Subspace sampling and relative-error matrix approximation

Petros Drineas

Rensselaer Polytechnic Institute
Computer Science Department

(joint work with M. W. Mahoney)

For papers, etc.

YAHOO! drineas

The CUR decomposition

Carefully chosen U

$$\underbrace{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}}_{m \times n} \approx \begin{pmatrix} \\ C \\ \\ \end{pmatrix} \cdot \begin{pmatrix} U \end{pmatrix} \cdot \begin{pmatrix} R \end{pmatrix}$$

O(1) columns

O(1) rows

Goal: make (some norm) of A-CUR small.

Why? After making two passes over A, we can compute provably good C, U, and R and store them ("sketch") instead of A: O(m+n) vs. O(mn) RAM space.

Why? Given a sample consisting of a few columns (C) and a few rows (R) of A, we can compute U and "reconstruct" A as CUR.

If the sampling probabilities are not "too bad", we get provably good accuracy.

Why? Given sufficient time, we can find C, U and R such that A – CUR is almost optimal.

This might lead to improved data interpretation.

Overview

- Background & Motivation
- Relative error CX and CUR
- Open problems

Singular Value Decomposition (SVD)

$$\underset{m \times n}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} = \underset{m \times \rho}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \cdot \underset{\rho \times \rho}{\begin{pmatrix} \Sigma \end{pmatrix}} \cdot \underset{\rho \times n}{\begin{pmatrix} V \end{pmatrix}}^T$$

: rank of *A*

U (V): orthogonal matrix containing the left (right) singular vectors of *A*.

S: diagonal matrix containing the singular values of *A*.

Exact computation of the SVD takes $O(\min\{mn^2, m^2n\})$ time.

The top k left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.

Singular Value Decomposition (SVD)

$$\underset{m \times n}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} = \underset{m \times \rho}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \cdot \underset{\rho \times \rho}{\begin{pmatrix} \Sigma \end{pmatrix}} \cdot \underset{\rho \times n}{\begin{pmatrix} V \end{pmatrix}}^T$$

: rank of A

U (V): orthogonal matrix containing the left (right) singular vectors of A.

S: diagonal matrix containing the singular values of A.

Pseudoinverse of A:

$$A^+ = V\,S^{-1}\,U^T$$

Rank k approximations (A_k)

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

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

U_k (V_k): orthogonal matrix containing the top k left (right) singular vectors of A.

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

A_k is a matrix of rank k such that $||A-A_k||_F$ is minimized over all rank k matrices.

Definition: $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$

U_k and V_k

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

The rows of V_k^T are linear combinations of **<u>all rows</u>** of *A*

The columns of U_k are linear combinations of **<u>all columns</u>** of *A*

U_k (V_k): orthogonal matrix containing the top k left (right) singular vectors of *A*.

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

Potential problems with SVD

Structure in the data is *not* respected by mathematical operations on the data:

- Reification - maximum variance directions are just that.
- Interpretability - what does a linear combination of 6000 genes mean.
- Sparsity - is destroyed by orthogonalization.
- Non-negativity - is a convex and not linear algebraic notion.

Do there exist "better" low-rank matrix approximations?

- "better" structural properties for certain applications.
- "better" at respecting relevant structure.
- "better" for interpretability and informing intuition.

CUR for data interpretation

Exploit **structural properties of CUR** to analyze human genomic data:

n loci in the genome (SNPs)

m subjects

$$A \approx C \cdot U \cdot R$$

The data sets are not very large: low-poly(m,n) time is acceptable.

We seek subjects and SNPs that capture most of the diversity in the data:

- Singular vectors are useless; linear combinations of humans and/or SNPs make no biological sense.
- CUR extracts a low-dimensional representation in terms of subjects and SNPs.

Human genomic data: Paschou (Yale U), Mahoney (Yahoo! Research),…, Kidd (Yale U), & D. '06.

