Subsampling Algorithms for Semidefinite Programming

Alexandre d'Aspremont
Princeton University

Support from NSF, DHS and Google.

Introduction

Focus on the following problem:

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(A^T y + c) - b^T y \\ \text{subject to} & y \in Q \end{array}$$

Sampling techniques

- Approximate leading eigenvalues and spectral radius with complexity $O(n)$.
- Smooth $\lambda_{\max}(X)$ by sampling gradients.

Stochastic Optimization

- Stochastic gradient algorithm using subsampling.
- Smooth optimization with approximate gradient.

First order algorithm

Complexity options. . .

$O(n)$	$O(n)$	$O(n^2)$	
			Memory
$O(1/\epsilon^2)$	$O(1/\epsilon)$	$O(\log(1/\epsilon))$	
First-order	Smooth	Newton IP	**Complexity**

Introduction

Simple illustrative example from Achlioptas & Mcsherry (2007). . .

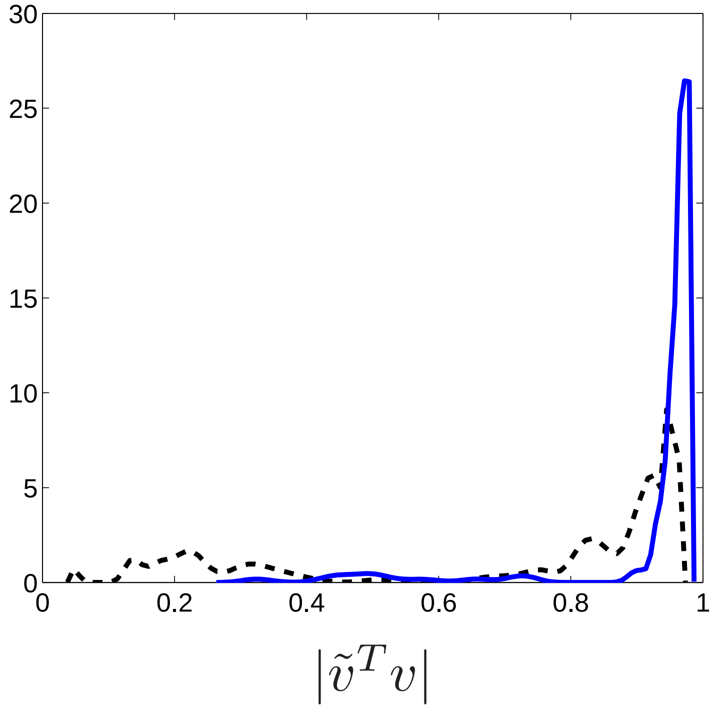
Given $p \in [0,1]$ and a symmetric matrix $A \in \mathbf{S}_n$, define:

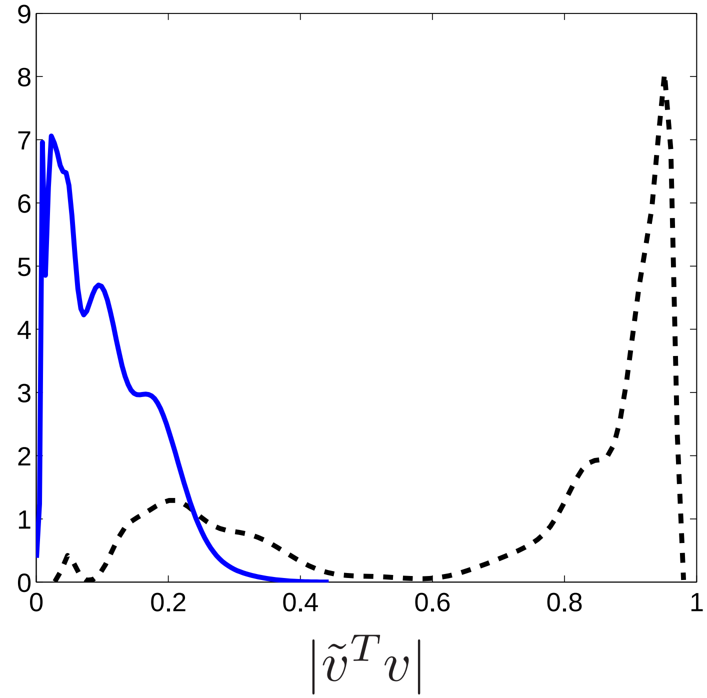
$$\tilde{A}_{ij} = \begin{cases} A_{ij}/p & \text{with probability p} \\ 0 & \text{otherwise} \end{cases}$$

