# Statistical Leverage and Improved Matrix Algorithms

**Michael W. Mahoney**

Stanford University

# Least Squares (LS) Approximation

$$\begin{pmatrix} \\ \\ A \\ \\ \\ \end{pmatrix}_{n \times d} \begin{pmatrix} \hat{x} \end{pmatrix} \approx \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix} \quad , \quad n \gg d$$

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

We are interested in over-constrained Lp regression problems, $n \gg d$.

- Typically, there is no $x$ such that $Ax = b$.
- Want to find the "best" x such that $Ax \approx b$.

Ubiquitous in applications & central to theory:

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

# Many applications of this!

- Astronomy: Predicting the orbit of the asteroid Ceres (in 1801!).

  Gauss (1809) -- see also Legendre (1805) and Adrain (1808).

  First application of "least squares optimization" and runs in $O(nd^2)$ time!

- Bioinformatics: Dimension reduction for classification of gene expression microarray data.
- Medicine: Inverse treatment planning and fast intensity-modulated radiation therapy.
- Engineering: Finite elements methods for solving Poisson, etc. equation.
- Control theory: Optimal design and control theory problems.
- Economics: Restricted maximum-likelihood estimation in econometrics.
- Image Analysis: Array signal and image processing.
- Computer Science: Computer vision, document and information retrieval.
- Internet Analysis: Filtering and de-noising of noisy internet data.
- Data analysis: Fit parameters of a biological, chemical, economic, social, internet, etc. model to experimental data.

# 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 $A$

# Modeling with Least Squares

Assumptions underlying its use:

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

Should always check to make sure these assumptions have not been (too) violated!

# Statistical Issues and Regression Diagnostics

| | |
|---|---|
| Model: $b = Ax+\varepsilon$ | b = response; $A^{(i)}$ = carriers; |
| | $\varepsilon$ = error process s.t.: mean zero, const. varnce, (i.e., E(e)=0 |
| | and $Var(e)=\sigma^2 I$), uncorrelated, normally distributed |
| $x_{opt} = (A^TA)^{-1}A^Tb$ | (what we computed before) |
| $b' = Hb$ | $H = A(A^TA)^{-1}A^T$ = "hat" matrix |
| | $H_{ij}$ - measures the leverage or influence exerted on $b'_i$ by $b_j$, |
| | *regardless* of the value of $b_j$ (since H depends only on A) |
| $e' = b-b' = (I-H)b$ | vector of residuals - note: $E(e')=0$, $Var(e')=\sigma^2(I-H)$ |
| | |
| Trace(H)=d | Diagnostic Rule of Thumb: Investigate if $H_{ii} > 2d/n$ |
| $H=UU^T$ | U is from SVD ($A=U\Sigma V^T$), or *any* orthogonal matrix for span(A) |
| $H_{ii} = \|U^{(i)}\|_2^2$ | leverage scores = row "lengths" of spanning orthogonal matrix |

# Hat Matrix and Regression Diagnostics

See: "The Hat Matrix in Regression and ANOVA," Hoaglin and Welsch (1978)

