

An Improved Approximation Algorithm for the Column Subset Selection Problem

Authors: Christos Boutsidis(RPI)

Michael W. Mahoney(Stanford)

Petros Drineas(RPI)

Presenter: Chia-Chun Tsai

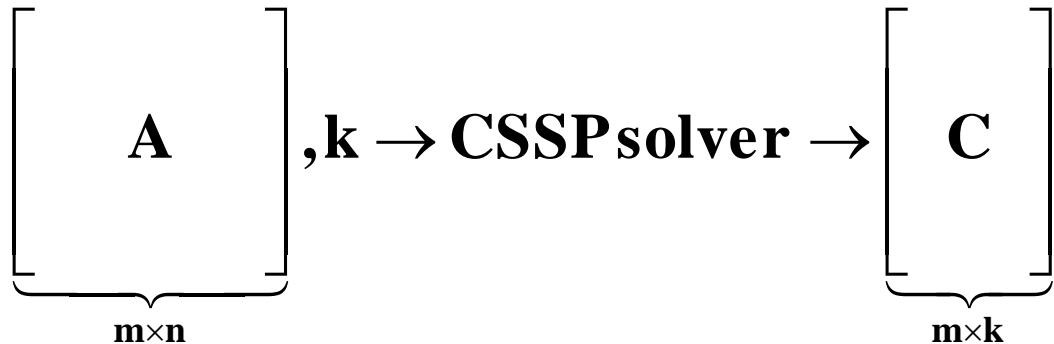
Department of Industrial and Systems Engineering

April, 29, 2010

Outline

- The Column Subset Selection Problem(CSSP)
 - Definition, complexity
 - Approximability framework
- Paper's contribution on the CSSP
 - Novel randomized algorithm
 - Improved approximability results

The Column Subset Selection Problem



Input : matrix $A \in \mathbb{R}^{m \times n}$, integer $k < n$

Output : matrix $C \in \mathbb{R}^{m \times k}$ with k columns of A

Combinatorial Optimization Problem :

$$\min_C \|A - CC^+ A\|_\xi$$

$\xi = 2, F$ stands for the spectral or the Frobenius norm

Complexity of the CSSP

- NP-hardness of the CSSP is an open question.
- There are $\binom{n}{k}$ for the matrix C
- Optimal solution can be found in $O(n^k m n)$ time.

Notation and Linear Algebra

SVD of a matrix $A \in \mathbf{R}^{m \times n}$ with $\text{rank}(A) = \rho$

$$A = U_A \Sigma_A V_A^T = \begin{pmatrix} U_k & U_{\rho-k} \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{\rho-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ V_{\rho-k}^T \end{pmatrix}$$

Why SVD?

$$A_k = U_k \Sigma_k V_k^T = \arg \left\{ \min_{Y \in R^{m \times n}, \text{rank}(Y) \leq k} \|A - Y\|_{\xi} \right\}$$

Pseudoinverse of C is $C^+ = V_c \Sigma_c^+ U_c^T$

Why $C^+ A$?

$$C^+ A = \arg \left\{ \min_{X \in R^{k \times n}} \|A - CX\|_{\xi} \right\}$$

Approximation Algorithm for the CSSP

SVD provides a lower bound for the CSSP:

$$\|A - A_k\|_{\xi} \leq \|A - C_{opt} C_{opt}^+ A\|_{\xi}$$

We seek algorithms with upper bounds of the form:

$$\|A - CC^+ A\|_{\xi} \leq p(k, n) \|A - A_k\|_{\xi}$$

p(k,n) is typically a polynomial function

Overview of best-existing and paper results

Spectralnorm :

Gu & Eisenstat, 1996 Deterministic, $O(mn^2)$ time

$$\|A - CC^+ A\|_2 \leq O(k^{\frac{1}{2}}(n-k)^{\frac{1}{2}}) \|A - A_k\|_2$$

Paper's Algorithm, 2009 Randomized, $O(mn^2)$ time

$$\|A - CC^+ A\|_2 \leq O(k^{\frac{3}{4}}(n-k)^{\frac{1}{4}}) \|A - A_k\|_2$$

Frobeniusnorm :