- By construction, $\tilde{A}$ has mean A with independent coefficients.
- Sparse: $\tilde{A}$ has $O(pn^2)$ nonzero entries on average.

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

Introduction

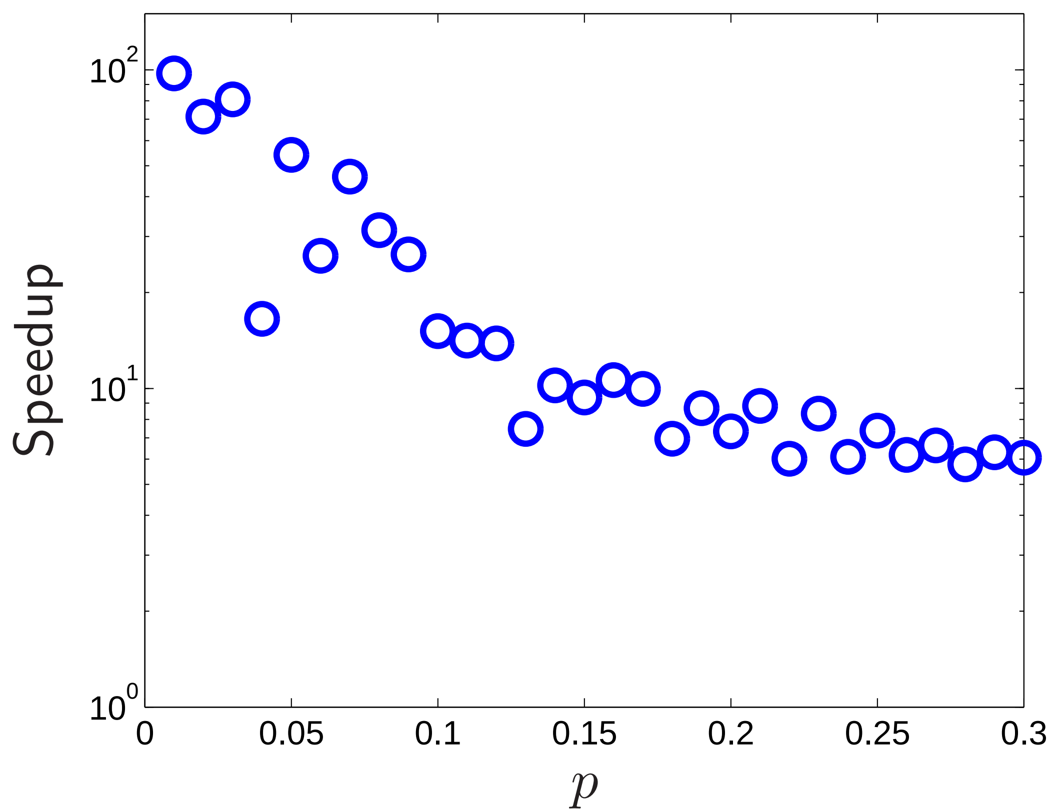




Left: Distribution of $|\tilde{v}^T v|$, with v the leading eigenvector of a structured covariance matrix and $\tilde{v}$ that of the randomly subsampled matrix, with $p = .25$ (solid line) and $p = .15$ (dashed line).

Right: Distribution of $|\tilde{v}^T v|$ where $\tilde{v}$ is computed on a subsampled matrix with $p = .15$, using a structured matrix (dashed line) or a Wishart matrix (solid line).

