2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Reconstructing matrices and tensors from few vectors

Cesar F. Caiafa[†,1] and Andrzej Cichocki[†,2]

†Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN
2-1 Hirosawa, Wako, Saitama 351-0198, JAPAN
Email: ccaiafa@brain.riken.jp, cia@brain.riken.jp

Abstract—We introduce a new algorithm, namely the `Greedy-CUR` algorithm for calculating a CUR decomposition of a given matrix. This deterministic algorithm allows one to obtain a low-rank approximation based only on the entries of a reduced set of rows and columns. The concept of a "greedy" algorithm is used to sample rows and columns of the matrix (or unfolded (matricized) tensor) by sequentially adding one row/column that minimizes the achieved error at every iteration. We also use `Greedy-CUR` to develop a method for approximating a 3D-tensor based only on the entries of few rows, columns and tubes fibers. Its extension to $N-$dimensional tensors is straightforward by using a hierarchical decomposition of a unfolded tensor and by applying CUR approximation sequentially. We analyze the quality of our CUR based approximations and show how the approximation error depends on the singular values distribution of corresponding matrices.

1. Introduction

It is known that, given a matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$ of rank k, one can perfectly reconstruct it by choosing only k rows and k columns which determine a non singular intersection submatrix $\mathbf{W}$ and by calculating the corresponding CUR decomposition, i.e. $\mathbf{Y} = \mathbf{CUR}$ where matrices $\mathbf{C} \in \mathbb{R}^{m\times k}$ and $\mathbf{R} \in \mathbb{R}^{k\times n}$ are composed by the selected rows and columns respectively and $\mathbf{U} = \mathbf{W}^{-1}$. This decomposition is also known as the skeleton decomposition of $\mathbf{Y}$ [1].

More interestingly, it has been proven that, for matrices with arbitrary rank, CUR decomposition can provide an approximation $\mathbf{Y} \approx \mathbf{C}\mathbf{W}_\tau^\dagger\mathbf{R}$ (pseudo-skeleton approximation), where $\mathbf{W}_\tau^\dagger$ is the robust Moore-Penrose pseudo-inverse i.e. by ignoring singular values smaller than τ in its Singular Value Decomposition (SVD). The quality of this approximation can be similar to the truncated SVD decomposition (optimal), i.e. providing an error which is proportional to the $(k+1)$-th singular value σ_{k+1} of matrix $\mathbf{Y}$ [2]. This is a strong result that allows one to obtain a good approximations of a whole matrix from the information contained only in a small subset of rows and columns.

In [1, 2, 3], theoretical bounds have been provided for the case where the submatrix $\mathbf{W}$ is of maximum volume (absolute value of its determinant) (See Table 1). However, the optimal selection of rows and columns subsets is a challenging task because we must avoid to test all possible combinations and even the search of a maximum volume submatrix is hard to solve. To alleviate this computational problem there are a class of heuristic algorithms known as *cross algorithms* which sequentially selects rows/columns by dynamically finding maximum absolute values within their residuals (see [4] and references therein for a description).

Table 1: Existing error bounds (in terms of Max Norm $\|\cdot\|_C$ and Spectral Norm $\|\cdot\|_2$) for the CUR (pseudo-skeleton) approximation of a square matrix $\mathbf{Y} \in \mathbb{R}^{n\times n}$ when core matrix $\mathbf{U}$ is calculated based only on the entries of the intersection matrix $\mathbf{W}$ of maximum volume.

Error Norm	Bound	Core Matrix
$\|\mathbf{Y}-\mathbf{CUR}\|_C$	$\leq \sigma_{k+1}(k+1)$ [3]	$\mathbf{U}=\mathbf{W}^{-1}$
$\|\mathbf{Y}-\mathbf{CUR}\|_2$	$\leq M_1 = \sigma_{k+1}\sqrt{1+k(n-k)}$ [2]	$\mathbf{U}=\mathbf{W}^{-1}$
	$\leq M_2 = \sigma_{k+1}\sqrt{1+k(n-k)}\left\{1+4\sqrt{k(n-k)}\right\}$ [1]	$\mathbf{U}=\mathbf{W}_\tau^\dagger$