Prior work: additive error CUR

(D. & Kannan '03, D., Mahoney, & Kannan '05)

Let A_k be the "best" rank k approximation to A. Then, **after two passes through A,** we can pick O(k/ ^{4}) rows and O(k/ ^{4}) columns, such that

$$\|A - CUR\|_F \leq \|A - A_k\|_F + \varepsilon \|A\|_F >> \|A - A_k\|_F$$

Additive error is **prohibitively large** in data analysis applications!

This "coarse" CUR **does not capture** the relevant structure in the data.

Theorem: relative error CUR

(D., Mahoney, & Muthukrishnan '05, '06)

For any k, $O(SVD_k(A))$ time suffices to construct C, U, and R s.t.

$$\|A - CUR\|_F \leq (1+\varepsilon)\left\|A - U_k \Sigma_k V_k^T\right\|_F = (1+\varepsilon)\|A - A_k\|_F$$

holds with probability at least 1- , by picking

O(k log k log(1/) / ^{2}) columns, and

O(k log^{2}k log(1/) / ^{6}) rows.

$O(SVD_k(A))$: time to compute the top k left/right singular vectors and values of A.

Applications: relative error CUR

Evaluation on:

- Microarray data (yeast), a (roughly) 6200 £ 24 matrix. (from O. Alter, UT Austin)
- Genetic marker data, 38 matrices, each (roughly) 60 £ 65 (with P. Paschou, Yale U.)
- HapMap SNP data, 4 matrices, each (roughly) 70 £ 800 (with P. Paschou, Yale U.)

For (small) k, in $O(SVD_k(A))$ time we can construct C, U, and R s.t.

$$\|A - CUR\|_F \;\leq\; (1 + .001)\,\|A - A_k\|_F$$

by typically picking at most (k+5) columns and at most (k+5) rows.

CX matrix decompositions

Create an approximation to A using columns of A

$$\begin{pmatrix} \\ & A & \\ \\ \end{pmatrix} \approx \begin{pmatrix} \\ C \\ \\ \end{pmatrix} \cdot \begin{pmatrix} & X & \end{pmatrix}$$

c=O(1) columns

Goal: Provide almost optimal bounds for some norm of A – CX.

1. How do we draw the columns of A to include in C?
2. How do we construct X? One possibility is

$$\min_{X \in \mathcal{R}^{c \times n}} \|A - CX\|_F = \|A - C\left(C^{+}A\right)\|_F$$

Subspace sampling

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

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

U_k (V_k): orthogonal matrix containing the top k left (right) singular vectors of A.

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

Subspace sampling

$$\begin{pmatrix} \\ V_k \\ \\ \end{pmatrix}$$

$$n \times k$$

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

Note: The columns of V_k are orthonormal vectors, **BUT** the rows of V_k (notation: $(V_k)_{(i)}$) are not orthonormal vectors.

<u>Subspace sampling</u> in $O(SVD_k(A))$ time

$$\forall i = 1, 2, \ldots, n \qquad p_i = \frac{\left\| (V_k)_{(i)} \right\|_2^2}{k}$$

Relative-error CX decomposition

Relative-error CX decomposition

- Compute the probabilities p_i;
- For each i = 1,2,…,n, pick the i-th column of A with probability $\min\{1, cp_i\}$
- Let C be the matrix containing the sampled columns;

(C has ·c columns in expectation)

Theorem: For any k, let A_k be the "best" rank k approximation to A.

In $O(SVD_k(A))$ we can compute p_i such that if c = O(k log k / ^{2}) then, with probability at least 1- ,

$$\min_{X \in \mathcal{R}^{\tilde{c} \times n}} \|A - CX\|_F = \|A - CC^+ A\|_F$$

$$\leq (1+\varepsilon)\|A - A_k\|_F$$

Inside subspace sampling

Let C = AS, where S is a sampling/rescaling matrix and let the SVD of A be A = U_A S_A V_A^T. Then,

$$\begin{aligned} A - C\left(C^{+}A\right) &= A - A\mathcal{S}\left(A\mathcal{S}\right)^{+}A \\ &= A - U_A\Sigma_A V_A^T\mathcal{S}\left(U_A\Sigma_A V_A^T\mathcal{S}\right)^{+}A \end{aligned}$$

$$\left\|A - C\left(C^{+}A\right)\right\|_F = \left\|\Sigma_A - \Sigma_A V_A^T\mathcal{S}\left(\Sigma_A V_A^T\mathcal{S}\right)^{+}\Sigma_A\right\|_F$$