Introduction



Ratio of average CPU time for computing the leading eigenvalue of a (sparse) subsampled matrix using ARPACK over average CPU time for computing the leading eigenvalue of the original (dense) matrix, for various values of the sampling rate p on a covariance matrix of dimension 2000.

Outline

- Introduction
- **Sampling techniques**
  - Subsampling
  - Smoothing
- Stochastic Optimization
  - Stochastic gradient
  - Smooth Optimization with Approximate Gradient.
- Numerical Experiments

Elementwise subsampling

Given $X \in \mathbf{S}_n$ and $\epsilon > 0$, define:

$$\tilde{X}_{ij} = \begin{cases} X_{ij}/p & \text{with probability } p, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose we set

$$p_\epsilon = \min\left\{1, \frac{16n\|X\|_\infty^2}{\epsilon^2}\right\} \tag{1}$$

then $\lambda_{\max}(X) \leq \mathbf{E}[\lambda_{\max}(\tilde{X})]$ and we have

$$\|X - \tilde{X}\|_2 \leq \epsilon \quad \text{and} \quad \lambda_{\max}(\tilde{X}) - \lambda_{\max}(X) \leq \epsilon,$$

with probability at least $1 - \exp(-19(\log n)^4)$. The average number of nonzero coefficients in $\tilde{X}$ is bounded by:

$$\frac{16n\|X\|_F^2}{\alpha\epsilon^2} \,\mathrm{mean}\left(\left\{\frac{\|X\|_\infty^2}{X_{[i]}^2}\right\}_{i=1,\ldots,\lceil \alpha n^2 \rceil}\right)$$

for any $\alpha \in [0, 1]$.

Columnwise subsampling

Another procedure from Drineas, Kannan & Mahoney (2006).

Let $X \in \mathbf{S}_n$ and $0 < k \leq s < n$. Define $p_i = \|X_i\|^2 / \|X\|_F^2$, for $i = 1, \ldots, n$. Pick $i_t \in [1, n]$ with $\mathbf{P}(i_t = u) = p_u$ for $t = 1, \ldots, s$ and define a matrix $C \in \mathbf{R}^{m \times s}$ with

$$C_t = \frac{X_{i_t}}{\sqrt{s p_{i_t}}}$$

Form the singular value decomposition of $C^T C = Y \Sigma Y^T$ and let

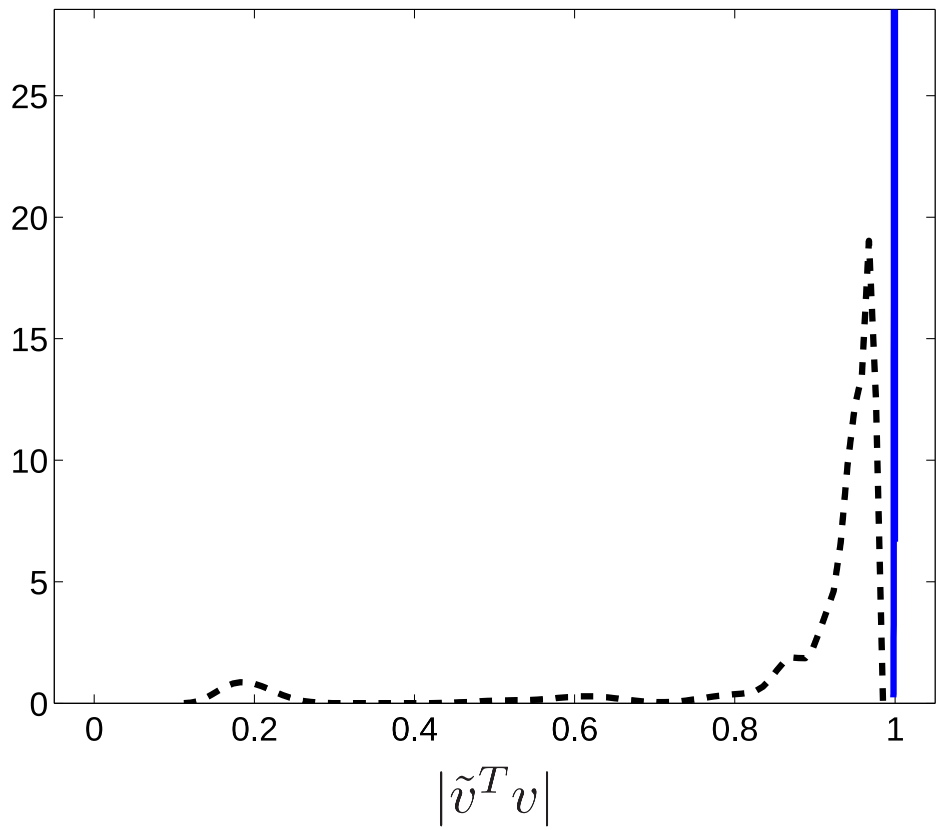
$$H_k = C Y_{[1,k]} \Sigma_{[1,k]}^{-1/2},$$

