Subsampling, Spectral Methods and Semidefinite Programming

Alexandre d'Aspremont, *Princeton University*

Joint work with **Noureddine El Karoui**, *U.C. Berkeley*.

Support from NSF, DHS and Google.

Introduction

- PCA, LSI, spectral clustering, etc., do not require a high precision.
- Idem for many optimization problems.
- Yet, we perform most linear algebra operations with **15 digits of accuracy**. . .

Central question in this talk: **How low can we go?**

Introduction

The **computing** environment is changing. . .

- Storage is extremely cheap.
- CPUs are cheap too (Amazon EC2: \$0.10/hour, Google, Yahoo, etc.).
- Clock speed is very expensive (physical limitations).
- Bandwidth and latency are also key limitations.

Solution: break large numerical problems into many small, independent tasks.

Introduction

Juditsky, Lan, Nemirovski & Shapiro (2009) solve

$$\min_{x \in X} \max_{y \in Y} x^T A y + b^T x + c^T y$$

where $X = \{x \in \mathbf{R}^n : x \geq 0,\ \mathbf{1}^T x = 1\}$ and $Y = \{y \in \mathbf{R}^n : y \geq 0,\ \mathbf{1}^T y = 1\}$, using a stochastic gradient algorithm.

- Linear algebra operations are performed using subsampling: at each iteration Ax is replaced by A_i where i is sampled from x.
- Use entropy based Bregman projections on the simplex.
- The total cost of getting a solution with relative accuracy ϵ and confidence $1 - \delta$ is then

$$O\left(\frac{n \log n + n \log(1/\delta)}{\epsilon^2}\right)$$

which is **negligible** compared to the data size $O(n^2)$!

Introduction

Focus on eigenvalues & eigenvectors.

Spectral methods.

- Computing leading eigenvectors using iterative methods costs $O(n^2)$.
- Approximate leading eigenvectors at a cost below $O(n^2)$?

Optimization.

- Subgradient techniques for semidefinite programming require leading eigenvectors.
- Use approximate eigenvectors in semidefinite programming while controlling complexity?

Introduction

Subsampling procedure from Achlioptas & McSherry (2007).

Given $p \in [0,1]$ and a symmetric matrix $A \in \mathbf{S}_n$, define

$$S_{ij} = \begin{cases} A_{ij}/p & \text{with probability p} \\ 0 & \text{otherwise.} \end{cases}$$

By construction

- S has mean A and independent coefficients.
- S is sparse, it has pn^2 nonzero entries on average when A is dense.

Because of independence, the impact of subsampling on the spectrum is both small and isotropic. . .

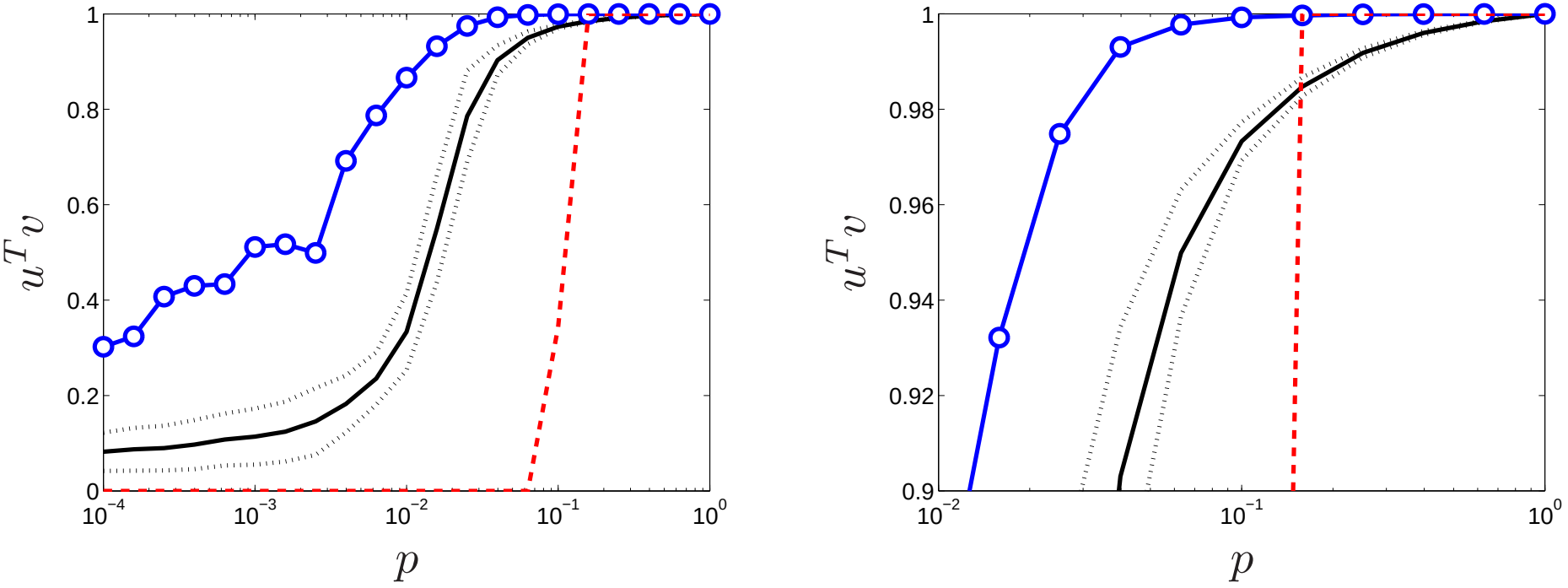
Introduction

A simple experiment.

- Covariance matrix of the 500 most active genes in the cancer data set from Alon, Barkai, Notterman, Gish, Ybarra, Mack & Levine (1999).
- Use the subsampling procedure described in the previous slide for various values of the sampling probability p.
- For each sample, measure the **alignment** $u^T v$ between the true eigenvector u and its approximation v.
- Average all subsampled vectors and test its quality too.

Introduction

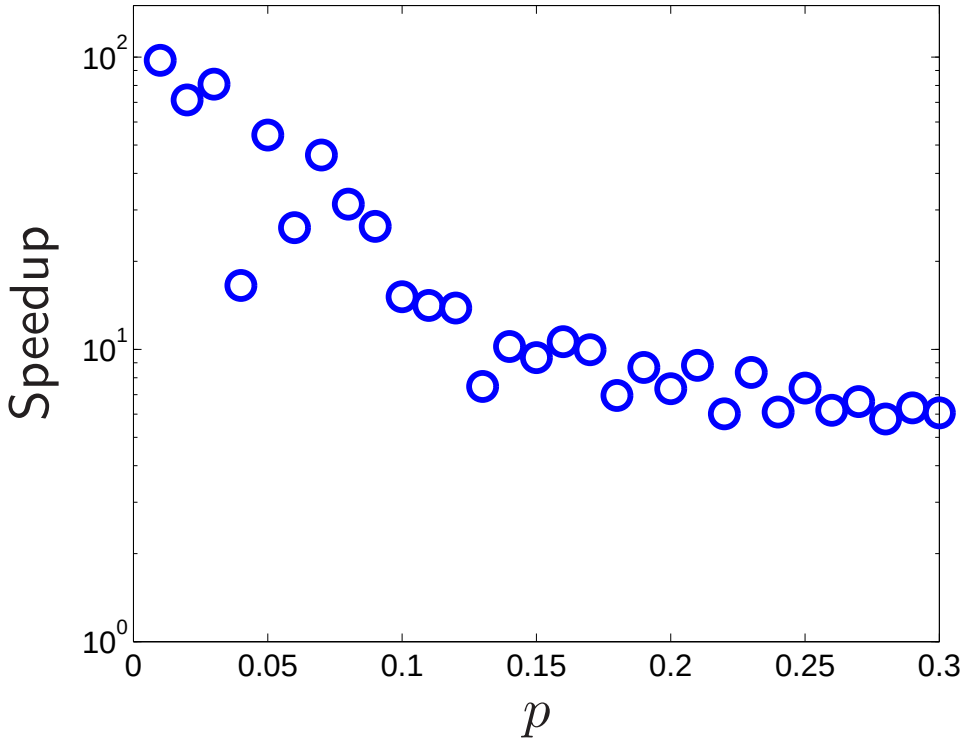
Phase transition.



