Computational and Sample Tradeoffs via Convex Relaxation

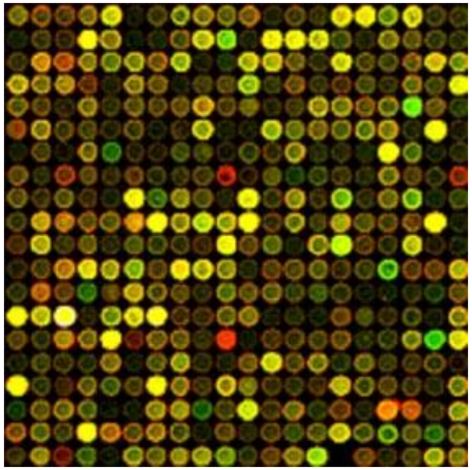
Venkat Chandrasekaran

Caltech

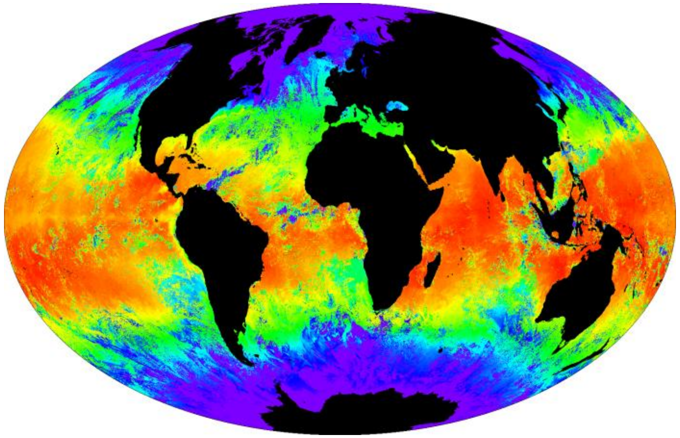
Joint work with **Michael Jordan**

High-dimensional Data

Gene microarray analysis

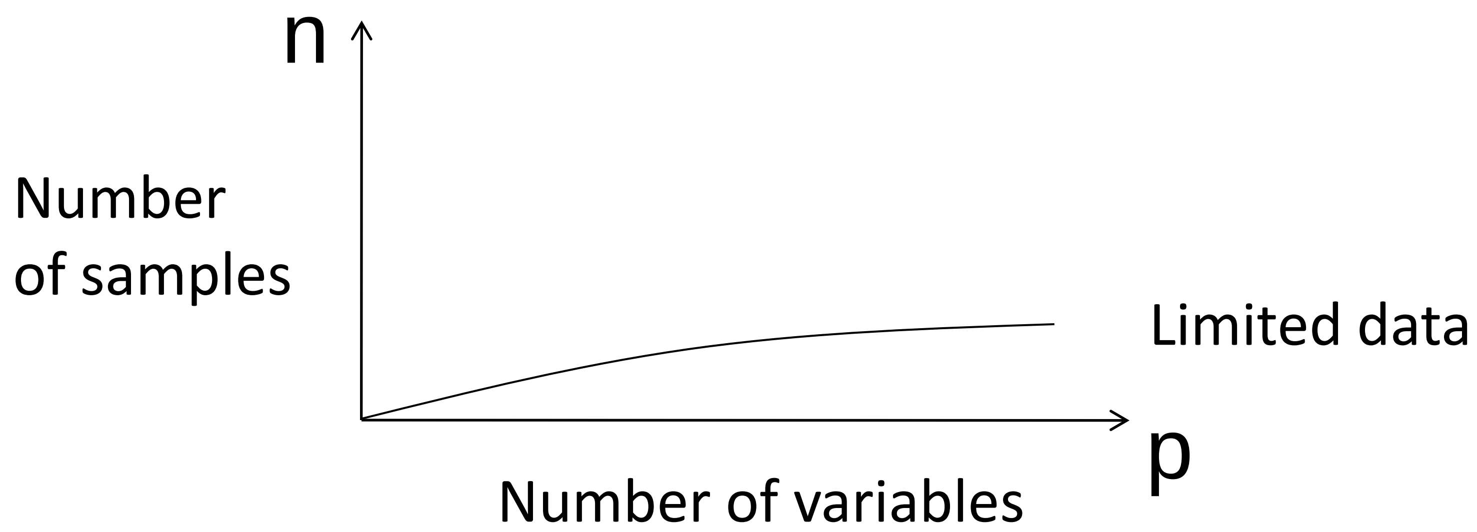


Global weather modeling



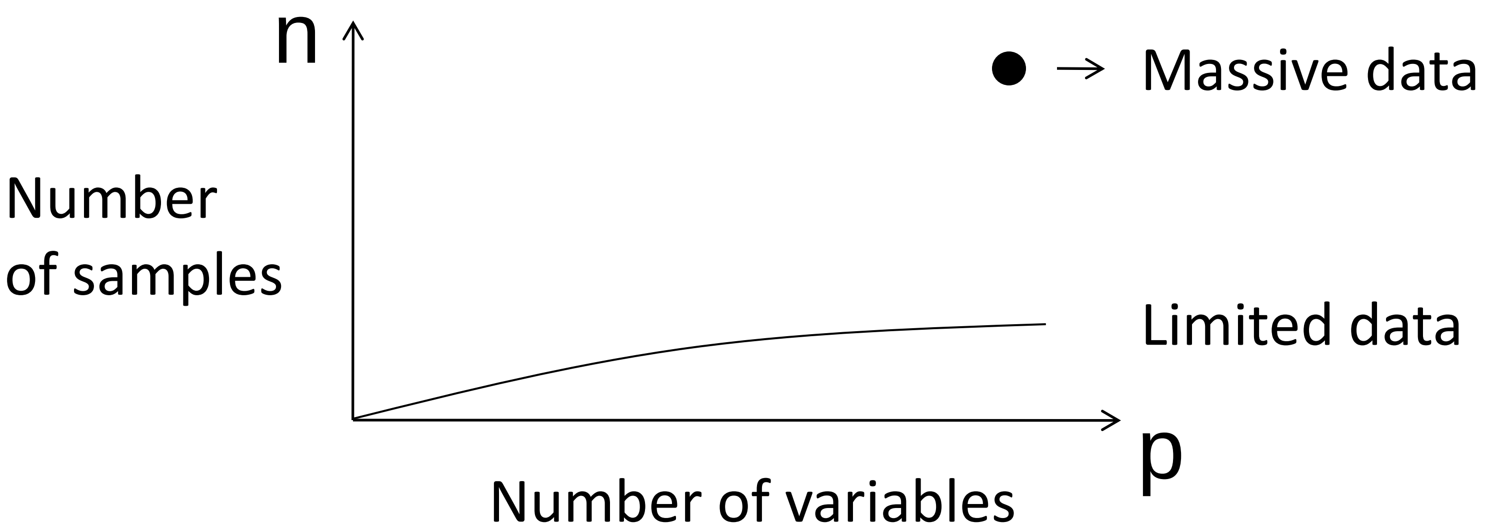
- Statistical inference with many variables
- Data in high-dimensional spaces
- E.g., images, Netflix, protein sequencing, …

High-dimensional Data



- A major success story in recent years
  - Role of ***structure***: sparsity, low-rank, …
  - Sophisticated ***computational*** techniques
