Prospects in Applied Mathematics
University of Chicago
October 19 & 20, 2014

The Interplay Between Computation and Estimation

John Lafferty
Department of Statistics &
Department of Computer Science
University of Chicago
“Morally,” it should be possible to do more work/computation and get an improvement in a statistical estimation problem.

Or get a worse solution by investing less computation.
“Morally,” it should be possible to do more work/computation and get an improvement in a statistical estimation problem.

Or get a worse solution by investing less computation.

*How can we trade off statistical efficiency for computational efficiency in a principled, controlled way?*
How Many Beans in the Jar?
How Many Beans in the Jar?
How Many Beans in the Jar?
How Many Beans in the Jar?
Outline

- The general problem of tradeoffs
- Some previous work
- Quantized estimation: Motivation from star gazing
- Quantized estimation: Minimax theory
- Sparsified estimation
- Discussion
Collaborators

Yuancheng Zhu

Dinah Shender
Edward Nelson
1. The general problem of tradeoffs

“Morally,” it should be possible to do more work/computation and get an improvement in a statistical estimation problem.

Or get a worse solution by investing less computation.

How can we trade off statistical efficiency for computational efficiency in a principled, controlled way?
Maximize expected return on portfolio, subject to constraint on the variance (risk):

\[
\min -\mathbf{p}^T \mathbf{x} + \lambda \mathbf{x}^T \Sigma \mathbf{x}
\]

\text{s.t.} \quad 1^T \mathbf{x} = 1

x \geq 0
In numerical optimization, it’s understood (in part) how to trade off computation for speed of convergence

- First order methods: linear cost, linear convergence
- Quasi-Newton methods: quadratic cost, superlinear convergence
- Newton’s method: cubic cost, quadratic convergence

Are similar tradeoffs possible in statistical estimation/learning?
A formal framework

Computational analysis

\[ T_n = \inf_A \sup_{I_n \in \mathcal{P}} T(A, I_n) \]
A formal framework

Computational analysis

\[ T_n = \inf_A \sup_{I_n \in \mathcal{P}} T(A, I_n) \]

Statistical analysis

\[ R_n = \inf_{\hat{f}_n \in \mathcal{H}} \sup_{f \in \mathcal{F}} R(\hat{f}_n, f) \]
A formal framework

Computational analysis

\[ T_n = \inf_{A} \sup_{I_n \in P} T(A, I_n) \]

Statistical analysis

\[ R_n = \inf_{\hat{f}_n \in \mathcal{H}} \sup_{f \in \mathcal{F}} R(\hat{f}_n, f) \]

Computation-constrained minimax:

\[ R_n(T_n) = \inf_{\hat{f}_n, T(\hat{f}_n) \leq T_n} \sup_{f \in \mathcal{F}} R(\hat{f}_n, f) \]
A formal framework

Computational analysis

\[ T_n = \inf_A \sup_{l_n \in \mathcal{P}} T(A, l_n) \]

Statistical analysis

\[ R_n = \inf_{\hat{f}_n \in \mathcal{H}} \sup_{f \in \mathcal{F}} R(\hat{f}_n, f) \]

Computation-constrained minimax:

\[ R_n(T_n) = \inf_{\hat{f}_n \in \mathcal{H}, T(\hat{f}_n) \leq T_n} \sup_{f \in \mathcal{F}} R(\hat{f}_n, f) \]

↑

estimators using \( T_n \) units of computation
Constrained minimax

Risk $R(T)$ vs. Computation $T$
2. Previous work in different areas

- Brief survey of some steps in this direction
- Little detail, biased selection
Nash Equilibria

Nash equilibrium does not take computation into account

Example (Halpern and Pass, 2008):

- You are given an \( n \)-bit number, and are asked if it is prime.
- You can either give an answer, or say nothing.
- If you guess and are correct, you get $10. If you are wrong, you lose $10. If you say nothing, you get $1
Nash Equilibria

Nash equilibrium does not take computation into account

Example (Halpern and Pass, 2008):

- You are given an $n$-bit number, and are asked if it is prime.
- You can either give an answer, or say nothing.
- If you guess and are correct, you get $10. If you are wrong, you lose $10. If you say nothing, you get $1

Nash equilibrium: Give the correct answer.

