Subset Selection

Deterministic vs. Randomized

Ilse Ipsen

North Carolina State University

Joint work with: Stan Eisenstat, Yale

Mary Beth Broadbent, Martin Brown, Kevin Penner

Subset Selection

Given: real or complex matrix A
integer k

Determine permutation matrix P so that

$$AP = (\underbrace{A_1}_{k} \quad A_2)$$

- Important columns A_1
  Columns of A_1 are 'very' linearly independent
- Redundant columns A_2
  Columns of A_2 are 'well' represented by A_1

Subset Selection Requirements

- **Important columns A_1**

  Smallest singular value $\sigma_k(A_1)$ should be 'large'

- **Redundant columns A_2**

  $\min_Z \|A_1 Z - A_2\|$ should be 'small' (two norm)

Subset Selection Requirements

- **Important columns A_1**

  Smallest singular value $\sigma_k(A_1)$ should be 'large'

  $$\sigma_k(A)/\gamma \leq \sigma_k(A_1) \leq \sigma_k(A)$$

  for some γ

- **Redundant columns A_2**

  $\min_Z \|A_1 Z - A_2\|$ should be 'small' (two norm)

  $$\sigma_{k+1}(A) \leq \min_Z \|A_1\, Z - A_2\| \leq \gamma\, \sigma_{k+1}(A)$$

  for some γ

Outline

- Deterministic algorithms: strong RRQR, SVD
- Randomized 2-phase algorithm
- Perturbation analysis of randomized algorithm
- Numerical experiments: randomized vs. deterministic
- New deterministic algorithm

Deterministic Subset Selection

Businger & Golub (1965) QR with column pivoting
Faddev, Kublanovskaya & Faddeeva (1968)
Golub, Klema & Stewart (1976)
Gragg & Stewart (1976)
Stewart (1984)
Foster (1986)
T. Chan (1987)

Hong & Pan (1992)
Chandrasekaran & Ipsen (1994)
Gu & Eisenstat (1996) Strong RRQR

First Deterministic Algorithm

Rank revealing QR decomposition

$$AP = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \qquad \text{where} \quad Q^T Q = I$$

- Important columns

$$Q \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} = A_1 \quad \text{and} \quad \sigma_i(A_1) = \sigma_i(R_{11}) \quad 1 \leq i \leq k$$

- Redundant columns

$$Q \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} = A_2 \qquad \min_Z \|A_1 Z - A_2\| = \|R_{22}\|$$

Strong RRQR (Gu & Eisenstat 1996)

Input: $m \times n$ matrix A, $m \geq n$, integer k

Output: $AP = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$ $\quad$ R_{11} is $k \times k$

- R_{11} is well conditioned

$$\frac{\sigma_i(A)}{\sqrt{1+k(n-k)}} \leq \sigma_i(R_{11}) \leq \sigma_i(A) \quad 1 \leq i \leq k$$

- R_{22} is small

$$\sigma_{k+j}(A) \leq \sigma_j(R_{22}) \leq \sqrt{1+k(n-k)}\, \sigma_{k+j}(A)$$

- Offdiagonal block not too large

$$\left| \left(R_{11}^{-1} R_{12} \right)_{ij} \right| \leq 1$$

Strong RRQR Algorithm

1. Compute some QR decomposition with column pivoting

$$AP_{\text{initial}} = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

2. Repeat

   Exchange a column of $\begin{pmatrix} R_{11} \\ 0 \end{pmatrix}$ with a column of $\begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix}$

   Update permutations P, retriangularize

   until $|\det(R_{11})|$ stops increasing

3. Output: $AP_{\text{final}} = \big(\underbrace{A_1}_{k} \quad \underbrace{A_2}_{n-k} \big)$

Second Deterministic Algorithm

Singular value decomposition

$$AP = \begin{pmatrix} \underbrace{A_1}_{k} & \underbrace{A_2}_{n-k} \end{pmatrix} = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

