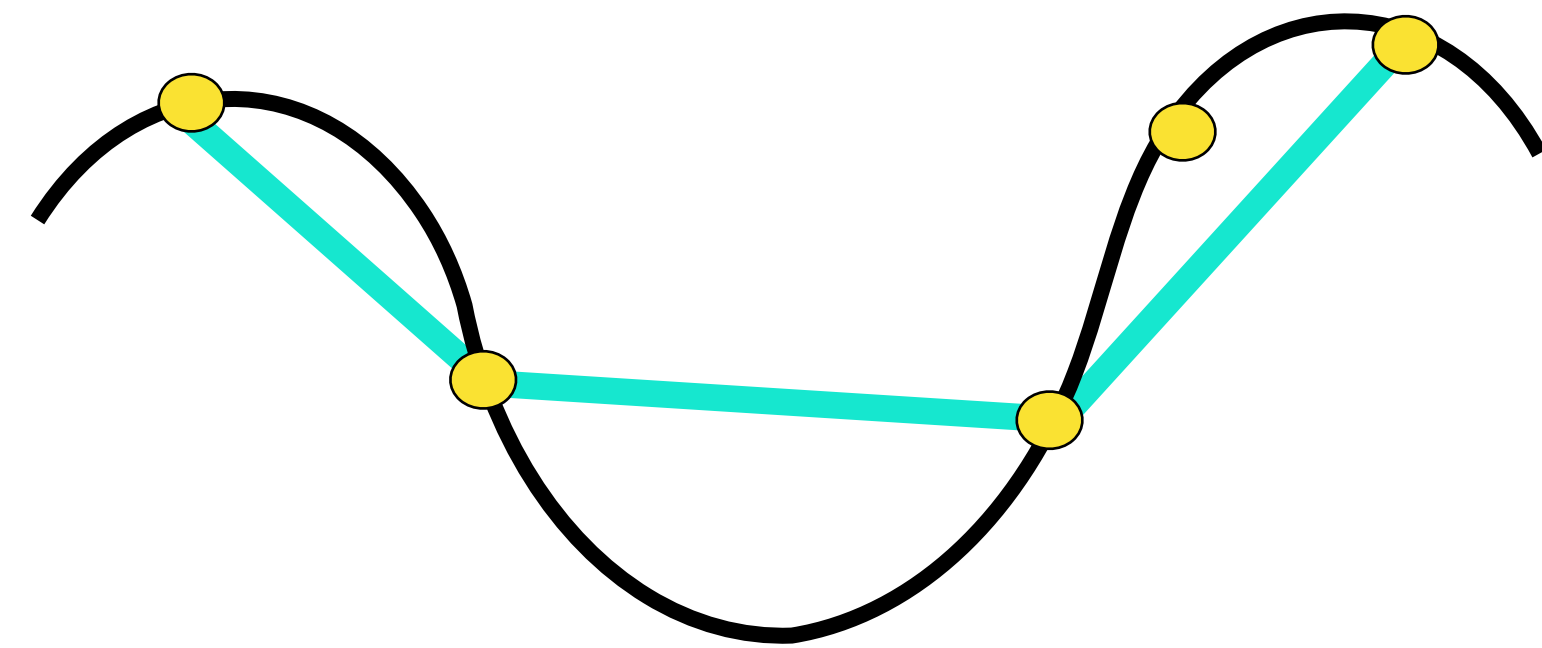
Can accept functions that are not L -Lipschitz convex.
Never reject functions that are L -Lipschitz convex.

One-sided testing

Empirical convex envelope $\bar{f}$: Largest convex, L -Lipschitz function lying below f on samples



Convex



Far from convex

Algorithm: Try to find a point where $\bar{f} \neq f$

f **convex** $\Rightarrow \bar{f}(y) = f(y)$ on all sampled points $y \Rightarrow$ accept w.p. 1

f **far from convex** $\Rightarrow \bar{f}(y) \neq f(y)$ on y whp $\Rightarrow$ reject whp

Summary

Learning

Upper bound: **Agnostic proper learner for Lipschitz convex functions**, $n^{O(L^2/\varepsilon^2)}$ **samples**

Lower bounds: $n^{\text{poly}(L/\varepsilon)}$ **CSQ queries**, $\Omega(n)$ **information-theoretic**

Testing

Two-sided tolerant tester: $n^{O(L^2/\varepsilon^2)}$ **samples**

One-sided: $O(\sqrt{nL}/\varepsilon)^n$ **samples**

Thank you!