Paths Ahead in the Science of Information and Decision Systems

Martin Wainwright

UC Berkeley
Departments of Statistics, and EECS

Paths Ahead Symposium, MIT

§1. High-dimensional data: Challenges and opportunities

- data sets: n samples in p dimensions
  - computational biology: p genes measured in n humans
  - computer vision: p textures or objects, n images
  - natural language processing: p word frequencies over n documents
  - financial engineering: p stocks sampled at n distinct times
  - social network analysis: p senators vote on n bills

§1. High-dimensional data: Challenges and opportunities

- data sets: n samples in p dimensions
  - computational biology: p genes measured in n humans
  - computer vision: p textures or objects, n images
  - natural language processing: p word frequencies over n documents
  - financial engineering: p stocks sampled at n distinct times
  - social network analysis: p senators vote on n bills

- modern data sets often "high-dimensional" in nature:
  - massive: n and p often very large
  - "large p and ~~small~~ less large n": i.e., $p \approx n$ or $p \gg n$

§1. High-dimensional data: Challenges and opportunities

- data sets: n samples in p dimensions
  - computational biology: p genes measured in n humans
  - computer vision: p textures or objects, n images
  - natural language processing: p word frequencies over n documents
  - financial engineering: p stocks sampled at n distinct times
  - social network analysis: p senators vote on n bills

- modern data sets often "high-dimensional" in nature:
  - massive: n and p often very large
  - "large p and ~~small~~ less large n": i.e., $p \approx n$ or $p \gg n$

- curses of dimensionality:
  - explosion in computational costs with dimension
  - statistical curse: sample size n required to achieve error δ grows quickly with p (often exponentially)

§1. High-dimensional data: Challenges and opportunities

- data sets: n samples in p dimensions
  - computational biology: p genes measured in n humans
  - computer vision: p textures or objects, n images
  - natural language processing: p word frequencies over n documents
  - financial engineering: p stocks sampled at n distinct times
  - social network analysis: p senators vote on n bills
- modern data sets often "high-dimensional" in nature:
  - massive: n and p often very large
  - "large p and ~~small~~ less large n": i.e., $p \approx n$ or $p \gg n$
- curses of dimensionality:
  - explosion in computational costs with dimension
  - statistical curse: sample size n required to achieve error δ grows quickly with p (often exponentially)
- blessings of dimensionality:
  - concentration of measure: high-dimensional quantities can be remarkably predictable
  - hidden "effective" dimensionalities: sparsity in vectors/matrices; eigen-decay in matrices/operators; Markov relations, latent variables etc.

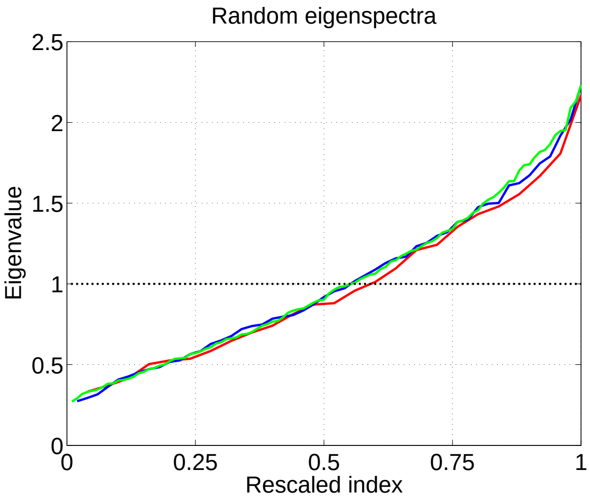
Example: Eigenanalysis in high-dimensions

Set-up: Collect n samples $\{Y_i\}_{i=1}^n$ of zero-mean random vector with covariance $\Sigma \in \mathbb{R}^{p \times p}$.

Goal: Estimate eigenstructure (eigenvalues and vectors) of Σ, say using the sample covariance $\widehat{\Sigma}_n = \frac{1}{n} \sum_{i=1}^n Y_i Y_i^T$. Look at scaling as $(n, p) \rightarrow +\infty$.

