Fast Monte-Carlo Low Rank Approximations for Matrices

Shmuel Friedland
University of Illinois at Chicago

joint work with M. Kaveh, A. Niknejad and H. Zare

IEEE SoSE 2006, LA, April 25, 2006

http://www.math.uic.edu/∼friedlan

1 Statement of the problem

Data is presented in terms of a matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

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_{ij}-the number of times webpage j was accessed from web page i

Object: condense data and storage it effectively

2 Matrix SVD

Let $A \in \mathbb{C}^{m\times n}$. Then $A : \mathbb{C}^n \to \mathbb{C}^m$. Assume $\mathbb{C}^n, \mathbb{C}^m$ equipped with standard inner product $\langle \mathbf{x}, \mathbf{y}\rangle := \mathbf{y}^*\mathbf{x}$.

Then $A = U\Sigma V^*$, where $U \in \mathrm{U}(m), V \in \mathrm{U}(n)$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)}) \in \mathbb{R}_+^{m\times n}$.
U, V transition matrices from $[\mathbf{u}_1, \ldots, \mathbf{u}_m], [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ to the standard bases in $\mathbb{C}^m, \mathbb{C}^n$ respectively.

For $k \le r$ let $\Sigma_k = diag(\sigma_1, \ldots, \sigma_k) \in \mathbb{R}^{k\times k}$, and $U_k \in \mathrm{U}(m, k), V_k \in \mathrm{U}(n, k)$ having the first k columns of U, V respectively. Then $A_k := U_k\Sigma_k V_k^*$ the best rank k approximation in Frobenius and operator norm of A:
$\min_{B\in\mathcal{R}(m,n,k)} ||A - B|| = ||A - A_k||$.

$A = U_r\Sigma_r V_r^*$ is Reduced SVD

$(r \ge)\ \nu$ numerical rank of A if $\frac{\sigma_{\nu+1}}{\sigma_\nu} \approx 0$.

A_ν is a noise reduction of A.

Noise reduction has many applications in image processing, DNA-Microarrays analysis, data compression.

3 SVD in inner product spaces

$\mathbf{U}_i$ is m_i-dimensional IPS over $\mathbb{C}$, with $\langle\cdot,\cdot\rangle_i, i = 1, 2$.
$T : \mathbf{U}_1 \to \mathbf{U}_2$ linear operator. $T^* : \mathbf{U}_2 \to \mathbf{U}_1$ the adjoint operator: $\langle T\mathbf{x}, \mathbf{y}\rangle_2 = \langle \mathbf{x}, T^*\mathbf{y}\rangle_1$.
$S_1 := T^*T : \mathbf{U}_1 \to \mathbf{U}_1$,
$S_2 := TT^* : \mathbf{U}_2 \to \mathbf{U}_2$.

S_1, S_2 self-adjoint: $S_1^* = S_1, S_2^* = S_2$ and nonnegative definite: $\langle S_i\mathbf{x}_i, \mathbf{x}_i\rangle_i \ge 0$.

$\sigma_1^2 \ge ... \ge \sigma_r^2 > 0$ positive eigenvalues of S_1 and S_2 and $r = \mathrm{rank}\, T = \mathrm{rank}\, T^*$. Let
$S_1\mathbf{v}_i = \sigma_i^2\mathbf{v}_i,\ \langle \mathbf{v}_i, \mathbf{v}_j\rangle_1 = \delta_{ij},\ i, j =, 1, ..., r$.
Define $\mathbf{u}_i := \sigma_i^{-1}T\mathbf{v}_i, i = 1, ..., r$. Then
$\langle \mathbf{u}_i, \mathbf{u}_j\rangle_2 = \delta_{ij}, i, j = 1, \ldots, r$.

Complete $\{\mathbf{v}_1, ..., \mathbf{v}_r\}$ and $\{\mathbf{u}_1, ..., \mathbf{u}_r\}$ to orthonormal bases $[\mathbf{v}_1, ..., \mathbf{v}_{m_1}]$ and $[\mathbf{u}_1, ..., \mathbf{u}_{m_2}]$ in $\mathbf{U}_1$ and $\mathbf{U}_2$.

