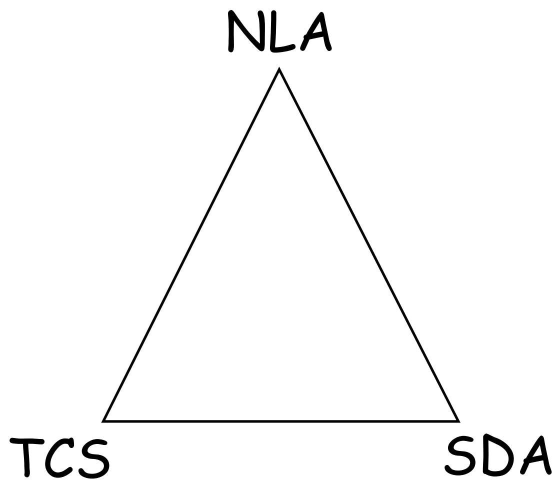
Statistical Leverage and Improved Matrix Algorithms

Michael W. Mahoney

Yahoo Research

(For more info, see: http://www.cs.yale.edu/homes/mmahoney)

Modeling data as matrices



Matrices often arise with data:

- n objects ("documents," genomes, images, web pages),
- each with m features,
- may be represented by an m x n matrix A.

Least Squares (LS) Approximation

$$\begin{pmatrix} \\ \\ A \\ \\ \\ \end{pmatrix} \begin{pmatrix} \hat{x} \end{pmatrix} \approx \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix}$$

$n \times d \quad , \quad n \gg d$

$$\begin{aligned} \mathcal{Z}_2 &= \min_{x \in \mathbb{R}^d} ||b - Ax||_2 \\ &= ||b - A\hat{x}||_2 \end{aligned}$$

We are interested in over-constrained L2 regression problems, *n* >> *d*.

- Typically, no *x* such that *Ax* = *b*.
- Want to find the "best" x such that *Ax* ≈ *b*.

Ubiquitous in applications & central to theory:

- Statistical interpretation: best linear unbiased estimator.
- Geometric interpretation: orthogonally project b onto span(A).

Exact solution to LS Approximation

Cholesky Decomposition:

If A is full rank and well-conditioned,

decompose $A^TA = R^TR$, where R is upper triangular, and

solve the normal equations: $R^TRx=A^Tb$.

QR Decomposition:

Slower but numerically stable, esp. if A is rank-deficient.

Write $A=QR$, and solve $Rx = Q^Tb$.

Singular Value Decomposition:

Most expensive, but best if A is very ill-conditioned.

Write $A=U\Sigma V^T$, in which case: $\mathbf{x_{OPT} = A^+b = V\Sigma^{-1}_k U^T b}$.

Complexity is O(nd²) for all of these, but constant factors differ.

$$\mathcal{Z}_2 = \min_{x \in R^d} ||b - Ax||_2$$
$$= ||b - A\hat{x}||_2$$

Projection of b on the subspace spanned by the columns of A

$$\mathcal{Z}_2^2 = ||b||_2^2 - ||AA^+b||_2^2$$
$$\hat{x} = A^+b$$

Pseudoinverse of A

LS and Statistical Modeling

Assumptions underlying its use:

- Relationship between "outcomes" and "predictors" is (approximately) linear.
- The error term ε has mean zero.
- The error term ε has constant variance.
- The errors are uncorrelated.
- The errors are normally distributed (or we have adequate sample size to rely on large sample theory).

Check to ensure these assumptions have not been (too) violated!

Statistical Issues and Regression Diagnostics

Statistical Model: $b = Ax+\varepsilon$

b = response; $A^{(i)}$ = carriers; ε = error process

$b' = A\,x_{opt} = A(A^TA)^{-1}A^Tb$

$H = A(A^TA)^{-1}A^T$ is the "hat" matrix, i.e. projection onto span(A)

Note: $H=UU^T$, where U is *any* orthogonal matrix for span(A)

Statistical Interpretation:

H_{ij} -- measures the leverage or influence exerted on b'_i by b_j,

H_{ii} -- leverage/influence score of the i-th constraint

Note: $H_{ii} = |U^{(i)}|_2^2$ = row "lengths" of spanning orthogonal matrix

Trace(H)=d -- Diagnostic Rule of Thumb: Investigate if $H_{ii} > 2d/n$

Overview

Statistical Leverage and the Hat Matrix

Faster Algorithms for Least Squares Approximation

Better Algorithm for Column Subset Selection Problem

Even better, both perform very well empirically!

