# Tensor-CUR Decompositions For Tensor-Based Data

Michael W. Mahoney *
Yahoo Research Labs
Sunnyvale, CA 94089
mahoney@yahoo-inc.com

Mauro Maggioni
Dept. of Mathematics
Yale University
New Haven, CT 06520
mauro.maggioni@yale.edu

Petros Drineas
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
drinep@cs.rpi.edu

## ABSTRACT

Motivated by numerous applications in which the data may be modeled by a variable subscripted by three or more indices, we develop a tensor-based extension of the matrix CUR decomposition. The tensor-CUR decomposition is most relevant as a data analysis tool when the data consist of one mode that is qualitatively different than the others. In this case, the tensor-CUR decomposition approximately expresses the original data tensor in terms of a basis consisting of underlying subtensors that are actual data elements and thus that have natural interpretation in terms of the processes generating the data. In order to demonstrate the general applicability of this tensor decomposition, we apply it to problems in two diverse domains of data analysis: hyperspectral medical image analysis and consumer recommendation system analysis. In the hyperspectral data application, the tensor-CUR decomposition is used to *compress* the data, and we show that classification quality is not substantially reduced even after substantial data compression. In the recommendation system application, the tensor-CUR decomposition is used to *reconstruct* missing entries in a user-product-product preference tensor, and we show that high quality recommendations can be made on the basis of a small number of basis users and a small number of product-product comparisons from a new user.

**Categories and Subject Descriptors:** E.m [Data] : Miscellaneous; H.m [Information Systems] : Miscellaneous

**General Terms:** Algorithms, Experimentation

**Keywords:** CUR Decomposition, Tensor CUR, Hyperspectral Image Analysis, Recommendation System Analysis

## 1. INTRODUCTION

Novel algorithmic methods to structure large data sets are of continuing interest. A particular challenge is presented by tensor-based data, i.e., data which are modeled by a variable subscripted by three or more indices [28, 19, 30, 39, 6]. Numerous examples suggest themselves, but to guide the discussion consider the following three. First, in internet data applications, if one is studying the properties of a large time-evolving graph, the data may consist of a graph or its adjacency matrix sampled at a large number of sequential time steps, in which case $\mathcal{A}_{ijk}$ may represent the weight of the edge between nodes $i$ and $j$ at time step $k$. Second, in biomedical data applications, if one is studying cancer diagnosis, the data may consist of a large number of hyperspectrally-resolved biopsy images, in which case $\mathcal{A}_{ijk}$ may represent the absorbed or transmitted light intensity of a biopsy sample at pixel $ij$ at frequency $k$. Third, in consumer data applications, if one is studying recommendation systems, the data may consist of product-product preference data for a large number of users, in which case $\mathcal{A}_{ijk}$ may be $\pm 1$, depending on whether product $i$ or $j$ is preferred by user $k$. Tensor-based data are particularly challenging due to their size and since many data analysis tools based on graph theory and linear algebra do not easily generalize.

When compared with algorithmic results for data modeled by either matrices or graphs, algorithmic results for data modeled by multi-mode tensors are modest. For example, even computing the rank of a general tensor $\mathcal{A}$ (defined as the minimum number of rank-one tensors into which $\mathcal{A}$ can be decomposed) is an NP-hard problem [20]. On the other hand, the model proposed by Tucker [39], as well as the related the "canonical decomposition" [6] or the "parallel factors" model [19], have a long history in applied data analysis [24, 25, 26, 28]. They provide exact or approximate decompositions for higher-order tensors. Recent research has focused on the relationship between these data tensor models and efforts to extend linear algebraic notions such as the Singular Value Decomposition to multi-mode data tensors [28, 29, 30, 31].

A seemingly unrelated line of work has focused on matrix CUR decompositions [10, 13, 12]. As discussed in more detail in Section 2.2, a matrix CUR decomposition provides a low-rank approximation of the form $A \approx \tilde{A} = CUR$, where $C$ is a matrix consisting of a small number of columns of $A$, $R$ is a matrix consisting of a small number of rows of $A$, and $U$ is an appropriately-defined low-dimensional encoding matrix [10]. Thus, a CUR matrix decomposition provides a dimensionally-reduced low-rank approximation to the original data matrix $A$ that is expressed in terms of a small number of actual columns and a small number of actual rows of the original data matrix, rather than, e.g., orthogonal linear combinations of those columns and rows.