Left: Alignment $u^T v$ between the true and the normalized average of 1000 subsampled eigenvectors (blue circles), median value of $u^T v$ over all sampled matrices (solid black line), with dotted lines at plus and minus one standard deviation, for various values of the sampling probability p on a gene expression covariance matrix. *Right:* Zoom on the the interval $p \in [10^{-2}, 1]$.

Introduction

The cost of computing a few leading eigenvectors of a sparse matrix using the power or Lanczos methods is proportional to the number of nonzero coefficients. The main benefit is **lower memory** usage however.



Ratio of average CPU time for computing the leading eigenvalue of a (sparse) subsampled matrix using ARPACK (directly) over average CPU time for the original (dense) matrix, versus sampling probability p, with $n = 2000$.

Introduction

References.

- Early subsampling results by Groh, Marshall, Kunz & Givens (1991) and Papadimitriou, Raghavan, Tamaki & Vempala (2000) who described algorithms based on subsampling and random projections.
- Explicit error estimates for columnwise and elementwise sampling strategies followed in Frieze, Kannan & Vempala (2004), Drineas, Kannan & Mahoney (2006), Achlioptas & McSherry (2007).
- More recently, Recht, Fazel & Parrilo (2007), Candes & Recht (2008), Candes & Tao (2009), Keshavan, Montanari & Oh (2009) focused on low-rank matrix reconstruction.
- Stability results in clustering and ranking by Ng, Zheng & Jordan (2001) and Huang, Yan, Jordan & Taft (2008) for example.
- In optimization: Bertsimas & Vempala (2004) solve convex feasibility problems using random walks, Arora & Kale (2007) focus on combinatorial relaxations and Juditsky & Nemirovski (2008) use subsampling to solve a matrix game.

Outline

- Introduction
- **Subsampling**
- Semidefinite programming

Elementwise subsampling

Subsampling: given $A \in \mathbf{S}_n$ and $p \in [0,1]$, define

$$S_{ij} = \begin{cases} A_{ij}/p & \text{with probability p} \\ 0 & \text{otherwise.} \end{cases}$$

Achlioptas & McSherry (2007, Th. 1.4) show that

$$\|A - S\|_2 \leq 4\sqrt{\frac{n}{p}}\ \max_{ij} |A_{ij}|,$$

holds with high probability for n large enough.

- What is the lowest reasonable p here?
- How does $\max_{ij} |A_{ij}| \sqrt{n/p}$ behave when $n \to \infty$?

Matrix Completion

Let's write

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

- Recent results in Recht et al. (2007), Candes & Recht (2008), Candes & Tao (2009), Keshavan et al. (2009) show that when

$$\|u_i\|_\infty \leq \frac{\mu}{\sqrt{n}}$$

with $\mu = O(1)$ and $\mathbf{Rank}(A) = O(1)$, and we sample more than $O(n \log^4 n)$ coefficients, then

$$\begin{array}{rlll} A = & \text{argmin} & \|X\|_* & \\ & \text{subject to} & X_{ij} = S_{ij}, & \text{when } S_{ij} \neq 0 \end{array}$$

with high probability.

- All the information we need on A is contained in S (which only has $O(n \log^4 n)$ nonzero coefficients), but is **expensive to extract**.

Elementwise subsampling

With

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

- Here, we measure the **incoherence** of the spectrum of A by

$$\mu(A, \alpha) = \sum_{i=1}^{n} |\lambda_i| n^{\alpha_i} \|u_i\|_\infty^2$$

  for some $\alpha \in [0,1]^n$.
- Because $\mathbf{NumCard}(x) = \|x\|_2^2/\|x\|_\infty^2$, the norm $\|u_i\|_\infty$ is a proxy for sparsity.
- When the eigenvector coefficients are uniformly distributed, we have

$$\|u_i\|_\infty \sim n^{-1/2}, \quad \text{when } n \to \infty$$

  we can take $\alpha_i = 1$ and $\mu(A, \alpha)$ will be bounded if $\|X\|_1$ remains bounded.

Error bounds

Main Result. *Suppose there is a vector* $\alpha \in [0,1]^n$ *for which*

$$\mu(M,\alpha) \leq \mu \quad \textit{and} \quad \mathbf{Card}(u_i) \leq \frac{\kappa}{2} n^{\alpha_i}, \quad i = 1, \ldots, n$$

as $n \to \infty$*, where* μ *and* κ *are absolute constants. Then*

$$\|A - S\|_2 \leq \frac{\kappa\mu}{\sqrt{pn^{\alpha_{\min}}}}$$

almost surely (asymptotically), where $\alpha_{\min} = \min_{i=1,\ldots,n} \alpha_i$.

Error bounds

Proof. (Sketch) Using e.g. Horn & Johnson (1991, Th. 5.5.19)

$$\begin{aligned} \|A - S\|_2 &= \sqrt{\frac{1-p}{p}} \left\| \sum_{i=1}^{n} \lambda_i C \circ (u_i u_i^T) \right\|_2 \\ &\leq \sqrt{\frac{1-p}{p}} \sum_{i=1}^{n} |\lambda_i| n^{\alpha_i/2} \|u_i\|_\infty^2 \left\| \frac{C_{\alpha_i}}{n^{\alpha_i/2}} \right\|_2 \end{aligned}$$

where C is i.i.d. Bernoulli with

$$C_{ij} = \begin{cases} \sqrt{(1-p)/p} & \text{with probability } p \\ -\sqrt{p/(1-p)} & \text{otherwise.} \end{cases}$$

with C_{α_i} is a sparse submatrix of C. Vu (2007, Th. 1.4) shows

$$\left\| \frac{C_{\alpha_i}}{n^{\alpha_i/2}} \right\|_2 \leq \kappa$$

almost surely (asymptotically), whenever p is greater than $(\log n)^{4+\delta}/n^{\alpha_i}$, for some $\delta > 0$. We need to control the max. of these quantities.

Error bounds

A few facts. . .

- The subsampled matrix must have more than

$$O(n^{(2-\alpha_{\min})} \log^4 n)$$

  nonzero coefficients to keep the error small ($\alpha_{\min} = 1$ for incoherent spectrum). The **sparsest eigenvector** determines error size through $\alpha_{\min}$.

- The smallest submatrix we can form using this result must have more than $O\left(n \log^4 n\right)$ nonzero coefficients.

- Because $\|C/n^{1/2}\|_2$ blows up if $p \leq (\log\, n)^{1-\delta}/n$, the best we can hope for here is $O(n \log n)$ nonzero coefficients.

Coupon Collector

Coupon collector. *Let C_n be the number of elements that need to be drawn from [1,n] with replacement until one first obtains a collection with n different elements. Then*

$$\mathbf{E}[C_n] = nH_n$$

where H_n is the n^{th} harmonic number, with $H_n \sim \log n$.

The $O(n \log n)$ lower bound on the size of the sampled matrix is natural since below this rate we are not guaranteed to sample from every row/column.

Averaging

- Kato (1995, Theorem II.3.9) shows that if

$$\|A - S\|_2 \leq (\lambda_1 - \lambda_2)/2$$

the subsampled matrix can be seen as a small **perturbation** of the original one.