- Important columns A_1

$$\frac{\sigma_i(A)}{\|V_{11}^{-1}\|} \leq \sigma_i(A_1) \leq \sigma_i(A) \qquad \text{for all} \quad 1 \leq i \leq k$$

- Redundant columns A_2

$$\sigma_{k+1}(A) \leq \min_Z \|A_1 Z - A_2\| \leq \|V_{11}^{-1}\| \, \sigma_{k+1}(A)$$

[Hong & Pan 1992]

Almost Strong RRQR Algorithm

In the spirit of Golub, Klema and Stewart (1976)

1. Compute SVD

$$A = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

2. Apply strong RRQR to V_1: $\quad V_1 P = \begin{pmatrix} \underbrace{V_{11}}_{k} & \underbrace{V_{12}}_{n-k} \end{pmatrix}$

$$\frac{1}{\sqrt{1 + k(n-k)}} \leq \sigma_i(V_{11}) \leq 1 \qquad 1 \leq i \leq k$$

3. Output: $\quad AP = \begin{pmatrix} \underbrace{A_1}_{k} & \underbrace{A_2}_{n-k} \end{pmatrix}$

Almost Strong RRQR

Produces permutation P so that

$$AP = (\underbrace{A_1}_{k} \quad \underbrace{A_2}_{n-k})$$

where

- Important columns A_1

$$\frac{\sigma_i(A)}{\sqrt{1+k(n-k)}} \leq \sigma_i(A_1) \leq \sigma_i(A) \qquad 1 \leq i \leq k$$

- Redundant columns A_2

$$\sigma_{k+1}(A) \leq \min_Z \|A_1 Z - A_2\| \leq \sqrt{1+k(n-k)}\,\sigma_{k+1}(A)$$

Deterministic Subset Selection

- Algorithms: strong RRQR, SVD
- Permuting columns of A corresponds to permuting right singular vector matrix V
- Perturbation bounds in terms of V

$$\sigma_{k+1}(A) \leq \min_Z \|A_1 Z - A_2\| \leq \|V_{11}^{-1}\| \, \sigma_{k+1}(A)$$

- Operation count for $m \times n$ matrix, $m \geq n$

$$\min_Z \|A_1 Z - A_2\| \leq \sqrt{1 + f^2 k(n-k)} \, \sigma_{k+1}(A)$$

in $\mathcal{O}\left(mn^2 + n^3 \log_f n\right)$ flops

Randomized Algorithms

Frieze, Kannan & Vempala 1998, 2004
Drineas, Kannan & Mahoney 2006
Deshpande, Rademacher, Vempala & Wang 2006
Rudelson & Vershynin 2007
Liberty, Woolfe, Martinsson, Rokhlin & Tygert 2007
Drineas, Mahoney & Muthukrishnan 2006, 2008
Boutsidis, Mahoney & Drineas 2008, 2009
Civril & Magdon-Ismail 2009

Survey paper:

Halko, Martinsson & Tropp 2009

2-Phase Randomized Algorithm

Boutsidis, Mahoney & Drineas 2009

1. **Randomized Phase:**
   Sample small number ($\approx k \log k$) of columns
2. **Deterministic Phase:**
   Apply rank revealing QR to sampled columns

With 70% probability:

- Two norm

$$\min_Z \|A_1 Z - A_2\|_2 \leq \mathcal{O}\left(k^{3/4} \log^{1/2} k \, (n-k)^{1/4}\right) \|\Sigma_2\|_2$$

- Frobenius norm

$$\min_Z \|A_1 Z - A_2\|_F \leq \mathcal{O}\left(k \log^{1/2} k\right) \|\Sigma_2\|_F$$