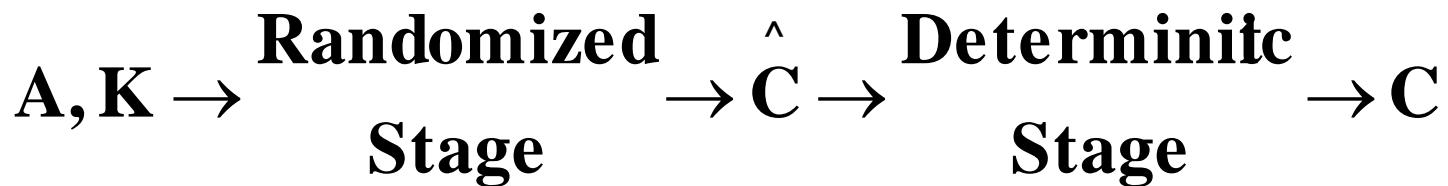
Deshpande & Vempala, 2006 Randomized, $O(mnk)$ time

$$\|A - CC^+ A\|_F \leq \sqrt{(K+1)!} \|A - A_k\|_F$$

Paper's Algorithm, 2009 Randomized, $O(mn^2)$ time

$$\|A - CC^+ A\|_F \leq O(K\sqrt{\log K}) \|A - A_k\|_F$$

Paper's novel randomized algorithm



2 stage column selection

RandomizedStage: $A \rightarrow \hat{C} \in R^{m \times c}, C = O(k \log k)$

DeterministicStage: $\hat{C} \rightarrow C \in R^{m \times k}$

Randomized Stage



Randomized Sampling:

1. You give a score to every column of A :

$$p_i = \frac{\frac{1}{2} \left\| (V_k)_{(i)} \right\|_2^2}{\sum_{j=1}^n \left\| (V_k)_{(j)} \right\|_2^2} + \frac{\frac{1}{2} \left\| (\sum_{\rho=k} V_{\rho-k}^T)^{(i)} \right\|_2^2}{\sum_{j=1}^n \left\| (\sum_{\rho=k} V_{\rho-k}^T)^{(j)} \right\|_2^2}$$

2. You select the i - th column with probability $\min\{1, cp_i\}$

Strong approximation guarantee ($\xi = F$) :

$$\left\| A - \hat{C} \hat{C}^+ A \right\|_F \leq (1 + \varepsilon) \|A - A_k\|_F$$

[Reference: Drineas, Mahoney, & Muthukrishnan, SIMAX 2008]

Deterministic Stage

Somemore notation:

$$C = O(k \log k)$$

$S_1 \in R^{m \times c}$ **is a sampling matrix**. $\hat{C} = AS_1$

$D_1 \in R^{c \times c}$ **is a diagonal matrix**. $D_{ii} = 1/\sqrt{\min\{1, cp_i\}}$

RRQR factorization on $V_k^T S_1 D_1 \in R^{k \times c}$

$$(V_k^T S_1 D_1) \Pi = QR$$

Extract k columns after this RRQR factorization.

[Reference Pan, LAA 2000, Gu & Eisenstat SISC 1996]

Intuition for 2-stgae algorithm

- The key idea:
 - Select k linearly independent columns from A
- Randomized Stage: The sampling scores are biased toward outlier columns
 - Leverage scores in the diagnostic regression analysis domain
 - Resistance scores in the graph sparsification domain
[Srivastava & Spielman, STOC 2008]
- Deterministic Stage: RRQR factorizations were explicitly designed to identify sets of linearly independent columns