Additionally, there has been an increased interest on CUR decompositions as a tool for data analysis as an alternative to the popular Singular Value Decomposition (SVD) or Principal Component Analysis (PCA), specially for cases with sparse datasets providing a representation of data as a linear combination of few "meaningful" components which are exact replicas of columns and rows of the data. Most of existing CUR algorithms for data analysis requires to access to all data matrix entries at least once for defining the strategy of row/column selection ([5, 6]). In this paper, we do not consider these methods since we are restricted to use only partial information of large scale matrices (or tensors), i.e. few rows, columns (and tube fibers).

The purpose of this paper is to introduce new algorithms for reconstructing matrices and tensors from a reduced set of their entries. We use the theory of pseudo-skeleton decompositions (CUR decomposition) but, instead of search for a maximum volume submatrix, we construct our selection of rows and columns by sequentially adding a row/column in order to minimize the error approximation at every iteration (`Greedy-CUR`).

[1] On leave from Engineering Faculty, University of Buenos Aires, Buenos Aires, C1063ACV, ARGENTINA.

[2] Also from Warsow University of Technology and Systems Research Institute, PAN, POLAND.

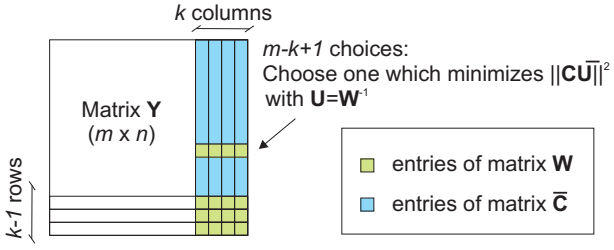

Figure 1: Optimal selection of an additional row given that k columns and $k-1$ rows are already selected. Note that the optimal choice requires to use only entries of already selected columns.

2. A greedy algorithm for selection of rows and columns

Let us first to introduce the notation we use throughout the paper. Given a sequence of k row indices $\mathcal{I} = [i_1, i_2, ..., i_k]$ (or k column indices $\mathcal{J} = [j_1, j_2, ..., j_k]$), we denote the $(m-k)$ row complement indices by $\overline{\mathcal{I}}$ ($(n-k)$ column complement indices by $\overline{\mathcal{J}}$), i.e. the indices of rows (columns) not selected. The column, row and intersection matrices are then defined by $\mathbf{C} = \mathbf{Y}(:, \mathcal{J})$, $\mathbf{R} = \mathbf{Y}(\mathcal{I}, :)$ and $\mathbf{W} = \mathbf{Y}(\mathcal{I}, \mathcal{J})$ respectively. Accordingly, we also define $\overline{\mathbf{C}} = \mathbf{Y}(\overline{\mathcal{I}}, \mathcal{J})$ and $\overline{\mathbf{R}} = \mathbf{Y}(\mathcal{I}, \overline{\mathcal{J}})$.

Our `Greedy-CUR` algorithm is inspired by the following Theorem due by Goreinov, Zamarashkin and Tyrtyshnikov (see proof on original paper [2]):

Theorem 1 (Goreinov, Zamarashkin and Tyrtyshnikov) *Let a square matrix $\mathbf{Y} \in \mathbb{R}^{n \times n}$ be nonsingular and a selection of row and column indices $\mathcal{I}$ and $\mathcal{J}$ such that the intersection submatrix $\mathbf{W}$ is non-degenerate. In this case, the following CUR approximation bounds are obtained:*

$$\|\mathbf{Y} - \mathbf{CUR}\|_2 \leq \sigma_{k+1} \sqrt{1 + \|\overline{\mathbf{C}}\mathbf{U}\|_2^2}, \quad (1)$$

$$\|\mathbf{Y} - \mathbf{CUR}\|_2 \leq \sigma_{k+1} \sqrt{1 + \|\mathbf{U}\overline{\mathbf{R}}\|_2^2}, \quad (2)$$

where σ_{k+1} is the $(k+1)$-th singular value of $\mathbf{Y}$ and the core matrix is defined by $\mathbf{U} = \mathbf{W}^{-1}$.