4 RANDOM k-SVD

Stable numerical algortihms of SVD introduced by Golub-Kahan 1965, Golub-Reinsch 1970:

Implicit QR Algo to reduce to upper bidiagonal form using Householder matrices, then Golub-Reinsch SVD algo to zero superdiagonal elements.

Complexity: $O(mn\min(m,n))$.

In applications for massive data:
$A \in \mathbb{R}^{m\times n}, m, n >> 1$ needed a good approximation $A_k = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{y}_i^T, \mathbf{x}_i \in \mathbb{R}^m, \mathbf{y}_i \in \mathbb{R}^n, i = 1, \ldots, k << \min(m,n)$.

Random A_k approximation algo:

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

Frieze-Kannan-Vempala FOCS 1998 suggest algo without updating.

5 FKNZ RANDOM ALGO [4]

Fast k-rank approximation and SVD algorithm

Input: positive integers m, n, k, l, N, $m \times n$ matrix A, $\epsilon > 0$.

Output: an $m \times n$ k-rank approximation B_f of A, with the ratios $\frac{||B_0||}{||B_t||}$ and $\frac{||B_{t-1}||}{||B_t||}$, approximations to k-singular values and k left and right singular vectors of A.

1. Choose k-rank approximation B_0 using k columns, (or rows), of A.

2. **for** $t = 1$ **to** N

- Select l columns, (or rows), from A at random and update B_{t-1} to B_t.

- Compute the approximations to k-singular values, and k left and right singular vectors of A.

- If $\frac{||B_{t-1}||}{||B_t||} > 1 - \epsilon$ let $f = t$ and finish.

Complexity: $O(mnk)$.

Each iteration $||A - B_{t-1}||_F \geq ||A - B_t||_F$.

6 DETAILS

Choose at random k columns of A. Apply modified Gram-Schmidt algo to obtain $\mathbf{x}_1, \ldots, \mathbf{x}_q \in \mathbb{R}^m, q \leq k$.
Set $B_0 := \sum_{i=1}^{q} \mathbf{x}_i (A^{\mathrm{T}} \mathbf{x}_i)^{\mathrm{T}}$.
$||A - B_0||_F^2 = \mathbf{tr}\, A^T A - \mathbf{tr}\, B_0^T B_0 =$
$\mathbf{tr}\, A^T A - \sum_{i=1}^{q} (A^{\mathrm{T}} \mathbf{x}_i)^{\mathrm{T}} (A^{\mathrm{T}} \mathbf{x}_i)$.

Choose at random another l columns of A: $\mathbf{w}_1, \ldots, \mathbf{w}_l$.
Apply modified Gram-Schmidt algo to
$\mathbf{x}_1, \ldots, \mathbf{x}_q, \mathbf{w}_1, \ldots, \mathbf{w}_l$ to obtain o.n.s.
$\mathbf{x}_1, \ldots, \mathbf{x}_q, \mathbf{x}_{q+1}, \ldots, \mathbf{x}_p$. Form
$C_0 := B_0 + \sum_{i=q+1}^{p} \mathbf{x}_i (A^T \mathbf{x}_i)^T$.

Find the first left k-o.n. left singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of C_0. Then $B_1 := \sum_{i=1}^{k} \mathbf{v}_i (A^T \mathbf{v}_i)^T$ and
$\mathbf{tr}\, B_0^T B_0 \leq \mathbf{tr}\, B_1^T B_1$.

Obtain B_t from B_{t-1} as above.