- Fundamental limits on n for consistent inference

A New Challenge



- Large p + large n
  - Social data, financial modeling, …
- n much larger than fundamental limits
- Significant ***computational*** challenge

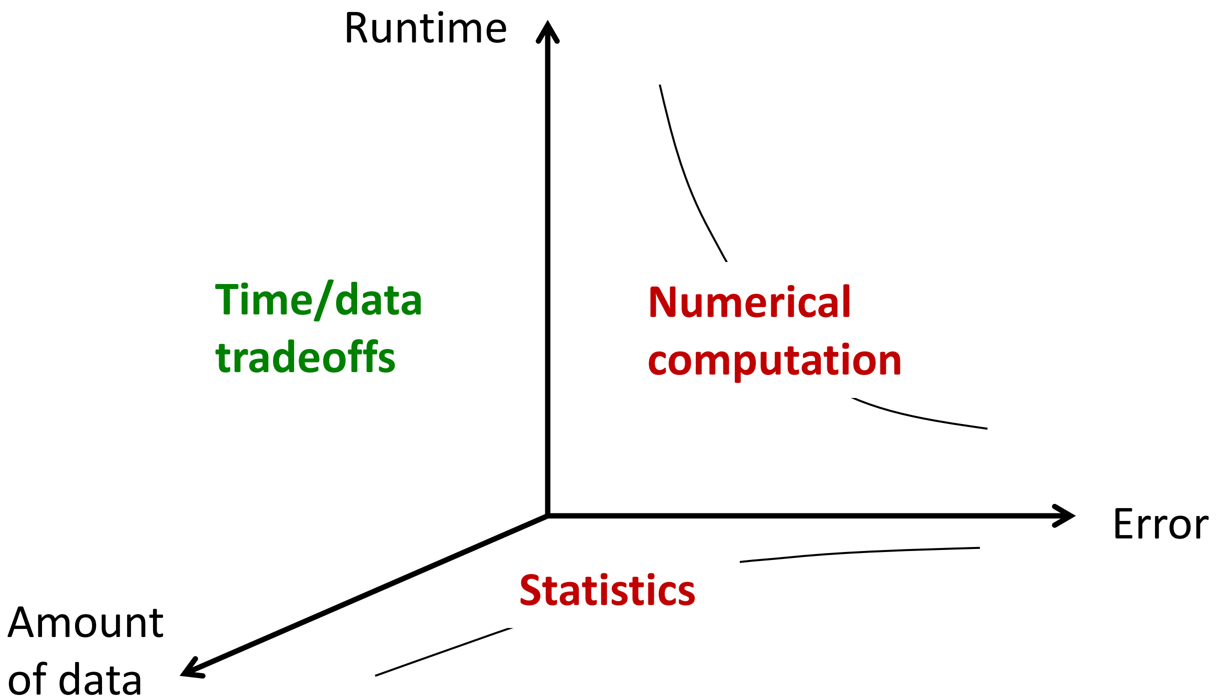
A Thought Experiment

- Consider a typical inference scenario
  - 1 hour for inference task with n = 5000, risk = 0.03
  - 20 days for same task with n = 500000, risk = 0.0003

- Suppose we don't care about such small improvements in risk
  - Statistical models are only approximations to reality

- More data useful for less computation?
  - Process larger datasets *more coarsely*?