Submatrices of orthogonal matrices

Important observation: our subspace sampling probabilities guarantee that SV_A is a full-rank, approx. orthogonal matrix:

$$(SV_A)^T (SV_A) ¼ I.$$

(Frieze, Kannan, Vempala '98, D., Kannan, Mahoney '01, 04', Rudelson, Virshyin '05 and even earlier by Bourgain, Kashin, and Tzafriri using uniform sampling.)

This property allows us to **completely capture** the subspace spanned by the top k right singular vectors of A.

Relative-error CX & low-rank approximations

November 2005: Drineas, Mahoney, and Muthukrishnan

- First relative-error CX matrix factorization algorithm.
- $O(SVD_k(A))$ time and $O(k^2)$ columns.

January 2006: Har-Peled

- $O(mn\, k^2 \log k)$ - "linear in mn" time to get 1+ approximation.

March 2006: Deshpande and Vempala

- $O(k \log k)$ passes, $O(Mk^2)$ time and $O(k \log k)$ columns.

April 2006: Drineas, Mahoney, and Muthukrishnan

- Improved the DMM November 2005 result to $O(k \log k)$ columns.

April 2006: Sarlos

- Relative-error low-rank approximation in just two passes with $O(k \log k)$ columns, after some preprocessing.

Relative-error CUR decomposition

Create an approximation to *A*, using rows and columns of *A*

$$A \approx C \cdot U \cdot R$$

Carefully chosen U

O(1) columns

O(1) rows

<u>Goal:</u> Provide very good bounds for some norm of *A* – CUR.

1. How do we draw the columns and rows of *A* to include in *C* and R?
2. How do we construct U?

Step 1: subspace sampling for C

Relative-error CX decomposition **(given A, construct C)**

- Compute the probabilities p_i;
- For each i = 1,2,…,n, pick the i-th column of A with probability $\min\{1, cp_i\}$
- Let C be the matrix containing the sampled columns;

(C has ·c columns in expectation)

Subspace sampling for R

$$\underset{m \times \tilde{c}}{\begin{pmatrix} \\ C \\ \\ \end{pmatrix}} = \underset{m \times \rho}{\begin{pmatrix} \\ U_C \\ \\ \end{pmatrix}} \cdot \underset{\rho \times \rho}{\begin{pmatrix} \Sigma_C \end{pmatrix}} \cdot \underset{\rho \times \tilde{c}}{\begin{pmatrix} V_C \end{pmatrix}}^T$$

U_C : orthogonal matrix containing the left singular vectors of *C*.

: **rank** of *C*.

Let $(U_C)_{(i)}$ denote the i-th row of U.

Subspace sampling for R

$$\underbrace{\begin{pmatrix} \\ U_C \\ \\ \end{pmatrix}}_{m \times \rho}$$

U_C : orthogonal matrix containing the left singular vectors of *C*.

: **rank** of *C*.

Let $(U_C)_{(i)}$ denote the i-th row of U.

<u>Subspace sampling</u> in $O(c^2 m)$ time

$$\forall i = 1, 2, \ldots, m \qquad q_i = \frac{\left\|(U_C)_{(i)}\right\|_2^2}{\rho}$$

Step 2: constructing U and R

Relative-error CX decomposition **(given A, construct C)**

- Compute the probabilities p_i;
- For each i = 1,2,…,n, pick the i-th column of A with probability $\min\{1, cp_i\}$;
- Let C be the matrix containing the sampled columns;

(C has ·c columns in expectation)