- If the matrix satisfies

$$\frac{\kappa\mu}{\sqrt{pn^{\alpha_{\min}}}} \leq (\lambda_1 - \lambda_2)/2,$$

we get the following expansion for the leading eigenvector v of the subsampled matrix compared to the true vector u

$$v = u - REu + R(E - u^T Eu\mathbf{I})RE + o_P(\|E\|_2^2)$$

where $E = A - S$ and R is the reduced resolvent of A, written

$$R = \sum_{j \neq 1} \frac{1}{\lambda_j - \lambda_1} u_j u_j^T.$$

Averaging

- **Phase transition** when

$$\|A - S\|_2 \leq (\lambda_1 - \lambda_2)/2$$

- Because

$$\mathbf{E}[R(A - S)u] = 0,$$

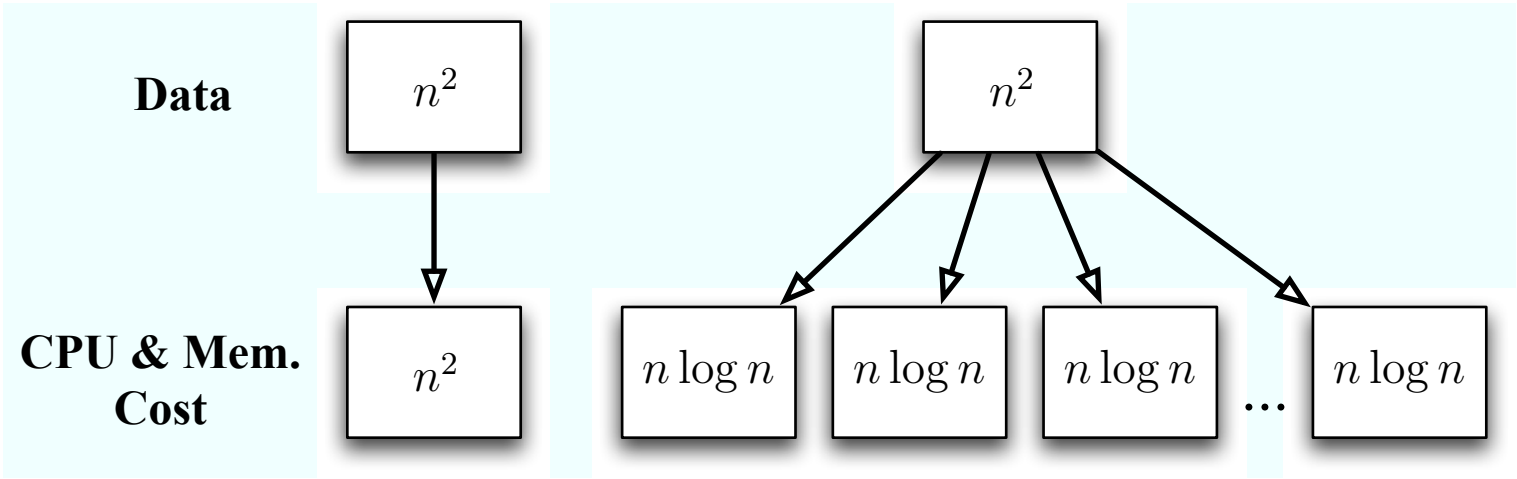
  averaging eigenvectors over many subsampled matrices in the perturbative regime means that the residual error will be of order $\|A - S\|_2^2$.
- The variance of the first order term is given by

$$\mathbf{E}[\|REu\|_2^2] \leq \frac{1}{(1 - \lambda_2/\lambda_1)^2} \|u_1\|_\infty^2 \frac{\mathbf{NumRank}(A)}{p}$$

  so the quality of the eigenvector approximation is a function of
  - The **spectral** gap λ_2/λ_1.
  - The **numerical sparsity** of u_1, measured by $\|u_1\|_\infty$.
  - The **numerical rank** of the matrix A.
  - The **sampling probability** p.

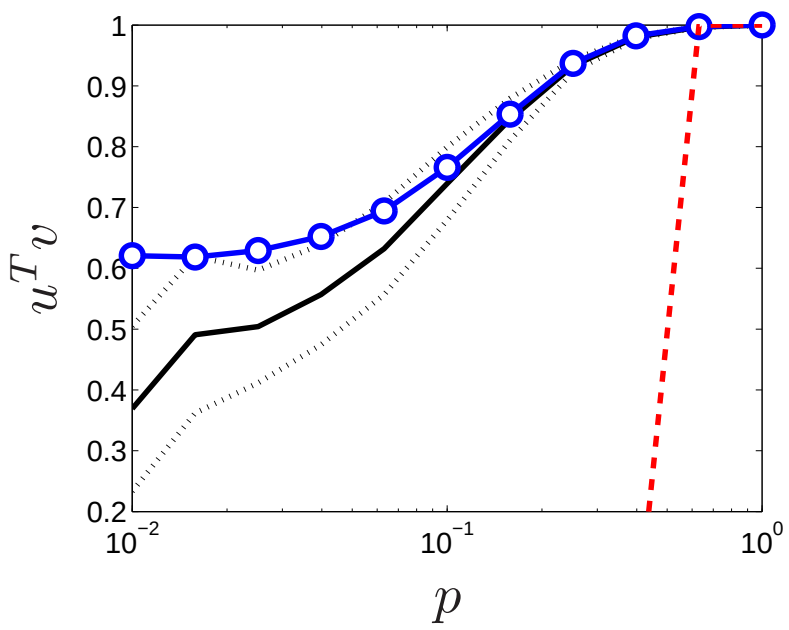
Computing model

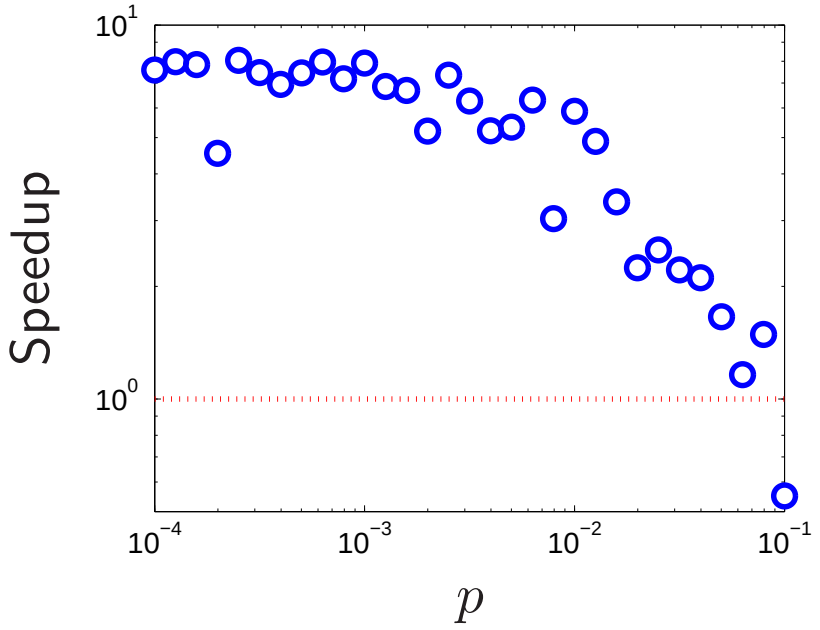
Higher granularity. . .



This works if getting additional CPUs is very cheap, but memory and bandwidth are limited.