Proof step by step

$S \in R^{n \times k}$ denotes a sampling operator such that

$$C = AS$$

The proof :

$$\begin{aligned}\|A - CC^+ A\|_2 &= \|A - AS(AS)^+ A\|_2 \\ &\leq \|A - AS(A_k S)^+ A_k\|_2 \\ &= \|(A_k + A_{\rho-k}) - (A_k + A_{\rho-k})S(A_k S)^+ A_k\|_2 \\ &\leq \|A_k - (A_k S)(A_k S)^+ A_k\|_2 + \|A_{\rho-k}\|_2 + \|A_{\rho-k}S(A_k S)^+ A_k\|_2\end{aligned}$$

$$\|A - CC^+ A\|_2 \leq \underbrace{\|A_k - (A_k S)(A_k S)^+ A_k\|_2}_1 + \underbrace{\|A_{\rho-k}\|_2}_2 + \underbrace{\|A_{\rho-k} S(A_k S)^+ A_k\|_2}_3$$

1. $\|A_k - (A_k S)(A_k S)^+ A_k\|_2 = \|A_k - I A_k\|_2 = 0$

(it is non - trivial to prove that $(A_k S)(A_k S)^+ = I$)

2. $\|A_{\rho-k}\|_2 = \|A - A_k\|_2$

(This trivial comes from the SVD of A)

3. $\|A_{\rho-k} S(A_k S)^+ A_k\|_2 \leq \|A_{\rho-k} S\|_2 \|(A_k S)^+ A_k\|_2$

(Standard property for matrix norms)

$$\|A_{\rho-k} S\|_2 \leq O((n-k)^{\frac{1}{4}}) \|A - A_k\|$$

(proof based on [Rudelson & Vershynin, JACM 2007])

$$\|(A_k S)^+ A_k\|_2 \leq O(\sqrt{k(k \log k - 1)})$$

(proof based on [Pan, LAA 2000])

Past work : Algorithm to bound $\|A - CC^+ A\|_2$

Deterministic algorithms

**Developed by the numerical linear algebra people
(G. Golub, G.W Stewart,..)**

**Best Result: The algorithm in [Gu & Eisenstat, SISC 1996]
runs in $O(mn^2)$ time and guarantees that**

$$\|A - CC^+ A\|_2 \leq O(\sqrt{1 + k(n - k)}) \|A - A_k\|_2$$

**Deep connections with the so-called Rank Revealing QR
(RRQR) factorization.**

Past work : Algorithm to bound $\|A - CC^+ A\|_F$

Randomized algorithms

**Developed by theoretical computer science people
(A. Frieze, R. Kannan, S. Vempala,..)**

**Best Result: The algorithm in [Deshpande & Vempala,
SODA 2006] runs in $O(mnk + kn)$ time and guarantees
that with high probability**

$$\|A - CC^+ A\|_F \leq \sqrt{(k+1)!} \|A - A_k\|_F$$

**Algorithm idea : Assign a probability to each column of
A, and then sample the columns of A based on these
probabilities.**

$$\text{Conclusion s : } \|A - CC^+ A\|_{\xi} \leq p(k,n) \|A - A_k\|_{\xi}$$

- Best Algorithm so far

| | Reference | $p(k,n)$ | Time |
|-----------|------------------------|---|-----------|
| $\xi = 2$ | [Gu,Eisenstat,96] | $O(k^{\frac{1}{2}}(n-k)^{\frac{1}{2}})$ | $O(mn^2)$ |
| $\xi = F$ | [Desphande,Vempala,06] | $\sqrt{(k+1)!}$ | $O(mnk)$ |

- Paper's Algorithm results

| | Reference | $p(k,n)$ | Time |
|-----------|-----------------|---|-----------|
| $\xi = 2$ | [This paper,09] | $O(k^{\frac{3}{4}}(n-k)^{\frac{1}{4}})$ | $O(mn^2)$ |
| $\xi = F$ | [This paper,09] | $O(k \sqrt{\log k})$ | $O(mn^2)$ |

Thanks