Uses/relevance: Principal components analysis, canonical correlation analysis, spectral clustering etc. ...

Example: Eigenanalysis in high-dimensions



Eigenspectrum concentrates on interval $[(1 - \sqrt{\frac{p}{n}})^2,\ (1 + \sqrt{\frac{p}{n}})^2]$.

(e.g., Marcenko & Pastur, 1967, Geman, 1980, Szarek, 1991)

Example: Eigenanalysis in high-dimensions

Set-up: Collect n samples $\{Y_i\}_{i=1}^n$ of zero-mean random vector with covariance $\Sigma \in \mathbb{R}^{p \times p}$.

Goal: Estimate eigenstructure (eigenvalues and vectors) of Σ, say using the sample covariance $\widehat{\Sigma}_n = \frac{1}{n} \sum_{i=1}^n Y_i Y_i^T$. Look at scaling as $(n, p) \to +\infty$.

Uses/relevance: Principal components analysis, canonical correlation analysis, spectral clustering etc. ...

Eigenanalysis with structural constraints:

1. Some models:
   - Spiked covariance models with sparse eigenvectors
   - Covariance matrices with rapid eigendecay
   - Inverse covariance matrices with Markov structure (i.e., Gaussian graphical models)

2. Some estimators:
   - thresholded versions of sample covariance
   - regularized M-estimators (based on solving convex programs)

Example: Learning graphical models

- random variable X_s at node s takes values in discrete space (e.g., $\mathcal{X} = \{-1, +1\}$)
- hierarchies of probability distributions:

Example: Learning graphical models

- random variable X_s at node s takes values in discrete space (e.g., $\mathcal{X} = \{-1, +1\}$)
- hierarchies of probability distributions:
  - Independence model (biased but independent coin flips):

$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\big\{ \sum_{s \in V} \theta_s x_s \big\}.$$

Example: Learning graphical models

- random variable X_s at node s takes values in discrete space (e.g., $\mathcal{X} = \{-1, +1\}$)

- hierarchies of probability distributions:
  - Independence model (biased but independent coin flips):

$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\Big\{ \sum_{s \in V} \theta_s x_s \Big\}.$$

  - Pairwise MRF (Ising model, 1923)

$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\Big\{ \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t \Big\}.$$

Example: Learning graphical models

- random variable X_s at node s takes values in discrete space (e.g., $\mathcal{X} = \{-1, +1\}$)

- hierarchies of probability distributions:
  - Independence model (biased but independent coin flips):

$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\Big\{\sum_{s \in V} \theta_s x_s\Big\}.$$

  - Pairwise MRF (Ising model, 1923)

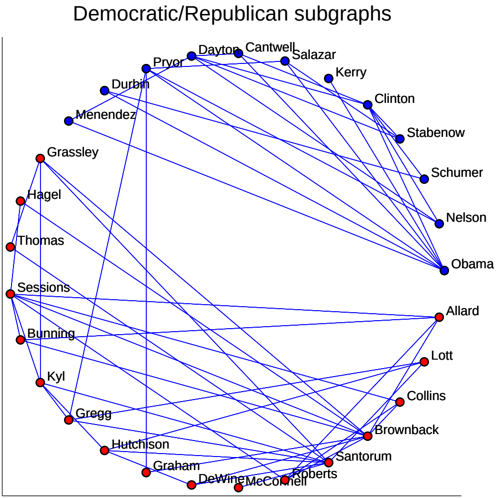
$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\Big\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t\Big\}.$$

  - Triplet MRF

$$\mathbb{P}(x_1, \ldots, x_p) = \frac{1}{Z(\theta)} \exp\Big\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E_2} \theta_{st} x_s x_t + \sum_{(s,t,u) \in E_3} \theta_{stu} x_s x_t x_u\Big\}.$$