*Part of this work was performed while at the Department of Mathematics, Yale University, New Haven, CT, USA 06520.


*KDD'06,* August 20–23, 2006, Philadelphia, Pennsylvania, USA.

In this paper, we extend a recently-developed and provably accurate matrix CUR decomposition to tensor-based data sets in which there is a "distinguished" mode, and we apply it to problems in two of the three data set domains mentioned previously. When applied to hyperspectral image data, we use tensor-CUR to perform compression in order to run a classification on a more concise input, and when applied to recommendation system data we use tensor-CUR to perform reconstruction in the absence of the full input.

By a "distinguished" mode, we mean a mode that is qualitatively different than the other modes in an application-dependent manner. The most appropriate data structure for a data set consisting of, e.g., a time-evolving internet graph or a set of hyperspectrally-resolved biopsy images or user-product-product preference data for consumers, depends on the application and is a matter of debate. Nevertheless, we will view such a data set as a tensor, albeit one in which one of the modes is "distinguished." For example, in these three applications, the distinguished mode would be the mode describing, respectively, the temporal evolution of the graph, the frequency or spectral variation in the images, and the users. The tensor-CUR decomposition computes an approximation to the original data tensor that is expressed as a linear combination of subtensors of the original data tensor. As we shall see, since these subtensors are actual data elements, rather than, e.g., more complex functions of data elements, in many cases they lend themselves more readily to application-specific interpretation.

# 2. REVIEW OF RELEVANT LINEAR AND MULTILINEAR ALGEBRA

## 2.1 Singular Value Decomposition (SVD)

The following theorem is a fundamental result from linear algebra that is widely-used (often via the related Principal Components Analysis) in data analysis.

Theorem 1. *If $A \in \mathbb{R}^{m\times n}$, then there exist orthogonal matrices $U = [u^1 u^2 \dots u^m] \in \mathbb{R}^{m\times m}$ and $V = [v^1 v^2 \dots v^n] \in \mathbb{R}^{n\times n}$, where $\{u^t\}_{t=1}^m \in \mathbb{R}^m$ and $\{v^t\}_{t=1}^n \in \mathbb{R}^n$ are such that*

$$U^T A V = \Sigma = \mathbf{diag}(\sigma_1, \dots, \sigma_\rho), \tag{1}$$

*where $\Sigma \in \mathbb{R}^{m\times n}$, $\rho = \min\{m, n\}$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_\rho \geq 0$. Equivalently,*

$$A = U\Sigma V^T. \tag{2}$$

The three matrices $U$, $V$, and $\Sigma$ constitute the Singular Value Decomposition (SVD) of $A$. If we define $r$ by $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_\rho = 0$, then $\text{rank}(A) = r$. In addition, if $k \leq r$ and we define

$$A_k = U_k \Sigma_k V_k^T = \sum_{t=1}^{k} \sigma_t u^t {v^t}^T, \tag{3}$$

then the distance (as measured, e.g., by the Frobenius norm $\|\cdot\|_F$, where $\|A\|_F^2 = \sum_{ij} A_{ij}^2$) between $A$ and any rank $k$ approximation to $A$ is minimized by $A_k$. More formally, we have the following theorem.

Theorem 2. *If $A \in \mathbb{R}^{m\times n}$ and $A_k \in \mathbb{R}^{m\times n}$ is defined by (3), then*

$$\|A - A_k\|_F^2 = \min_{D\in\mathbb{R}^{m\times n}: rank(D)\leq k} \|A - D\|_F^2. \tag{4}$$

For more details about these results, see [16].