CUR Algorithm **(given A and C, return U and R)**

- Compute the probabilities q_i;
- For each i = 1,2,…,m pick the i-th row of A with probability $\min\{1, rq_i\}$;
- **Let R** be the matrix containing the sampled rows;
- Let W be the intersection of C and R;
- **Let U** be a (rescaled) pseudo-inverse of W;

(R has · r rows in expectation)

Overall decomposition

columns of A

diagonal rescaling matrix

rows of A

$$A \approx C \cdot \left[D \left(W \right) \right]^{+} D \cdot \left(R \right)$$

$$m \times n \qquad m \times \tilde{c} \qquad \tilde{r} \times n$$

"intersection" of C and R

$$\tilde{r} \times \tilde{c}$$

Analyzing Step 2 of CUR

CUR Algorithm (given A and C, return U and R)

- Compute the probabilities q_i;
- For each i = 1,2,…,m pick the i-th row of A with probability $\min\{1, rq_i\}$;
- **Let R** be the matrix containing the sampled rows;
- Let W be the intersection of C and R;
- **Let U** be a (rescaled) pseudo-inverse of W;

(R has · r rows in expectation)

Theorem: Given C, in $O(c^2 m)$ time, we can compute q_i such that

$$\left\| A - C \underbrace{(DW)^{+} D}_{U} R \right\|_F \leq (1+\epsilon) \left\| A - C\left(C^{+}A\right) \right\|_F$$

holds with probability at least 1- , if r= O(c log c / ^{2}) rows.

Putting the two theorems together

Thm 1: For any k, let A_k be the "best" rank k approximation to A.

Then, in $O(SVD_k(A))$ we can pick (in expectation) $c = O(k \log k / \ ^2)$ columns of A such that, with probability at least 1- ,

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

Thm 2: Given A and C, in $O(c^2 m)$ time, we can pick (in expectation) $r = O(c \log c / \ ^2)$ rows of A such that, with probability at least 1- ,

$$\left\|A - C\underbrace{(DW)^{+}D}_{U}R\right\|_F \;\leq\; (1+\epsilon)\left\|A - C\left(C^{+}A\right)A\right\|_F$$

Relative error CUR

For any k, O(SVD$_k$(A)) time suffices to construct C, U, and R s.t.

$$\left\| A - C \underbrace{(DW)^{+} D}_{U} R \right\|_F \leq (1+\varepsilon) \left\| A - A_k \right\|_F$$

holds with probability at least 1- , by picking

O(k log k / ^{2}) columns, and

O(k log^{2}k / ^{6}) rows.

CUR decompositions: a summary

G.W. Stewart (Num. Math. '99, TR '04)	C: variant of the QR algorithm R: variant of the QR algorithm U: minimizes $\|\|A\text{-}CUR\|\|_F$	No a priori bounds Solid experimental performance
Goreinov, Tyrtyshnikov, & Zamarashkin (LAA '97, Cont. Math. '01)	C: columns that span max volume U: W^+ R: rows that span max volume	Existential result Error bounds depend on $\|\|W^+\|\|_2$ Spectral norm bounds!
Williams & Seeger (NIPS '01)	C: uniformly at random U: W^+ R: uniformly at random	Experimental evaluation A is assumed PSD Connections to Nystrom method
D., Kannan, & Mahoney (SODA '03, '04)	C: w.r.t. column lengths U: in linear/constant time R: w.r.t. row lengths	Randomized algorithm Provable, a priori, bounds Explicit dependency on $A - A_k$
D., Mahoney, & Muthukrishnan ('05, '06)	C: depends on singular vectors of A. U: (almost) W^+ R: depends on singular vectors of C	(1+) approximation to $A - A_k$ Computable in $SVD_k(A)$ time.

Open problem

Is it possible to construct a CUR decomposition satisfying bounds similar to ours deterministically?

• Gu and Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization", SIAM J. Sci. Computing, 1996.

Main algorithm: there exist k columns of A, forming a matrix C, such that the smallest singular value of C is "large".

We can find such columns in $O(mn^2)$ time *deterministically*!