If $n$ large, not worth it?
Hierarchical Testing Designs


- Formalization of “Twenty Questions”
- Find true pattern $Y$ from large set of possibilities $\mathcal{Y}$.
- Consider hierarchical (“course-to-fine”) family of tests
  - Zero type I error
  - Candidate tests $A \subset \mathcal{Y}$ form nested partitions
  - Tests evaluated according to power (type II error) and computational cost
Detecting geometric objects


- Detect line segments in two dimensional image data with Gaussian noise
- Compare multiscale geometric analysis to likelihood ratio test
Upper bounds for normal means
(Chandrasekaran and Jordan, 2012)

Risk is fixed at some level $\epsilon$. Methods have different sample complexities and runtimes.
Preference Learning

Hazan, Kale and Shalev-Shwartz, COLT 2012.

\[ \mathcal{X} = [d] \times [d], \mathcal{Y} = \{0, 1\}. \text{ Given } (i, j) \in \mathcal{X} \text{ predict if } i \text{ is preferable to } j. \]

Hypothesis class \( \mathcal{H} \) all permutations \( \mathfrak{S}_d \).

<table>
<thead>
<tr>
<th>Method</th>
<th>Comput.</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>( d! )</td>
<td>( O(d) )</td>
</tr>
<tr>
<td>HKS</td>
<td>( d^4 \log^3 d )</td>
<td>( O(d \log^3 d) )</td>
</tr>
<tr>
<td>ERM(( \mathcal{H}^{(n)} ))</td>
<td>( d^2 )</td>
<td>( O(d^2) )</td>
</tr>
</tbody>
</table>

\[ \text{Computation} \Rightarrow \text{Samples} \]
Known tradeoffs for sparse PCA (d’Aspremont et al. 2008, Amini and Wainwright, 2009)

<table>
<thead>
<tr>
<th>Method</th>
<th>Computation</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>$\binom{p}{k}$</td>
<td>$k \log(p-k)$</td>
</tr>
<tr>
<td>SDP relax</td>
<td>$np^2 + p^4 \log p$</td>
<td>$k \log(p-k)$</td>
</tr>
<tr>
<td>Thresh.</td>
<td>$np + p \log p$</td>
<td>$k^2 \log(p-k)$</td>
</tr>
</tbody>
</table>

SDP assumes existence of rank-1 solution. Berthet and Rigollet (2013) show $O(p^2 \log p)$ lower bound for polynomial time procedures.
Known tradeoffs for sparse PCA (d’Aspremont et al. 2008, Amini and Wainwright, 2009)

<table>
<thead>
<tr>
<th>Method</th>
<th>Computation</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>$\binom{p}{k}$</td>
<td>$k \log(p - k)$</td>
</tr>
<tr>
<td>SDP relax</td>
<td>$np^2 + p^4 \log p$</td>
<td>$k \log(p - k)$</td>
</tr>
<tr>
<td>Thresh.</td>
<td>$np + p \log p$</td>
<td>$k^2 \log(p - k)$</td>
</tr>
</tbody>
</table>

SDP assumes existence of rank-1 solution. Berthet and Rigollet (2013) show $O(k^2 \log p)$ lower bound for polynomial time procedures.
3. Quantized estimation: Motivation from Kepler

95 megapixel camera, (42 CCDs at $2200 \times 1024$ pixels) CCDs are read out every six seconds, and co-added on board for 30 minutes. The raw pixels are more data than can be stored and sent back to Earth.
One reading for a single star
Sample lightcurve for one star (among 150,000)
Task: Detrend, threshold, score

Figure 2. Kepler-29 lightcurves. Upper panel: the quarter-normalized, calibrated Kepler photometry (PA); lower panel: the detrended, normalized flux. The transit times of each planet are indicated by dots at the bottom of each panel.