An (expensive) LS sampling algorithm

Drineas, Mahoney, and Muthukrishnan (SODA, 2006)

Algorithm

1. Randomly sample r constraints according to probabilities p_i.
2. Solve the induced least-squares problem.

Theorem:

Let: $r = O\left(d \log(d) \log(1/\delta)/(\beta\epsilon^2)\right)$

If the p_i satisfy:

$$p_i \geq \frac{\beta \left\|U_{(i)}\right\|_2^2}{\sum_{i=1}^n \left\|U_{(i)}\right\|_2^2} = \frac{\beta \left\|U_{(i)}\right\|_2^2}{d}$$

p_i are statistical leverage scores!

$U_{(i)}$ are *any* orthogonal basis for span(A).

for some $\beta \in (0,1]$, then w.p. $\geq 1-\delta$,

$$\begin{aligned} ||A\tilde{x}_{opt} - b||_2 &\leq (1+\epsilon)\mathcal{Z}, \text{ and} \\ ||x_{opt} - \tilde{x}_{opt}||_2 &\leq \sqrt{\epsilon}\left(\kappa(A)\sqrt{\gamma^{-2}-1}\right)||x_{opt}||_2 \end{aligned}$$

A structural lemma

Drineas, Mahoney, Muthukrishnan, and Sarlos (2007)

Approximate: $\mathcal{Z} = \min_{x \in \mathbb{R}^d} ||Ax - b||_2$

by: $\tilde{\mathcal{Z}} = \min_{x \in \mathbb{R}^d} ||\mathcal{X}(Ax - b)||_2$

($\mathcal{X}$: *any* matrix.)

(U_A: *any* orthonormal basis for span(A).)

Lemma: Assume that: $\sigma_{min}(\mathcal{X}U_A) \geq 9/10$; and

$$||U_A^T \mathcal{X}^T \mathcal{X} b^{\perp}||_2^2 \leq \epsilon \mathcal{Z}^2/2$$

Then, we get "relative-error" approximation:

$$\begin{aligned} ||A\tilde{x}_{opt} - b||_2 &\leq (1+\epsilon)\mathcal{Z}, \text{ and} \\ ||x_{opt} - \tilde{x}_{opt}||_2 &\leq \sqrt{\epsilon}\left(\kappa(A)\sqrt{\gamma^{-2} - 1}\right)||x_{opt}||_2 \end{aligned}$$

A “fast” LS sampling algorithm

Drineas, Mahoney, Muthukrishnan, and Sarlos (2007)

Algorithm:

1. Pre-process A and b with a “*randomized Hadamard transform*”.
2. Uniformly sample $r = O\left(d \log(n) \log(d \log(n)/\epsilon)\right)$ constraints.
3. Solve the induced problem:

$$\mathcal{Z}_{2,s} = \min_{x \in \mathbb{R}^d} ||\mathcal{SH}(b - Ax)||_2 = ||\mathcal{SH}(b - A\hat{x})||_2$$

Main theorem:

- *(1±ε)-approximation*
- *in* $O\left(nd \log\left(d \log(n)/\epsilon\right) + d^3 \log(n) \log(d \log n)/\epsilon\right)$ *time!!*

Randomized Hadamard preprocessing

Facts implicit or explicit in: Ailon & Chazelle (2006), or Ailon and Liberty (2008).

H_n = *n-by-n deterministic* Hadamard matrix, and
D_n = *n-by-n* {+1/-1} *random* Diagonal matrix.

Fact 1: Multiplication by H_nD_n doesn't change the solution:

$$||Ax - b||_2 = ||H_n D_n A x - H_n D_n b||_2$$

Fact 2: Multiplication by H_nD_n is fast - only *O(n log(r)) time*, where r is the number of elements of the output vector we need to "touch".