## 2.2 Matrix CUR Decomposition

Recent work in theoretical computer science, numerical linear algebra, and statistical learning theory [10, 12, 37, 38, 3, 18, 17, 40, 13] has focused on low-rank matrix decompositions with structural properties that satisfy the following definition:

Definition 1. *Let $A$ be an $m \times n$ matrix. In addition, let $C$ be an $m \times c$ matrix whose columns consist of a small number $c$ of columns of the matrix $A$, let $R$ be an $r \times n$ matrix whose rows consist of a small number $r$ of rows of the original matrix $A$, and let $U$ be a $c \times r$ matrix. Then $\tilde{A}$ is a* column-row-based low-rank approximation, *or a* CUR approximation, *to $A$ if it may be explicitly written as*

$$\tilde{A} = CUR. \tag{5}$$

Several things should be noted about this definition. First, for data applications, we prefer not to provide too precise a characterization of what we mean by a "small" number of columns and/or rows, but one should think of $r, c \ll m, n$. For example, they could be constant, independent of $m$ and $n$, logarithmic in the size of $m$ and $n$, or simply a large constant factor less than $m, n$. Second, since the approximation is expressed in terms of a small number of columns and rows of the original data matrix, it will provide a low-rank approximation to the original matrix, although one with structural properties that are quite different than those provided by truncating the SVD. Third, a CUR approximation approximately expresses all of the columns of $A$ in terms of a linear combination of a small number of "basis columns," and similarly for the rows.

Finally, and most relevant for the present paper, note that a CUR matrix decomposition has structural properties that are auspicious for its use as a tool in the analysis of large data sets. For example, if the data matrix $A$ is large and sparse but well-approximated by a low-rank matrix, then $C$ and $R$ (consisting of actual columns and rows) are sparse, whereas the matrices consisting of the top left and right singular vectors will not in general be sparse. In addition, in many applications, interpretability is important; practitioners often have an intuition about the actual columns and rows that they fail to have about linear combinations of (up to) all the columns or rows.

The following algorithmic result regarding a matrix CUR approximation was recently proven [10].

Theorem 3. *There exists a randomized algorithm (see the* LinearTimeCUR *algorithm of [10]) that takes as input an $m \times n$ matrix $A$ and a fixed rank parameter $k$, and that returns as output an $m \times c$ matrix $C$ consisting of $c$ columns of $A$, an $r \times n$ matrix $R$ consisting of $r$ rows of $A$, and an $c \times r$ matrix $U$. The columns/rows are randomly sampled in $c/r$ independent trials according to a judiciously-chosen probability distribution depending on the Euclidean norm of the corresponding column/row. If $c = O(k \log(1/\delta)/\epsilon^4)$ and $r = O(k/\delta^2\epsilon^2)$, then*

$$\|A - CUR\|_F \leq \|A - A_k\|_F + \epsilon \|A\|_F \tag{6}$$

*holds with probability at least $1 - \delta$. The algorithm requires $O(m+n)$ additional time and scratch space after reading the matrix $A$ twice from external storage.*

Our two tensor-CUR algorithms are tensor-based extensions of this matrix algorithm. For more details about these results, see [8, 9, 10, 13].

## 2.3 Tensor-Based Extension on the SVD

Tensors are a natural generalization of matrices. We shall use calligraphic letters to denote higher-order or multi-mode tensors with $d > 2$ modes. For example, let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a $d$-mode tensor of size $n_1 \times n_2 \times \cdots \times n_d$. In addition, let $\alpha \in \{1, \ldots, d\}$ be a particular mode and let $N_\alpha = \prod_{i \neq \alpha} n_i$.

Consider the following definitions: Define the matrix $A_{[\alpha]} \in \mathbb{R}^{n_\alpha \times N_\alpha}$, where the columns of the matrix consist of varying the $\alpha^{th}$ coordinate of $\mathcal{A}$ while leaving the rest fixed. We refer to the (usually implicit) construction of $A_{[\alpha]}$ as *matricizing* or *unfolding* $\mathcal{A}$ along mode $\alpha$ and define the *$\alpha$-rank* of the tensor $\mathcal{A}$ to be the rank of the matrix $A_{[\alpha]}$. Given any $n_\alpha \times c_\alpha$ matrix $B$, define the *$\alpha$-mode tensor-matrix product* to be the $d$-mode tensor of size $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ whose $i_1 \cdots i_d$ element is

$$\left(\mathcal{A} \otimes_\alpha B\right)_{i_1 \cdots i_d} = \sum_{i=1}^{n_\alpha} \mathcal{A}_{i_1 \cdots i_{\alpha-1} i i_{\alpha+1} \cdots i_d} B_{i i_\alpha}. \tag{7}$$