This theorem shows us that we can control the approximation error based only on the entries of selected columns (or rows) since bound (1) depends exclusively on them.

Let us consider a simple case of having already selected $k-1$ rows and k columns and we want to optimally select an additional row in order to complete a CUR decomposition with k rows/columns. Theorem 1 gives us a clue on how to select it without needing to test all entries of all possible additional rows. In this case, we minimize the bound (1) by testing all possible selections which requires to use only the entries of the selected columns (see Figure 1).

The suggested algorithm requires to calculate the inverse of the intersection matrix for all possible selections which are not guaranteed to exist or can be bad conditioned so we replace the inverse by the adapted Moore-Penrose pseudoinverse $\mathbf{W}_\tau^\dagger$ for a small τ (as implemented in `MATLAB`). Note also that, when this pseudoinverse is used, we are allowed to apply this algorithm even if the intersection matrix is not square, that is, different number of rows and columns. For a fast implementation of this algorithm we also use the Frobenius norm ($\|\cdot\|_F$) instead of spectral norm ($\|\cdot\|_2$). This basic algorithm, called `ADD_ROW`($\mathbf{C}$, $\mathcal{I}$), is shown as Algorithm 1.

Algorithm 1 `ADD_ROW` algorithm

INPUT: Matrix $\mathbf{C}$ and indices of already selected rows $\mathcal{I}$.
OUTPUT: Index of the best row to add i.
1: $i = \min_{i \notin \mathcal{I}} \|\overline{\mathbf{C}}\mathbf{C}(\mathcal{I} \cup \{i\}, :)^\dagger\|_F^2$;
2: **return** i

The idea of our `Greedy-CUR` algorithm is to start from the selection of an arbitrary single column and successively add single rows and single columns which are optimal at each iteration step $r = 1, 2, ..., k$. It is interesting to note that only in the first step (when only one column was previously selected), the `Greedy-CUR` algorithm is identical to the 2D cross algorithm by selecting rows/columns corresponding to the maximum absolute value entry [4]. The detailed `Greedy-CUR` algorithm is presented as Algorithm 2.

Algorithm 2 `GREEDY_CUR` algorithm

INPUT: initial column index j_1 and the number of rows/columns to be selected k.
OUTPUT: Indexes of selected k rows/columns $\mathcal{I}_k$ and $\mathcal{J}_k$.
1: $\mathcal{J}_1 = \{j_1\}$, $\mathcal{I}_0 = \{\emptyset\}$;
2: $r = 1$;
3: **while** $r < k$ **do**
4: **if** $r \neq 1$ **then**
5: $\mathbf{R}_r = \mathbf{Y}(\mathcal{I}_r, :)$;
6: $\mathcal{J}_r = \mathcal{J}_{r-1} \cup$ `ADD_ROW`$(\mathbf{R}_r^T, \mathcal{J}_{r-1})$;
7: **end if**
8: $\mathbf{C}_r = \mathbf{Y}(:, \mathcal{J}_r)$;
9: $\mathcal{I}_r = \mathcal{I}_{r-1} \cup$ `ADD_ROW`$(\mathbf{C}_r, \mathcal{I}_{r-1})$;
10: r=r+1;
11: **end while**
12: **return** $\mathcal{I}_r$ and $\mathcal{J}_r$;

In Fig. 2 we illustrate a Monte Carlo analysis by applying our `Greedy-CUR` algorithm to synthetic date. We generated a class of matrices as $\mathbf{Y} = \mathbf{V}_1 \mathbf{\Lambda} \mathbf{V}_2^T$ where $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{R}^{100 \times 100}$ are randomly generated orthogonal matrices and $\mathbf{\Lambda} \in \mathbb{R}^{100 \times 100}$ is diagonal with its main diagonal elements (singular values) obeying the following decay law $\sigma_k = 1/k$. From these results we conclude that: a) Greedy-CUR achieves better quality of reconstruction compared to the *cross algorithm*; b) bounds of equations (1) and (2) are tight allowing us to effectively minimize the approximation error, i.e. this allows us to decide when to stop adding new rows/columns if a desired approximation quality is achieved[1]; and c) Bounds based on the Maximum