From Fabrycky et al., 2012.
Motivation for quantized estimation

- On the telescope, all the captured data cannot be transmitted to earth. Data are aggregated on board, subsampled and then transmitted.
- Received light curves are \( \approx 1 \) terabyte. When we process them in our Amazon AWS system, storing and moving the data in Amazon S3 is limiting cost in the computation.
Suppose we represent the estimator with $B$ bits. In an optimal representation, how much do we lose in terms of minimax risk?

We answer this by revisiting nonparametric minimax theory from perspective of rate-distortion theory.
Normal means model

Normal means is the archetypal nonparametric problem. Captures essentials of nonparametric estimation.

Observe $X_i \sim N(\theta_i, \sigma_n^2)$, for $i = 1, 2, \ldots, n$.

Goal: Estimate means $\theta_i$ to minimize the risk $R(\hat{\theta}, \theta) = \mathbb{E}\|\hat{\theta} - \theta\|^2$. 
Normal means model

Normal means is the archetypal nonparametric problem. Captures essentials of nonparametric estimation.

Observe $X_i \sim N(\theta_i, \sigma_n^2)$, for $i = 1, 2, \ldots, n$.

Goal: Estimate means $\theta_i$ to minimize the risk $R(\hat{\theta}, \theta) = \mathbb{E}\|\hat{\theta} - \theta\|^2$.

**Pinsker’s Theorem.** Let $\sigma_n = \sigma/\sqrt{n} \equiv \varepsilon$. Over the $L_2$ ball $\Theta_n(c)$ of radius $c$, the asymptotic minimax risk is

$$R(\sigma, c) = \liminf_{n \to \infty} \inf_{\hat{\theta}} \sup_{\theta \in \Theta_n(c)} R(\hat{\theta}, \theta) = \frac{\sigma^2 c^2}{\sigma^2 + c^2}$$
Quantized normal means

We wish to limit the number of bits in our estimator:

\[ X_1, X_2, \ldots, X_n \mapsto \tilde{\theta}_1, \tilde{\theta}_2 \ldots, \tilde{\theta}_n \]

Classical rate-distortion setting:

minimize number of bits subject to \( \mathbb{E}(X - \tilde{X})^2 \leq D \)

In our estimation setting:

minimize number of bits subject to \( \inf_{\tilde{\theta}} \mathbb{E}(\tilde{\theta}(X) - \theta)^2 \leq R \)

We are quantizing with respect to the risk, or estimation error — the distortion in our estimation of an unknown constant
Rate distortion vs. quantized estimation

\[ X^n \rightarrow \text{Encoder} \varphi_n \xrightarrow{\varphi_n(X^n) \in C(B)} \text{Decoder} \psi_n \rightarrow \tilde{X}^n = \psi_n(\varphi_n(X^n)) \]
Rate distortion vs. quantized estimation

\[ X^n \rightarrow \text{Encoder} \ 
\varphi_n \ 
\xrightarrow{\varphi_n(X^n) \in C(B)} \ 
\text{Decoder} \ 
\psi_n \ 
\xrightarrow{} \ \hat{X}^n = \psi_n(\varphi_n(X^n)) \]

\[ \theta^n \rightarrow X^n \rightarrow \text{Encoder} \ 
\varphi_n \ 
\xrightarrow{\varphi_n(X^n) \in C(B)} \ 
\text{Decoder} \ 
\psi_n \ 
\xrightarrow{} \ \hat{\theta}^n = \psi_n(\varphi_n(X^n)) \]
Quantized minimax formulation

Quantized minimax risk

\[ R(\sigma, c, B) = \lim_{n \to \infty} \inf_{\tilde{\theta}^n \in \mathcal{Q}_n(B)} \sup_{\theta^n \in \Theta_n(c)} R_n(\tilde{\theta}^n, \theta^n). \]
Quantized minimax formulation