Numerical results: Averaging

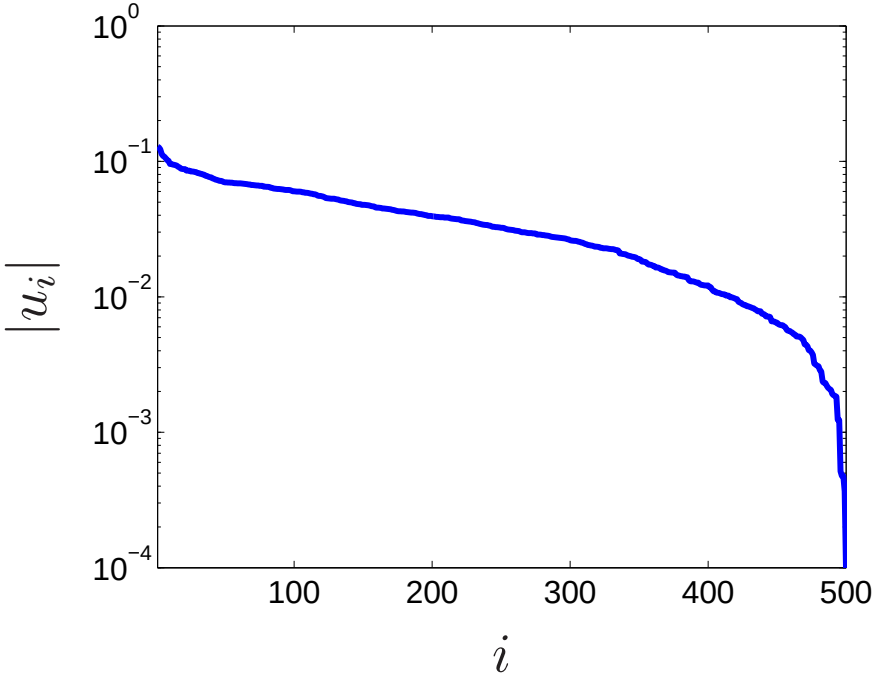


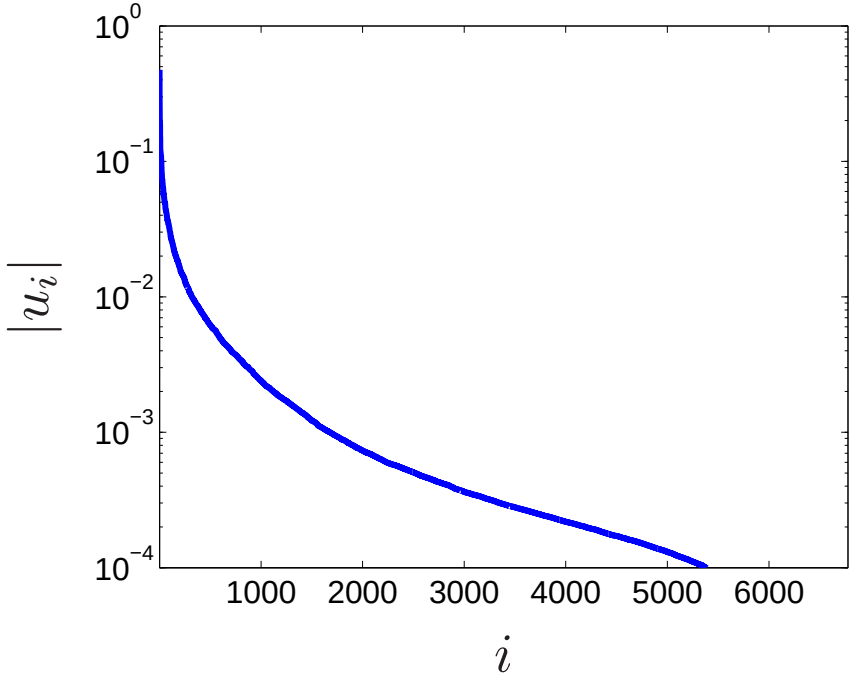


Left: Alignment $u^T v$ between the true and the normalized average of 1000 subsampled left eigenvectors (blue circles), median value (solid black line) and proportion of samples satisfying the perturbation condition (dashed red line), for various values of p on a term document matrices with dimensions 6779×11171.

Right: Speedup in computing leading eigenvector of a gene expression data, including subsampling time, for various values of the sampling probability p. (Memory scales linearly with p).

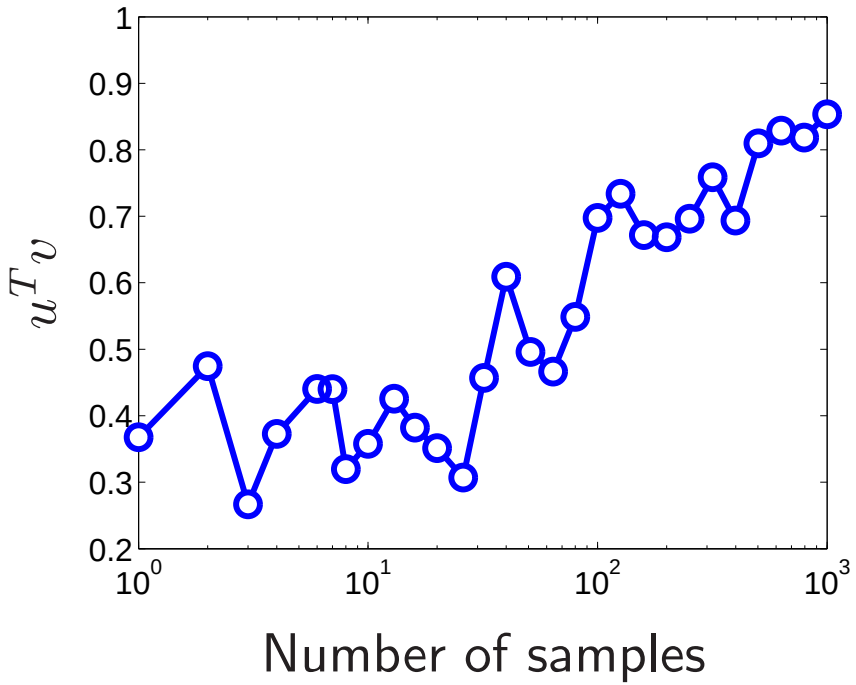
Numerical results: Averaging

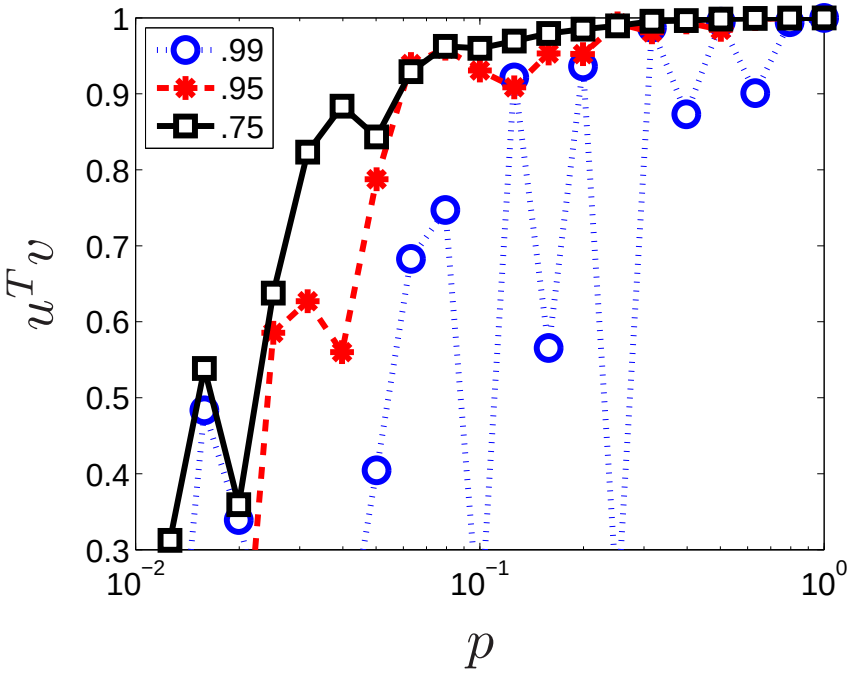




Magnitude of eigenvector coefficients $|u_i|$ in decreasing order for both the leading eigenvector of the gene expression covariance matrix (left) and the leading left singular vector of the 6779×11171 term document matrix (right).

Numerical results: Averaging





Left: Alignment $u^T v$ between the true leading eigenvector u and the normalized average leading eigenvector versus number of samples, on the gene expression covariance matrix with subsampling probability $p = 10^{-2}$.