Denote the SVD of $A_{[\alpha]}$ by

$$A_{[\alpha]} = U_{A_{[\alpha]}} \Sigma_{A_{[\alpha]}} V_{A_{[\alpha]}}^T = U_{[\alpha]} \Sigma_{[\alpha]} V_{[\alpha]}^T, \tag{8}$$

where, e.g., $U_{[\alpha]}$ is an $n_\alpha \times \operatorname{rank}(A_{[\alpha]})$ matrix and $U_{[\alpha],k_\alpha}$ is a $n_\alpha \times k_\alpha$ matrix consisting of the left singular vectors corresponding to the top $k_\alpha$ singular values of $A_{[\alpha]}$. Define the (square of the) *Frobenius norm* to be

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}_{i_1 \cdots i_d}^2. \tag{9}$$

Let us refer to as *slabs* each of the $n_\alpha$ $d-1$-mode tensors of size $n_1 \times \cdots \times n_{\alpha-1} \times n_{\alpha+1} \times \cdots \times n_d$ constructed by fixing the $\alpha^{th}$ coordinate to some particular value $i_\alpha \in \{1, \ldots, n_\alpha\}$. Similarly, let us refer to as *fibers* each of the $N_\alpha$ vectors (mode-one tensors) of size $n_\alpha$ constructed by fixing each of the other coordinates to a particular value.

For more details about these results, see [28, 29, 12].

# 3. A TENSOR-BASED EXTENSION OF THE CUR MATRIX DECOMPOSITION

## 3.1 A Tensor-CUR Decomposition for $(2+1)$-Data Tensors

In this subsection, for simplicity of exposition and in light of the two applications we will consider, we restrict ourselves to tensors that are subscripted by three indices, i.e., so-called three-mode tensors.

Consider an $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, defined as the collection of elements

$$\{\mathcal{A}_{ijk} | i = 1, \ldots, n_1; j = 1, \ldots, n_2; k = 1, \ldots, n_3\}.$$

The elements may be thought of as a data cube, i.e, a three-dimensional block such that index $i$ runs along the vertical axis, index $j$ runs along the horizontal axis, and index $k$ runs along the "depth" axis. Since by assumption there is a "distinguished" mode, we are considering the special case of a *(2+1)-tensor*, i.e., an $n_1 \times n_2 \times n_3$ tensor in which two modes (without loss of generality, we will assume they are the first two) are similar in some application-dependent manner and the third is qualitatively different. See Figure 1 for a pictorial description of a $(2+1)$-data tensor. In this case, we refer to each of the $n_3$ different $n_1 \times n_2$ matrices as "slabs" and each of the $n_1 n_2$ different $n_3$-vectors as "fibers."

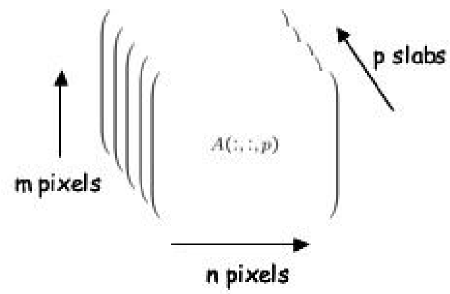


**Figure 1: Pictorial representation of a $(2+1)$-data tensor.**

With this in mind, consider the $(2+1)$-Tensor-CUR algorithm, described in Figure 2. This algorithm takes as input an $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, a probability distribution $\{p_i\}_{i=1}^{n_3}$ over the slabs, a probability distribution $\{q_i\}_{i=1}^{n_1 n_2}$ over the fibers, a number $c$ of slabs to choose, and a number $r$ of fibers to choose. (Without loss of generality, we have assumed that the preferred mode $\alpha \in \{1, 2, 3\}$ is the third mode.) The tensor $\mathcal{A}$ is decomposed along this mode in a manner analogous to the original CUR matrix decomposition [10]. More precisely, this algorithm computes the approximation by performing the following: first, choose $c$ slabs (2-mode subtensors, i.e., matrices) in independent random trials and choose $r$ fibers (1-mode subtensors, i.e., vectors) in independent random trials according to the input probability distributions; second, define the $n_1 \times n_2 \times c$ tensor $\mathcal{C}$ to consist of the $c$ chosen slabs and also define the $r \times n_3$ matrix $\mathcal{R}$ to consist of the chosen fibers; third, let $\mathcal{U}$ be an an appropriately-defined and easily-computed (given $\mathcal{C}$ and $\mathcal{R}$) $c \times r$ matrix.