Deterministic vs. Randomized Algorithms

- Want: permutation P so that $AP = \begin{pmatrix} \underbrace{A_1}_{k} & \underbrace{A_2}_{n-k} \end{pmatrix}$
- Compute SVD

$$A = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

- Obtain P from k dominant right singular vectors V_1

Deterministic vs. Randomized Algorithms

- Want: permutation P so that $\quad AP = \begin{pmatrix} \underbrace{A_1}_{k} & \underbrace{A_2}_{n-k} \end{pmatrix}$

- Compute SVD

$$A = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

- Obtain P from k dominant right singular vectors V_1

- Deterministic:
  Apply RRQR to all columns of matrix V_1

- Randomized:
  Apply RRQR to subset of columns of scaled matrix $V_1\ D$

2-Phase Randomized Algorithm

1. Compute SVD

$$A = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

2. Randomized phase:
   Scale: $V_1 \longrightarrow V_1 D$
   Sample c columns: $(V_1 D)\, P_s = (\underbrace{V_{1s} D_s}_{\hat{c}} \quad *)$

3. Deterministic phase:
   Apply RRQR to $V_{1s}D_s$: $(V_{1s}D_s)\, P_d = (\underbrace{V_{11} D_1}_{k} \quad *)$

4. Output: $AP_sP_d = (\underbrace{A_1}_{k} \quad \underbrace{A_2}_{n-k})$

Perturbation Bounds

$$\textbf{SVD :} \qquad AP = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \qquad V_{11} \text{ is } k \times k$$

- **For deterministic algorithms**

$$\min_Z \|A_1 Z - A_2\| \le \|\Sigma_2\| / \sigma_k(V_{11})$$

or

$$\min_Z \|A_1 Z - A_2\| \le \|\Sigma_2\| + \|\Sigma_2 V_{21}\| / \sigma_k(V_{11})$$

Perturbation Bounds

SVD : $$AP = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$ V_{11} is $k \times k$

- For deterministic algorithms

$$\min_Z \|A_1 Z - A_2\| \leq \|\Sigma_2\| / \sigma_k(V_{11})$$

or

$$\min_Z \|A_1 Z - A_2\| \leq \|\Sigma_2\| + \|\Sigma_2 V_{21}\| / \sigma_k(V_{11})$$

- For randomized 2-phase algorithm

$$\min_Z \|A_1 Z - A_2\| \leq \|\Sigma_2\| + \|\Sigma_2 V_{21} D_1\| / \sigma_k(V_{11} D_1)$$

for any nonsingular matrix D_1

Perturbation Bounds for Randomized Algorithm

- D_1 is scaling matrix

$$\min_Z \|A_1 Z - A_2\| \le \|\Sigma_2\| + \|\Sigma_2 V_{21} D_1\| / \sigma_k(V_{11} D_1)$$

- $V_{11}D_1$ comes from RRQR: $\underbrace{(V_{1s}D_s)}_{\hat{c}} P_d = (\underbrace{V_{11}D_1}_{k} \quad *)$

$$\min_Z \|A_1 Z - A_2\| \le$$

$$\|\Sigma_2\| + \sqrt{1 + k(\hat{c} - k)}\, \|\Sigma_2 V_{21} D_1\| / \sigma_k(V_{1s} D_s)$$

Perturbation Bounds for Randomized Algorithm

- D_1 is scaling matrix

$$\min_Z \|A_1 Z - A_2\| \leq \|\Sigma_2\| + \|\Sigma_2 V_{21} D_1\| / \sigma_k(V_{11} D_1)$$

- $V_{11} D_1$ comes from RRQR: $\underbrace{(V_{1s} D_s)}_{\hat{c}} P_d = (\underbrace{V_{11} D_1}_{k} \quad *)$

$$\min_Z \|A_1 Z - A_2\| \leq$$
$$\|\Sigma_2\| + \sqrt{1 + k(\hat{c} - k)}\, \|\Sigma_2 V_{21} D_1\| / \sigma_k(V_{1s} D_s)$$

