Fast low rank approximations of matrices and tensors

S. Friedland, V. Mehrmann, A. Miedlar and M. Nkengla

Univ. Illinois at Chicago & Technische Universität Berlin

Gene Golub memorial meeting, Berlin, February 29, 2008

Overview

- Statement of the problem
- Matrix SVD
- Best rank k-approximation
- Random low rank approximations
- CUR approximation I – VII
- Nonnegative CUR-approximation
- Algorithm for $\tilde{U}_{opt}$
- Extension to 3-tensors I–III
- Simulations
- Conclusions

Statement of the problem

Data is presented in terms of a matrix

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}.$$

Examples

1. digital picture: 512×512 matrix of pixels,
2. DNA-microarrays: $60,000 \times 30$
   (rows are genes and columns are experiments),
3. web pages activities:
   $a_{i,j}$-the number of times webpage j was accessed from web page i.

Objective: condense data and store it effectively.

Matrix SVD

$$A = U\Sigma V^\top$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_n \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in \mathbb{R}^{m\times n}, \quad U \in \mathbb{O}(m), \quad V \in \mathbb{O}(n)$$

Best rank k-approximation

For $k \leq r = \mathrm{rank}\, A$ let $\Sigma_k = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_k \end{bmatrix} \in \mathbb{R}^{k\times k}$

$A_k := U_k \Sigma_k V_k^*$ is the best rank k approximation in Frobenius and operator norm of A

$$\min_{B\in\mathcal{R}(m,n,k)} ||A - B||_F = ||A - A_k||_F.$$

Reduced SVD $A = U_r \Sigma_r V_r^*$ where $(r \geq)\ \nu$ numerical rank of A if

$$\frac{\sum_{i\geq\nu+1} \sigma_i^2}{\sum_{i\geq 1} \sigma_i^2} \approx 0, (0.01).$$

A_ν is a noise reduction of A. Noise reduction has many applications in image processing, DNA-Microarrays analysis, data compression.
Full SVD: $O(mn\min(m, n))$, k- SVD: $O(kmn)$.

Random low rank approximations

Find a good algorithm by reading l rows or columns of A at random and update the approximations.

- Frieze-Kannan-Vempala FOCS 1998 suggest algorithm assuming a probabilistic distribution of entries of A, without updating. (Later versions included several passes of the algorithm with updating.)
- Friedland-Kaveh-Niknejad-Zare [2] proposed randomized k-rank approximation by reading l rows or columns of A at random and updating the approximations.

The main feature of this algorithm is that each update is a better rank k-approximation.

Each iteration: $||A - B_{t-1}||_F \geq ||A - B_t||_F$.
Complexity $O(kmn)$.

CUR approximation-I

From $A \in \mathbb{R}^{m\times n}$ choose submatrices consisting of p-columns $C \in \mathbb{R}^{m\times p}$ and q rows $R \in \mathbb{R}^{q\times n}$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ a_{4,1} & a_{4,2} & a_{4,3} & \dots & a_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \\ a_{m-1,1} & a_{m-1,2} & a_{m-1,3} & \dots & a_{m-1,n} \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,n} \end{bmatrix},$$

R - red - blue, $\quad C$ - red - magenta.

Approximate A using C, R.

CUR approximation-II

Let $\langle m \rangle := \{1, \ldots, m\}$, $\langle n \rangle := \{1, \ldots, n\}$.
For $A = [a_{i,j}] \in \mathbb{R}^{m \times n}$ we define

$$\|A\|_{\infty,e} := \max_{i \in \langle m \rangle, j \in \langle n \rangle} |a_{i,j}|.$$

Let $I = \{1 \leq \alpha_1 < \ldots < \alpha_q \leq m\}$, $J = \{1 < \beta_1 < \ldots < \beta_p \leq n\}$.

$$A_{I,J} := [a_{i,j}]_{i \in I, j \in J},$$

$$R = A_{I,\langle n \rangle} = [a_{\alpha_k,j}],\ k = 1, \ldots, q, \quad j = 1, \ldots, n,$$

$$C = A_{\langle m \rangle,J} = [a_{i,\beta_l}],\ i = 1, \ldots, m, \quad l = 1, \ldots, p.$$

The set of read entries of A

$$\mathcal{S} := \langle m \rangle \times \langle n \rangle \backslash ((\langle m \rangle \backslash I) \times (\langle n \rangle \backslash J)),\ \#\mathcal{S} = mp + qn - pq.$$

Goal: approximate A by CUR for appropriately chosen C, R and $U \in \mathbb{R}^{p \times q}$.