Computer Science v.s. Statistics

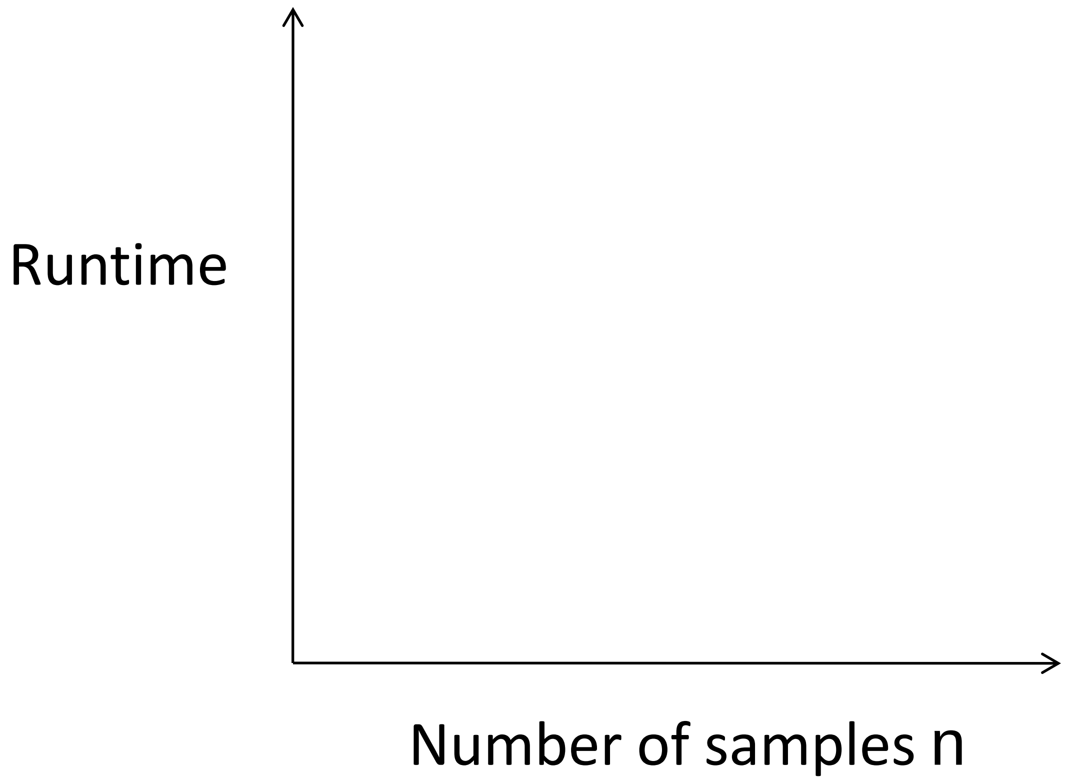


Outline

- What can we expect from time-data tradeoffs?
- A simple statistical inference problem
- Convex programming based estimation
- Tradeoffs via convex relaxation

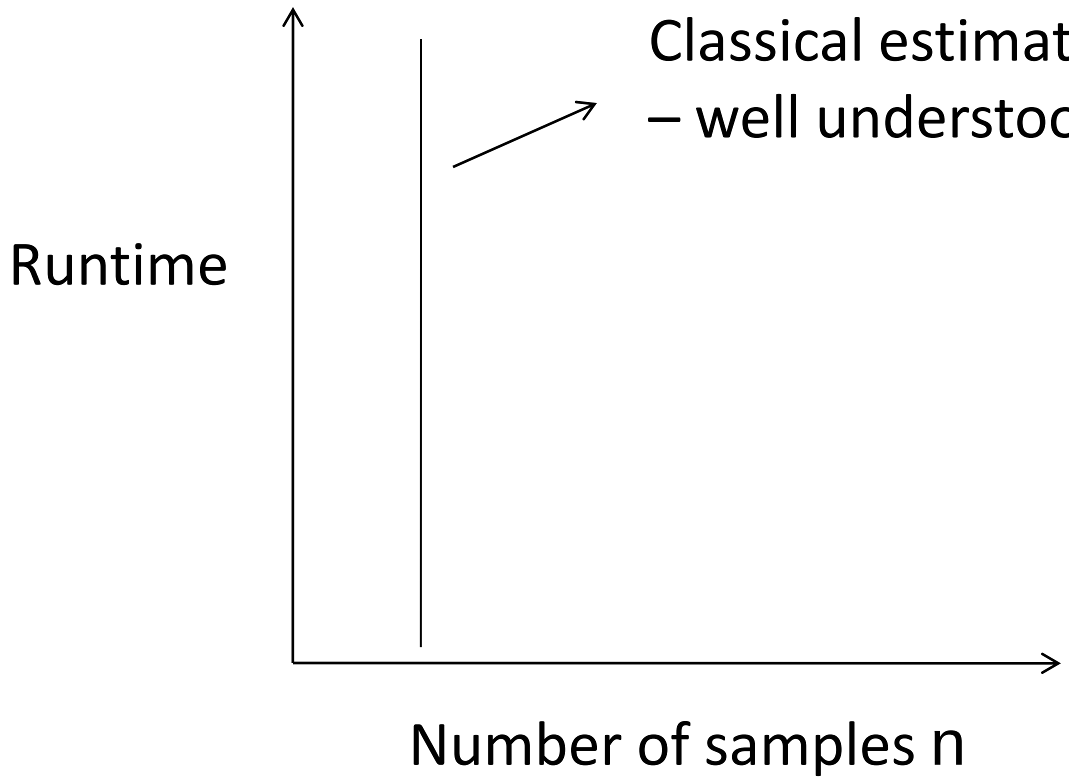
Time-Data Tradeoffs

- Consider an inference problem with ***fixed*** risk
- Inference procedures viewed as points in plot