Right: Alignment $u^T v$ for various values of the spectral gap $\frac{\lambda_2}{\lambda_1} = 0.75, 0.95, 0.99$.

Numerical results: Ranking

Suppose we are given an adjacency matrix for a **web graph**

$$A_{ij} = 1, \quad \text{if there is a link from } i \text{ to } j$$

with $A \in \mathbf{R}^{n \times n}$. We normalize it into a stochastic matrix

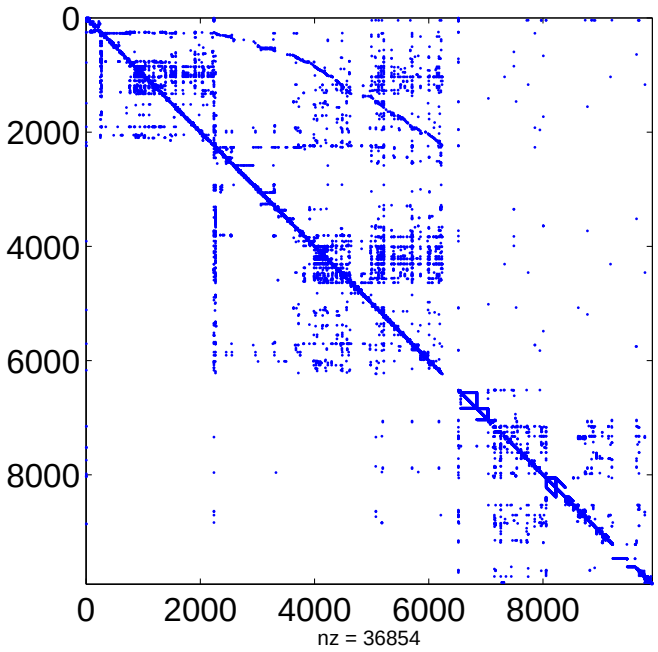
$$P^g_{ij} = \frac{A_{ij}}{\deg_i}$$

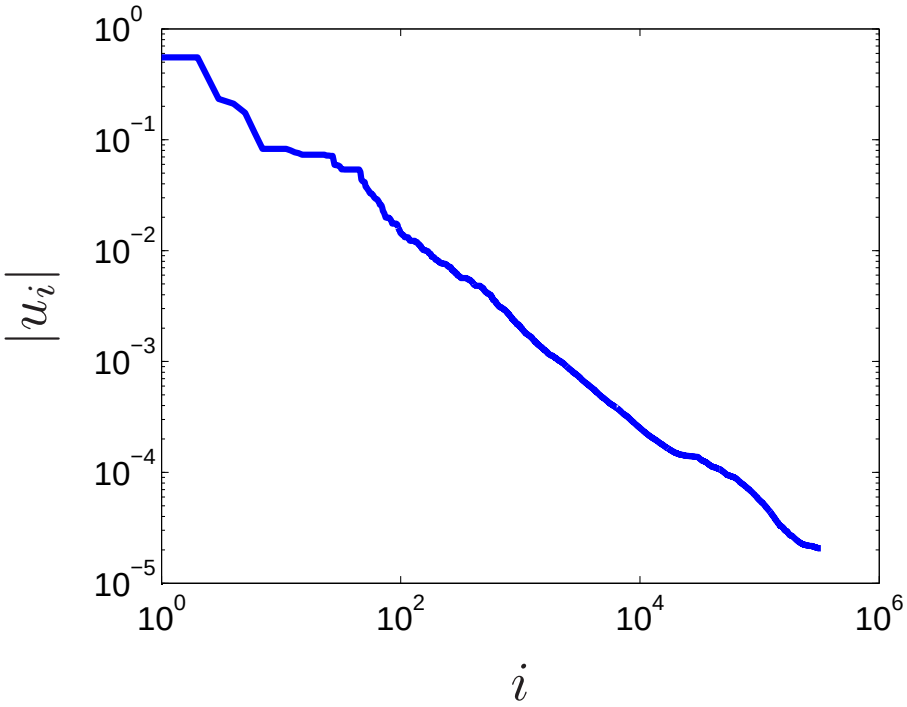
- The matrix P is the transition matrix of a Markov chain on the graph. If we set

$$P = cP^g + (1-c)\mathbf{1}\mathbf{1}^T/n$$

  for $c \in [0, 1]$, this Markov chain will be irreducible.
- The leading (Perron-Frobenius) eigenvector of this matrix is called the **Pagerank** vector (see Page, Brin, Motwani & Winograd (1998)).
- The spectral gap is at least c. . .

Numerical results: Ranking





Left: The `wb-cs.stanford` graph (9914 nodes and 36854 edges).

Right: Loglog plot of the Pagerank vector coefficients for the `cnr-2000` graph (325,557 nodes and 3,216,152 egdes).

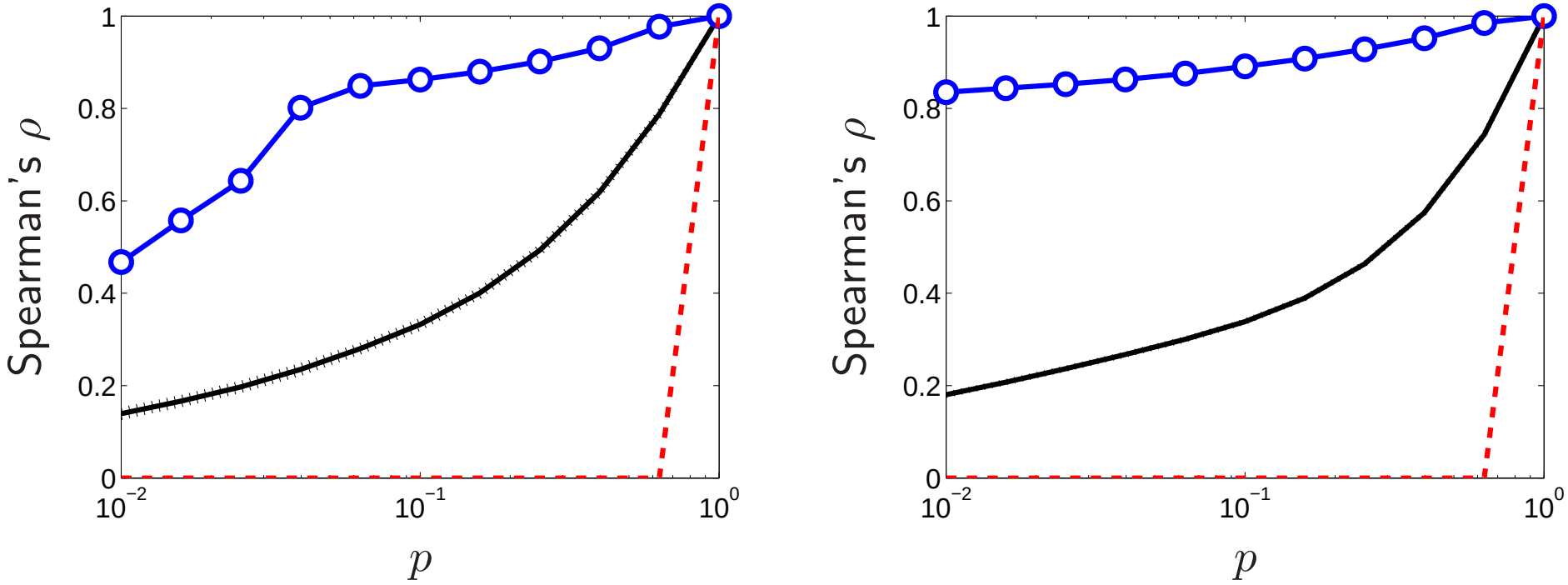
Numerical results: Ranking

Only the order of the coefficients matters in ranking, so measuring Pearson correlation between pagerank vectors is pointless. For ranking performance, we can use **Spearman's** ρ.