Fact 3: Multiplication by H_nD_n approximately *uniformizes all leverage scores*:

$$||U_{(i)}{}_{H_n D_n A}||_2 = ||\,(H_n D_n U_A)_{(i)}\,||_2 \leq O\left(\sqrt{\frac{d \log n}{n}}\right)$$

Overview

Statistical Leverage and the Hat Matrix

Faster Algorithms for Least Squares Approximation

Better Algorithm for Column Subset Selection Problem

Even better, both perform very well empirically!

Column Subset Selection Problem (CSSP)

Given an m-by-n matrix *A* and a rank parameter k, choose *exactly k columns* of *A* s.t. the m-by-k matrix *C* minimizes the error over all O(n^k) choices for *C*:

$$\min ||A - P_C A||_2 = \min ||A - CC^+ A||_2,$$

$$\text{where } ||X||_2 = \max_{x \in \mathbb{R}^n : |x|=1} |Xx|$$

$$\min ||A - P_C A||_F = \min ||A - CC^+ A||_F,$$

$$\text{where } ||X||_F^2 = \sum_{ij} X_{ij}^2$$

Notes:

- $P_C = CC^+$ is the projector matrix onto span(*C*).
- The "best" rank-k approximation from the SVD gives a lower bound.
- Complexity of the problem? O($n^k mn$) trivially works; NP-hard if *k* grows as a function of *n*. (Civril & Magdon-Ismail '07)

Prior work in NLA

Numerical Linear Algebra algorithms for the CSSP

- Deterministic, typically greedy approaches.
- Deep connection with the Rank Revealing QR factorization.
- Strongest results so far (spectral norm): in *$O(mn^2)$* time

$$\|A - P_C A\|_2 \leq O(k^{1/2}(n-k)^{1/2}) \left\|A - P_{U_k} A\right\|_2$$

(more generally, some function p(k,n))

- Strongest results so far (Frobenius norm): in *$O(n^k)$* time

$$\|A - P_C A\|_F \leq \sqrt{k\,(n-k)}\,\|A - P_{U_k} A\|_2$$

Working on p(k,n): 1965 - today

Year	Reference	Authors	$p(k,n)$	Complexity	Software
1965	[23]	Golub	-	$O(mn^2)$	[13, 2, 31]
1986	[19]	Foster	-	$O(mn^2)$	[18]
1987	[7]	Chan	$\sqrt{(n-k)}\|\|W\|\|_2$	$O(mn^2)$	[18]
1990	[8]	Chan-Hansen	$\sqrt{n(n-k)}2^{n-k}$	$O(mn^2)$	[18]
1991	[3]	Bischof-Hansen	$\sqrt{n(n-k)}2^{n-k}$	$O(mn^2)$	-
1992	[27]	Hong-Pan	$\sqrt{k(n-k)+\min(k,n-k)}$	$O(n^k)$	-
1994	[10]	Chan-Hansen	$\sqrt{nk}2^k$	$O(mn^2)$	[18]
1994	[11]	Chandrasekaran-Ipsen	$\sqrt{(k+1)(n-k)}$	$O(n^k)$	-
1996	[25]	Gu-Eisenstat	$\sqrt{k(n-k)+1}$	$O(n^k)$	-
			$\sqrt{k(n-k)+1}$	$O(mn^2)$	-
1998	[6]	Bischof-Orti	-	$O(mn^2)$	[4]
		modification of [11]	$\sqrt{(k+1)(n-k)}$	$O(mn^2)$	[4, 20]
		modification of [30]	$\sqrt{(k+1)^2(n-k)}$	$O(mn^2)$	[4, 20]
1999	[30]	Pan-Tang	$\sqrt{(k+1)(n-k)}$	$O(mn^2)$	-
			$\sqrt{(k+1)^2(n-k)}$	$O(mn^2)$	-
			$\sqrt{(k+1)^2(n-k)}$	$O(mn^2)$	-
2000	[29]	Pan	$\sqrt{k(n-k)+1}$	$O(mn^2)$	-

Theoretical computer science contributions

<u>Theoretical Computer Science algorithms for the CSSP</u>

1. Randomized approaches, with some failure probability.

2. More than k columns are picked, e.g., O(poly(k)) columns chosen.

3. Very strong bounds for the Frobenius norm in low polynomial time.

4. Not many spectral norm bounds.

Prior work in TCS

Drineas, Mahoney, and Muthukrishnan 2005,2006 - *"subspace sampling"*

- $O(mn^2)$ time, $O(k^2/\varepsilon^2)$ columns -> $(1\pm\varepsilon)$-approximation.
- $O(mn^2)$ time, $O(k \log k/\varepsilon^2)$ columns -> $(1\pm\varepsilon)$-approximation.