CUR-approximation III

CUR approximation introduced by Goreinov, Tyrtyshnikov and Zmarashkin [7, 8].
Suppose that $A, F \in \mathbb{R}^{m\times n}$ and $\mathrm{rank}\,(A-F) \le p$. Then there exists p rows and columns of A: $R \in \mathbb{R}^{p\times n}$, $C \in \mathbb{R}^{m\times p}$and $U \in \mathbb{R}^{p\times p}$ such that

$$||A - CUR||_2 \le ||F||_2(1 + 2\sqrt{pn} + 2\sqrt{pm})$$

Good choice of C, R, U: Goreinov and Tyrtyshnikov[6]:

$$\mu_p := \max_{I\subset\langle m\rangle, J\subset\langle n\rangle, \#I=\#J=p} |\det A_{I,J}| > 0.$$

Suppose that
$|\det A_{I,J}| \ge \delta\mu_p, \delta \in (0,1], I \subset \langle m\rangle, J \subset \langle n\rangle, \#I = \#J = p.$
Then

$$||A - CA_{I,J}^{-1}R||_{\infty,e} \le \frac{p+1}{\delta}\sigma_{p+1}(A).$$

CUR-approximations: IV

Random A_k approximation algorithm:

II. Find a good algorithm by reading q rows and p columns of A at random and update the approximations. View the rows and the columns read as the corresponding matrices

$$R \in \mathbb{R}^{q\times n}, C \in \mathbb{R}^{m\times p}.$$

Then a low rank approximation is of the form

$$A_k = CUR,$$

for a properly chosen $U \in \mathbb{R}^{p\times q}$.

Drineas, Kannan and Mahoney gave in [1] a way to compute U using a probabilistic distribution of entries of A, without updating, or 2-3 updates.

The aim of this talk is to introduce an algorithm for finding an optimal U_{opt}, corresponding to $F := CU_{\mathrm{opt}}R$ and an optimal k-rank approximation B of F, if needed by updating the approximations.

Complexity $O(k^2 \max(m, n))$.

CUR-approximations: V

$$U_{\text{opt}} \in \arg \min_{U \in \mathbb{R}^{p \times q}} \sum_{(i,j) \in \mathcal{S}} (a_{i,j} - (CUR)_{i,j})^2,$$

$$F = CU_{\text{opt}} R.$$

If $\min(p, q) \le k$ then $B = F$

If $\min(p, q) > k$ then B is best rank k-approximation of F using SVD of F.

Average error

$$\text{Error}_{\text{av}}(B) = \Big(\frac{1}{\#\mathcal{S}} \sum_{(i,j) \in \mathcal{S}} (a_{i,j} - b_{,ij})^2\Big)^{\frac{1}{2}}.$$

Read additional rows and columns of A. Repeat the above process to obtain new k-rank approximation B_1 of A. Continue until either the error $\text{Error}_{\text{av}}(B_k)$ stabilizes, or we exceeded the allowed number of computational work.

CUR-approximations: VI

Given A the best choice of U is

$$U_b \in \arg \min_{U \in \mathbb{R}^{p \times q}} ||A - CUR||_F,$$

$$U_b = C^\dagger A R^\dagger,$$

$X^\dagger$ Moore-Penrose inverse of $X \in \mathbb{R}^{m \times n}$.
Complexity: $O(pqmn)$.
In [3] we characterize for $r \leq \min(p, q)$

$$U_{b,r} \in \arg \min_{U \in \mathcal{C}_r(p,q)} ||A - CUR||_F.$$

SVD of CUR can be fast found by Golub-Reinsch SVD algorithm.
Complexity: $O(rpq \max(m, n))$.

Least squares solution

$$U_{\text{opt}} \in \arg \min_{U \in \mathbb{R}^{p \times q}} \sum_{(i,j) \in \mathcal{S}} (a_{i,j} - (CUR)_{i,j})^2.$$

Example:

Cameraman: $n = m = 256$, $p = q = 80$.
Number of variables: $pq = 6400$.
Number of equations: $2 \times 256 \times 80 - 6400 = 34,560$.

Problems with executing least squares with Matlab:
very long time of execution time and poor precision.

Nonnegative *CUR*-approximation

$A \geq 0$: entries of A are nonnegative

$$U_{\mathrm{opt}} \in \arg \min_{U \in \mathbb{R}^{p \times q}} \sum_{(i,j) \in \mathcal{S}} (a_{i,j} - (CUR)_{i,j})^2,$$

subject to constrains: $(CUR)_{i,j} \geq 0, \ (i,j) \in \mathcal{S}$.
Or

$$U_b \in \arg \min_{U \in \mathbb{R}^{p \times q}} ||A - CUR||_F,$$

subject to constrains: $(CUR) \geq 0$.