- Suppose we get two pagerank vectors $u, v \in \mathbf{R}^n$, we first convert them into ranking vectors $x, y \in [1, n]^n$ such that u_{x_i}, u_{y_i}, $i = 1, \ldots, n$ are both decreasing.
- We then compute the Pearson correlation between the ranking vectors

$$\rho(u, v) = \mathbf{corr}(x, y)$$

Numerical results: Ranking



Ranking correlation between true and averaged pagerank vector (blue circles), median value of the correlation over all subsampled matrices (solid black line), proportion of samples satisfying the perturbation condition (dashed red line), for various values of the sampling probability p.

Left: On the `wb-cs.stanford` graph: 9914 nodes and 36854 edges.
Right: On the `cnr-2000` graph: 325,557 nodes and 3,216,152 edges.

Outline

- Introduction
- Subsampling
- **Semidefinite programming**

Semidefinite programming

Consider the following **spectral radius minimization** problem

$$\begin{array}{ll} \text{minimize} & \left\| \sum_{j=1}^{p} y_j A_j + C \right\|_2 - b^T y \\ \text{subject to} & y \in Q, \end{array}$$

in the variable $y \in \mathbf{R}^p$, with parameters $A_j \in \mathbf{S}_n$, for $j = 1, \ldots, p$, $b \in \mathbf{R}^p$ and $C \in \mathbf{S}_n$, where Q is a compact convex set.

- Subgradients here are given by leading eigenvectors.
- Can we use subsampling to reduce the cost of each iteration in a stochastic gradient algorithm, as in Juditsky et al. (2009)?

The answer is yes, using a slightly different subsampling method. . .

Semidefinite programming

Matrix multiplication algorithm in Drineas et al. (2006)

Matrix multiplication

Input: $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{n \times p}$ and s such that $1 \leq s \leq n$.

1: Define a probability vector $p \in \mathbf{R}^n$ such that

$$p_i = \frac{\|A^{(i)}\|_2 \|B_{(i)}\|_2}{\sum_{j=1}^n \|A^{(j)}\|_2 \|B_{(j)}\|_2}, \quad i = 1, \ldots, n.$$

2: Define subsampled matrices $C \in \mathbf{R}^{m \times s}$ and $R \in \mathbf{R}^{s \times p}$ as follows.
3: **for** $i = 1$ to s **do**
4: Pick $j \in [1, n]$ with $\mathbf{P}(j = l) = p_l$.
5: Set $C^{(i)} = A^{(j)}/\sqrt{sp_j}$ and $R_{(i)} = B_{(j)}/\sqrt{sp_j}$.
6: **end for**

Output: Matrix product CR approximating AB.

Semidefinite programming

- By construction
$$\mathbf{E}[CR] = AB$$
- This algorithm can produce low-rank approximations of $X \in \mathbf{R}^{m \times n}$ such that
$$\|SS^T - XX^T\|_F$$
is small, where $S \in \mathbf{R}^{m \times s}$ is a scaled submatrix of X.
- Computing singular values of S using iterative methods is much faster because of its size.

Semidefinite programming

Low-rank approximation algorithm in Drineas et al. (2006)

Low rank approximation

Input: $X \in \mathbf{R}^{m \times n}$ and k, s such that $1 \leq k \leq s < n$.

1: Define a probability vector $p \in \mathbf{R}^n$ such that $p_i = \|X^{(i)}\|_2^2 / \|X\|_F^2$, for $i = 1, \ldots, n$.
2: Define a subsampled matrix $S \in \mathbf{R}^{m \times s}$ as follows.
3: **for** $i = 1$ to s **do**
4: Pick an index $j \in [1, n]$ with $\mathbf{P}(j = l) = p_l$.
5: Set $S^{(i)} = X^{(j)} / \sqrt{s p_j}$.
6: **end for**
7: Form the eigenvalue decomposition $S^T S = Y \, \mathbf{diag}(\sigma) Y^T$ where $Y \in \mathbf{R}^{s \times s}$ and $\sigma \in \mathbf{R}^s$.
8: Form a matrix $H \in \mathbf{R}^{m \times k}$ with $H^{(i)} = S Y^{(i)} / \sigma_i^{1/2}$.

Output: Approximate singular vectors $H^{(i)}$, $i = 1, \ldots, k$.

Semidefinite programming

Error bounds. *Let $X \in \mathbf{R}^{m\times n}$ and $\beta \in [0,1]$. Given a precision target $\epsilon > 0$, construct a matrix $S \in \mathbf{R}^{m\times s}$ by subsampling the columns of X. Let $\eta = 1 + \sqrt{8\log(1/\beta)}$ and*

$$s = \eta^2 \frac{\|X\|_2^2}{\epsilon^2} \mathbf{NumRank}(X)^2 \tag{1}$$

we have

$$\mathbf{E}[|\|S\|_2 - \|X\|_2|] \leq \epsilon$$

and

$$|\|S\|_2 - \|X\|_2| \leq \epsilon$$

with probability at least $1-\beta$.

Semidefinite programming

Let's come back to the spectral radius minimization problem

$$\min_{y \in Q} f(y) \equiv \mathbf{E}\left[\left\|\pi^{(s)}\left(\sum_{j=1}^{p} y_j A_j + C\right)\right\|_2\right] - b^T y$$

in the variable $y \in \mathbf{R}^p$ and parameters $A_j \in \mathbf{S}_n$, for $j = 1, \ldots, p$, $b \in \mathbf{R}^p$ and $C \in \mathbf{S}_n$, with $1 \leq s \leq n$ controlling the sampling rate

- For $X \in \mathbf{S}_n$, we have written $\pi^{(s)}(X)$ the subsampling/scaling operation

$$\pi^{(s)}(X) = S$$

  where $0 < s < n$ controls the sampling rate and $S \in \mathbf{R}^{n \times s}$.
- The function $\|\pi^{(s)}(\sum_{j=1}^{p} y_j A_j + C)\|_2$ and a subgradient with respect to y are computed using subsampling.

Spectral norm minimization using subsampling

Input: Matrices $A_j \in \mathbf{S}_n$, for $j = 1, \ldots, p$, $b \in \mathbf{R}^p$ and $C \in \mathbf{S}_n$, sampling rates s_1 and s_2.

1: Pick initial $y_0 \in Q$
2: **for** $l = 1$ to N **do**
3: Compute $v \in \mathbf{R}^n$, the leading singular vector of the matrix $\pi^{(s_1)}(\sum_{j=1}^p y_{l,j} A_j + C)$, subsampled with $k = 1$ and $s = s_1$.
4: Compute the approximate subgradient $g_l = \pi^{(s_2)}(\mathcal{A}^T)\ \pi_{(s_2)}(\mathbf{vec}(vv^T)) - b$, by subsampling the matrix product with $s = s_2$.
5: Set $y_{l+1} = P_{y_l}^{Q,\omega}(\gamma_l g_l)$.
6: Update the running average $\tilde{y}_N = \sum_{k=0}^N \gamma_l y_l / \sum_{k=0}^N \gamma_l$.
7: **end for**

Output: An approximate solution $\tilde{y}_N \in \mathbf{R}^p$ with high probability.

Semidefinite programming

Complexity. . .

- On line 3: Computing the **leading singular vector** v.
  - Computing the probabilities p_i at a cost of $c_1 n^2$ operations.
  - Forming the matrix $S = \pi^{(s_1)}(\sum_{j=1}^{p} y_{l,j} A_j + C)$ costs pns_1 flops.
  - Computing the leading singular vector of S using the Lanczos method at a cost of $c_2 n s_1$.

  Here $c_1 \ll c_2$, as c_2 is the number of iterations in the Lanczos method.