Quantized minimax risk

\[ R(\sigma, c, B) = \lim_{n \to \infty} \inf_{\tilde{\theta}^n \in \mathcal{Q}_n(B)} \sup_{\theta^n \in \Theta_n(c)} R_n(\tilde{\theta}^n, \theta^n). \]

\[ \uparrow \]

estimators using \( B \) bits

How does this risk depend on \( B \)?
Characterizing the tradeoff

**Theorem.** The asymptotic quantized minimax risk satisfies

\[
R(\sigma, c, B) \geq \frac{\sigma^2 c^2}{\sigma^2 + c^2} + \frac{c^4}{\sigma^2 + c^2} 2^{-2B}
\]

Moreover, this minimax lower bound is achievable by a random coding algorithm, which is adaptive to \(\|\theta\|\).
Quantization-risk tradeoffs: Pareto curve

\[ R(B) = \frac{\sigma^2 c^2}{\sigma^2 + c^2} + \frac{c^4 2^{-2B}}{\sigma^2 + c^2} \]
Random Coding Scheme

Step 1. *Generating codebooks.*

- Generate codebook $\mathcal{B} = \{1/\sqrt{n}, 2/\sqrt{n}, \ldots, [c^2 \sqrt{n}]/\sqrt{n}\}$.
- Generate codebook $\mathcal{X}$ of $2^{nB}$ i.i.d. random vectors from uniform distribution on $\mathbb{S}^{n-1}$.
Random Coding Scheme

Step 1. *Generating codebooks.*
- Generate codebook $\mathcal{B} = \{1/\sqrt{n}, 2/\sqrt{n}, \ldots, [c^2 \sqrt{n}]/\sqrt{n}\}$.
- Generate codebook $\mathcal{X}$ of $2^{nB}$ i.i.d. random vectors from uniform distribution on $\mathbb{S}^{n-1}$.

Step 2. *Encoding.*
- Encode $\tilde{b}^2 = \frac{1}{n} \|X\|^2 - \sigma^2$ by
  $$\tilde{b}^2 = \arg\min\{|b^2 - \tilde{b}^2| : b^2 \in \mathcal{B}\}.$$  
- Encode $X^n$ by
  $$\tilde{X}^n = \arg\max\{\langle X^n, x^n \rangle : x^n \in \mathcal{X}\}.$$
Random Coding Scheme

Step 1. **Generating codebooks.**
- Generate codebook $\mathcal{B} = \{1/\sqrt{n}, 2/\sqrt{n}, \ldots, [c^2\sqrt{n}]/\sqrt{n}\}$. 
- Generate codebook $\mathcal{X}$ of $2^{nB}$ i.i.d. random vectors from uniform distribution on $S^{n-1}$.

Step 2. **Encoding.**
- Encode $\tilde{b}^2 = \frac{1}{n} \|X\|^2 - \sigma^2$ by
  $$\tilde{b}^2 = \arg\min\{|b^2 - \tilde{b}^2| : b^2 \in \mathcal{B}\}.$$
- Encode $X^n$ by
  $$\tilde{X}^n = \arg\max\{\langle X^n, x^n \rangle : x^n \in \mathcal{X}\}.$$

Step 3. **Decoding.** Estimate $\theta$ by
  $$\tilde{\theta}^n = \sqrt{\frac{n\tilde{b}^4(1 - 2^{-2B})}{\tilde{b}^2 + \sigma^2}} \cdot \tilde{X}^n.$$
Optimality

**Theorem.** For a sequence of vectors $\theta^n$ satisfying $\|\theta^n\|^2 = b^2 \leq c^2$ we have

$$
\mathbb{P} \left( \|\theta^n - \tilde{\theta}^n\|^2 > \frac{\sigma^2 b^2}{\sigma^2 + b^2} + \frac{b^4 2^{-2B}}{\sigma^2 + b^2} + C \sqrt{\log \frac{n}{\sigma^2 + b^2}} \right) \rightarrow 0
$$

Thus, the quantized Pinsker bound is achievable.
Connecting Shannon and Kolmogorov

Donoho’s 1997 Wald Lectures show beautiful interplay between rate distortion, Kolmogorov’s metric entropy, and minimax theory.