- We need with high probability:

$$\|\Sigma_2 V_{21} D_1\| \approx \|\Sigma_2\| \qquad \sigma_k(V_{1s} D_s) \gg 0$$

Probabilistic Bounds: Frobenius Norm

Column i of X sampled with probability p_i
Scaling matrix $D_{ii} = 1/\sqrt{p_i}$ with probability p_i

Probabilistic Bounds: Frobenius Norm

Column i of X sampled with probability p_i
Scaling matrix $D_{ii} = 1/\sqrt{p_i}$ with probability p_i

- Frobenius norm $\|X\,D\|_F^2 = \mathrm{trace}(X\,D^2\,X^T)$
- Linearity $E\left[\|X\,D\|_F^2\right] = \mathrm{trace}(X\;\underbrace{E\left[D^2\right]}_{I}\;X^T)$
- Scaling $E\left[D_{ii}^2\right] = p_i * \frac{1}{p_i} + (1-p_i) * 0 = 1$
- Expected value $E\left[\|X\,D\|_F^2\right] = \|X\|_F^2$

Probabilistic Bounds: Frobenius Norm

Column i of X sampled with probability p_i
Scaling matrix $D_{ii} = 1/\sqrt{p_i}$ with probability p_i

- Frobenius norm $\|X\,D\|_F^2 = \mathrm{trace}(X\,D^2\,X^T)$
- Linearity $E\left[\|X\,D\|_F^2\right] = \mathrm{trace}(X\,\underbrace{E\left[D^2\right]}_{I}\,X^T)$
- Scaling $E\left[D_{ii}^2\right] = p_i * \frac{1}{p_i} + (1-p_i) * 0 = 1$
- Expected value $E\left[\|X\,D\|_F^2\right] = \|X\|_F^2$
- Markov's inequality

$$\mathrm{Prob}\left[\|X\,D\|_F^2 \leq \alpha\,\|X\|_F^2\right] \geq 1 - \frac{1}{\alpha}$$

Randomized Subset Selection: Frobenius Norm

- Perturbation bound

$$\min_{Z} \|A_1 Z - A_2\|_F \leq$$
$$\|\Sigma_2\|_F + \sqrt{1 + k(\hat{c} - k)}\ \|\Sigma_2 V_{21} D_1\|_F / \sigma_k(V_{1s} D_s)$$

- With probability $1 - \frac{1}{\alpha}$

$$\|\Sigma_2 V_{21} D_1\|_F \leq \sqrt{\alpha}\ \|\Sigma_2 V_2\|_F = \sqrt{\alpha}\ \|\Sigma_2\|_F$$

- Holds for any probability distribution

Randomized Subset Selection: Frobenius Norm

- Perturbation bound

$$\min_{Z} \|A_1 Z - A_2\|_F \leq$$
$$\|\Sigma_2\|_F + \sqrt{1 + k(\hat{c} - k)}\ \|\Sigma_2 V_{21} D_1\|_F / \sigma_k(V_{1s} D_s)$$

- With probability $1 - \frac{1}{\alpha}$

$$\|\Sigma_2 V_{21} D_1\|_F \leq \sqrt{\alpha}\ \|\Sigma_2 V_2\|_F = \sqrt{\alpha}\ \|\Sigma_2\|_F$$

- Holds for any probability distribution

- Still to show: $\sigma_k(V_{1s} D_s) \gg 0$ with high probability

Sampling: Frobenius Norm

When is $\sigma_k(V_{1s}D_s) \gg 0$ with high probability?

- Expected number of sampled columns: c
- Column i of V_1 sampled with "probability"

$$p_i = \min\{1, c\, q_i\} \qquad \text{where} \quad q_i = \|(V_1)_i\|_2^2/k$$

Try to sample columns with large norm

- Scaling matrix $D = \begin{pmatrix} 1/\sqrt{p_1} & \dots & 1/\sqrt{p_n} \end{pmatrix}$
- If $c = \Theta\,(k \log k)$ then with probability $\geq .9$

$$\sigma_k(V_{1s}D_s) \geq 1/2$$

[Boutsidis, Mahoney & Drineas 2009]

Sampling: Frobenius Norm

How many columns should V_{1s} actually have?