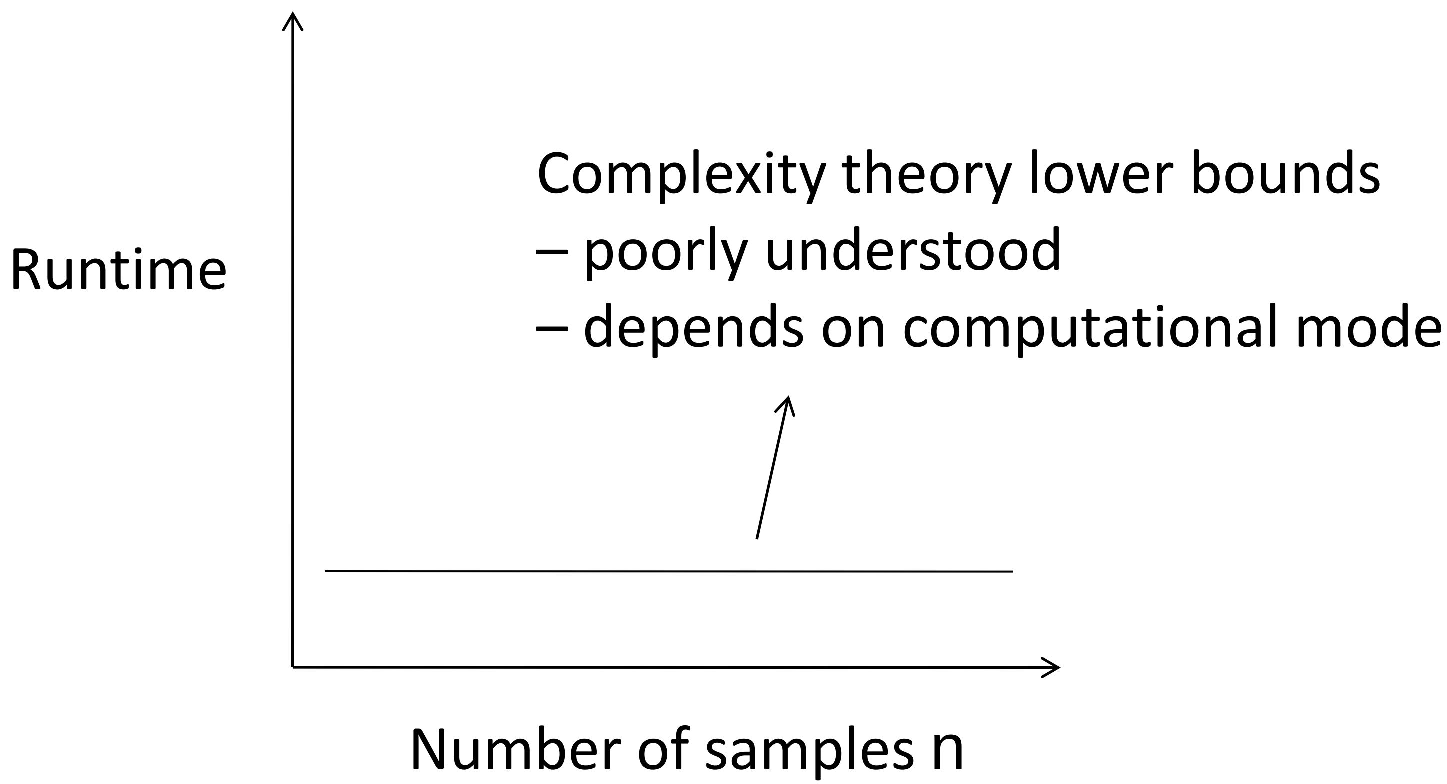
Time-Data Tradeoffs

- Consider an inference problem with ***fixed*** risk
- Vertical lines



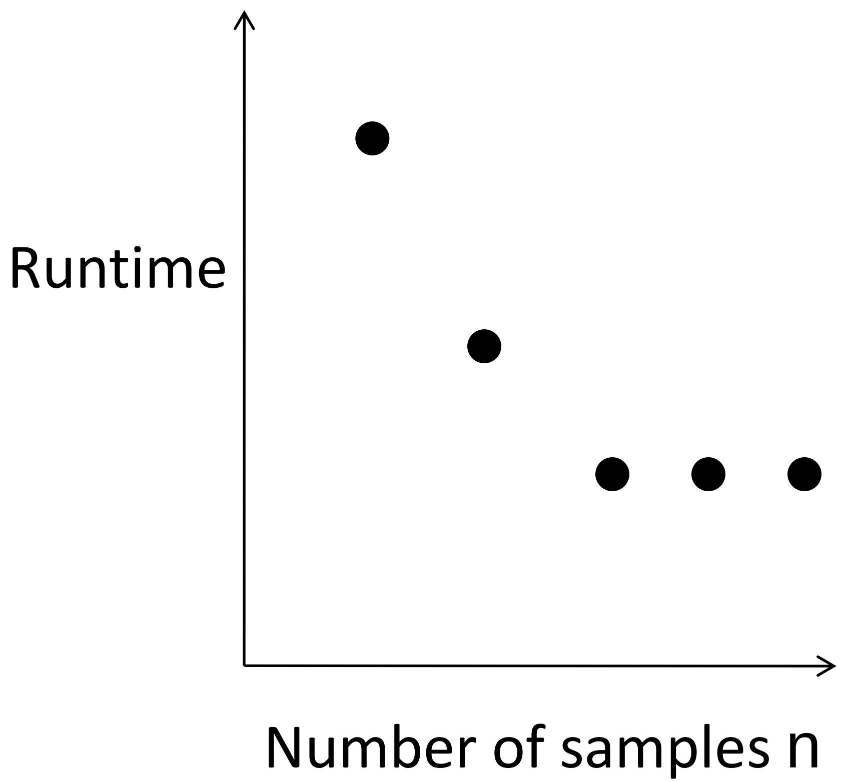
Time-Data Tradeoffs

- Consider an inference problem with ***fixed*** risk
- Horizontal lines



Time-Data Tradeoffs

- Consider an inference problem with ***fixed*** risk



- Need **"weaker"** algorithms for larger datasets
- At some stage, throw away data
- Tradeoff runtime ***upper bounds***
  - More data means smaller runtime upper bound

An Estimation Problem

- Signal $\mathbf{x}^* \in \mathcal{S} \subset \mathbb{R}^p$ from known (bounded) set
- Noise $\mathbf{z} \sim \mathcal{N}(0, I_{p\times p})$
- Observation model

$$\mathbf{y} = \mathbf{x}^* + \sigma \mathbf{z}$$

- Observe n i.i.d. samples $\{\mathbf{y}_i\}_{i=1}^n$

Convex Programming Estimator

- Sample mean $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i$ is sufficient statistic

- Natural estimator