[1]This requires to know in advance or make an assumption about the decay law associated to singular values σ_k

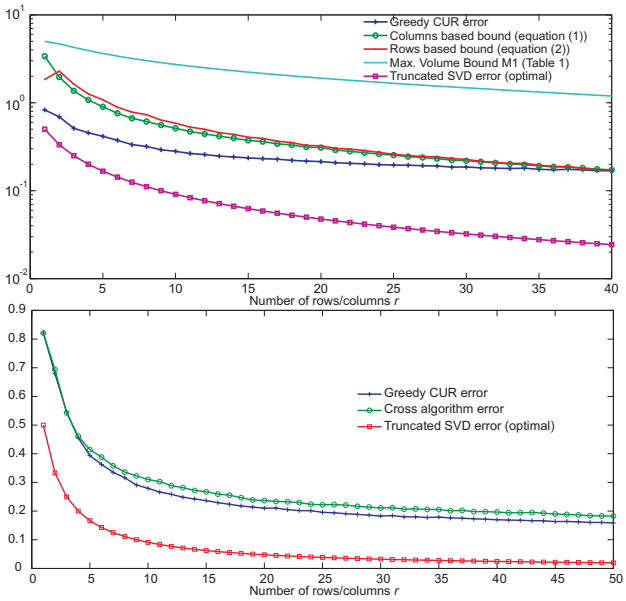

Figure 2: Error of `Greedy-CUR` algorithm and related bounds (top), and comparison of errors obtained with `Greedy-CUR` and a classical *2D-cross algorithm* [4] (bottom). Bound M2 is much larger than M1 and it is not shown. We averaged the results over 100 realizations

Volume concept (see Table 1) are considerably higher than obtained errors with Greedy-CUR suggesting that Maximum Volume property is not required in practice.

3. Extensions to tensors

In this section we are faced to the problem of obtaining good approximations of tensors (multiway arrays [7]) based only on few entries of them. Let us first consider, for a clear presentation, the simplest case of a 3D tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{n\times n\times n}$ where entries are denoted by y_{ijk} and the indices i, j, k take values in the range: $1, 2, ..., n$. A procedure that transforms a 3D tensor into a matrix is called *unfolding* (or *matricization*) and consists on mapping a pair of indices to a new index. As a result, a matrix with the same elements as the original tensor is obtained. The three possible (modes) unfolded matrices ($n \times n^2$) are: $\mathbf{Y}_{(1)} = [y_{i(jk)}]$, $\mathbf{Y}_{(2)} = [y_{j(ik)}]$, and $\mathbf{Y}_{(3)} = [y_{k(ij)}]$.

Then, we can approximate a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{n\times n\times n}$ by approximating any of its unfolded matrix versions, for example $\mathbf{Y}_{(1)}$, which can be done by applying `Greedy-CUR` algorithm arriving to a selection of rows and columns of $\mathbf{Y}_{(1)}$. Furthermore, it is easy to see that any row of $\mathbf{Y}_{(1)}$ corresponds to a vectorized version of a horizontal slice in the original tensor $\underline{\mathbf{Y}}$ which can be itself approximated by the CUR decomposition (see Fig. 3).

Therefore, we can obtain an approximation of a 3D tensor by applying `Greedy-CUR` to one unfolded matrix and applying again `Greedy-CUR` to approximate selected slices. In summary, this procedure requires to use only partial information of the tensor, more specifically, it requires

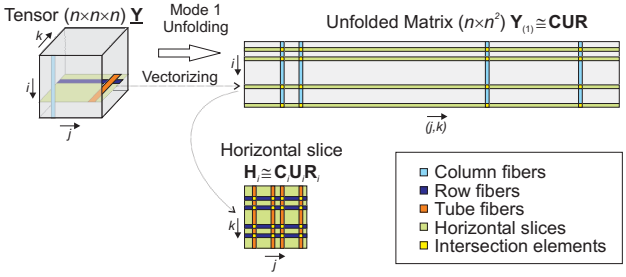

Figure 3: Application of CUR approximation to mode 1 unfolded matrix $\mathbf{Y}_{(1)} \cong \mathbf{CUR}$ and selected horizontal slices $\mathbf{H}_i = \mathbf{C}_i\mathbf{U}_i\mathbf{R}_i$.