<table>
<thead>
<tr>
<th>Shannon</th>
<th>Kolmogorov</th>
</tr>
</thead>
<tbody>
<tr>
<td>library</td>
<td>X stochastic</td>
</tr>
<tr>
<td>representers</td>
<td>codebook ( C )</td>
</tr>
<tr>
<td>fidelity</td>
<td>( \mathbb{E} \min_{X' \in C} | X - X' |^2 )</td>
</tr>
<tr>
<td>complexity</td>
<td>( \log</td>
</tr>
</tbody>
</table>

\[
H_\epsilon(\mathcal{F}) = \sup \left\{ R(\epsilon^2, X) : \mathbb{P}(X \in \mathcal{F}) = 1 \right\} (1 + o(1))
\]
White noise model

Classical model for nonparametric regression:

$$dY(t) = f(t)dt + \varepsilon dW(t)$$

where $f$ lies in the Sobolev space

$$\tilde{W}_m(c) = \left\{ f \in L_2[0, 1] : \{\theta_j\} \in \Theta(m, c) \right\}$$

where $\theta_j = \langle f, \varphi_j \rangle$ for the trigonometric basis, and $\Theta(m, c)$ is the ellipsoid

$$\Theta(m, c) = \left\{ \theta : \sum_{j=1}^{\infty} j^{2m} \theta_j^2 \leq \frac{c^2}{\pi^{2m}} \right\}$$
White noise model

Classical model for nonparametric regression:

\[ dY(t) = f(t)\,dt + \varepsilon\,dW(t) \]

where \( f \) lies in the Sobolev space

\[ \widetilde{W}_m(c) = \{ f \in L_2[0,1] : \{\theta_j\} \in \Theta(m, c) \} \]

where \( \theta_j = \langle f, \varphi_j \rangle \) for the trigonometric basis, and \( \Theta(m, c) \) is the ellipsoid

\[ \Theta(m, c) = \left\{ \theta : \sum_{j=1}^{\infty} j^{2m} \theta_j^2 \leq \frac{c^2}{\pi^{2m}} \right\} \]

We observe data

\[ Y_j = \int_0^1 \varphi_j(t)\,dY(t) = \theta_j + \varepsilon\xi_j \]

where \( \xi_j \sim N(0, 1) \).
Minimax risk for Sobolev ellipsoids

Minimax risk at noise level $\varepsilon$:

$$R_\varepsilon(m, c) = \inf_{\hat{\theta}} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$

Pinsker minimax bound:

$$R(m, c) = \lim_{\varepsilon \to 0} \inf_{\varepsilon^{\frac{-4m}{2m+1}}} R_\varepsilon(m, c) \geq \left( \frac{c}{\pi m} \right)^{\frac{2}{2m+1}} (2m+1)^{\frac{1}{2m+1}} \left( \frac{m}{m+1} \right)^{\frac{2m}{2m+1}}$$
Minimax risk for Sobolev ellipsoids

Minimax risk at noise level \( \varepsilon \): 

\[
R_{\varepsilon}(m, c) = \inf_{\hat{\theta}} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2
\]

Pinsker minimax bound:

\[
R(m, c) = \liminf_{\varepsilon \to 0} \varepsilon^{\frac{-4m}{2m+1}} \frac{c^2}{\pi m} \frac{(2m+1)^{1/2m+1}}{m+1} \left( \frac{m}{m+1} \right)^{\frac{2m}{2m+1}}
\]

\[\text{Pinsker constant } P_m(c)\]
Minimax risk at noise level $\varepsilon$ and quantization level $B_\varepsilon$:

$$R_\varepsilon(m, c, B) = \inf_{\hat{\theta} \in \mathcal{M}(B)} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$
Minimax risk at noise level $\varepsilon$ and quantization level $B_\varepsilon$:

$$R_\varepsilon(m, c, B) = \inf_{\hat{\theta} \in \mathcal{M}(B)} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$