Deshpande and Vempala 2006 - *"volume" and "iterative" sampling*

- $O(mnk^2)$ time, $O(k^2 \log k/\varepsilon^2)$ columns -> $(1\pm\varepsilon)$-approximation.
- They also prove the existence of *k* columns of *A* forming a matrix *C*, s.t.

$$\|A - P_C A\|_F \leq \sqrt{k}\|A - P_{U_k} A\|_F$$

- Compare to prior best existence result:

$$\|A - P_C A\|_F \leq \sqrt{k}\sqrt{n-k}\,\|A - A_k\|_2$$

The strongest Frobenius norm bound

Drineas, Mahoney, and Muthukrishnan (2006)

Theorem:

Given an m-by-n matrix A, there exists an $O(mn^2)$ algorithm that picks

at most $O(k \log k / \varepsilon^2)$ columns of A

such that with probability at least $1-10^{-20}$

$$\|A - P_C A\|_F \leq (1+\epsilon)\left\|A - P_{U_k} A\right\|_F$$

Algorithm:

Use subspace sampling probabilities /leverage score probabilities to sample $O(k \log k / \varepsilon^2)$ columns.

Subspace sampling probabilities

Subspace sampling probs:

in $O(mn^2)$ time, compute:

$$p_j = \frac{\left|\left(V_k^T\right)^{(i)}\right|^2}{k}$$

These p_i are statistical leverage scores!

$V_{k(i)}$ are *any* orthogonal basis for span(A_k).

NOTE: The rows of V_k^T are orthonormal, but its columns $(V_k^T)^{(i)}$ are not.

$$\underset{m \times n}{\Big(A_k \Big)} = \underset{m \times k}{\Big(U_k \Big)} \cdot \underset{k \times k}{\Big(\Sigma_k \Big)} \cdot \underset{k \times n}{\Big(V_k^T \Big)}$$

V_k: orthogonal matrix containing the top k right singular vectors of A.

Σ_k: diagonal matrix containing the top k singular values of A.

Other work bridging NLA/TCS

Woolfe, Liberty, Rohklin, and Tygert 2007

(also Martinsson, Rohklin, and Tygert 2006)

- O(mn log k) time, k columns
- Same spectral norm bounds as prior work
- Application of the Fast Johnson-Lindenstrauss transform of Ailon-Chazelle
- Nice empirical evaluation.

Question: *How to improve bounds for CSSP?*

- *Not obvious that bounds improve if allow NLA to choose more columns.*
- *Not obvious how to get around TCS need to over-sample to O(k log(k)) to preserve rank.*

A hybrid two-stage algorithm

Boutsidis, Mahoney, and Drineas (2007)

** Not so simple … Actually, run QR on the down-sampled k-by-O(k log k) version of V_k^T.*

Algorithm: Given an m-by-n matrix A and rank parameter k:

- (Randomized phase)

  Randomly select *c = O(k logk)* columns according to "leverage score probabilities".

- (Deterministic phase)

  Run a deterministic algorithm on the above columns* to pick exactly *k* columns of *A*.