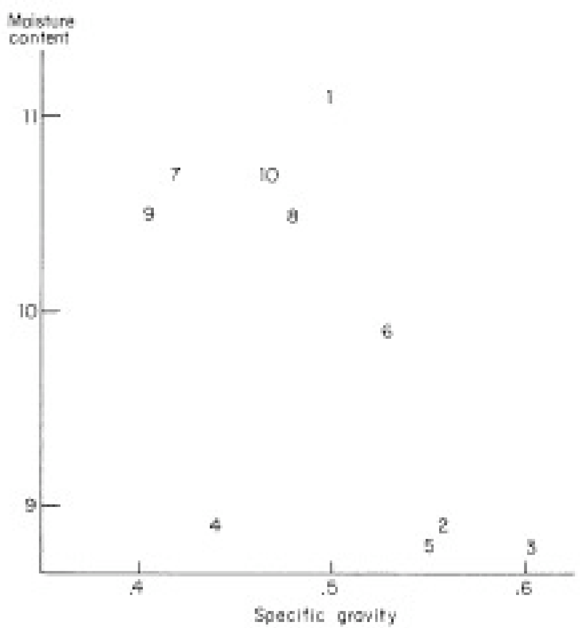



**Figure A.** The Two Carriers for the Wood Beam Data (Plotting symbol is beam number.).

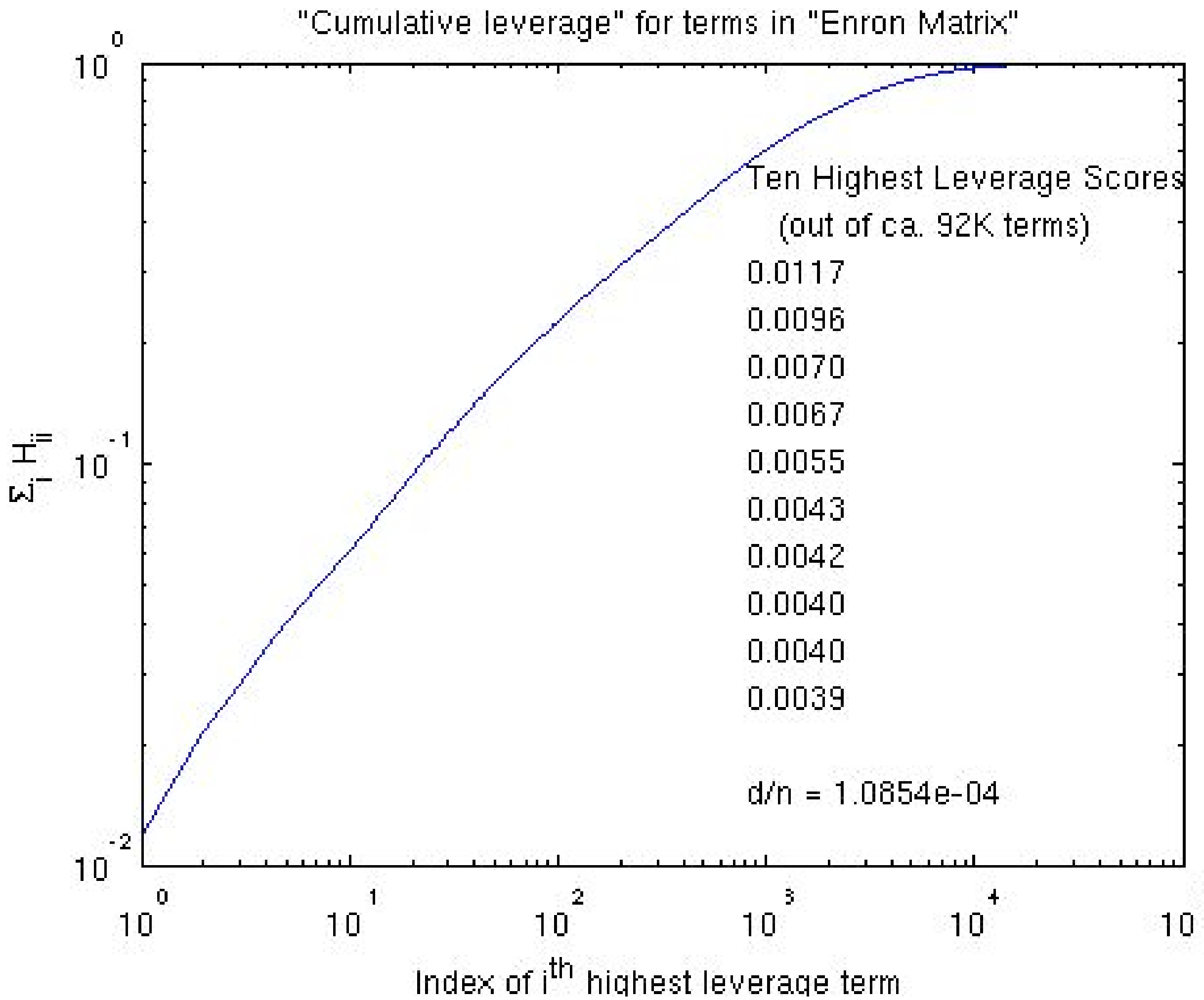
*2. The Hat Matrix for the Wood Beam Data (lower triangle omitted by symmetry)*

| i \ j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | .418 | -.002 | .079 | -.274 | -.046 | .181 | .128 | .222 | .050 | .242 |
| 2 | | .242 | .292 | .136 | .243 | .128 | -.041 | .033 | -.035 | .004 |
| 3 | | | .417 | -.019 | .273 | .187 | -.126 | .044 | -.153 | .004 |
| 4 | | | | .604 | .197 | -.038 | .168 | -.022 | .275 | -.028 |
| 5 | | | | | .252 | .111 | -.030 | .019 | -.010 | -.010 |
| 6 | | | | | | .148 | .042 | .117 | .012 | .111 |
| 7 | | | | | | | .262 | .145 | .277 | .174 |
| 8 | | | | | | | | .154 | .120 | .168 |
| 9 | | | | | | | | | .315 | .148 |
| 10 | | | | | | | | | | .187 |

## Examples of things to note:

- Point 4 is a bivariate outlier - and $H_{4,4}$ is largest, just exceeds 2p/n=6/10.
- Points 1 and 3 have relatively high leverage - extremes in the scatter of points.
- $H_{1,4}$ is moderately negative - opposite sides of the data band.
- $H_{1,8}$ and $H_{1,10}$ moderately positive - those points mutually reinforce.
- $H_{6,6}$ is fairly low - point 6 is in central position.

# Statistical Leverage and Large Internet Data



# Overview

## Faster Algorithms for Least Squares Approximation:

Sampling algorithm *and* projection algorithm.

Gets a $(1+\varepsilon)$-approximation in $o(nd^2)$ *time*.

Uses Randomized Hadamard Preprocessing from the recent "Fast" JL Lemma.

## Better Algorithm for Column Subset Selection Problem:

Two-phase algorithm to approximate the CSSP.

For spectral norm, improves best previous bound (Gu and Eisenstat, etc. and the RRQR).

For Frobenius norm, $O((k \log k)^{1/2})$ worse than best existential bound.

## Even better, both perform very well empirically!

Apply algorithm for CSSP to Unsupervised Feature Selection.

Application of algorithm for Fast Least Squares Approximation.

# Original (expensive) sampling algorithm

Drineas, Mahoney, and Muthukrishnan (SODA, 2006)

**Algorithm**

1. Fix a set of probabilities $p_i$, i=1,…,n.
2. Pick the i-th row of A and the i-th element of b with probability

   min {1, $rp_i$},

   and rescale by $(1/\min\{1, rp_i\})^{1/2}$.
3. Solve the induced problem.

**Theorem**:

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

If the $p_i$ satisfy:

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

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

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

- These probabilities $p_i$'s are statistical leverage scores!
  - "Expensive" to compute, $O(nd^2)$ time, these $p_i$'s!