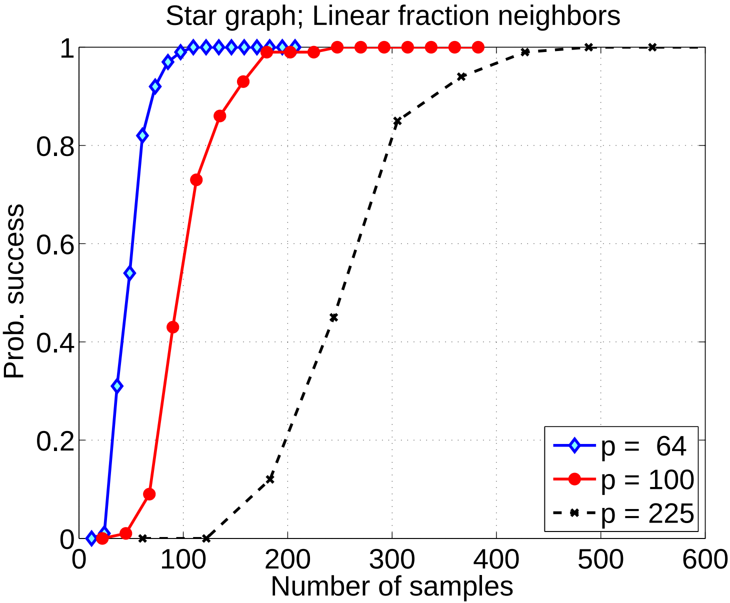
- (hyper)graph structure enforces that $\theta_{uv} = 0$ for all $(uv) \notin E$

Example: Learning social network structure

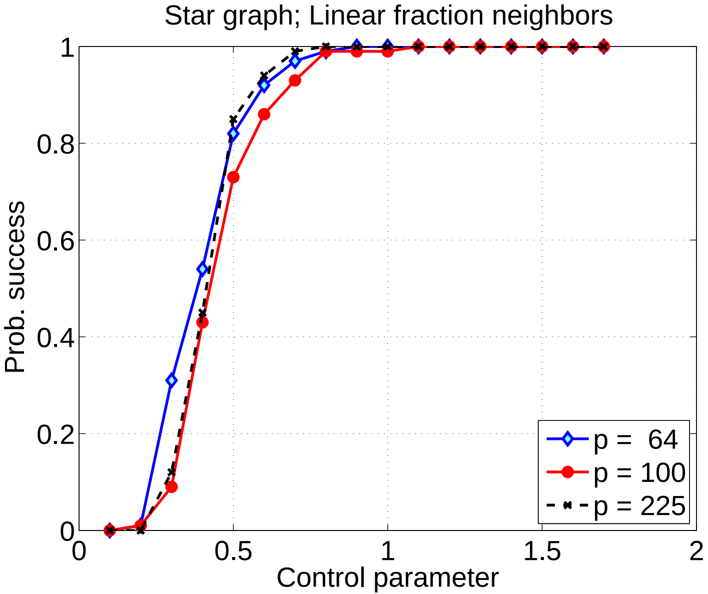


Graphical model fit to voting records of US senators (using technique from Ravikumar et al., 2008)

Empirical behavior: Unrescaled plots



Empirical behavior: Appropriately rescaled



§2. Trade-offs between computation and statistics

- given a fixed set of resources (storage, communication, processing), two different types of costs:
  - costs associated with collecting data (i.e., running experiments, simulations, MCMC sampling etc.)
  - costs associated with performing statistical inference

§2. Trade-offs between computation and statistics

- given a fixed set of resources (storage, communication, processing), two different types of costs:
  - costs associated with collecting data (i.e., running experiments, simulations, MCMC sampling etc.)
  - costs associated with performing statistical inference

- for many problems, there are hierarchies of methods ordered by computational complexity:
  - "naive" methods (e.g., greedy search, thresholding, heuristics etc.)
  - relaxation hierarchies (e.g., via LP, SDP and other convex programs)
  - optimal procedures (may require exponential time or space)