7 Lifting body original

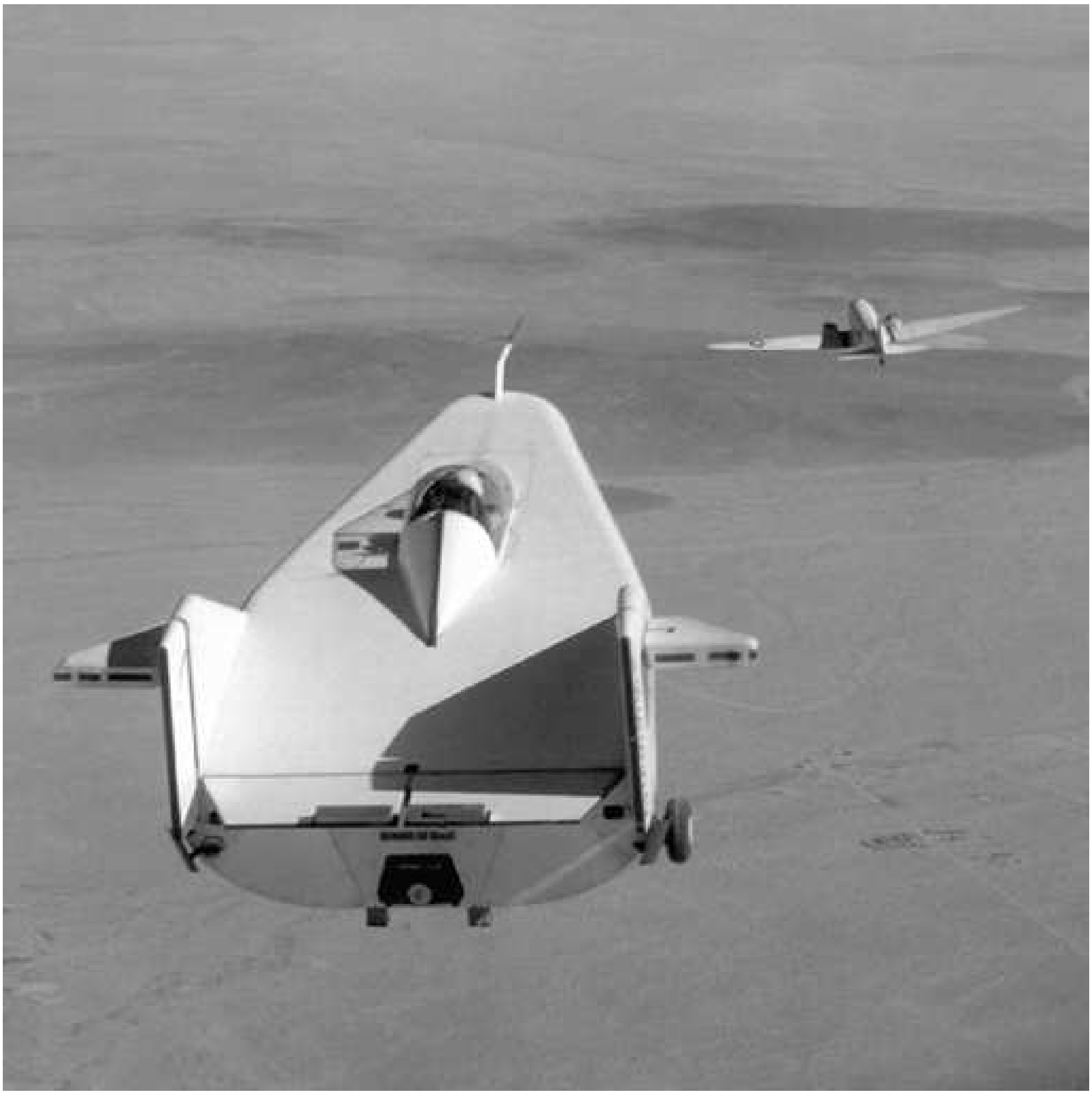

Figure 1: Lifting body image 512×512.