- **Expected** number of sampled columns from V_1: c
- **Actual** number of columns in V_{1s}: $\hat{c}$

$$E[\hat{c}] \leq c$$

Sampling: Frobenius Norm

How many columns should V_{1s} actually have?

- Expected number of sampled columns from V_1: c
- Actual number of columns in V_{1s}: $\hat{c}$

$$E[\hat{c}] \leq c$$

- If $c\, q_i \leq 1$ for all i, and $\hat{c} \geq k$ then

$$\sigma_k(V_{1s}D_s) \leq \sqrt{\frac{\hat{c}}{c}}$$

- If $\hat{c} < c/4$ then $\sigma_k(V_{1s}D_s) < 1/2$

Make sure that enough columns are actually sampled

Randomized Subset Selection: Two Norm

$$\min_{Z} \|A_1 Z - A_2\|_2 \leq$$
$$\|\Sigma_2\|_2 + \sqrt{1 + k(\hat{c} - k)}\ \|\Sigma_2 V_{21} D_1\|_2 / \sigma_k(V_{1s} D_s)$$

- Probability distribution $p_i = \min\{1, c\, q_i\}$

$$q_i = \frac{1}{2}\ \frac{\|(V_1)_i\|_2^2}{k} + \frac{1}{2}\left(\frac{\|(\Sigma_2 V_2)_i\|_2}{\|\Sigma_2 V_2\|_F}\right)^2$$

- With probability $\geq .9$

$$\|\Sigma_2 V_{21} D_1\|_2 \leq \gamma\ \left(\frac{(n-k+1)\log c}{c}\right)^{1/4}\ \|\Sigma_2\|_2$$

[Boutsidis, Mahoney & Drineas 2009]

Numerical Experiments

Compare strong RRQR and randomized 2-phase algorithm

- Subset selection for 2 norm
- Matrix orders $n \leq 500, 2000$
- $0 \leq k \leq 240$
- Matrices:

  Kahan, random, scaled random, triangular
  numerical rank k

- Randomized algorithm: run 40 times
- Iterative determination of c

  $c = 2k$
  while $\sigma_k(V_{1s}D_s) < 1/2$ do $c = 2 * c$

Accuracy

Residuals $\min_Z \|A_1 Z - A_2\|_2$ for $n = 500$ and $k = 20$

	RRQR	Randomized		
Matrix		max	min	mean
Kahan	7×10^0	2×10^0	5×10^{-1}	5×10^{-1}
num. rank k	7×10^0	2×10^1	8×10^0	1×10^1
triangular	3×10^0	1×10^0	1×10^0	1×10^0
random	3×10^1	3×10^1	3×10^1	3×10^1
scaled rand	4×10^1	5×10^1	4×10^1	5×10^1

No significant difference in accuracy between deterministic and randomized algorithms

Number of Sampled Columns

Values of c for $n = 500$ and $k = 20$

Matrix	c values tried	most frequent	mean $\hat{c}$
Kahan	$40, 80, 160, 320, 640$	$2k = 40$	56
num. rank k	$40, 80, 160$	$4k = 80$	83
triangular	$40, 80, 160$	$4k = 80$	97
random	$40, 80, 160$	$4k = 80$	93
scaled rand	$40, 80, 160$	$4k = 80$	97

$c = 4k$ seems to be a good value

Different Probability Distributions

- **Two norm**

$$q_i = \frac{1}{2}\,\frac{\|(V_1)_i\|_2^2}{k} + \frac{1}{2}\left(\frac{\|(\Sigma_2 V_2)_i\|_2}{\|\Sigma_2 V_2\|_F}\right)^2$$

expensive

$$q_i = \frac{1}{2}\,\frac{\|(V_1)_i\|_2^2}{k} + \frac{1}{2}\,\frac{\|(A)_i\|_2^2 - \|(AV_1^T V_1)_i\|_2^2}{\|A\|_F^2 - \|AV_1^T V_1\|_F^2}$$

numerically unstable (can be negative)