# 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!!*

# A structural lemma

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

where $\mathcal{X}$ *is any matrix.*

Let $U_A$ be the matrix of left singular vectors of A.

assume that $\gamma$-fraction of mass of b lies in 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:

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

# Randomized Hadamard preprocessing

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

Let $H_n$ be an *n-by-n* deterministic Hadamard matrix, and
Let $D_n$ be an *n-by-n* random diagonal matrix with +1/-1 chosen u.a.r. on the diagonal.

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

$$||Ax - b||_2 = ||H_n D_n Ax - H_n D_n b||_2 = ||\mathcal{H}Ax - \mathcal{H}b||_2$$

(since $H_n$ and $D_n$ are orthogonal matrices).

**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*:

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

# Fast LS via *sparse* projection

Drineas, Mahoney, Muthukrishnan, and Sarlos (2007) - sparse projection matrix from Matousek's version of Ailon-Chazelle 2006

**Algorithm**

1. Pre-process A and b with a randomized Hadamard transform.
2. Multiply preprocessed input by sparse random k x n matrix T, where

$$\mathcal{T}_{ij} = \begin{cases} +\sqrt{\frac{1}{kq}} & \text{, with probability } q/2 \\ -\sqrt{\frac{1}{kq}} & \text{, with probability } q/2 \\ 0 & \text{, with probability } 1-q, \end{cases}$$

   and where $k=O(d/\varepsilon)$ and $q=O(d \log^2(n)/n+d^2\log(n)/n)$ .

3. Solve the induced problem:

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

- Dense projections will work, but it is "slow."
- Sparse projection is "fast," but will it work?

-> YES! Sparsity parameter q of T related to non-uniformity of leverage scores!

# Overview

## Faster Algorithms for Least Squares Approximation:

Sampling algorithm *and* projection algorithm.

Gets a $(1+\varepsilon)$-approximation in $o(nd^2)$ *time*.

Uses Randomized Hadamard Preprocessing from the recent "Fast" JL Lemma.

## Better Algorithm for Column Subset Selection Problem:

Two-phase algorithm to approximate the CSSP.

For spectral norm, improves best previous bound (Gu and Eisenstat, etc. and the RRQR).

For Frobenius norm, $O((k \log k)^{1/2})$ worse than best existential bound.

## Even better, both perform very well empirically!

Apply algorithm for CSSP to Unsupervised Feature Selection.

Application of algorithm for Fast Least Squares Approximation.

# 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*:

$$
\begin{aligned}
\min ||A - P_C A||_2 &= \min ||A - CC^+ A||_2, \\
&\quad \text{where } ||X||_2 = \max_{x \in \mathbb{R}^n : |x|=1} |Xx| \\
\min ||A - P_C A||_F &= \min ||A - CC^+ A||_F, \\
&\quad \text{where } ||X||_F^2 = \sum_{ij} X_{ij}^2
\end{aligned}
$$

$P_C = CC^+$ is the projector matrix on the subspace spanned by the columns of *C*.

Complexity of the problem? $O(n^k mn)$ trivially works; NP-hard if *k* grows as a function of *n*. (NP-hardness in Civril & Magdon-Ismail '07)

# A lower bound for the CSS problem

For any m-by-k matrix C consisting of at most k columns of A

$$||A - P_{U_k} A||_\xi \leq ||A - P_C A||_\xi$$

$A_k$

$$\min_{\Phi \in \mathcal{R}^{m\times k}, X \in \mathcal{R}^{k\times n}} \left\| \underbrace{\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix}}_{m \times n} - \underbrace{\begin{pmatrix} & & \\ & \Phi & \\ & & \end{pmatrix}}_{m \times k} \cdot \underbrace{\begin{pmatrix} & X & \end{pmatrix}}_{k \times n} \right\|_F^2$$

Given $\Phi$, it is easy to find X from standard least squares.

That we can find the optimal $\Phi$ is intriguing!

Optimal $\Phi = U_k$, optimal $X = U_k^T A$.

$$A_k = U_k U_k^T A = A V_k V_k^T$$

# 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

Theoretical Computer Science algorithms for the CSSP

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

- $O(mn^2)$ 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

- $O(mnk^2)$ 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 to sample O(k log k / $\varepsilon^2$ ) columns.

# Subspace sampling probabilities

<u>Subspace sampling probs:</u>

in $O(mn^2)$ time, compute: $$p_j = \frac{\left|\left(V_k^T\right)^{(i)}\right|^2}{k}$$

Normalization s.t. the $p_j$ sum up to 1

**NOTE**: Up to normalization, these are just statistical leverage scores!

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

$$\begin{pmatrix} & & \\ & A_k & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & U_k & \\ & & \end{pmatrix} \cdot \begin{pmatrix} \Sigma_k \end{pmatrix} \cdot \begin{pmatrix} V_k^T \end{pmatrix}$$

$$m \times n \qquad m \times k \qquad k \times k \qquad k \times n$$

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

$\Sigma_k$: diagonal matrix containing the top k singular values of A.

# Prior 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.

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)

Given an m-by-n matrix A (assume m ¸ n for simplicity):

- (Randomized phase) Run a randomized algorithm to pick *c = O(k logk)* columns.
- (Deterministic phase) Run a deterministic algorithm on the above columns* to pick exactly *k* columns of *A* and form an m-by-k matrix *C*.

* Not so simple …

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

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

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

# Randomized phase: O(k log k) columns

Randomized phase: c = O(k log k) via "subspace sampling" .