8 Lifting body compressed

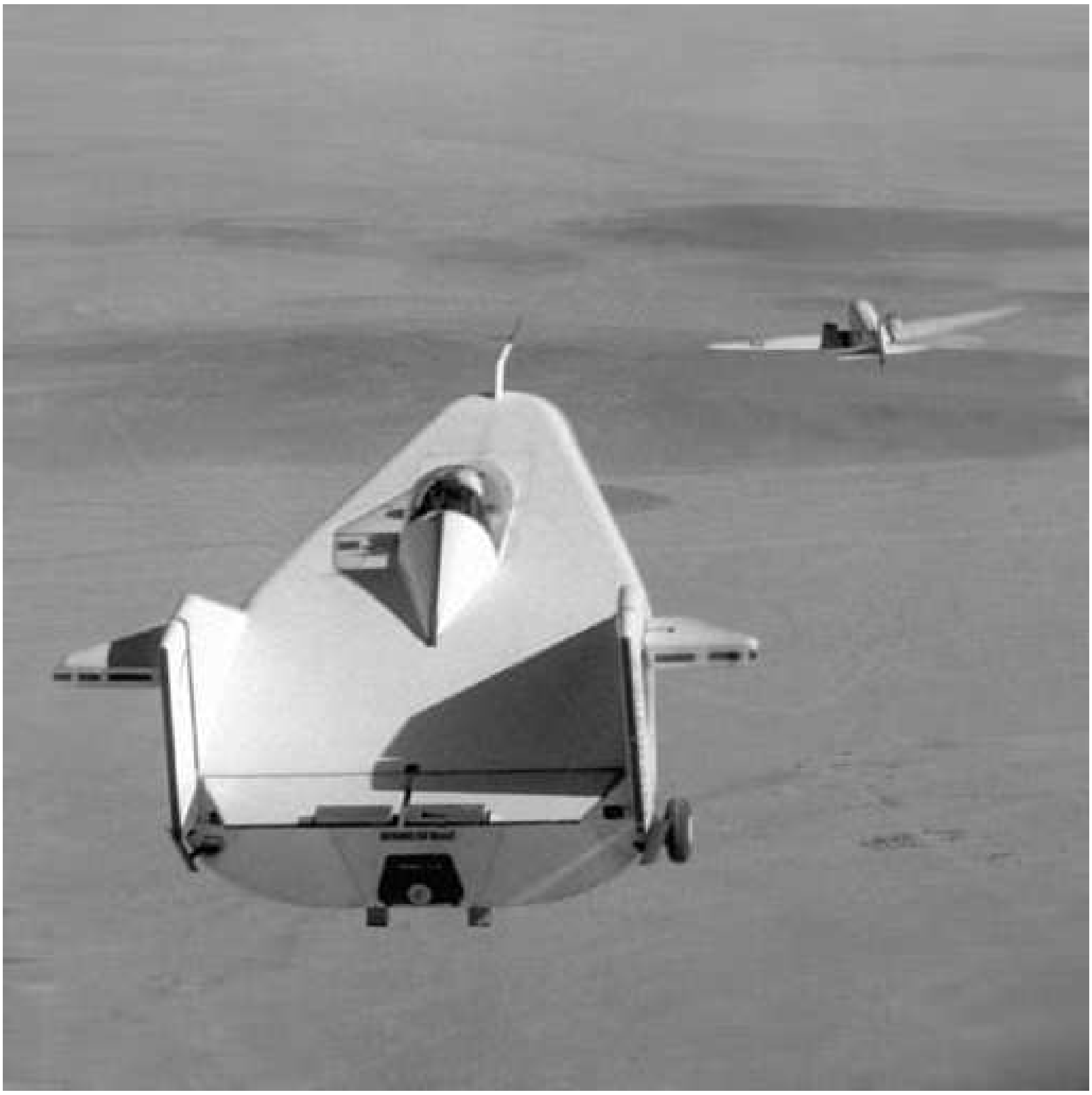

Figure 2: 80-rank approximation of Lifting body image 512×512.