Minimization of strictly convex quadratic function in a convex polytope.

Algorithm for $\tilde{U}_{\mathrm{opt}}$

Thm $U_{\mathrm{opt}} = A^{\dagger}_{I,J}$.
Suppose that $\#I = \#J = p$ and $A_{I,J}$ is invertible. Then $U_{\mathrm{opt}} = A^{-1}_{I,J}$ is the exact solution of the least square problem

$$(CUR)_{I,\langle n\rangle} = A_{I,\langle n\rangle}, \ (CUR)_{\langle m\rangle,J} = A_{\langle m\rangle,J},$$

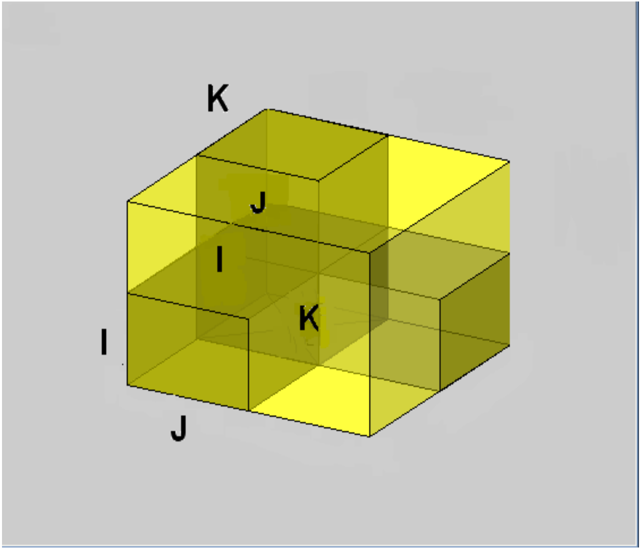
back to Goreinov-Tyrtyshnykov.
Instead of finding $A_{I,J}$ with maximum determinant we try several $I \subset \langle m\rangle, J \subset \langle n\rangle, \#I = \#J = p$, from which we chose the best I, J:

- $A_{I,J}$ has maximal numerical rank r_p,
- $\prod_{i=1}^{r_p} \sigma_i(A_{I,J})$ is maximal.

$$\tilde{U}_{\mathrm{opt}} := A^{\dagger}_{I,J,r_p}$$

A_{I,J,r_p} is the best rank r_p approximation of A_{IJ}.
A is approximated by $C\tilde{U}_{\mathrm{opt}}R$.

Extension to tensors: I



Extensions to tensors: II

$\mathcal{A} = [a_{i,j,k}] \in \mathbb{R}^{m\times n\times \ell}$ - 3-tensor
given $I \subset \langle m\rangle, J \subset \langle n\rangle, K \subset \langle \ell\rangle$ define
$R := \alpha_{\langle m\rangle,J,K} = [a_{i,j,k}]_{\langle m\rangle,J,K} \in \mathbb{R}^{m\times(\#J\cdot\#K)}$,
$C := \alpha_{I,\langle n\rangle,K} \in \mathbb{R}^{\langle n\rangle\times(\#I\cdot\#K)}$,
$D := \alpha_{I,J,\langle \ell\rangle} \in \mathbb{R}^{l\times(\#I\cdot\#J)}$
Problem: Find 3-tensor $\mathcal{U} =\in \mathbb{R}^{(\#J\cdot\#K)\times(\#I\cdot\#K)\times(\#I\cdot\#J)}$
such that $\mathcal{A}$ is approximated by the Tucker tensor
$\mathcal{V} = \mathcal{U} \times_1 C \times_2 \times_3 D$
where $\mathcal{U}$ is the least squares solution

$$\mathcal{U}_{\rm opt} \in \arg \min_{\mathcal{U}\in\mathbb{R}^{\rm three\ tensor}} \sum_{(i,j,k)\in\mathcal{S}} \left(a_{i,j,k} - (\mathcal{U} \times_1 C \times_2 \times_3 D)_{i,j,k}\right)^2$$

$\mathcal{S} = (\langle m\rangle \times J \times K) \cup (I \times \langle n\rangle \times K) \cup (I \times J \times \langle \ell\rangle)$

Extension to tensors: III

For $\#I = \#J = p,\ \#K = p^2,\ I \subset \langle m\rangle, J \subset \langle n\rangle, K \subset \langle \ell\rangle$
generically there is an exact solution to
$\mathcal{U}_{\rm opt} \in \mathbb{R}^{p^3\times p^3\times p^2}$
obtained by flattening:
View $\mathcal{A}$ as $A \in \mathbb{R}^{(mn)\times\ell}$ by identifying
$\langle m\rangle \times \langle n\rangle \equiv \langle mn\rangle,\ I_1 = I \times J,\ J_1 = K$ and apply CUR again.