then for a given precision target $\epsilon > 0$ and if $s \geq 4/\epsilon^2$ we have

$$\mathbf{E}[\|X - H_k H_k^T X\|_2^2] \leq \|X - X_k\|_2^2 + \epsilon \|X\|_F^2$$

where X_k is the best rank k approximation of X.

Columnwise subsampling



Distribution of the scalar product $|\tilde{v}^T v|$ where v is the leading eigenvector of a structured covariance matrix and $\tilde{v}$ is the leading eigenvector of the subsampled matrix, using the elementwise subsampling procedure (dotted line) and the columnwise procedure (continuous line), with a subsampling rate of 20% in both cases. Maximum eigenvalue and spectral radius coincide on this covariance matrix.

Smoothing by gradient sampling

Let $U \in \mathbf{S}_n$, with $U_{ij} \sim \mathcal{N}(0, \sigma/\sqrt{2})$ for $i \neq j$ and $U_{ii} \sim \mathcal{N}(0, \sigma)$.

$$f(X) = \mathbf{E}[\lambda_{\max}(X + U)] \tag{2}$$

with $X \in Q$, satisfies

$$\lambda_{\max}(X) \leq f(X) \leq \lambda_{\max}(X) + 2\sigma n^{1/2+\nu}$$

for any $\nu > 0$. Its gradient is Lipschitz continuous on Q with Lipschitz constant:

$$L = \frac{2(M_Q + D_{F,Q})}{\sigma^2} + \frac{3(e^\gamma n(n+1))^{1/2}}{\sigma}$$

with $\gamma = 0.577...$ the Euler-Mascheroni constant,

$$M_Q = \max_{X \in Q} \|X\|_F \quad \text{and} \quad D_{F,Q} = \max_{X,Y \in Q} \|X - Y\|_F,$$

with $D_{F,Q}$ the Euclidean diameter of Q.

Outline

- Introduction
- Sampling techniques
    - Subsampling
    - Smoothing
- **Stochastic Optimization**
    - Stochastic gradient
    - Smooth Optimization with Approximate Gradient.
- Numerical Experiments

Complexity

- Stochastic gradient algorithm in Juditsky, Lan, Nemirovski & Shapiro (2007) with subsampled iterates.
- Smooth optimization algorithm in d'Aspremont (2005) using smooth sampled gradient.

	Iterations	Cost per Iteration
Subsampled Stochastic Grad.	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{n}{\epsilon^2}\right)$
Stochastic Gradient	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n^2\right)$
Smooth Opt. with Grad. Sampling	$O\left(\frac{1}{\epsilon^{3/2}}\right)$	$O\left(\frac{n^2}{\epsilon^2}\right)$
Smooth Optimization	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n^3\right)$

Stochastic Gradient Algorithm

Stochastic gradient algorithm.

Starting from $y_0 \in Q$. For $k = 0, \ldots, N-1$,

1. Set $y_{k+1} = \pi_{y_k}^{Q,\omega}(\gamma_k g_k)$, where $g_k \in \partial \lambda^{\max}(\tilde{A}^T y + \tilde{c}) - b$.