9 SIMULATIONS 1

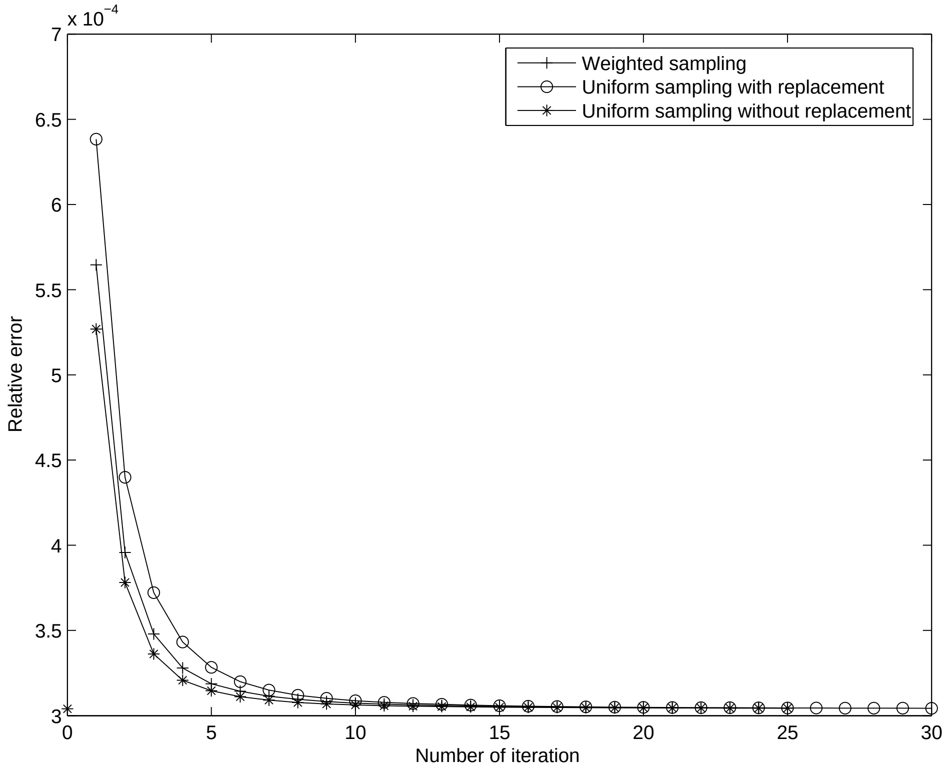

Figure 3: Convergence property of the Monte-Carlo method for Liftingbody image($\mathbf{512 \times 512}$), $\boldsymbol{k = 80}$.