- Compute probabilities $p_j$ (below) summing to 1
- Pick the j-th column of $V_k^T$ with probability min{1,$cp_j$}, for each j = 1,2,…,n.
- Let $(V_k^T)_{S1}$ be the (rescaled) matrix consisting of the chosen columns from $V_k^T$ .

(At most c columns of $V_k^T$ - in expectation, at most 10c w.h.p. - are chosen.)

$$\underset{m \times n}{\begin{pmatrix} \\ A_k \\ \\ \end{pmatrix}} = \underset{m \times k}{\begin{pmatrix} \\ U_k \\ \\ \end{pmatrix}} \cdot \underset{k \times k}{\begin{pmatrix} \Sigma_k \end{pmatrix}} \cdot \underset{k \times n}{\begin{pmatrix} V_k^T \end{pmatrix}}$$

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

$\Sigma_k$: diagonal matrix containing the top k singular values of A.

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

$$p_j = \frac{\left|(V_k^T)^{(i)}\right|^2 + \left(\left|(A)^{(i)}\right|^2 - \left|(A_k)^{(i)}\right|^2\right)}{\mathcal{N}}$$

# Deterministic phase: *exactly k columns*

Deterministic phase

- Let $S_1$ be the set of indices of the columns selected by the randomized phase.
- Let $(V_k^T)_{S1}$ denote the set of columns of $V_k^T$ with indices in $S_1$,

  (Technicality: the columns of $(V_k^T)_{S1}$ must be rescaled.)
- Run a deterministic NLA algorithm on $(V_k^T)_{S1}$ to select *exactly k columns*.

  (Any algorithm with $p(k,n) = k^{1/2}(n-k)^{1/2}$ will do.)
- Let $S_2$ be the set of indices of the selected columns.

  (The cardinality of $S_2$ is exactly k.)
- Return $A_{S2}$ as the final ourput.

  (That is, return the columns of A corresponding to indices in $S_2$.)

# Analysis of the two-stage algorithm

Lemma 1: $\sigma_k(V_k^T S_1 D_1) \geq 1/2$.

(By matrix perturbation lemma, subspace sampling, and since c=O(k log(k).)

Lemma 2: $||A-P_C A||_\xi \leq ||A-A_k||_\xi + \sigma_k^{-1}(V_k^T S_1 D_1 S_2)||\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1||_\xi$.

Lemma 3: $\sigma_k^{-1}(V_k^T S_1 D_1 S_2) \leq 2(k(c-k+1))^{1/2}$.

(Combine Lemma 1 with the NLA bound from the deterministic phase on the c - not n - columns of $V_k^T S_1 D_1$.)

Lemma 4&5: $||\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1||_\xi \approx ||A-A_k||_\xi$, for $\xi$=2,F.

# 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 $1\text{-}10^{-20}$,

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

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

## Faster Algorithms for Least Squares Approximation:

- Sampling algorithm *and* projection algorithm.
- Gets a $(1+\varepsilon)$-approximation in *$o(nd^2)$ time*.
- Uses Randomized Hadamard Preprocessing from the recent "Fast" JL Lemma.

## Better Algorithm for Column Subset Selection Problem:

- Two-phase algorithm to approximate the CSSP.
- For spectral norm, improves best previous bound (Gu and Eisenstat, etc. and the RRQR).
- For Frobenius norm, $O((k \log k)^{1/2})$ worse than best existential bound.

## Even better, both perform very well empirically!

- Apply algorithm for CSSP to Unsupervised Feature Selection.
- Application of algorithm for Fast Least Squares Approximation -- Tygert and Rohklin 2008!

# Empirical Evaluation: Data Sets

S&P 500 data:

- historical stock prices for ≈500 stocks for ≈1150 days in 2003-2007
- very low rank (so good methodological test), but doesn't classify so well in low-dim space

TechTC term-document data:

- benchmark term-document data from the Open Directory Project (ODP)
- hundreds of matrices, each ≈200 documents from two ODP categories and ≥10K terms
- sometimes classifies well in low-dim space, and sometimes not

DNA SNP data from HapMap:

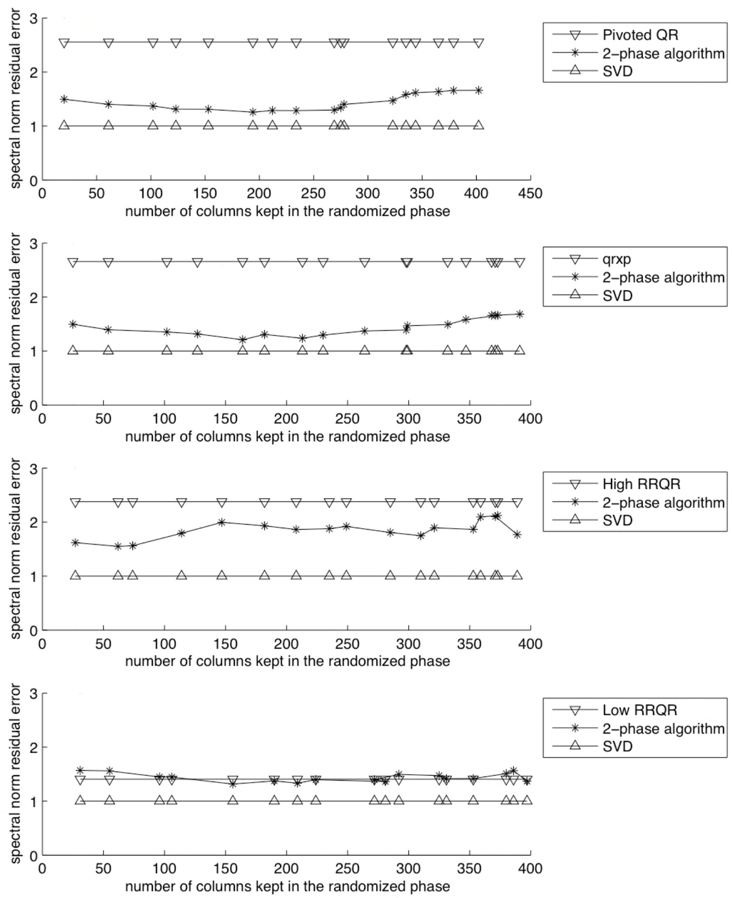
- Single nucleotide polymorphism (i.e., genetic variation) data from HapMap
- hundreds of individuals and millions of SNPs - often classifies well in low-dim space