- **Frobenius norm**

$$q_i = \frac{\|(V_1)_i\|_2^2}{k}$$

numerically stable and cheap

Different Probability Distributions

Residuals $\min_Z \|A_1 Z - A_2\|_2$ for $n = 500$ and $k = 20$

Matrix	P	max	min	mean	$\hat{c}$
num. rank k	2	2×10^1	8×10^0	1×10^1	83
	F	1×10^1	7×10^0	1×10^1	88
triangular	2	1×10^0	1×10^0	1×10^0	97
	F	1×10^0	1×10^0	8×10^0	91
random	2	3×10^1	3×10^1	3×10^1	93
	F	3×10^1	3×10^1	3×10^1	87
scaled rand	2	5×10^1	4×10^1	5×10^1	97
	F	5×10^1	4×10^1	5×10^1	93
Kahan	2	2×10^0	5×10^{-1}	5×10^{-1}	56
	F	7×10^{-1}	5×10^{-1}	5×10^{-1}	42

No difference in accuracy between 2 norm and Frobenius norm probability distributions

Ideas from Randomized Algorithm

$$AP = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \qquad V_1 = (\underbrace{V_{11}}_{k} \quad \underbrace{V_{12}}_{n-k})$$

- Order columns of V_1 in order of decreasing norms
- Apply strong RRQR to columns of largest norm

Intuition

$$\|V_{11}\|_F^2 = k - \|V_{12}\|_F^2$$

This means: $\|V_{11}\|_F$ large implies $\|V_{12}\|_F$ small

$$\sigma_k(V_{11})^2 = 1 - \|V_{12}\|_2^2$$

This means: $\|V_{12}\|_2$ small implies $\sigma_k(V_{11})$ large

New 2-Phase Deterministic Algorithm

1. Compute SVD

$$A = U \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

2. **Ordering phase:**
   Order according to decreasing column norms

$$V_1 \longrightarrow V_1 P_O$$

Select leading 4k columns: $\quad A P_O = (\underbrace{A_S}_{4k} \quad *)$

3. **Rank revealing phase:**
   Apply strong RRQR to A_S: $\quad A_S P_R = (\underbrace{A_1}_{k} \quad *)$

Results for New Deterministic Algorithm

- $n = 2000$ and $k = 40$
- Residuals $\min_Z \|A_1 Z - A_2\|_2$
- Time ratio $TR = \text{time(new algorithm)}/\text{time(RRQR)}$

Matrix	Residuals		$\sigma_k(A_1)$		TR
	RRQR	new	RRQR	new	
Kahan	4×10^0	4×10^0	8×10^{-2}	8×10^{-2}	0.11
random	9×10^1	9×10^1	1×10^1	1×10^1	0.03
s. rand	1×10^2	1×10^2	2×10^1	2×10^1	0.07
triang	4×10^0	3×10^1	3×10^{-1}	3×10^{-1}	0.02

New algorithm appears to be as accurate as RRQR
and possibly faster

Summary

Subset selection

Given: real or complex matrix A, integer k

Want: $AP = (\underbrace{A_1}_{k} \quad A_2)$ with

$$\sigma_k(A_1) \approx \sigma_k(A) \qquad \min_Z \|A_1 Z - A_2\| \approx \sigma_{k+1}(A)$$

- Deterministic algorithms: strong RRQR, SVD
- Randomized 2-phase algorithm
- Randomized algorithm: no more accurate than strong RRQR for matrices of order ≤ 2000
- Numerical issues with randomized algorithm
- New deterministic algorithm:
  As accurate as strong RRQR and perhaps faster