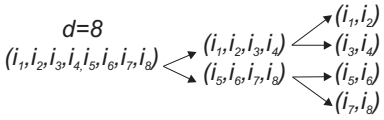

Figure 4: Example of recursive unfolding of a tensor with $N = 8$ dimensions. At every step of the recursion a matrix is obtained which can be approximated by its CUR decomposition

to sequentially sample few rows, columns and tubes fibers of a 3D tensor

We can easily extend this procedure to the case of $N-$dimensional tensors, i.e. $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$ by noting that any tensor can be written as a matrix by just grouping its indices in two sets. This generalizes the concept of unfolding a 3D tensor to arrays of general dimensions N as recently introduced in [8]. For example, if N is an even number, we can decompose N indices as union of 2 groups of $N/2$ indices each and this procedure can be recursively repeated until reach to the leaves of a hierarchical tree with only two indices. In Fig. 4 an example of such recursive grouping of indices for a tensor with $N = 8$ indices is shown.

Another way to reconstruct a tensor from the entries of selected few fibers is given by the following theorem (recently provided in [9]) which is based in the Tucker model [10] where the n-mode product of a tensor by a matrix is denoted by $\times_n$:

Theorem 2 *Given an N-way tensor* $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$ *having an exact representation by the Tucker model of order R, i.e. there exist a set of matrices* $\mathbf{A}_n \in \mathbb{R}^{I_n\times R}$, $n = 1, 2, ..., N$ *and a core tensor* $\underline{\mathbf{G}} \in \mathbb{R}^{R\times R\times ...\times R}$ *such that*

$$\underline{\mathbf{Y}} = \underline{\mathbf{G}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 ... \times_N \mathbf{A}_N, \tag{3}$$

and, given a selection of P indices for each dimension ($P \geq R$) $\mathcal{I}_n$, $n = 1, 2, ..., N$ *such that all the unfolding matrix modes of the intersection subtensor* $\underline{\mathbf{W}} = \underline{\mathbf{Y}}(\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N)$ *have rank R. In this case, the following exact tensor representation is obtained:*

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}} \times_1 \mathbf{C}_1 \times_2 \mathbf{C}_2 ... \times_N \mathbf{C}_N, \tag{4}$$

where matrices $\mathbf{C}_n \in \mathbb{R}^{I_n\times P^{(N-1)}}$ *are matrices contained the corresponding n-mode fibers, i.e.* $\mathbf{C}_n = \mathbf{Y}_{(n)}(:, \bigotimes_{p\neq n} \mathcal{I}_p)$ *and the core tensor is* $\underline{\mathbf{U}} = \underline{\mathbf{W}} \times_1 \mathbf{W}_{(1)}^{\dagger} \times_2 \mathbf{W}_{(2)}^{\dagger} ... \times_N \mathbf{W}_{(N)}^{\dagger}$.

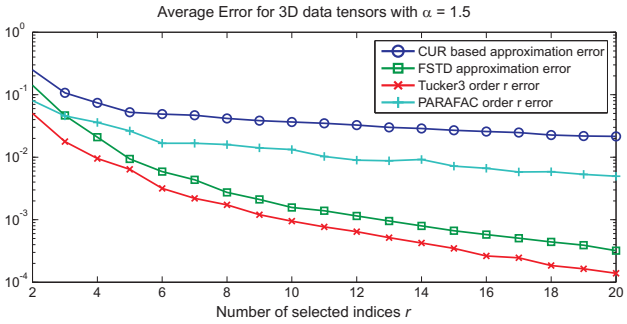

Figure 5: Error of approximation of 3D tensors ($100 \times 100 \times 100$) with our CUR based method and FSTD method for a decay law $\sigma_l = 1/l^{\alpha}$ with $\alpha = 1.5$. As a reference, the approximation errors obtained by applying Tucker3 model order r and PARAFAC model with r components are also shown. We averaged the results over 20 realizations.