# Empirical Evaluation: Algorithms

| Method | Reference | $\|A - P_C A\|_2 \leq$ | $Time =$ | Eval. |
|---|---|---|---|---|
| Pivoted QR | [Golub, 1965] | $\sqrt{(n-k)2^k}\|A - A_k\|_2$ | $O(mnk)$ | ✓ |
| High RRQR | [Foster, 1986] | $\sqrt{n(n-k)}2^{n-k}\|A - A_k\|_2$ | $O(mn^2)$ | |
| High RRQR | [Chan, 1987] | $\sqrt{n(n-k)}2^{n-k}\|A - A_k\|_2$ | $O(mn^2)$ | ✓ |
| RRQR | [Hong and Pan, 1992] | $\sqrt{k(n-k)+k}\|A - A_k\|_2$ | $O(n^k)$ | |
| Low RRQR | [Chan and Hansen, 1994] | $\sqrt{(k+1)n2^{k+1}}\|A - A_k\|_2$ | $O(mn^2)$ | ✓ |
| Hybrid-I RRQR | [Chandrasekaran and Ipsen, 1994 | $\sqrt{(k+1)(n-k)}\|A - A_k\|_2$ | $O(n^k)$ | |
| Hybrid-II RRQR | | $\sqrt{(k+1)(n-k)}\|A - A_k\|_2$ | $O(n^k)$ | |
| Hybrid-III RRQR | | $\sqrt{(k+1)(n-k)}\|A - A_k\|_2$ | $O(n^k)$ | |
| Strong RRQR | [Gu and Eisenstat, 1996] | $\sqrt{k(n-k)+1}\|A - A_k\|_2$ | $O(n^k)$ | |
| Strong RRQR | | $O(\sqrt{k(n-k)+1})\|A - A_k\|_2$ | $O(mn^2)$ | |
| DGEQPY (LAPACK) | [Bischof and Orti, 1998] | $O(\sqrt{(k+1)^2(n-k)})\|A - A_k\|_2$ | - | ✓ |
| DGEQPX (LAPACK) | | $O(\sqrt{(k+1)(n-k)})\|A - A_k\|_2$ | $O(n^k)$ | ✓ |
| SPQR | [Stewart, 1999] | - | - | ✓ |
| PT Algorithm 1 | [Pan and Tang, 1999] | $O(\sqrt{(k+1)(n-k)})\|A - A_k\|_2$ | - | |
| PT Algorithm 2 | | $O(\sqrt{(k+1)^2(n-k)})\|A - A_k\|_2$ | - | |
| PT Algorithm 3 | | $O(\sqrt{(k+1)^2(n-k)})\|A - A_k\|_2$ | - | |
| Gauss RRQR | [Pan, 2000] | $O(\sqrt{k(n-k)+1})\|A - A_k\|_2$ | $O(mn^2)$ | |

- Empirical Evaluation Goal: Unsupervised Feature Selection

# S&P 500 Financial Data




| Stock symbol | Stock Name | Sector |
|---|---|---|
| TE | TECO Energy | Utilities |
| RDC | Rowan Cos. | Energy |
| CTXS | Citrix Systems | Inf. Tech. |
| AFL | AFLAC Inc. | Financials |
| TER | Teradyne Inc. | Inf. Tech. |
| PCAR | PACCAR Inc. | Industrials |
| TYC | Tyco | Industrials |
| CHRW | C.H. Robinson | Industrials |
| CAT | Caterpillar Inc. | Industrials |
| SWK | Stanley Works | Consumer Disc |

- S&P data is a test - it's low rank but doesn't cluster well in that space.