Clearly, $\tilde{\mathcal{A}} = \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}$, where $\otimes_3$ is a tensor-matrix multiplication, is a $n_1 \times n_2 \times n_3$ tensor. Thus, by using the $(2+1)$-Tensor-CUR algorithm, we make the following approximation:

$$\mathcal{A} \approx \tilde{\mathcal{A}} = \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}. \tag{10}$$

Thus, in particular, if $i \in 1, \ldots, n_3$ is one of the slabs that is not randomly selected, then by using the $(2+1)$-Tensor-CUR algorithm, we make the following approximation:

$$\mathcal{A}(:, :, i) \approx \sum_{\xi \in C} \mathcal{A}(:, :, \xi) X(\xi, i), \tag{11}$$

where $\mathcal{A}(:, :, i)$ is the $n_1 \times n_2$ matrix formed from $\mathcal{A}$ by fixing the value of the third mode to be $i$, $C$ is a set indicating which $c$ indices were randomly chosen, and $X(:, i)$ is a vector consisting of the $i^{th}$ column of the matrix $\mathcal{U}\mathcal{R}$.

See Figure 3 for a pictorial description of the action of the algorithm and this approximation. In particular, note that a small number of slabs are sampled, and every other slab is approximately reconstructed using the information in those slabs as a basis and the information in a small number of fibers (depicted as the dashed lines). The extent to which (10) or (11) is a good approximation has to do with the selection of slabs and fibers. In Sections 4 and 5, we show that (10) holds empirically for our two applications if the slabs and fibers are chosen uniformly and/or nonuniformly

**Input:** An $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, a probability distribution $\{p_i\}_{i=1}^{n_3}$, a probability distribution $\{q_i\}_{i=1}^{n_1 n_2}$, and positive integers $c$ and $r$.
**Output:** An $n_1 \times n_2 \times c$ tensor $\mathcal{C}$, a $c \times r$ matrix $\mathcal{U}$, and a $r \times n_3$ matrix $\mathcal{R}$.

1. Select $c$ slabs in $c$ i.i.d. trials according to $\{p_i\}_{i=1}^{n_3}$.
   (a) Let $\mathcal{C}$ be the $n_1 \times n_2 \times c$ tensor consisting of the chosen slabs.
   (b) Let $D_C$ be the $c \times c$ diagonal scaling matrix with $(D_C)_{tt} = \frac{1}{\sqrt{cp_{i_t}}}$ if the $i_t$-th slab is chosen in the $t$-th independent trial.
2. Select $r$ fibers in $r$ i.i.d. trials according to $\{q_i\}_{i=1}^{n_1 n_2}$.
   (a) Let $\mathcal{R}$ be the $r \times n_3$ matrix consisting of the chosen fibers.
   (b) Let $D_R$ be the $r \times r$ diagonal scaling matrix with $(D_R)_{tt} = \frac{1}{\sqrt{rq_{j_t}}}$ if the $j_t$-th slab is chosen in the $t$-th independent trial.
3. Let the $r \times c$ matrix $W$ be the matricized intersection between $\mathcal{C}$ and $\mathcal{R}$.
4. Define the $c \times r$ matrix $\mathcal{U} = D_C \left(D_R W D_C\right)^+ D_R$.

**Figure 2: The $(2+1)$-Tensor-CUR Algorithm**

with probabilities that depend on the Frobenius norms of slabs and Euclidean norms of fibers, respectively. See the proof of Theorem 4 in Section 3.2 and also [8, 9, 10] for a discussion of the algorithmic justification for this sampling.