Symmetric situation for 4-tensors $\mathcal{A} \in \mathbb{R}^{m\times n\times l\times q}$.
Choose $I \subset \langle m\rangle, J \subset \langle n\rangle, K \subset \langle \ell\rangle, L \subset \langle q\rangle$
$\#I = \#J = \#K = \#L = p$
$\mathcal{V}$ the Tucker approximation
$\mathcal{V} = \mathcal{U} \times_1 C \times_2 R \times_3 D \times_4 H$
$\mathcal{U} \in \mathbb{R}^{p^3\times p^3\times p^3\times p^3}$ is the least squares solution
$\mathcal{U}_{\rm opt} \in \arg\min_{\mathcal{U}\in\mathbb{R}^{\text{four tensor}}} \sum_{(i,j,k,r)\in\mathcal{S}} \left(a_{i,j,k,r} - (\mathcal{U}\times_1 C\times_2 R\times_3 D\times_4 H)_{i,j,k,r}\right)^2$
At least 6 solutions in generic case.
More solutions?

Simulations: Tire I

Figure: Tire image compression (a) original, (b) SVD approximation, (c) CLS approximation, $t_{\max} = 100$.

Figure 1 portrays the original image of the Tire picture from the Image Processing Toolbox of MATLAB, given by a matrix $A \in \mathbb{R}^{205 \times 232}$ of rank 205, the image compression given by the SVD (using the MATLAB function `svds`) of rank 30 and the image compression given by $B_b = CU_bR$.

Simulations: Tire II

The corresponding image compressions given by the approximations B_{opt_1}, B_{opt_2} and $\tilde{B}_{opt}$ are displayed respectively in Figure 2. Here, $t_{\max} = 100$ and $p = q = 30$. Note that the number of trials $t_{\max}$ is set to the large value of 100 for all simulations in order to be able to compare results for different (small and large) matrices.

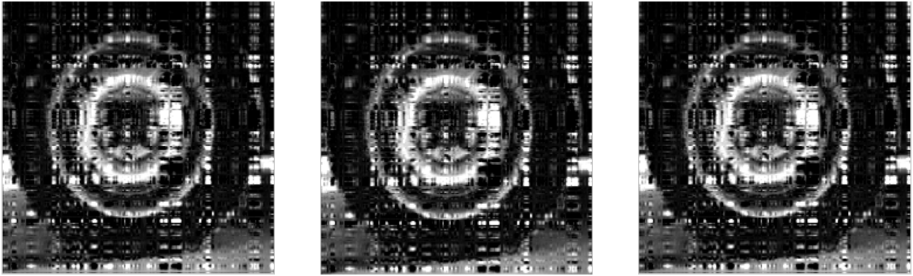


Figure: Tire image compression with (a) B_{opt_1}, (b) B_{opt_2}, (c) $\tilde{B}_{opt}$, $t_{\max} = 100$.

Simulations: Table 1

In Table 1 we present the S-average and total relative errors of the image data compression. Here, $B_b = CU_bR$, $B_{opt_2} = CU_{opt_2}R$ and $\tilde{B}_{opt} = C\tilde{U}_{opt}R$. Table 1 indicates that the less computationally costly FSVD with B_{opt_1}, B_{opt_2} and $\tilde{B}_{opt}$ obtains a smaller S-average error than the more expensive complete least squares solution CLS and the SVD. On the other hand, CLS and the SVD yield better results in terms of the total relative error. However, it should be noted that CLS is very costly and cannot be applied to very large matrices.

	rank	SAE	TRE
B_{svd}	30	0.0072	0.0851
B_b	30	0.0162	0.1920
B_{opt_1}	30	$1.6613 \cdot 10^{-26}$	0.8274
B_{opt_2}	30	$3.2886 \cdot 10^{-29}$	0.8274
$\tilde{B}_{opt}$	30	$1.9317 \cdot 10^{-29}$	0.8274

Table: Comparison of rank, S-average error and total relative error.

Simulations: Cameraman 1

Figure 3 shows the results for the compression of the data for the original image of a camera man from the Image Processing Toolbox of MATLAB. This data is a matrix $A \in \mathbb{R}^{256\times 256}$ of rank 253 and the resulting image compression of rank 69 is derived using the SVD and the complete least square approximation CLS given by $B_b = CU_bR$.