Equation (4) provides the so called Fiber Based Tensor Decomposition (FBTD) or Fiber Sampling Tensor Decomposition (FSTD) which gives an exact calculation of a tensor based on a subset of intelligently selected n-mode fibers and can be used as an approximation tool when the original tensor is approximated by a Tucker model [9].

4. Numerical results analysis for tensors with $N = 3$

In Fig. 5 a Monte Carlo analysis of applying our CUR based method and the FSTD algorithm to a class of 3D tensors with $n = 100$ generated as $\underline{\mathbf{Y}} = \sum_{l=1}^{n} \sigma_l \mathbf{u}_l \circ \mathbf{v}_l \circ \mathbf{w}_l$ where vectors $\mathbf{u}_l, \mathbf{v}_l, \mathbf{w}_l$ are the columns of randomly generated orthogonal matrices $\mathbf{U}, \mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times n}$, the operation $\circ$ is the outer product and the coefficients obey the following decay law $\sigma_l = 1/l^{\alpha}$. We note that this data corresponds to PARAFAC model with orthogonal vectors or Tucker3 model with orthogonal matrices and a diagonal core tensor [11]. We also compare the obtained approximation with those ones achieved by standard tensor decomposition methods like PARAFAC and Tucker3 which uses the information of all the entries of the tensor (we have used the MATLAB Tensor Toolbox [12]) for different values of parameter α (decay law).

5. Conclusions and discussions

`Greedy-CUR` algorithm allows one to achieve better approximations (lower error) than cross algorithms. As a drawback it can be shown that it runs in quadratic time $O(n^2)$ in contrast with linear time ($O(n)$) of cross algorithms [4]. It is also important to note that the complexity increases more than linearly in k (number rows/columns to be selected) and calculations can be expensive for large k. In fact, the key factor for obtaining good CUR approximation is to have singular values with abrupt decay laws as our experimental results showed. Our results also suggest that Maximum Volume condition of the intersection matrix $\mathbf{W}$ can be relaxed and special attention should be paid to the matrices $\overline{\mathbf{C}}$ and $\overline{\mathbf{R}}$ in order to minimize the error as our `Greedy-CUR` algorithm does. We have developed two methods for for the approximation of $N-$dimensional tensors which are based on the information contained only in few fibers (for $N = 3$ it corresponds to few row, column and tube fibers): the first method is a based on the CUR decomposition of matrices applied to one unfolding mode and the second method is based on the FSTD formula recently provided in [9]. This methods can be used as representation models of large tensor (dataset) since they are able to reconstruct the whole dataset from the indices of the selected fibers. This can result in enormous saving of memory when singular values of associated matrices have sharp decays.

References

[1] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, "A theory of pseudoskeleton approximations," *Linear Algebra and its Applications*, vol. 261, pp. 1–21, 1997.

[2] ——, "Pseudo-skeleton approximations by matrices of maximal volume," *Mathematical Notes*, vol. 62, no. 4, pp. 515–519, 1997.

[3] S. A. Goreinov and E. E. Tyrtyshnikov, "The maximal-volume concept in approximation by low-rank matrices," *Contemporary Mathematics*, vol. 280, pp. 47–51, 2001.

[4] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, "Tucker dimensionality reduction of three-dimensional arrays in linear time," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, p. 939, 956 2008.

[5] G. W. Stewart, "Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix," *Numerische Mathematik*, vol. 83, no. 2, pp. 313–323, 1999.

[6] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, January 2009.

[7] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*. Wiley, September 2009, in press.

[8] I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse od dimensionality, or how to use SVD in many dimensions," ICM HKBU, Report 09-03, 2009.

[9] C. F. Caiafa and A. Cichocki, "Generalizing the CUR matrix decomposition to multi-ways arrays," June 2009, submited.

[10] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," in *Problems in Measuring Change*, C. W. Harris, Ed. University of Wisconsin Press, 1963, pp. 122–137.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," 2009, to appear in SIAM Review (accepted June 2008).

[12] B. W. Bader and T. G. Kolda, "Matlab tensor toolbox version 2.2," January 2007.