$$\hat{\mathbf{x}}_n(\mathcal{S}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \ \tfrac{1}{2}\|\bar{\mathbf{y}} - \mathbf{x}\|_{\ell_2}^2 \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{S}$$

- Convex programming estimator

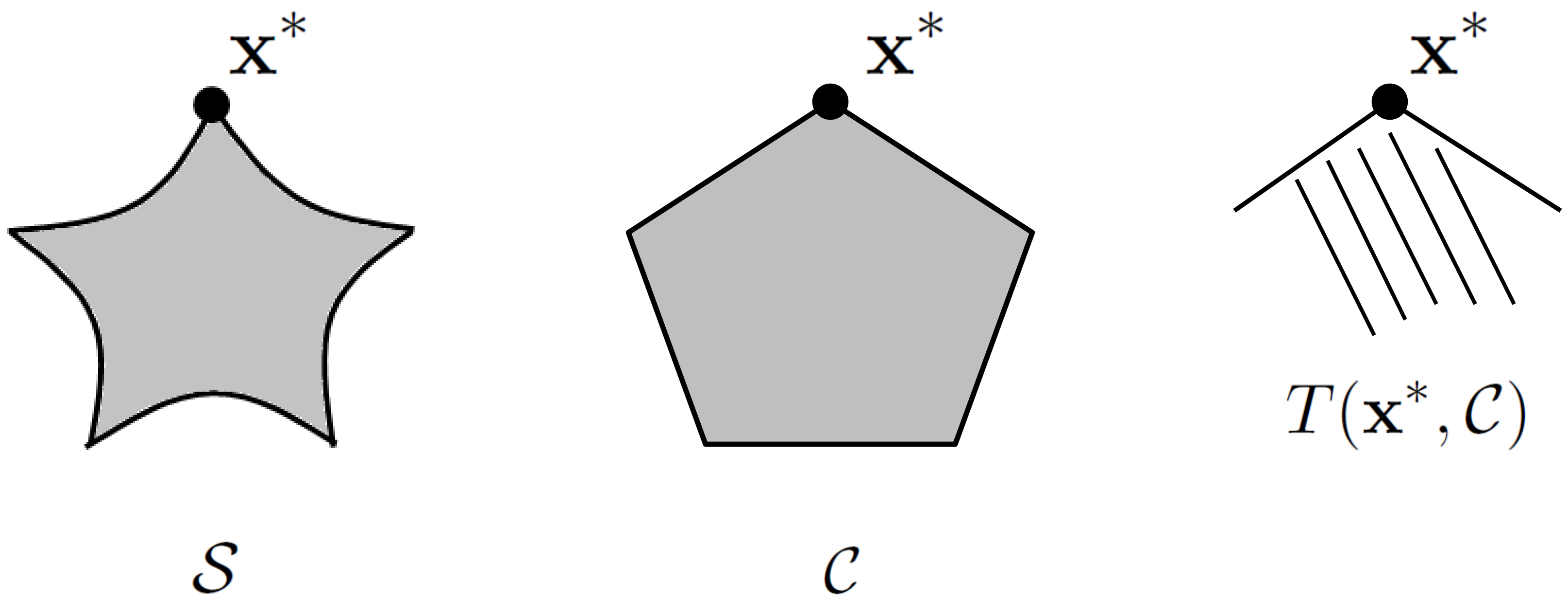
$$\hat{\mathbf{x}}_n(C) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \ \tfrac{1}{2}\|\bar{\mathbf{y}} - \mathbf{x}\|_{\ell_2}^2 \quad \text{s.t.} \quad \mathbf{x} \in C$$

  – C is a **convex** set such that $\mathcal{S} \subset C$

Statistical Performance of Estimator

- <u>Defn 1</u>: The ***cone of feasible directions*** into a convex set C is defined as

$$T(\mathbf{x}^*, C) = \text{cone}\{w - \mathbf{x}^* | w \in C\}$$



Statistical Performance of Estimator

- Defn 1: The ***cone of feasible directions*** into a convex set C is defined as

$$T(\mathbf{x}^*, C) = \text{cone}\{w - \mathbf{x}^* | w \in C\}$$

- Defn 2: The ***Gaussian (squared) complexity*** of a cone T is defined as

$$g(T) = \mathbb{E}\left[\sup_{\delta \in T, ||\delta||_{\ell_2} \leq 1} \langle \mathbf{z}, \delta \rangle^2\right]$$