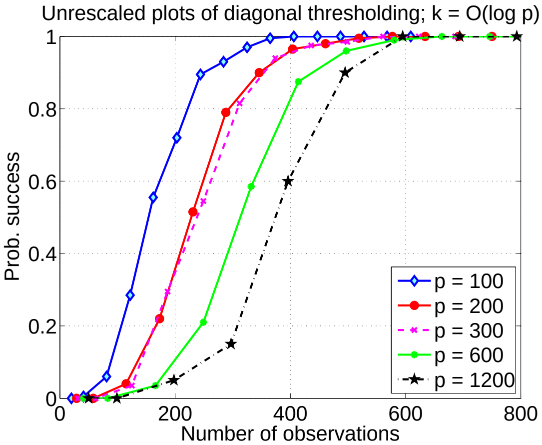
§2. Trade-offs between computation and statistics

- given a fixed set of resources (storage, communication, processing), two different types of costs:
  - costs associated with collecting data (i.e., running experiments, simulations, MCMC sampling etc.)
  - costs associated with performing statistical inference

- for many problems, there are hierarchies of methods ordered by computational complexity:
  - "naive" methods (e.g., greedy search, thresholding, heuristics etc.)
  - relaxation hierarchies (e.g., via LP, SDP and other convex programs)
  - optimal procedures (may require exponential time or space)

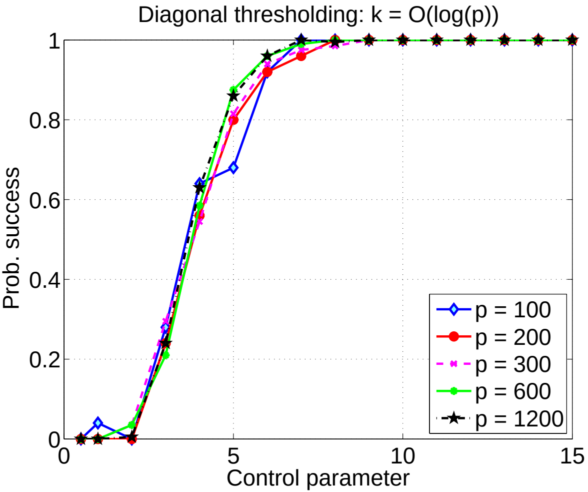
- some open questions:
  - for fixed sample size n, when does more computation guarantee greater accuracy?
  - can we derive fundamental limits that include upper bounds on computation?
  - does computational complexity versus sample size n exhibit non-monotonic behavior?

Empirical performance of thresholding



- spiked covariance model $\Sigma = zz^T + \sigma^2 I$
- model selection: find k-sized subset $S \subset \{1, \ldots, p\}$ where $z \in \mathbb{R}^p$ is non-zero
- plot the success probability $\mathbb{P}[\widehat{S} = S^*]$ versus sample size n.

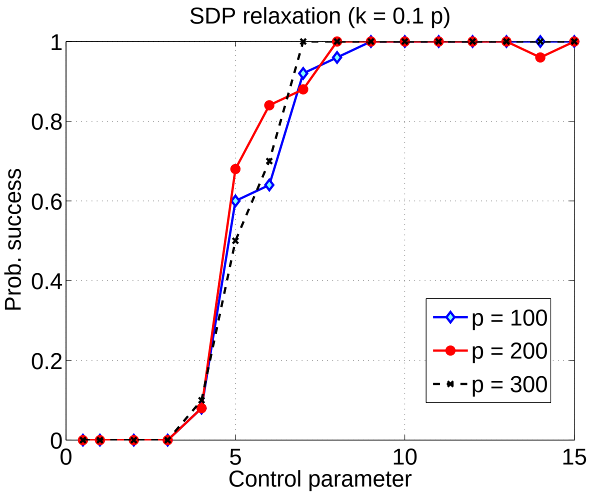
Empirical performance of thresholding



- success prob. versus rescaled sample size:

$$\theta_{\text{thr}}(n, p, k) = \frac{n}{k^2 \log(p-k)}.$$

More computationally expensive SDP relaxation



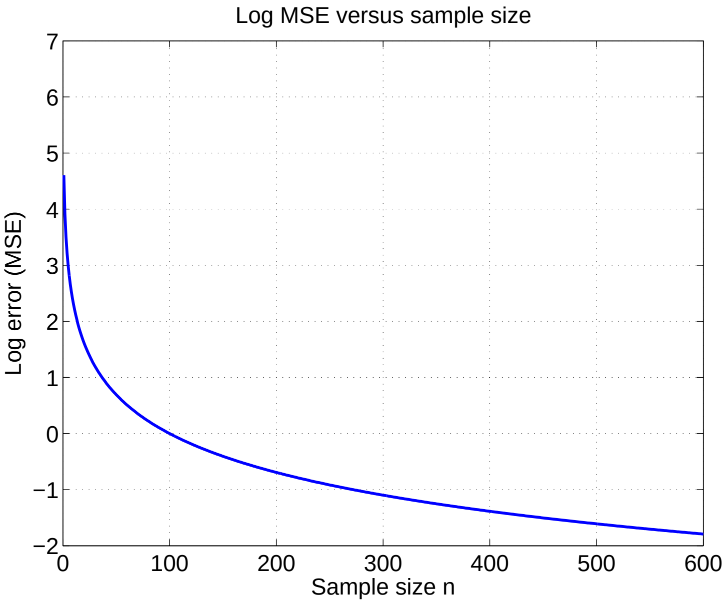
Probability versus rescaled sample size

$$\theta_{\mathrm{sdp}}(n, p, k) = \frac{n}{k \log(p-k)}.$$

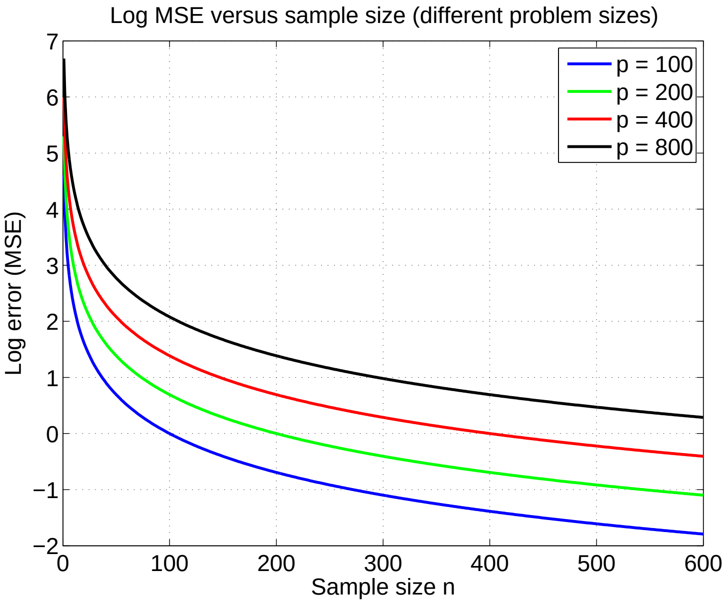
Summary

- contributions possible/required from multiple disciplines:
  - electrical engineering
  - computer science
  - statistics
  - applied mathematics
- optimization theory and statistics:
  - need theory for analyzing random ensembles of optimization problems
  - need algorithms for solving large-scale instances
- control theory and statistics:
  - on-line learning introduces interesting dynamical aspects
  - stochastic approximation
- information-theoretic methods in learning:
  - statistical inference $\equiv$ (non-orthodox) communication channel:
    - codewords/codebook $\equiv$ parameter θ in set Θ
    - drawing samples $\equiv$ using channel
  - fundamental lower bounds via Fano and other methods
- applied probability and statistics:
  - large deviations; concentration of measure
  - empirical process theory