10 SIMULATIONS 2

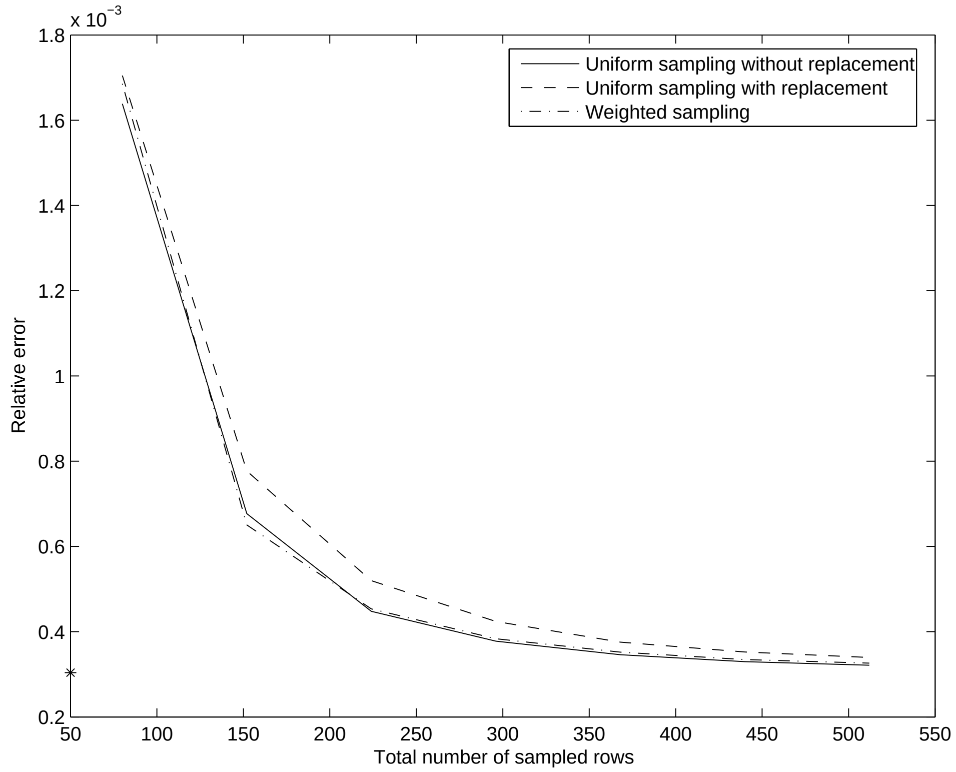

Figure 4: Liftingbody: relative errors versus total number of sampled rows, $k = 100$

11 Camera man original

Figure 5: Camera man image $\mathbf{256} \times \mathbf{256}$.

12 Camera man compressed

Figure 6: 80-rank approximation of Camera man $\mathbf{256} \times \mathbf{256}$.

13 SIMULATIONS 3

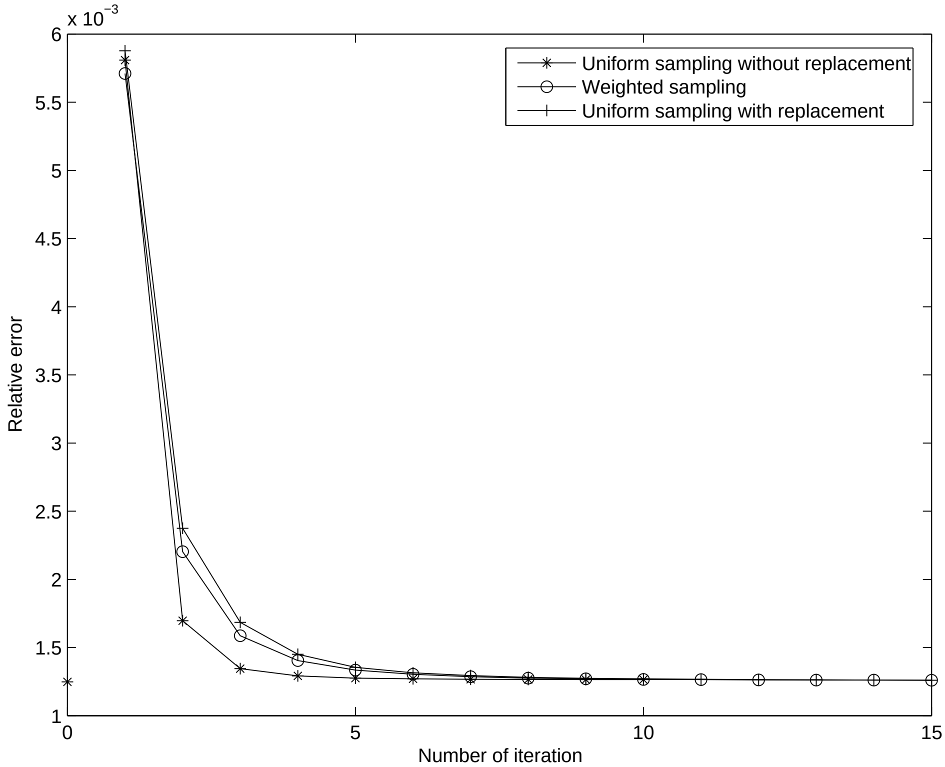

Figure 7: Convergence property of the Monte-Carlo method for Cameraman image(256×256), $k = 80$.