2. Set $\bar{y}_N = \sum_{k=0}^{N-1} \gamma_k y_k / \sum_{k=0}^{N-1} \gamma_k$.

Stochastic Gradient Algorithm

Given $\epsilon > 0$, and a sampling rate $p \in [0, 1]$. Suppose that p satisfies:

$$p \geq \frac{16\|X\|_F^2}{n\alpha\epsilon^2} \,\mathrm{mean}\left(\left\{\frac{\|X\|_\infty^2}{X_{[i]}^2}\right\}_{i=1,\ldots,\lceil\alpha n^2\rceil}\right) \tag{3}$$

for $X = A^T y^* - c$ and some $\alpha \in [0, 1]$, then after

$$N = \frac{4M_*^2 D_{\omega,Q}^2}{\alpha\epsilon^2(1-\beta)^2}$$

iterations, the stochastic gradient algorithm with constant step size $\gamma = \alpha\epsilon/\sqrt{2}M_*^2$ will produce an iterate in problem satisfying:

$$\mathbf{P}[f(\bar{y}_N) - f(y^*) \geq \epsilon] \leq (1-\beta) + \exp(-19(\log n)^4).$$

The average number of nonzero coefficients in $\tilde{X}$ is pn^2.

Smooth Optimization

Smooth minimization with approximate gradient.

Starting from x_0, the prox center of the set Q, we iterate:

1. compute $\tilde{\nabla} f(x_k)$,
2. compute $y_k = \mathrm{argmin}_{y \in Q}\{\langle \tilde{\nabla} f(x), y - x\rangle + \frac{1}{2}L\|y-x\|^2\}$,
3. compute $z_k = \mathrm{argmin}_{x \in Q}\{\frac{L}{\eta}d(x) + \sum_{i=0}^{k}\alpha_i[f(x_i) + \langle \tilde{\nabla} f(x_i), x - x_i\rangle]\}$,
4. update x using $x_{k+1} = \tau_k z_k + (1-\tau_k)y_k$,

Smooth Optimization

Consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \lambda^{\max}(A^T y + c) - b^T y \\ \text{subject to} & y \in Q, \end{array}$$

in the variable $y \in \mathbf{R}^m$, with parameters $A \in \mathbf{R}^{m \times n^2}$, $b \in \mathbf{R}^m$ and $c \in \mathbf{R}^{n^2}$.

Let $U \in \mathbf{S}_n$ be a random symmetric matrix with Gaussian coefficients $U_{ij} \sim \mathcal{N}(0, \sigma/\sqrt{2})$ for $i \neq j$ and $U_{ii} \sim \mathcal{N}(0, \sigma)$. Let

$$f(y) = \mathbf{E}[\lambda_{\max}(\mathrm{mat}(A^T y + c) + U)]$$

and suppose we sample k matrices U_i as above to define:

$$\tilde{\nabla} f(y) = \frac{1}{k} \sum_{i=1}^{k} \nabla \lambda_{\max}(\mathrm{mat}(A^T y + c) + U_i)$$

where $\epsilon > 0$ is the target precision.

Smooth Optimization

Then, with probability $1-\beta$, the smooth optimization algorithm will produce a 2ϵ solution in at most:

$$N(n,\epsilon)=\frac{4\|A\|_{2,2}d(y^*)^{1/2}}{\epsilon}\left(\frac{8n(M_Q+D_{F,Q})n^{2\nu}}{\tau\epsilon}+\frac{6e^{\gamma}(n+1)n^{1+\nu}}{\tau}\right)^{1/2}$$

iterations, having defined:

$$M_Q=\max_{X\in Q}\|X\|_F \quad \text{and} \quad D_{F,Q}=\max_{X,Y\in Q}\|X-Y\|_F,$$

where $D_{F,Q}$ is the Euclidean diameter of Q, provided that:

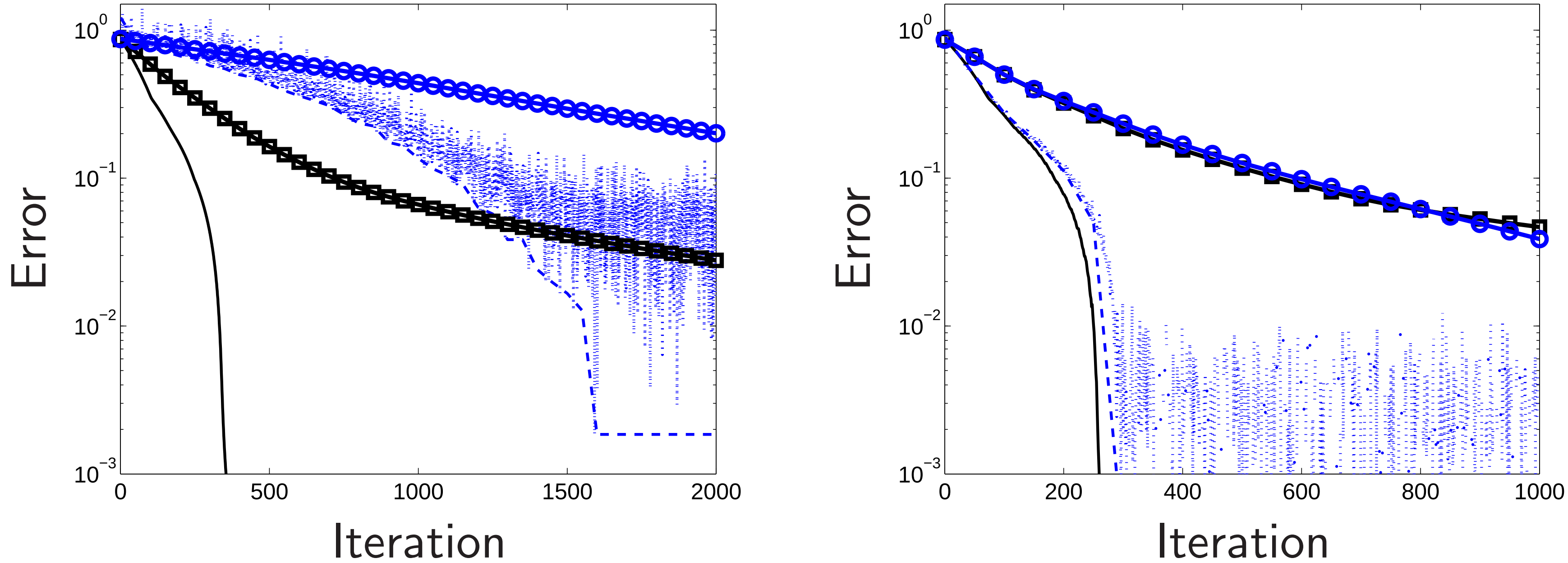
$$k\geq\frac{m\|A\|_F^2}{\epsilon^2}\log\left(\frac{mN(n,\epsilon)}{\beta}\right)$$

with each iteration requiring k maximum eigenvalue computations.

Outline

- Introduction
- Sampling techniques
  - Subsampling
  - Smoothing
- Stochastic Optimization
  - Stochastic gradient
  - Smooth Optimization with Approximate Gradient.
- **Numerical Experiments**

Stochastic Gradient



Left: Current distance to optimality for the averaged iterates of the stochastic gradient algorithm with exact gradients (squares) and elementwise subsampled gradients (circles) with a $p = .2$ sampling rate on a maximum eigenvalue minimization problem of dimension 2000.
Right: Same plot on a spectral radius minimization problem, using exact gradients (squares) and columnwise subsampled gradients (circles) with a 20% sampling rate.

References

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

d'Aspremont, A. (2005), 'Smooth optimization with approximate gradient', *ArXiv: math.OC/0512344* .

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.

Juditsky, A., Lan, G., Nemirovski, A. & Shapiro, A. (2007), 'Stochastic approximation approach to stochastic programming', *Working Paper* .