↑

estimators using $B$ bits
Quantized Minimax risk for Sobolev ellipsoids

Minimax risk at noise level $\varepsilon$ and quantization level $B_\varepsilon$:

$$R_\varepsilon(m, c, B) = \inf_{\hat{\theta} \in \mathcal{M}(B)} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$

estimators using $B$ bits

Quantized minimax bound:

$$R(m, c, B) = \liminf_{\varepsilon \to 0} r(B_\varepsilon) R_\varepsilon(m, c, B_\varepsilon) \geq Q_m(c, B)$$
Minimax risk at noise level $\varepsilon$ and quantization level $B_\varepsilon$:

$$R_\varepsilon(m, c, B) = \inf_{\hat{\theta} \in \mathcal{M}(B)} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$

estimators using $B$ bits

Quantized minimax bound:

$$R(m, c, B) = \lim_{\varepsilon \to 0} \inf R_\varepsilon(m, c, B_\varepsilon) \geq Q_m(c, B)$$

rate of convergence
Quantized Minimax risk for Sobolev ellipsoids

Minimax risk at noise level $\varepsilon$ and quantization level $B_\varepsilon$:

$$R_\varepsilon(m, c, B) = \inf_{\hat{\theta} \in \mathcal{M}(B)} \sup_{\theta \in \Theta(m, c)} \mathbb{E} \| \hat{\theta} - \theta \|^2$$

↑

estimators using $B$ bits

Quantized minimax bound:

$$R(m, c, B) = \lim_{\varepsilon \to 0} \inf \ r(B_\varepsilon) R_\varepsilon(m, c, B_\varepsilon) \geq Q_m(c, B)$$

↑

rate of convergence  Pinsker-Zhu constant
Regime Change

Four regimes of quantization:

1. Lots of bits
2. Just enough to preserve rate
3. Too few, suffer rate loss
4. Constant
Quantized minimax estimation

**Theorem.**

1. If \( B_{\varepsilon} \frac{2}{2m+1} \to \infty \),

\[
\liminf_{\varepsilon \to 0} \varepsilon^{-\frac{4m}{2m+1}} R_{\varepsilon}(c, B) \geq P_m(c)
\]
Quantized minimax estimation

**Theorem.**

1. If \( B_{\varepsilon}^{\frac{2}{2m+1}} \rightarrow \infty \),

\[
\liminf_{\varepsilon \to 0} \varepsilon^{-\frac{4m}{2m+1}} R_\varepsilon(c, B) \geq P_m(c)
\]

2. If \( B_{\varepsilon}^{\frac{2}{2m+1}} \rightarrow b \),

\[
\liminf_{\varepsilon \to 0} \varepsilon^{-\frac{4m}{2m+1}} R_\varepsilon(c, B) \geq Q_m(c, b)
\]

where \( Q_m(c, b) \) is the solution of a cubic equation in \( b \).
Quantized minimax estimation

Theorem (continued)

3. If \( B \in 2^{m+1} \rightarrow 0 \) and \( B \rightarrow \infty \)

\[
\lim_{\varepsilon \rightarrow 0} B^{2m} R_{\varepsilon}(c, B) \geq \frac{c^2}{\pi^{2m}} m^{2m}
\]
Quantized minimax estimation

*Theorem* (continued)

3. If $B_{\varepsilon}^{\frac{2}{2m+1}} \to 0$ and $B \to \infty$

$$\lim_{\varepsilon \to 0} \inf B^{2m} R_{\varepsilon}(c, B) \geq \frac{c^2}{\pi^{2m}} m^{2m}$$