Theorem: Let C be the m-by-k matrix of the selected columns. Our algorithm runs in $O(mn^2)$ and satisfies, w.p. $\geq 1-10^{-20}$,

$$\|A - P_C A\|_F \;\leq\; O\left(k \log^{1/2} k\right) \|A - P_{U_k} A\|_F$$

$$\|A - P_C A\|_2 \;\leq\; O\left(k^{3/4} \log^{1/2} k (n-k)^{1/4}\right) \|A - P_{U_k} A\|_2$$

Comparison: spectral norm

Our algorithm runs in $O(mn^2)$ and satisfies, with probability at least $1-10^{-20}$,

$$\|A - P_C A\|_2 \;\leq\; O\left(k^{3/4}\log^{1/2} k(n-k)^{1/4}\right) \|A - P_{U_k} A\|_2$$

1. Our running time is comparable with NLA algorithms for this problem.
2. Our spectral norm bound grows as a function of $(n-k)^{1/4}$ instead of $(n-k)^{1/2}$!
3. Do notice that with respect to k our bound is $k^{1/4}\log^{1/2}k$ worse than previous work.
4. To the best of our knowledge, **our result is the first asymptotic improvement of the work of Gu & Eisenstat** 1996.

Comparison: Frobenius norm

Our algorithm runs in $O(mn^2)$ and satisfies, with probability at least $1-10^{-20}$,

$$\|A - P_C A\|_F \leq O\left(k \log^{1/2} k\right) \|A - P_{U_k} A\|_F$$

1. We provide an efficient algorithmic result.
2. We guarantee a Frobenius norm bound that is at most $(k \log k)^{1/2}$ worse than the best known *existential* result.

Overview

Statistical Leverage and the Hat Matrix

Faster Algorithms for Least Squares Approximation

Better Algorithm for Column Subset Selection Problem

Even better, both perform very well empirically!

TechTC Term-document data

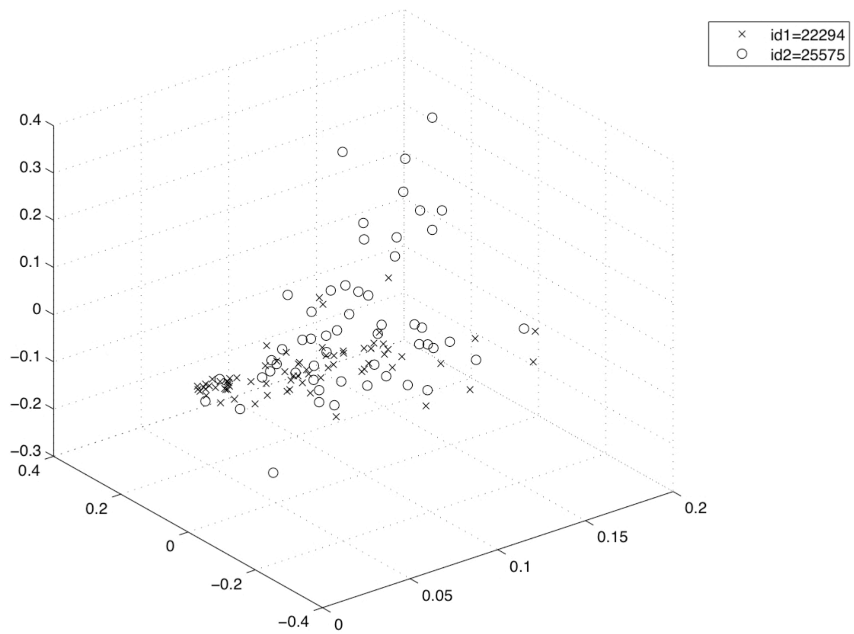
	id1	id2	#docs × #terms
(i)	10567 [1]	11346 [2]	139 × 15170
(ii)	10567 [1]	12121 [3]	138 × 11859
(iii)	11346 [2]	22294 [4]	125 × 14392
(iv)	11498 [5]	14517 [6]	125 × 15485
(v)	14517 [6]	186330 [7]	130 × 18289
(vi)	20186 [8]	22294 [4]	130 × 12708
(vii)	22294 [4]	25575 [9]	127 × 10012
(viii)	332386 [10]	61792 [11]	159 × 15860
(ix)	61792 [11]	814096 [12]	159 × 16066
(x)	85489 [13]	90753 [14]	154 × 14780