14 SIMULATIONS 4

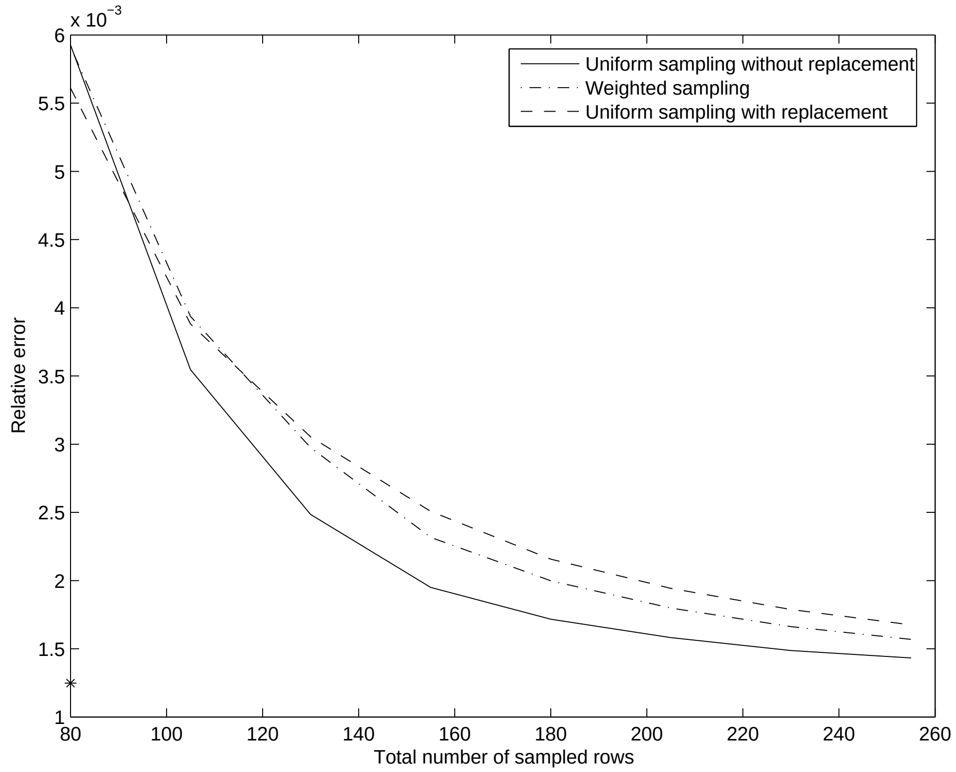

Figure 8: Cameraman: Relative error versus total number of sampled rows, $k = 80$.