# TechTC Term-document data

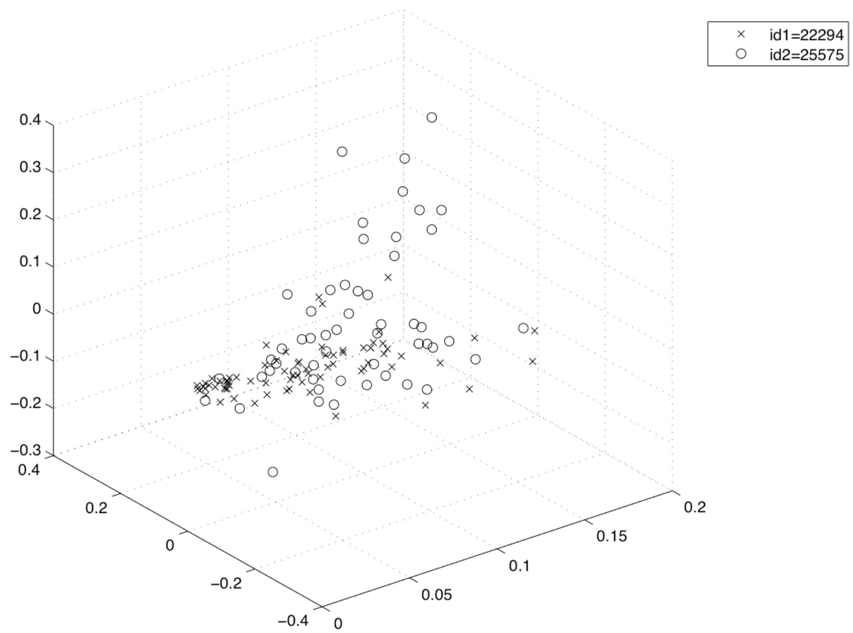
| | id1 | id2 | #docs × #terms |
|---|---|---|---|
| (i) | 10567 [1] | 11346 [2] | 139 × 15170 |
| (ii) | 10567 [1] | 12121 [3] | 138 × 11859 |
| (iii) | 11346 [2] | 22294 [4] | 125 × 14392 |
| (iv) | 11498 [5] | 14517 [6] | 125 × 15485 |
| (v) | 14517 [6] | 186330 [7] | 130 × 18289 |
| (vi) | 20186 [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 |

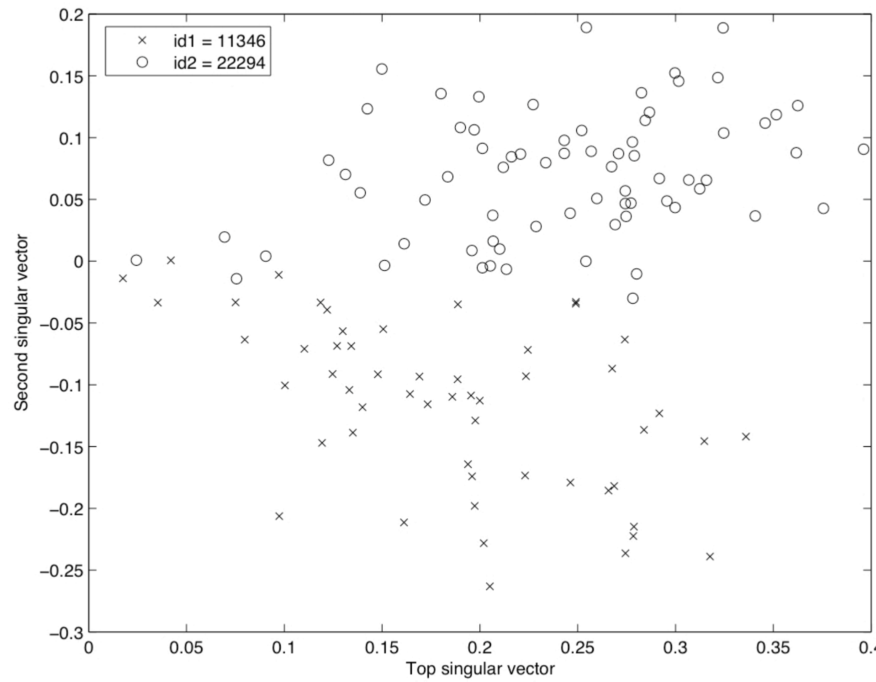
[1] US: Indiana: Evansville
[2] US: Florida
[3] California: San Diego: Business, economy
[4] Canada: British Columbia: Nanaimo
[5] California: Politics: Candidates, campaigns
[6] US: Arkansas
[7] US: Illinois
[8] US: Texas: Dallas
[9] Asia: Taiwan: Business and Economy
[10] Shopping: Vehicles
[11] US: 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.