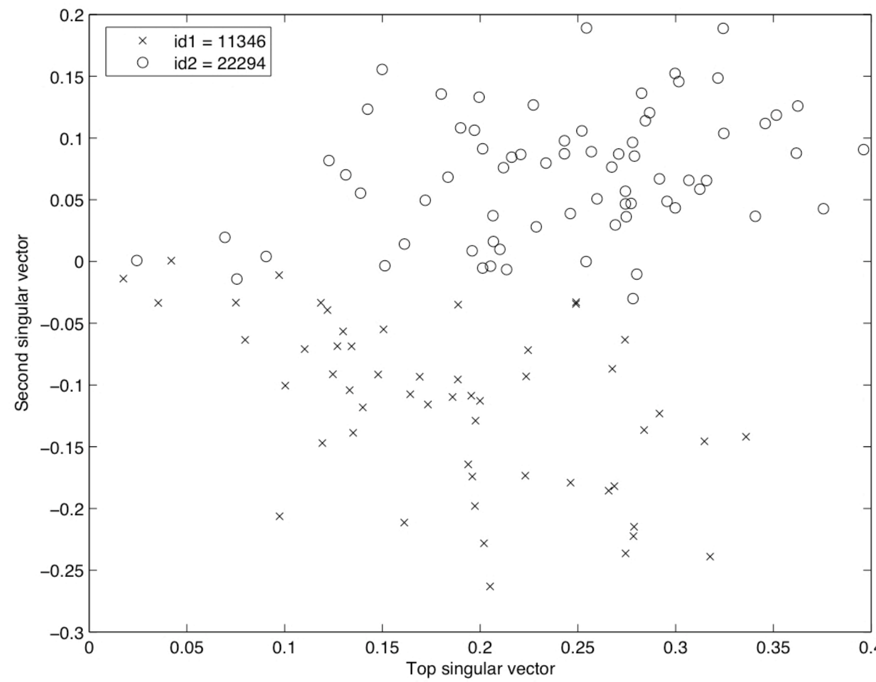
[1] US: Indiana: Evansville
[2] US: Florida
[3] California: San Diego: Business, economy
[4] Canada: British Columbia: Nanaimo
[5] California: Politics: Candidates, campaigns
[6] US: Arkansas
[7] US: Illinois
[8] US: Texas: Dallas
[9] Asia: Taiwan: Business and Economy
[10] Shopping: Vehicles
[11] US: California
[12] Europe: Ireland: Dublin
[13] Canada: Business and Economy: Industries
[14] Materials and Supplies: Masonry and Stone

(i) **florida**, **evansville**, their, consumer, reports
(ii) **diego**, **evansville**, pianos, which, services
(iii) **florida**, **nanaimo**, served, expensive, other
(iv) **eureka**, **california**, cobbler, which, insurance
(v) **eureka**, reliable, coldwell, rosewood, information
(vi) **dallas**, **nanaimo**, untitled, buffet, included
(vii) **nanaimo**, **taiwan**, megahome, great, states
(viii) **agent**, topframe, spacer, order, during
(ix) **dublin**, **beach**, estate, spacer, which
(x) **canada**, **stone**, mainframe, spacer, other

- Representative examples that cluster well in the low-dimensional space.

TechTC Term-document data





- Representative examples that cluster well in the low-dimensional space.

Conclusion

Statistical Leverage and the Hat Matrix

Faster Algorithms for Least Squares Approximation

Better Algorithm for Column Subset Selection Problem

Even better, both perform very well empirically!

Workshop on "Algorithms for Modern Massive Data Sets"

(http://mmds.stanford.edu)

Stanford University and Yahoo! Research, June 25-28, 2008

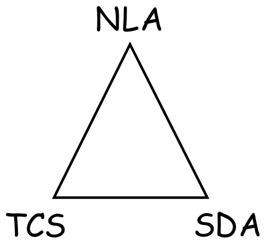
Objectives:

- Address algorithmic, mathematical, and statistical challenges in modern statistical data analysis.

- Explore novel techniques for modeling and analyzing massive, high-dimensional, and nonlinear-structured data.

- Bring together computer scientists, mathematicians, statisticians, and data analysis practitioners to promote cross-fertilization of ideas.

Organizers: M. W. Mahoney, L-H. Lim, P. Drineas, and G. Carlsson.



Sponsors: NSF, Yahoo! Research, PIMS, DARPA.