We emphasize that, as with the matrix CUR decomposition, when this tensor-CUR decomposition is applied to data there is a natural interpretation in terms of underlying data elements. For our imaging application, a "slab" corresponds to an image at a given frequency step and a "fiber" corresponds to a time- or frequency-resolved pixel. Similarly, for our recommendation system application, a "slab" corresponds to a product-product preference matrix for a single user and a "fiber" corresponds to preference information from every user about a single product-product pair.

## 3.2 A General Tensor-CUR Decomposition for Very Large Data Tensors

In this subsection, to provide a theoretical justification for the tensor-CUR decomposition of Section 3.1, we present our main algorithmic result. Our main algorithmic result is a generalization the $(2+1)$-Tensor-CUR algorithm and an associated provable quality-of-approximation bound for the Frobenius norm of the error tensor $\mathcal{A} - \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}$.

The Tensor-CUR algorithm, described in Figure 4, takes as input a $d$-mode tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, a "distinguished" mode $\alpha \in \{1, \ldots, d\}$, a rank parameter $k_\alpha$, an error parameter $\epsilon > 0$, and a failure probability $\delta \in (0,1)$. The algorithm returns as output three carefully constructed subtensors that, when multiplied together, are an approximation $\tilde{\mathcal{A}}$ to $\mathcal{A}$. Both the number of slabs $c_\alpha$ and the number of fibers $r_\alpha$ that are randomly sampled depend on the rank parameter $k_\alpha$, an error parameter $\epsilon$, and a failure probability $\delta$. The subtensors $\mathcal{C}$ and $\mathcal{R}$ are chosen by sampling according to a carefully-constructed nonuniform probability distribution. In order to obtain the provable quality-of-approximation bounds of Theorem 4, the probability distribution depends on the Frobenius norms of the slabs and the Euclidean norms of the fibers, respectively. Intuitively, this biases the random sampling toward the subtensors that are of most interest; see [8, 9, 10] for details.

**Input:** An $n_1 \times n_2 \times \ldots n_d$ tensor $\mathcal{A}$, a mode $\alpha \in \{1, \ldots, d\}$, a rank parameter $k_\alpha$, an error parameter $\epsilon > 0$, and a failure probability $\delta \in (0,1)$.
**Output:** An $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \ldots \times n_d$ tensor $\mathcal{C}$, a $c_\alpha \times r_\alpha$ matrix $\mathcal{U}$, and a $r_\alpha \times n_\alpha$ matrix $\mathcal{R}$.

1. Let $c_\alpha = 4k_\alpha \left(1 + \sqrt{8\log(2/\delta)}\right)^2 / \epsilon^4$, $r_\alpha = 4k_\alpha/\delta^2\epsilon^2$, and $N_\alpha = \prod_{i \neq \alpha} n_i$.
2. Define $\{p_i\}_{i=1}^{n_\alpha}$ to be $p_i = \frac{\left|(A_\alpha)^{(i)}\right|^2}{\|\mathcal{A}\|_F^2}$.
3. Define $\{q_j\}_{j=1}^{N_\alpha}$ to be $q_j = \frac{\left|(A_\alpha)_{(j)}\right|^2}{\|\mathcal{A}\|_F^2}$.
4. Select $c_\alpha$ slabs in $c_\alpha$ i.i.d. trials according to the probability distribution $\{p_i\}_{i=1}^{n_\alpha}$.
   (a) Let $\mathcal{C}$ be the $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \ldots \times n_d$ tensor consisting of the chosen slabs.
   (b) Let $D_C$ be the $c_\alpha \times c_\alpha$ diagonal scaling matrix with $(D_C)_{tt} = \frac{1}{\sqrt{cp_{i_t}}}$ if the $i_t$-th slab is chosen in the $t$-th independent trial.
5. Select $r_\alpha$ fibers in $r_\alpha$ i.i.d. trials according to the probability distribution $\{q_i\}_{i=1}^{N_\alpha}$.
   (a) Let $\mathcal{R}$ be the $r_\alpha \times n_\alpha$ matrix consisting of the chosen fibers (from all the slabs).
   (b) Let $\Psi$ be the $r_\alpha \times c_\alpha$ matrix consisting of the chosen fibers (from the chosen slabs).
   (c) Let $D_R$ be the $r_\alpha \times r_\alpha$ diagonal scaling matrix with $(D_R)_{tt} = \frac{1}{\sqrt{rq_{j_t}}}$ if the $j_t$-th slab is chosen in the $t$-th independent trial.
6. Let $\Phi$ be the best rank-$k$ approximation to the Moore-Penrose generalized inverse of $\left(\mathcal{C} \otimes_\alpha D_C\right)_{[\alpha]}^T \left(\mathcal{C} \otimes_\alpha D_C\right)_{[\alpha]}$.
7. Define the $c_\alpha \times r_\alpha$ matrix $\mathcal{U} = \Phi \left(D_R \Psi\right)^T$.