4. If $B$ is constant then

$$\lim_{\varepsilon \to 0} \inf R_{\varepsilon}(c, B) \geq \frac{c^2}{\pi^{2m}} \left( \exp \left( \frac{B}{m} \right) \ell! \right)^{2m/\ell}$$

where $\ell$ is the integer satisfying

$$\frac{\ell^\ell}{\ell!} < \exp \left( \frac{B}{m} \right) \leq \frac{(\ell + 1)^{(\ell+1)}}{(\ell + 1)!}$$
Summary of quantized minimax estimation

- Computation-risk tradeoffs for communication/storage constraints can be sharply characterized.
- Current coding schemes are exponential time.
- Promising future work: Coding/compression using sparse regression (Barron and Joseph, 2012; Venkataramanan et al., 2013).
Sparsified linear regression

We have been studying fine-grained tradeoffs (upper bounds) in the setting of large scale linear regression

Dinah Shender
Covariance-thresholded ridge regression

Ridge regression: \( \hat{\beta} = \arg \min_{\beta} \| Y - X\beta \|^2 + \lambda \| \beta \|^2 \)

\[
\begin{pmatrix}
\hat{\Sigma} \\
\end{pmatrix} + \lambda I \\
\begin{pmatrix}
\hat{\beta}_{\lambda} \\
\end{pmatrix} = \frac{1}{n} X^T Y
\]
Covariance-thresholded ridge regression

Ridge regression: \( \hat{\beta} = \arg \min_{\beta} \| Y - X\beta \|^2 + \lambda \| \beta \|^2 \)

Our estimator:

\[
\begin{pmatrix}
(\hat{\Sigma} + \lambda I) \\
T_t (\hat{\Sigma}) + \lambda I
\end{pmatrix} = \begin{pmatrix}
\frac{1}{n} X^T \\
\frac{1}{n} X^T
\end{pmatrix}
\]
Computation-risk tradeoffs for linear regression

Standard ridge estimator solves

\[
\left( \frac{1}{n} X^T X + \lambda_n I \right) \hat{\beta}_\lambda = \frac{1}{n} X^T Y
\]

Sparsify sample covariance to get estimator

\[
\left( T_t[\hat{\Sigma}] + \lambda_n I \right) \hat{\beta}_{t,\lambda} = \frac{1}{n} X^T Y
\]

where \( T_t[\hat{\Sigma}] \) is hard-thresholded sample covariance:

\[
T_t([s_{ij}]) = [s_{ij} \mathbf{1}(|s_{ij}| > t)]
\]

Solving SDD systems (Spielman et al.; Miller et al.; Kelner et al., 2009–2013)

\[
\tilde{O}(m \log p \log \varepsilon^{-1})
\]
Risk decomposition

Define

\[
\tilde{\beta}_{t,\lambda} = \mathbb{E}[\hat{\beta}_{t,\lambda} | X] = (T_t(\hat{\Sigma}) + \lambda I)^{-1}\hat{\Sigma}\beta^*
\]

\[
\beta_\lambda = \mathbb{E}\hat{\beta}_\lambda = (\Sigma + \lambda I)^{-1}\Sigma\beta^*
\]

Then we can decompose the risk as

\[
\|\tilde{\beta}_{t,\lambda} - \beta^*\|^2_\Sigma \leq 3 \left( \|\beta_{t,\lambda} - \tilde{\beta}_{t,\lambda}\|^2_\Sigma + \|\tilde{\beta}_{t,\lambda} - \beta_\lambda\|^2_\Sigma + \|\beta_\lambda - \beta^*\|^2_\Sigma \right)
\]

\(\text{variance}\)  \(\text{random design and thresholding}\)  \(\text{bias}^2\)
Risk decomposition