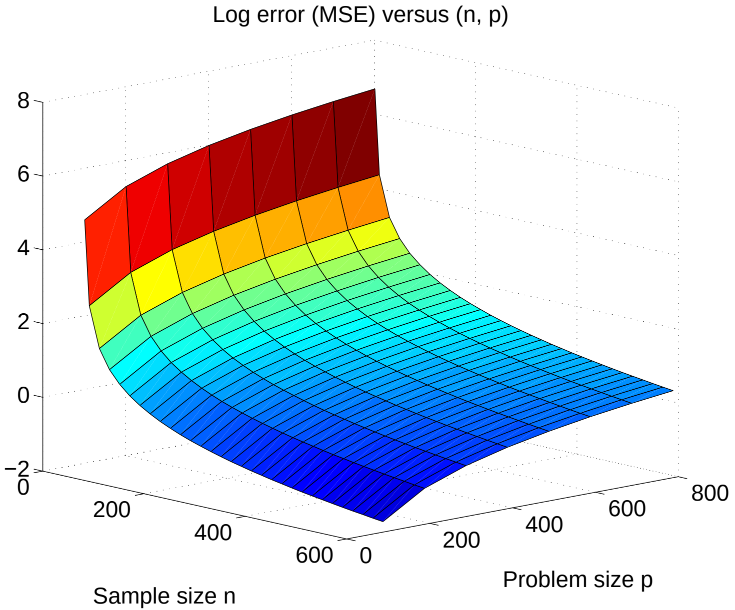
Traditional asymptotics



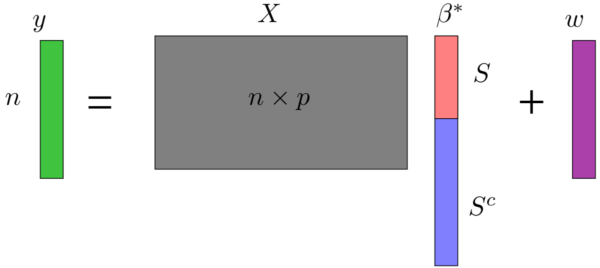
Challenge: High-dimensional scaling laws



Challenge: High-dimensional scaling laws



Example: (Sparse) linear regression



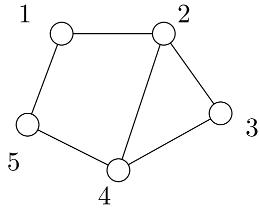
Set-up:
- vector $\beta^* \in \mathbb{R}^p$ with at most $k \ll p$ non-zero entries
- noisy observations $y = X\beta^* + w$

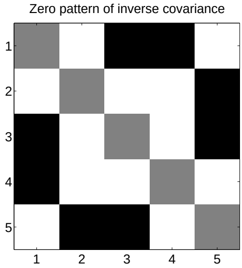
Goal: Generate "good" estimate $\widehat{\beta}$ of β^* (various loss functions: prediction, ℓ_2-loss, model selection)

Applications: Imaging; data-base sketching; compressed sensing.

Some relevant work: Portnoy, 1984; Tibshirani, 1996; Chen et al., 1998; Donoho/Xuo, 2001; Tropp, 2004; Fuchs, 2004; Meinshausen/Buhlmann, 2005; Candes/Tao, 2005; Donoho, 2005; Haupt & Nowak, 2006; Zhao/Yu, 2006; Wainwright, 2006; Tsybakov et al., 2008

Example: Structured matrix estimation





Set-up: Samples from random vector with structured covariance Σ, or structured inverse covariance Θ.

Goal: Produce estimates $\widehat{\Sigma}$ (or $\widehat{\Theta}$) close in Frobenius or spectral norm.

Applications: Social network analysis, computer vision, financial time series analysis, geostatistics....

<u>Some relevant work:</u> Marcenko & Pastur, 1967; Geman, 1980; Bai, 1999; Ledoit & Wolf, 2003; Bickel & Levina, 2006, 2007; d'Asprémont et al., 2007; El Karoui, 2007; Rothman et al., 2007; Yuan & Lin, 2007; Ravikumar et al., 2008