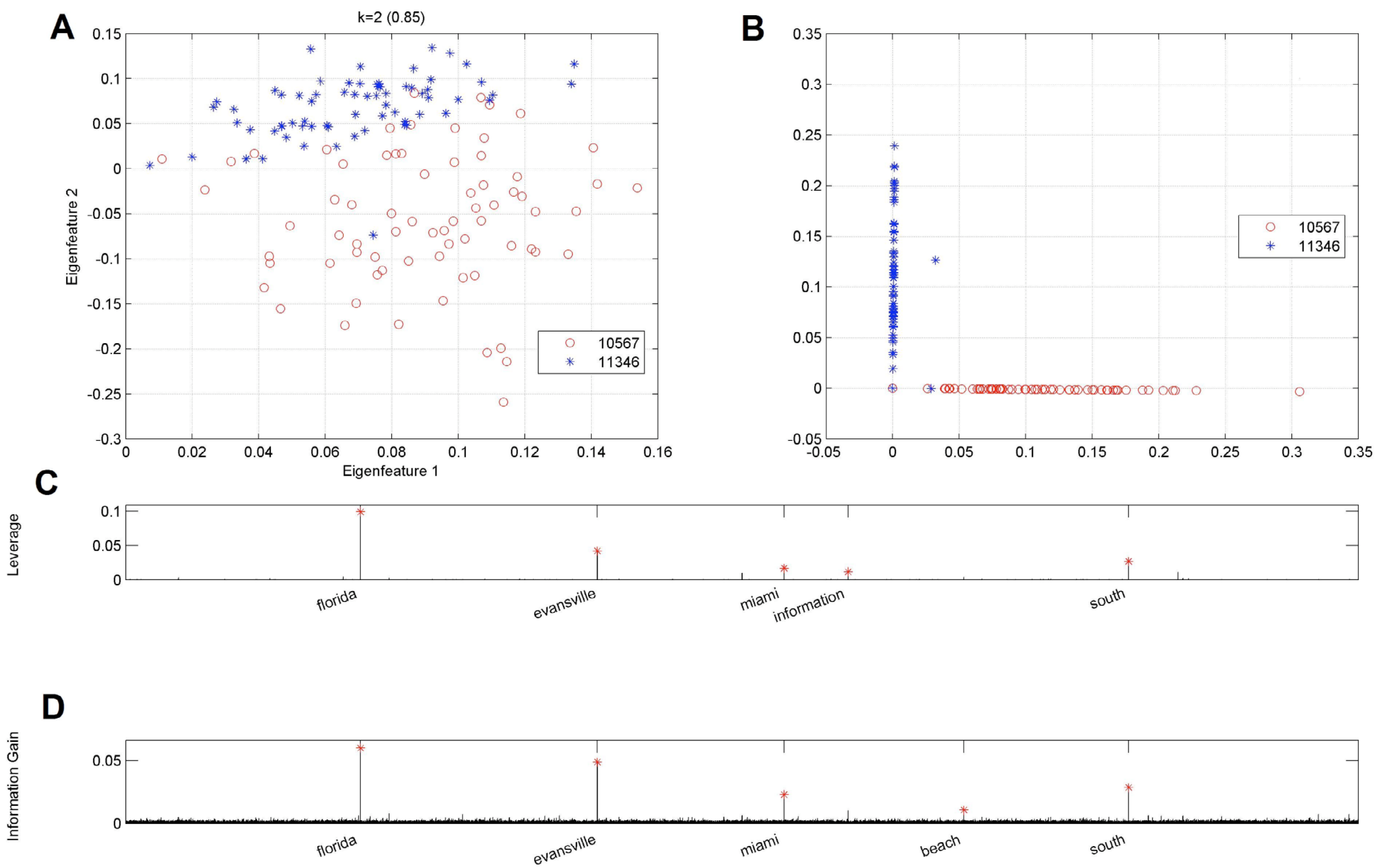
# TechTC Term-document data



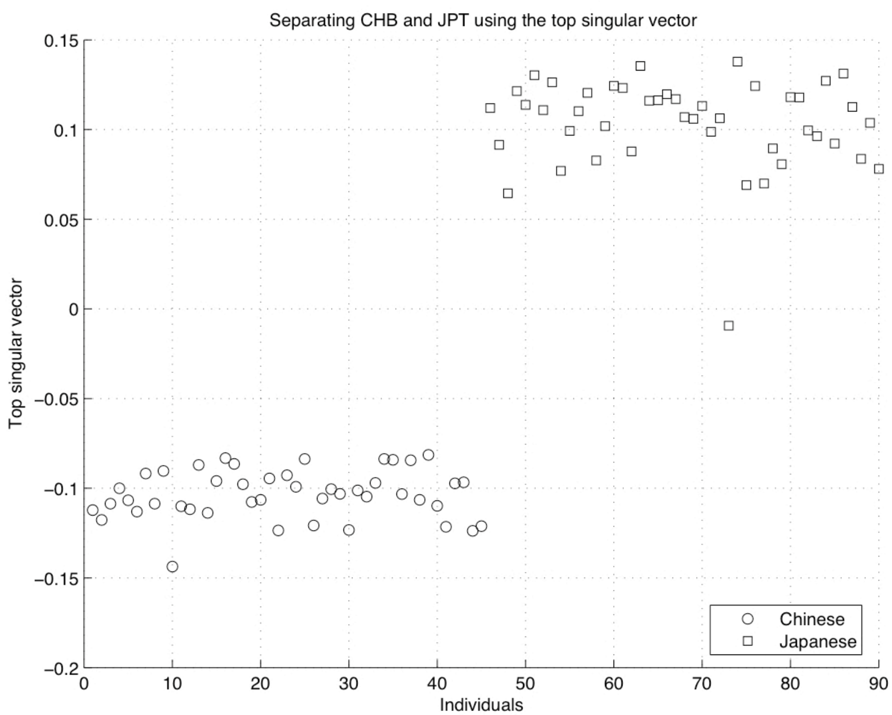



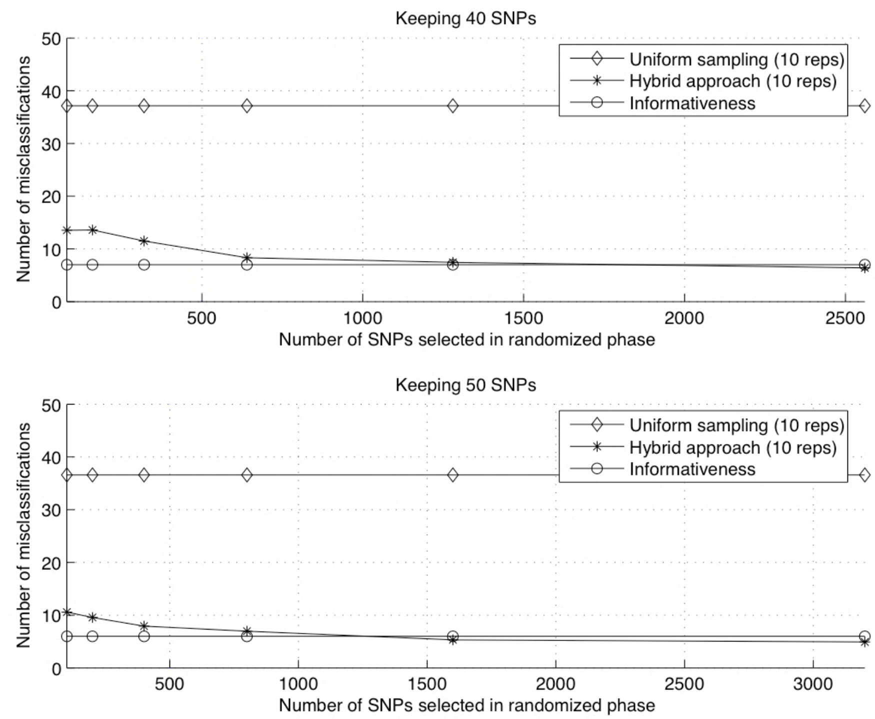

- Representative examples that cluster well in the low-dimensional space.