Statistical Performance of Estimator

- <u>Prop</u>: The risk of the estimator $\hat{\mathbf{x}}_n(C)$ is

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_n(C) - \mathbf{x}^*\|_{\ell_2}^2\right] \leq \frac{\sigma^2}{n}\, g\Big(T(\mathbf{x}^*, C)\Big)$$

- <u>Proof</u>: Apply optimality conditions

- Intuition: Only consider error in feasible cone

Statistical Performance of Estimator

- E.g.: the risk of the estimator $\hat{\mathbf{x}}_n(\mathbb{R}^p)$ is

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_n(\mathbb{R}^p) - \mathbf{x}^*\|_{\ell_2}^2\right] \quad \leq \quad \frac{\sigma^2}{n} p$$

- Can generalize proposition in several ways
  - Obtain better bias-variance tradeoffs
  - Similar results for non-Gaussian noise

Weakening via Convex Relaxation

- Prop: The risk of the estimator $\hat{\mathbf{x}}_n(C)$ is

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_n(C) - \mathbf{x}^*\|_{\ell_2}^2\right] \leq \frac{\sigma^2}{n}\, g\Big(T(\mathbf{x}^*, C)\Big)$$

- Corr: To obtain risk of at most 1,

$$n \geq \sigma^2\, g\Big(T(\mathbf{x}^*, C)\Big)$$

Weakening via Convex Relaxation

- Corr: To obtain risk of at most 1,

$$n \geq \sigma^2 \underbrace{g\Big(T(\mathbf{x}^*, C)\Big)}_{\text{Monotonic in C}}$$

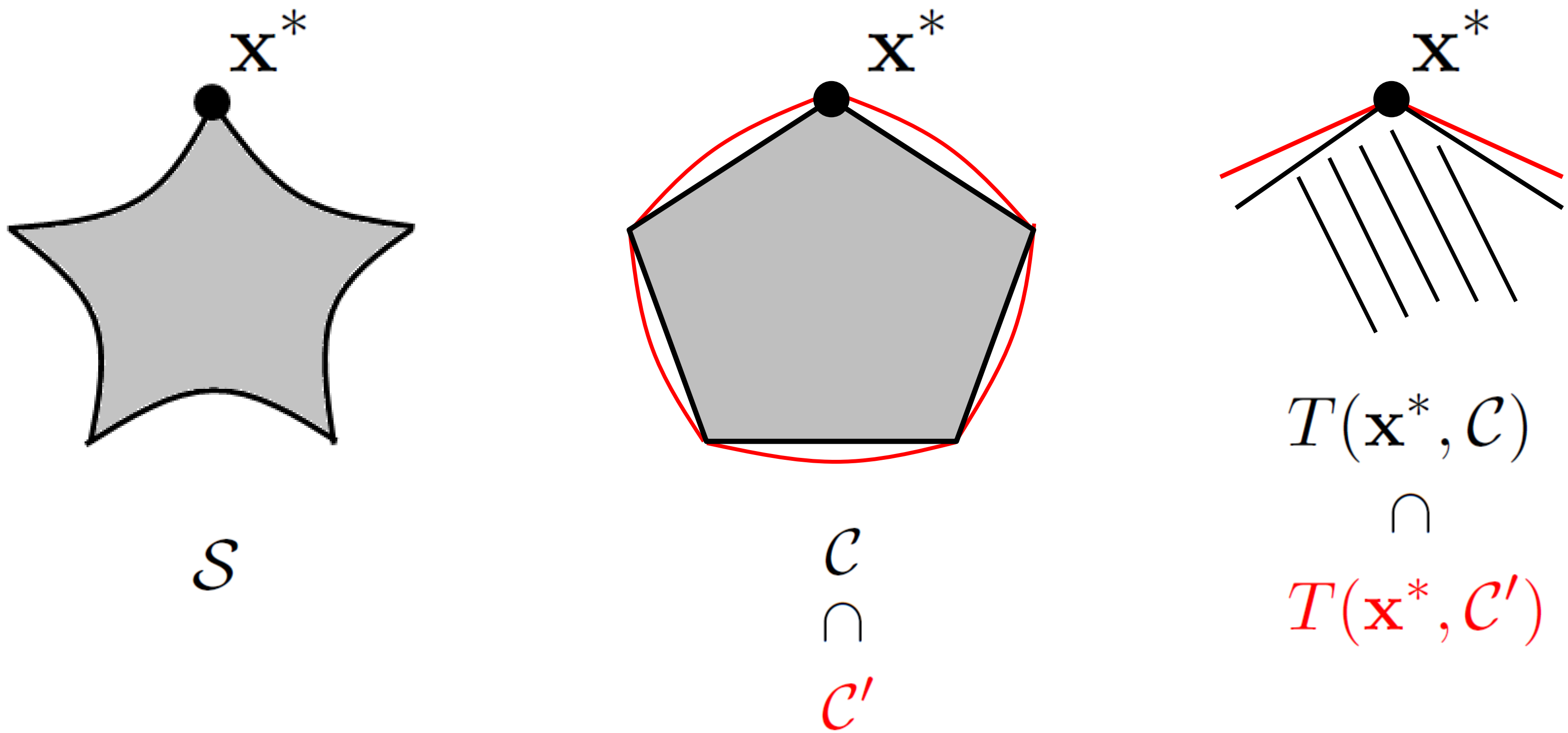
- Key point:

If we have access to larger n, can use larger C

Weakening via Convex Relaxation

If we have access to larger n, can use larger C

→ Obtain "weaker" estimation algorithm



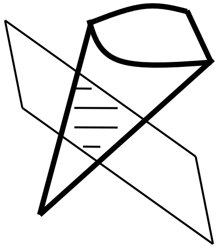
Hierarchy of Convex Relaxations

- If $\mathcal{S}$ "algebraic", then one can obtain family of outer convex approximations

$$\operatorname{conv}(\mathcal{S}) \subseteq \cdots \subset C_3 \subset C_2 \subset C_1$$

  – Polyhedral, semidefinite, hyperbolic relaxations (Sherali-Adams, Parrilo, Lasserre, Garding, Renegar)

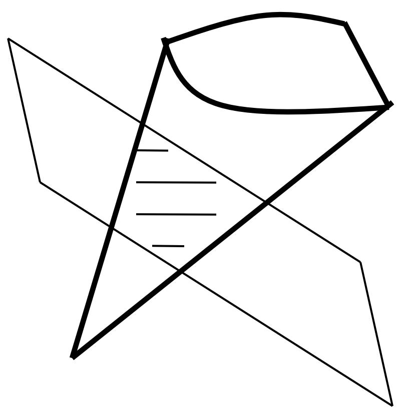
- Sets $\{C_i\}$ ordered by *computational complexity*
  – Central role played by **lift-and-project**



Hierarchy of Convex Relaxations

$$\mathrm{conv}(\mathcal{S}) \subseteq \cdots \subset C_3 \subset C_2 \subset C_1$$

- Concept of **lift-and-project**
  - Sets expressed as projection of affine slice of cone
  - Orthant (linear programming)
  - PSD cone (semidefinite programming)



- Larger dimensional lifts
  - Better approximation
  - Greater computational cost

Contrast to Previous Work

- Binary classifier learning
  - Decatur et al. [1998], Servedio [2000], Shalev-Shwarz & Srebro [2008], Perkins & Hallett [2010], Shalev-Shwarz et al. [2012]
  - Lots of extra data required for simpler algorithms
  - Our examples: modest extra data for simpler algorithms
- Sparse PCA, clustering, network inference
  - Amini & Wainwright [2009], Kolar et al. [2011]

- **Our work**: Emphasis on ***algorithm weakening***
  - Convex relaxation: principled, general way to do this

Before we get to examples …

- How do we calculate runtime?

- Total runtime = np + # ops for projection

Computing sample mean

With more data, this ***increases***

Subsequent processing

With more data, this ***decreases***

Before we get to examples …

- Estimating Gaussian complexity
  - General techniques: covering numbers, Dudley's integral formula (1967), …
  - Usually not sharp

- Thm: If a convex cone T has a dual with relative volume μ , then

$$g(T) \leq 20 \log(\frac{1}{4\mu})$$

- Proof: Appeal to Gaussian isoperimetry

Example 1

- $\mathcal{S}$ consists of cut matrices

$$\mathcal{S} = \{\mathbf{a}\mathbf{a}' \mid \mathbf{a} \text{ consists of } \pm 1's\}$$

- E.g., collaborative filtering, clustering

C	Runtime	n
conv($\mathcal{S}$) (cut polytope)	super-poly(p)	$c_1\sqrt{p}$
elliptope	$p^{2.25}$	$c_2\sqrt{p}$
nuclear norm ball	$p^{1.5}$	$c_3\sqrt{p}$

$(c_1 < c_2 < c_3)$

Example 2

- Banding estimators for covariance matrices
  - Bickel-Levina (2007), many others
  - Assume known variable ordering
- Stylized problem: let M be known tridiagonal matrix
- Signal set $\mathcal{S} = \{\Pi M \Pi' \mid \Pi \text{ a permutation}\}$

C	Runtime	n
$\mathrm{conv}(\mathcal{S})$	super-poly(p)	$c_1 \sqrt{p} \log(p)$
scaled ℓ_1 norm ball	$p^{1.5} \log(p)$	$c_2 \sqrt{p} \log(p)$

$(c_1 < c_2)$

Example 3

- Signal set $\mathcal{S}$ consists of all perfect matchings in complete graph
- E.g., network inference

C	Runtime	n
$\mathrm{conv}(\mathcal{S})$	p^5	$c_1\sqrt{p}\log(p)$
hypersimplex	$p^{1.5}\log(p)$	$c_2\sqrt{p}\log(p)$

$(c_1 < c_2)$

Example 4

- $\mathcal{S}$ consists of all adjacency matrices of graphs with only a clique on square-root of the nodes
- E.g., sparse PCA, gene expression patterns
- Kolar et al. (2010)

C	Runtime	n
$\mathrm{conv}(\mathcal{S})$	super-poly(p)	$\sim p^{0.25}\log(p)$
nuclear norm ball	$p^{1.5}$	$\sim \sqrt{p}$

Example 4

C	Runtime	n
$\mathrm{conv}(\mathcal{S})$	super-poly(p)	$\sim p^{0.25}\log(p)$
nuclear norm ball	$p^{1.5}$	$\sim \sqrt{p}$