15 SIMULATIONS 5

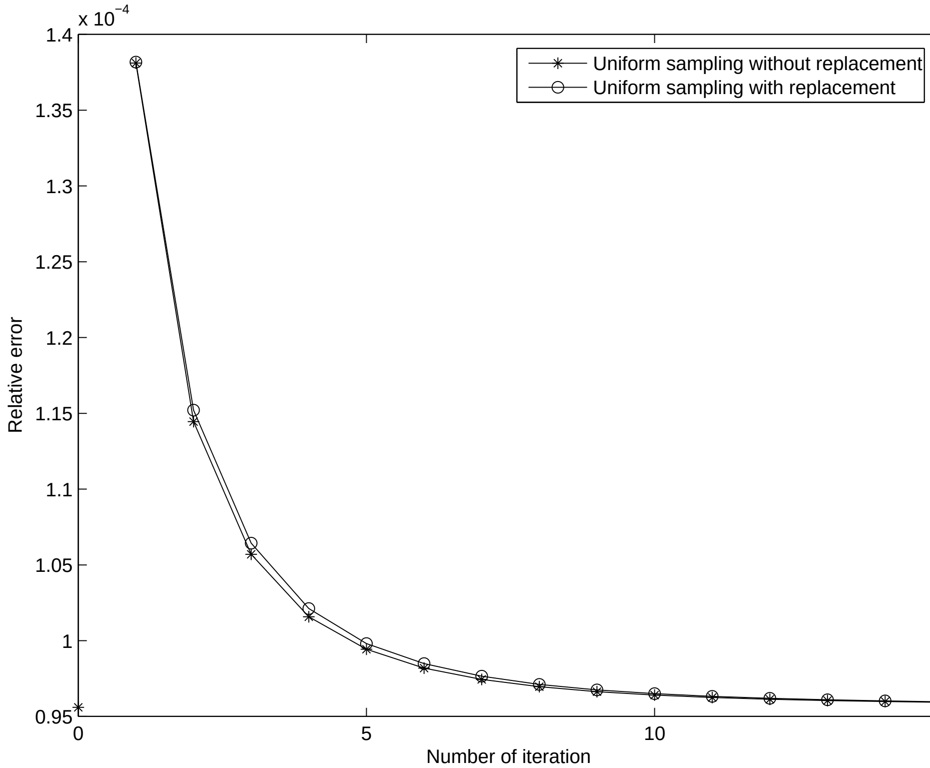

Figure 9: Convergence property of the Monte-Carlo method for random data matrix(3000×500) $k = l = 100$.

16 COMPARISONS

Table 1: Comparison of relative error and speed up of our algorithm with optimum k-rank approximation algorithm

Data sets	Speed up	Re. ratio
Cameraman(256×256), $k = 80$	1.145	1.083
Liftingbody (512×512), $k = 100$	8	1.08
Map image(627×865) $k = 200$	3.33	1.067
Random matrix(8000×200) $k = 100$	42	1.1

17 Choosing columns of A

Frieze, Kannan and Vempala [8] suggest to choose column $\mathbf{c}_i(A)$ with probability $\frac{||\mathbf{c}_i(A)||^2}{||A||_F^2}$.

If $s \geq k$ are chosen then the k-approximation satisfies A_k $||A - A_k||_F^2 \leq \sum_{i=k+1}^{m} \sigma_i(A)^2 + \frac{10k}{s}||A||_F^2$.

If $s \geq \frac{k}{10\epsilon}$ then

$$||A - A_k||_F^2 \leq \sum_{i=k+1}^{m} \sigma_i(A)^2 + \epsilon||A||_F^2.$$

Deshpande, Rademacher, Vempala and Wang [2] improved the sampling by modifying the sampling $\mathbf{c}_i(A)$ according to new probabilities $\frac{||\mathbf{c}_i(A-A_k)||^2}{||A-A_k||_F^2}$.

Perhaps our algorithm can be combined with above sampling of columns to get better results.

References

[1] O. Alter, P.O. Brown and D. Botstein, Singular value decomposition for genome-wide expression data processing and modelling, *Proc. Nat. Acad. Sci. USA* 97 (2000), 10101-10106.

[2] A. Deshpande, L. Rademacher, S. Vemapala and G. Wang, Matrix Approximation and Projective Clustering via Volume Sampling, *SODA*, 2006.

[3] S. Friedland, A New Approach to Generalized Singular Value Decomposition, *SIMAX* 27 (2005), 434-444.

[4] S. Friedland, M. Kaveh, A. Niknejad and H. Zare, Fast Monte-Carlo Low Rank Approximations for Matrices, *Proc. IEEE SoSE*, 2006, 6 pp., to appear.

[5] S. Friedland, M. Kaveh, A. Niknejad and H. Zare, An Algorithm for Missing Value Estimation for DNA Microarray Data, *Proceedings of ICASSP 2006*, Toulouse, France, 4 pp., to appear.

[6] S. Friedland, A. Niknejad and L. Chihara, A simultaneous reconstruction of missing data in DNA

microarrays, to appear in *Linear Algebra and Its Applications*.

[7] S. Friedland, J. Nocedal and M. Overton, The formulation and analysis of numerical methods for inverse eigenvalue problems, SIAM J. Numer. Anal. 24 (1987), 634-667.

[8] 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.

[9] G.H. Golub and C.F. Van Loan, *Matrix Computation*, John Hopkins Univ. Press, 3rd Ed., 1996.