Define

\[ \tilde{\beta}_{t,\lambda} = \mathbb{E}[\hat{\beta}_{t,\lambda} | X] = (T_t(\hat{\Sigma}) + \lambda I)^{-1}\hat{\Sigma}\beta^* \]

\[ \beta_\lambda = \mathbb{E}\hat{\beta}_\lambda = (\Sigma + \lambda I)^{-1}\Sigma\beta^* \]

Then we can decompose the risk as

\[ \|\tilde{\beta}_{t,\lambda} - \beta^*\|^2_\Sigma \leq 3\left( \|\hat{\beta}_{t,\lambda} - \tilde{\beta}_{t,\lambda}\|^2_\Sigma + \|\tilde{\beta}_{t,\lambda} - \beta_\lambda\|^2_\Sigma + \|\beta_\lambda - \beta^*\|^2_\Sigma \right) \]

\[ O_P\left( \frac{\sigma^2}{n} \right) \quad O(\lambda^2\|\beta^*\|^2) \]
Bounding the excess risk

The term $\| \tilde{\beta}_{t, \lambda} - \beta_{\lambda} \|_{\Sigma}^2$ is the excess risk due to sparsification—which allows faster optimization.

It is controlled by $\| T_t(\hat{\Sigma}) - \Sigma \|$ and $\| \hat{\Sigma} - \Sigma \|$.

We work over class of covariance matrices with rows in sparse $\ell_q$ balls, $q < 1$, as studied by Bickel and Levina.
Suppose $\lambda = O(n^{-1/2})$. Then the excess risk of $\hat{\beta}_{t,\lambda}$ satisfies

$$\|\hat{\beta}_{t,\lambda} - \beta^*\|_2^2 = O_P \left( (t^{2(1-q)} + \frac{t^{-2q}}{n} + \lambda^2) \|\beta^*\|^2 + \frac{\sigma^2}{n} \right)$$

Assuming $T_t(\hat{\Sigma}) + \lambda I$ is diagonally dominant, the estimator can be computed in time

$$T(m_{n,t}, p) = \tilde{O}(m_{n,t} \log p \log n)$$

where $m_{n,t}$ is the number of nonzero entries in the thresholded covariance matrix $T_t(\hat{\Sigma})$. 
Computation-risk tradeoffs for linear regression

- Combined with the computational bounds for SDD systems, this gives us an explicit, fine-grained risk/computation tradeoff.
- Dinah has also studied variants of locality-sensitive hashing for efficient, approximate kernel regression.
Current work: Sparsified designs

\[
Y = \underbrace{\begin{bmatrix}
\beta \\
\vdots \\
\beta 
\end{bmatrix}}_{p \times 1} + \underbrace{\begin{bmatrix}
\varepsilon \\
\varepsilon \\
\varepsilon 
\end{bmatrix}}_{n \times 1}
\]

original data matrix \(X\)
Current work: Sparsified designs

\[ \mathbf{Y} = \mathbf{X} \mathbf{\beta} + \mathbf{\varepsilon} \]

\( \mathbf{Y} \) \( n \times 1 \)
\( \mathbf{X} \) \( n \times p \)
\( \mathbf{\beta} \) \( p \times 1 \)
\( \mathbf{\varepsilon} \) \( n \times 1 \)

\( \mathbf{X} \) is the sparsified data matrix.
Motivation:

- Sparsified problem can be solved in time $O(\text{nnz}(X) + \rho^3)$ using “subspace embedding” algorithms (Clarkson and Woodruff, 2012; Nelson and Nguyen, 2012)

- Connection to “drop-out” method in Deep Learning (Hinton et al., 2012, Wager et al., 2013)
Illustrated two facets of the interplay between estimation and computation:

- Storage-risk tradeoffs in nonparametric estimation
- Time-risk tradeoffs in sparsified linear regression

Motivated by modern concerns of massive data analysis.

Theoretical frameworks bringing together computational, statistical, and information theoretic perspectives.
Thank you!

arXiv:1409.6833 (NIPS 2014)
jmlr.org/proceedings/papers/v28/shender13.html (ICML 2013)