- On line 4: Computing the **approximate subgradient** $g_l = \pi^{(s_2)}(\mathcal{A}^T)\ \pi_{(s_2)}(\mathbf{vec}(vv^T)) - b$, by subsampling the matrix product.
  - This means forming the vector p at a cost of $O(n^2)$.
  - Computing the subsampled matrix vector product then costs $O(ps_2)$.

  Both of these complexity bounds have low constants.

Semidefinite programming

Complexity	Stoch. Approx. with Subsampling
Per Iter.	$c_1 n^2 + c_3 p s_2 + c_2 n \eta^2 \frac{\|Y^*\|_2^2}{\epsilon^2} \mathbf{NumRank}(Y^*)^2 + c(p)$
Num. Iter.	$\frac{2 D_{\omega,Q}^2 \delta^*(p)^2 \left(\frac{\|\mathcal{A}\|_F^2}{s_2} + \|b\|_2^2 \right)}{\alpha \epsilon^2 \beta^2}$

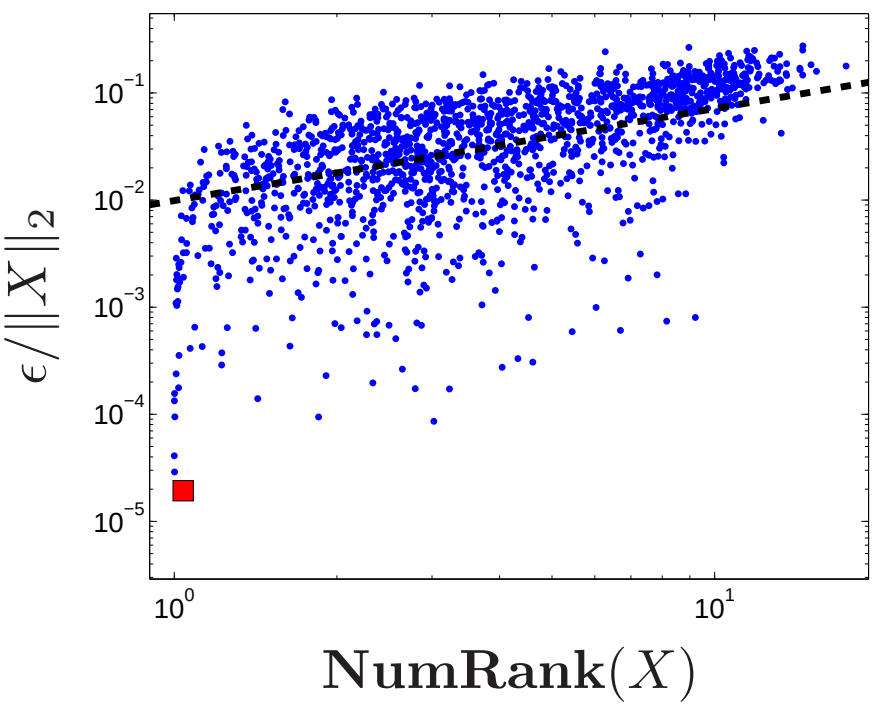
Complexity of solving min. spectral norm problem using subsampled stochastic approximation method versus original algorithm. Here $c_1, \ldots, c_4$ are absolute constants with $c_1, c_3 \ll c_2, c_4$.

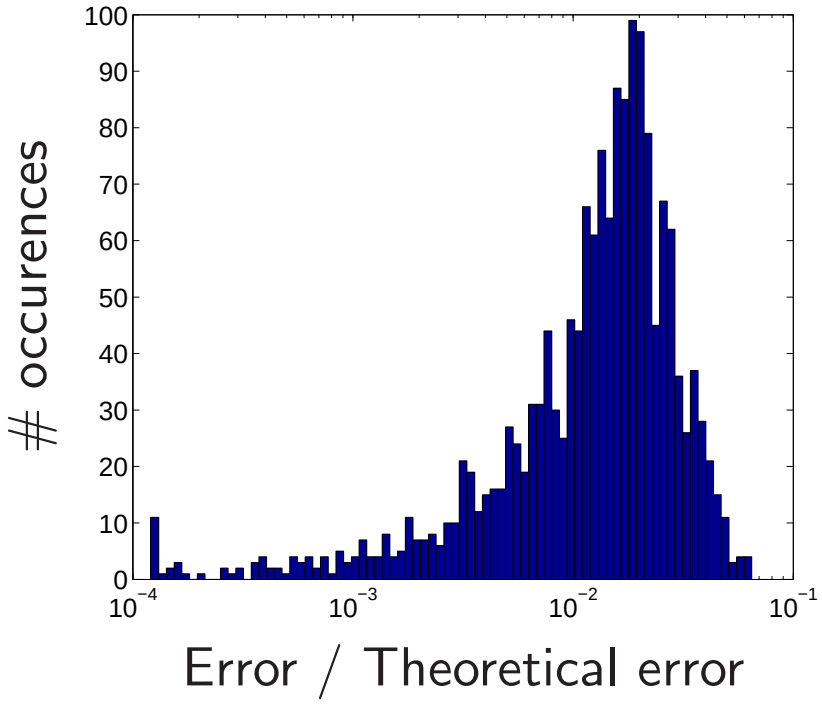
- Here, the main term in this complexity estimate is

$$n \eta^2 \frac{\|Y^*\|_2^2}{\epsilon^2} \mathbf{NumRank}(Y^*)^2$$

  with $Y^* = \sum_{j=1}^p y_j^* A_j + C$. This means that the complexity of the algorithm grows as the square of the **complexity of the solution**.
- The optimal susbampling rate is not known a priori, but if duality gap can be computed cheaply, getting a good s implies at most $\log_2 n$ restarts.

Semidefinite programming





Left: Loglog plot of relative error $\epsilon/\|X\|_2$ versus numerical rank $\mathbf{NumRank}(X)$ with 20% subsampling and $n = 500$ on random matrices (blue dots) and gene expression covariance (red square). The dashed line has slope one in loglog scale.

Right: Histogram plot in semilog scale of relative error $\epsilon/\|X\|_2$ over theoretical bound $\eta\,\mathbf{NumRank}(X)/\sqrt{s}$ for random matrices with $n = 500$.

Semidefinite programming

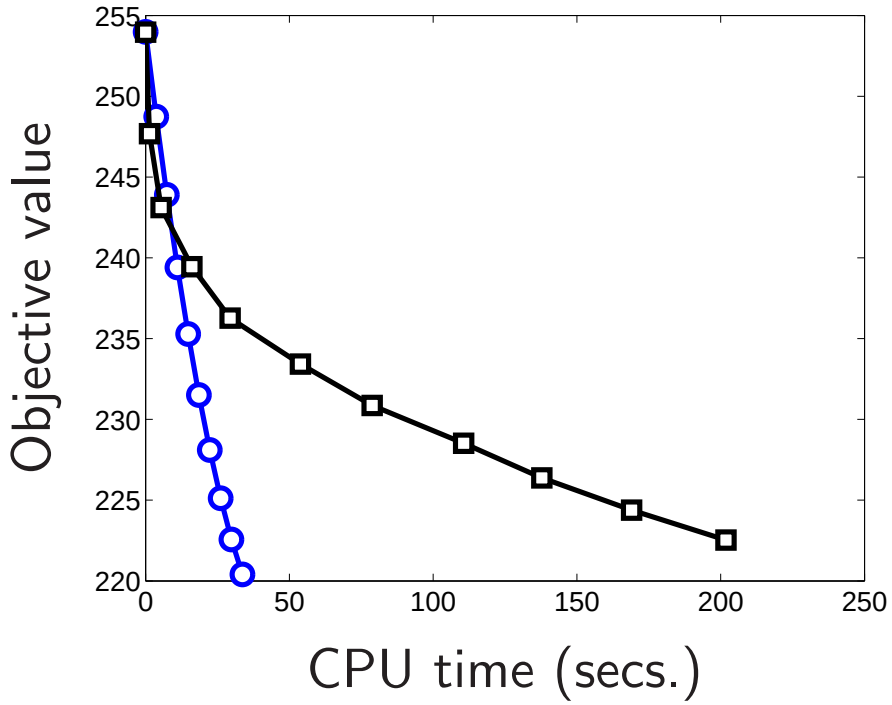
n	Deterministic	Subsampling	Speedup factor
500	5	5	0.92
750	19	13	1.40
1000	32	24	1.31
1500	107	58	1.84
2000	281	120	2.34