- What if we use an even weaker relaxation?
  - E.g., (properly scaled) Euclidean ball

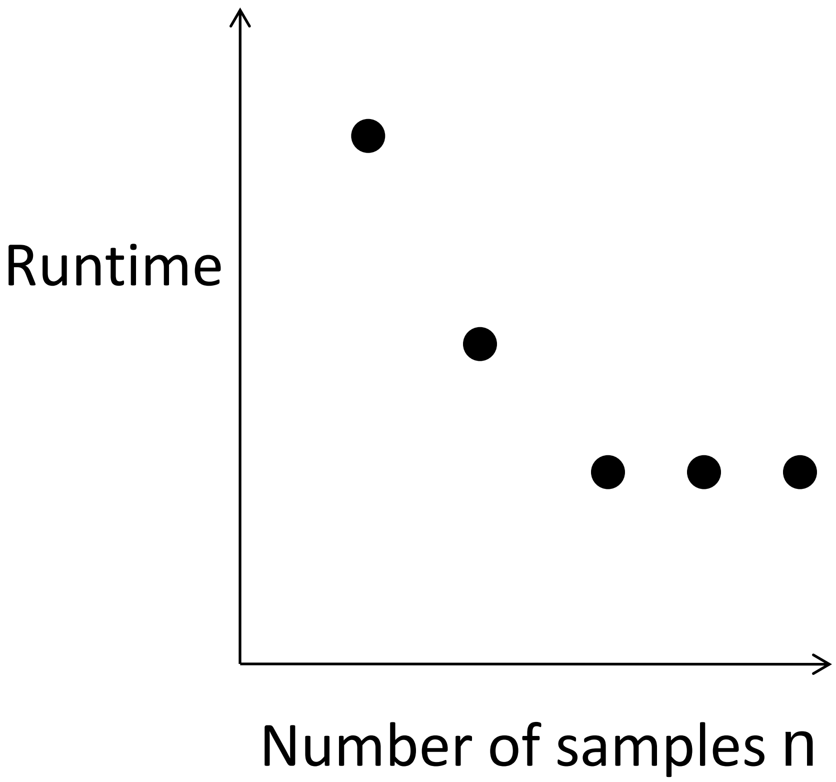
Example 4

C	Runtime	n
$\mathrm{conv}(\mathcal{S})$	super-poly(p)	$\sim p^{0.25}\log(p)$
nuclear norm ball	$p^{1.5}$	$\sim \sqrt{p}$

- What if we use an even weaker relaxation?
  - E.g., (properly scaled) Euclidean ball

- Require $\mathcal{O}(p)$ samples $\Rightarrow$ Runtime $= np + \mathcal{O}(p) = \mathcal{O}(p^2)$
- In this case, makes sense to throw away data …

Recall Plot …



- At some stage, throw away data

Some Questions

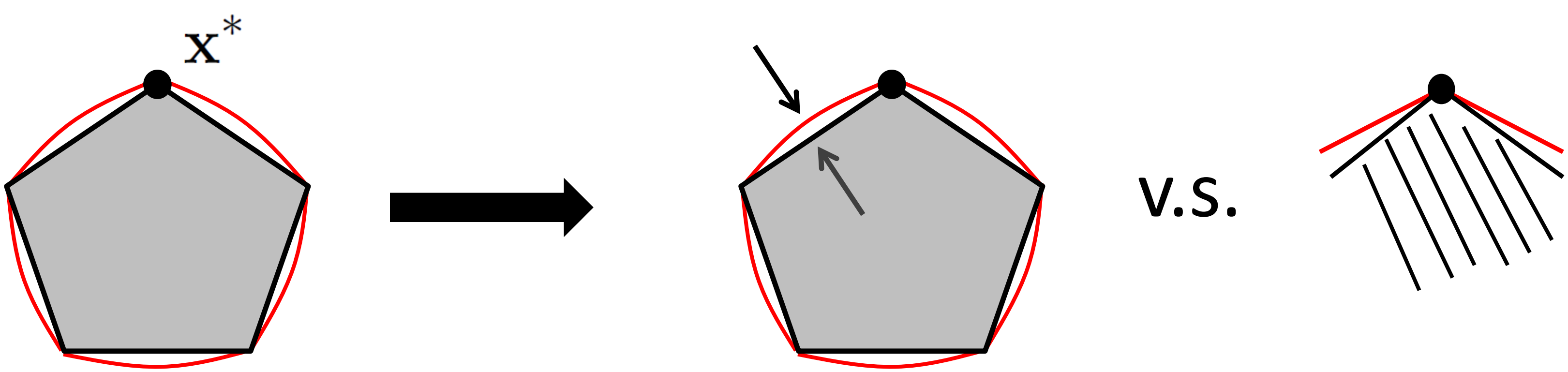
- In several examples, not too many extra samples required for really simple algorithms

- Approximation ratio might be bad, but doesn't matter as much for statistical inference

- Understand Gaussian complexities of LP/SDP hierarchies in contrast to theoretical CS

Some Questions

- Measuring the quality of approximation of convex sets
  - ***Approximation ratio*** is focus in theoretical CS
  - ***Gaussian complexities*** of interest in statistical inference



Approximation ratio in CS

Gaussian complexity in statistics

Summary

- Challenges with massive datasets
- Considered simple denoising problem
- Time-data tradeoffs via convex relaxation

- Future work:
  - Other methods to “weaken” algorithms
  - More complex statistical inference problems