Figure: Camera man image compression (a) original, (b) SVD approximation, (c) CLS approximation, $t_{\max} = 100$.

Simulations: Cameraman 2

Figure 4 is FSVD approximation $B_{opt_2} = CU_{opt_2}R$ and $\tilde{B}_{opt} = C\tilde{U}_{opt}R$. Here $t_{\max} = 100$ and $p = q = 80$. Table 2 gives S-average and total relative errors.

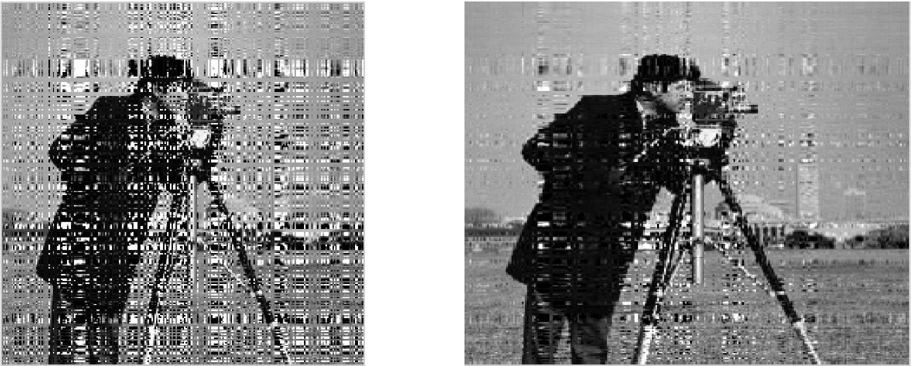


Figure: Camera man image compression. FSVD approximation with (a) $B_{opt_2} = CU_{opt_2}R$, (b) $\tilde{B}_{opt} = C\tilde{U}_{opt}R$. $t_{\max} = 100$.

Simulations: Table 2

	rank	SAE	TRE
B_{svd}	69	0.0020	0.0426
B_b	80	0.0049	0.0954
B_{opt_1}	—	—	—
B_{opt_2}	80	$3.7614 \cdot 10^{-27}$	1.5154
$\tilde{B}_{opt}$	69	$7.0114 \cdot 10^{-4}$	0.2175

Table: Comparison of rank, S-average error and total relative error.

Canal at night 1

Figure: Canal image (a) original, (b) SVD approximation, $t_{\max} = 100$.

Canal at night 2

Figure: Canal image compression (a) CLS approximation, (b) FSVD with $\tilde{B}_{opt}$, $t_{\max} = 100$.

Conclusions

Fast low rank approximation using *CUR* approximation of A of dimension $m \times n$, $C \in \mathbb{R}^{m\times p}$, $R \in \mathbb{R}^{q\times n}$ are submatrices of A $U \in \mathbb{R}^{p\times q}$ computable by least squares to fit best the entries of C and R.
Advantage: low complexity $O(pq\max(m,n))$.
Disadvantage: problems with computation time and approximation error
Drastic numerical improvement when using $\tilde{U}_{\mathrm{opt}}$.
Least squares can be straightforward generalized to tensors

References

- P. Drineas, R. Kannan, and M. W. Mahoney, Fast Monte Carlo algorithms for matrices I-III: computing a comppressed approximate matrix decompositon, *SIAM J. Comput.* 36 (2006), 132 – 206.
- S. Friedland, M. Kaveh, A. Niknejad and H. Zare, Fast Monte-Carlo Low Rank Approximations for Matrices, *Proc. IEEE SoSE*, 2006, 218 – 223.
- S. Friedland and A. Torokhti, Generalized rank-constrained matrix approximations, *SIAM J. Matrix Anal. Appl.* 29 (2007), 656 – 659.
- A. Frieze, R. Kannan and S. Vempala, Fast Monte-Carlo algorithms for finding low rank approximations, *Proceedings of the 39th Annual Symposium on Foundation of Computer Science*, 1998.
- G.H. Golub and C.F. Van Loan, *Matrix Computation*, John Hopkins Univ. Press, 3rd Ed., 1996.
- S.A. Goreinov and E.E. Tyrtyshnikov, The maximum-volume concept in approximation by low-rank matrices, *Contemporary Mathematics* 280 (2001), 47-51.
- S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zmarashkin, Pseudo-skeleton approximations of matrices, *Reports of the Russian Academy of Sciences* 343(2) (1995), 151-152.
- S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zmarashkin, A theory of pseudo-skeleton approximations of matrices, *Linear Algebra Appl.* 261 (1997), 1-21.