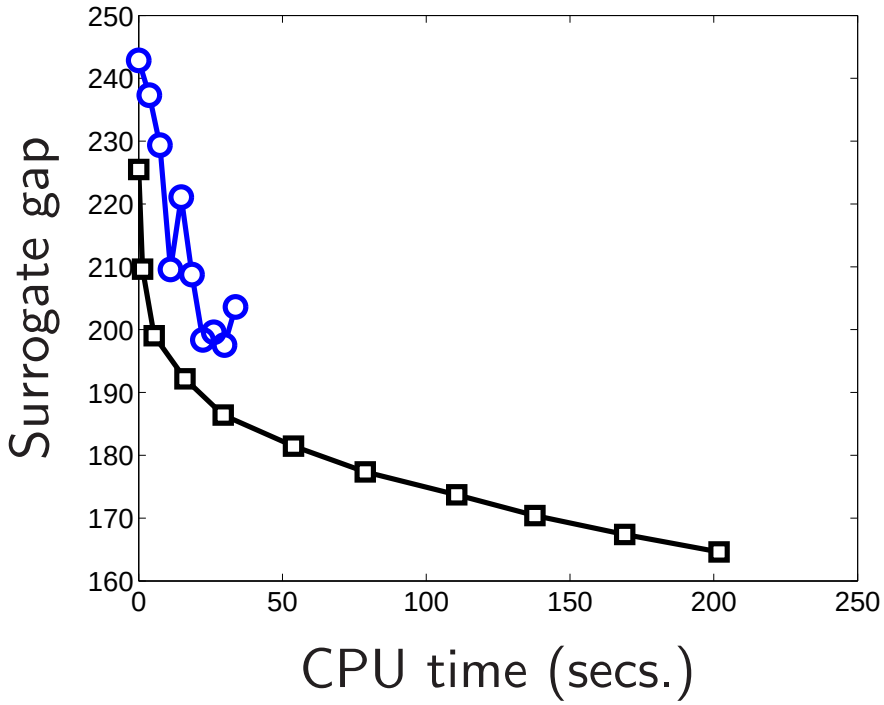
CPU time (in seconds) versus problem dimension n for deterministic and subsampled stochastic approximation algorithms on spectral norm minimization problems.

n	Deterministic	Subsampling	Speedup factor
100	154	23	6.67
200	766	63	12.2
500	4290	338	12.7

Median CPU time (in seconds) versus problem dimension n for deterministic and subsampled stochastic approximation algorithms on collaborative filtering problems.

Semidefinite programming





Left: Objective value versus CPU for a sample matrix factorization problem in dimension 100, using a deterministic gradient (squares) or a subsampled gradient with subsampling rate set at 20% (circles).

Right: Surrogate duality gap versus CPU time on the same example.

Conclusion

- Subsampling/averaging works surprisingly well, in particular the perturbation regime often holds for low sampling rates.
- Can be used to lower cost per iteration of stochastic gradient algorithms for semidefinite optimization.

What next?

- Reduce the complexity/memory requirements of smooth optimization methods.
- Volume sampling produces **relative accuracy** sampling bounds

$$\|A - S_k\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$$

  But computing the sampling probabilities is a hard combinatorial problem. See Kannan & Vempala (2009) for details.

Slides, paper, source code, binaries available at

`www.princeton.edu/~aspremon`

References

Achlioptas, D. & McSherry, F. (2007), 'Fast computation of low-rank matrix approximations', *Journal of the ACM* **54**(2).

Alon, A., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. & Levine, A. J. (1999), 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Cell Biology* **96**, 6745–6750.

Arora, S. & Kale, S. (2007), A combinatorial, primal-dual approach to semidefinite programs, *in* 'Proceedings of the thirty-ninth annual ACM symposium on Theory of computing', pp. 227–236.

Bertsimas, D. & Vempala, S. (2004), 'Solving convex programs by random walks', *J. ACM* **51**(4), 540–556.

Candes, E. & Recht, B. (2008), 'Exact matrix completion via convex optimization', *preprint* .

Candes, E. & Tao, T. (2009), 'The Power of Convex Relaxation: Near-Optimal Matrix Completion', *arXiv:0903.1476* .

Donoho, D. & Tsaig, Y. (2006), 'Fast solution of ℓ_1-norm minimization problems when the solution may be sparse', *Preprint* .

Drineas, P., Kannan, R. & Mahoney, M. (2006), 'Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix', *SIAM Journal on Computing* **36**, 158.

Frieze, A., Kannan, R. & Vempala, S. (2004), 'Fast monte-carlo algorithms for finding low-rank approximations', *Journal of the ACM (JACM)* **51**(6), 1025–1041.

Groh, D. J., Marshall, R. A., Kunz, A. B. & Givens, C. R. (1991), 'An approximation method for eigenvectors of very large matrices', *Journal of Scientific Computing* **6**(3), 251–267.

Horn, R. & Johnson, C. (1991), *Topics in matrix analysis*, Cambridge university press.

Huang, L., Yan, D., Jordan, M. & Taft, N. (2008), 'Spectral Clustering with Perturbed Data', *Advances in Neural Information Processing Systems (NIPS)* .

Juditsky, A., Lan, G., Nemirovski, A. & Shapiro, A. (2009), 'Stochastic approximation approach to stochastic programming', *SIAM Journal on Optimization* **19**(4), 1574–1609.

Juditsky, A. & Nemirovski, A. (2008), 'On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization', *ArXiv:0809.2650* .

Kannan, R. & Vempala, S. (2009), *Spectral algorithms*.
URL: *http://www.cc.gatech.edu/∼vempala/spectralbook.html*

Kato, T. (1995), *Perturbation theory for linear operators*, Springer.

Keshavan, R., Montanari, A. & Oh, S. (2009), 'Matrix Completion from a Few Entries', *arXiv:0901.3150* .

Ng, A., Zheng, A. & Jordan, M. (2001), Stable algorithms for link analysis, *in* 'ACM SIGIR', ACM New York, NY, USA, pp. 258–266.

Page, L., Brin, S., Motwani, R. & Winograd, T. (1998), 'The pagerank citation ranking: Bringing order to the web', *Stanford CS Technical Report* .

Papadimitriou, C., Raghavan, P., Tamaki, H. & Vempala, S. (2000), 'Latent semantic indexing: a probabilistic analysis', *Journal of Computer and System Sciences* **61**(2), 217–235.

Recht, B., Fazel, M. & Parrilo, P. (2007), 'Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization', *Arxiv preprint arXiv:0706.4138* .

Vu, V. (2007), 'Spectral norm of random matrices', *Combinatorica* **27**(6), 721–736.

Semidefinite programming

Trace norm minimization

$$\begin{array}{ll} \text{minimize} & \left\| \sum_{j=1}^{p} y_j A_j + C \right\|_{\text{tr}} - b^T y \\ \text{subject to} & y \in Q, \end{array}$$

in the variable $y \in \mathbf{R}^p$ where Q is a low dimension norm ball for example and the matrices A_j have a block format with only a few nonzero coefficients.

The sampling rate is written

$$s = \eta^2 \frac{\|Y^*\|_{\text{tr}}^2}{\epsilon^2} \kappa(Y^*)^2 \, \mathbf{Rank}(Y^*)$$

Simple solutions mean lower complexity (observed empirically in Donoho & Tsaig (2006) for the LASSO).