# TechTC Term-document data



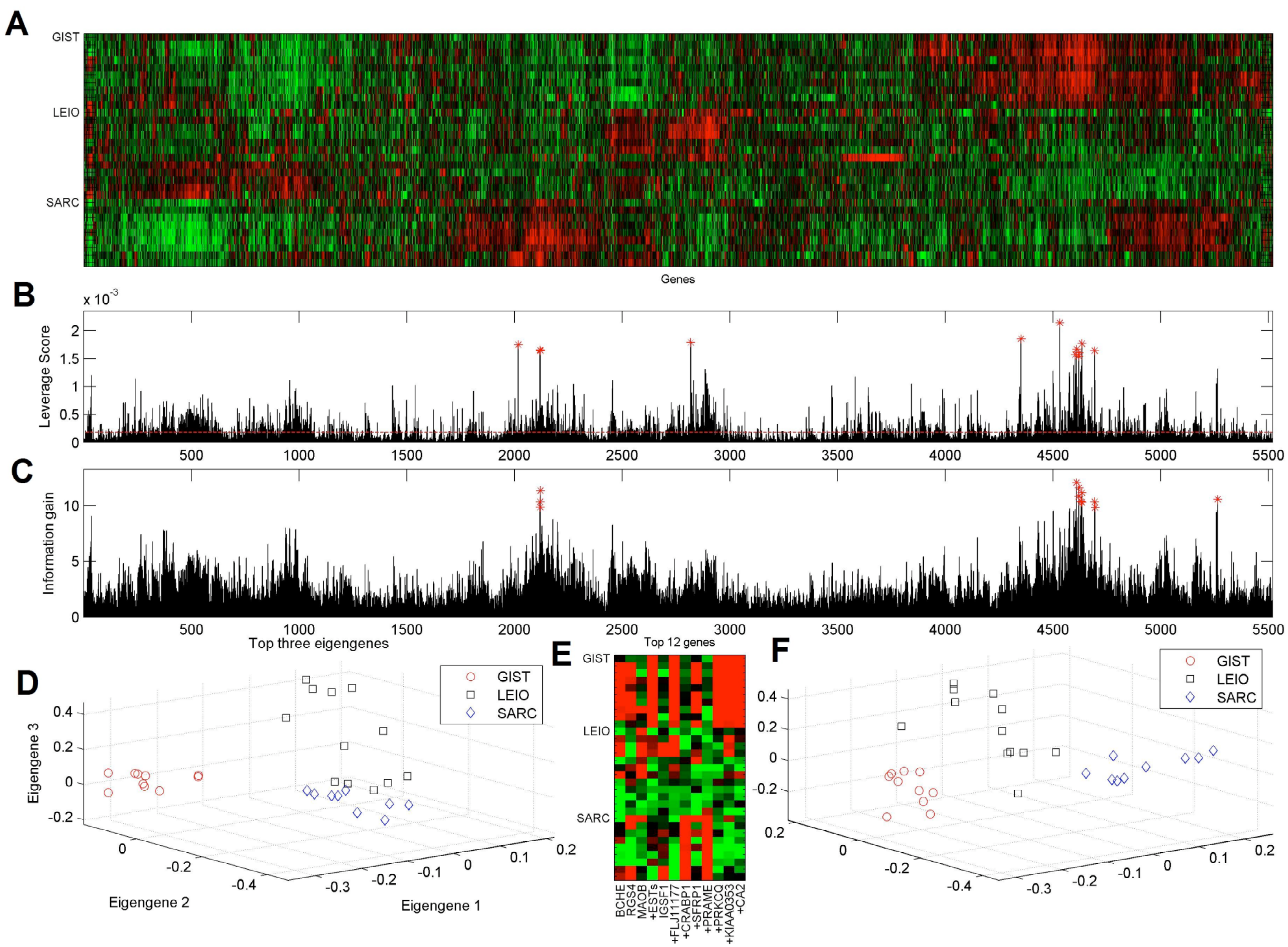
# DNA HapMap SNP data




- Most NLA codes don't even run on this 90 x 2M matrix.
- Informativeness is a state of the art supervised technique in genetics.

# DNA HapMap SNP data



# Conclusion

## Faster Algorithms for Least Squares Approximation:

Sampling algorithm *and* projection algorithm.

Gets a $(1+\varepsilon)$-approximation in $o(nd^2)$ *time*.

Uses Randomized Hadamard Preprocessing from the recent "Fast" JL Lemma.

## Better Algorithm for Column Subset Selection Problem:

Two-phase algorithm to approximate the CSSP.

For spectral norm, improves best previous bound (Gu and Eisenstat, etc. and the RRQR).

For Frobenius norm, $O((k \log k)^{1/2})$ worse than best existential bound.

## Even better, both perform very well empirically!

Apply algorithm for CSSP to Unsupervised Feature Selection.

Application of algorithm for Fast Least Squares Approximation.