**Figure 4: The Tensor-CUR Algorithm**

In more detail, the approximation $\tilde{\mathcal{A}}$ is computed by performing the following: first, form (implicitly) each of the $n_\alpha$ subtensors (slabs of mode $d-1$) defined by fixing $i \in \{1, \ldots, n_\alpha\}$ and also form (implicitly) each of the $N_\alpha = \prod_{i \neq \alpha} n_i$ subtensors (fibers of mode 1, i.e., vectors) defined by fixing a value for each of the modes $i \neq \alpha$; second, construct nonuniform probability distributions with respect to which to sample the slabs and the fibers; third, choose $c_\alpha$ of the $d-1$-mode slabs in independent random trials and also choose $r_\alpha$ of the 1-mode fibers in independent random trials; fourth, define the tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d}$ to consist of the $c_\alpha$ chosen $d-1$-mode slabs, and also define the tensor $\mathcal{R} \in \mathbb{R}^{r_\alpha \times n_\alpha}$ to consist of the $r_\alpha$ chosen 1-mode fibers; and finally, let $\mathcal{U} \in \mathbb{R}^{c_\alpha \times r_\alpha}$ be an appropriately-

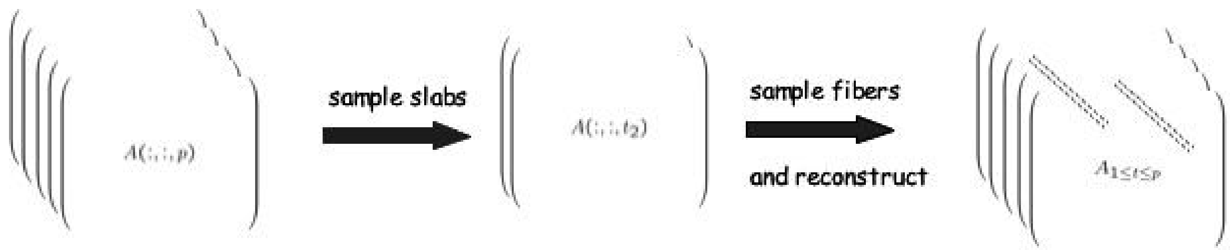


**Figure 3: Pictorial representation of the action of the tensor-CUR decomposition.**

defined and easily-computed (given $\mathcal{C}$ and $\mathcal{R}$) tensor of mode 2 (i.e., matrix). Then, we may define

$$\tilde{\mathcal{A}} = \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}, \tag{12}$$

where $\mathcal{C}\otimes_\alpha\mathcal{U}\mathcal{R}$ is the $\alpha$-mode tensor-matrix product between $\mathcal{C}$ and $\mathcal{U}\mathcal{R}$ to be an $n_1 \times \cdots \times n_{\alpha-1} \times n_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ tensor that is an approximation to the original tensor $\mathcal{A}$. (The awkward form of $\mathcal{U}$ is currently necessary for our provable results. Nevertheless, $\mathcal{U}$ is a subspace-perturbation of the Moore-Penrose generalized inverse of matricized intersection between $\mathcal{C}$ and $\mathcal{R}$. Thus, for the $(2+1)$-Tensor-CUR algorithm and for the applications described in Sections 4 and 5 we have taken it to be exactly this quantity.)

Our main quality-of-approximation bound for the Tensor-CUR algorithm is given by the following theorem, in which we bound the Frobenius norm of the error tensor $\mathcal{A} - \tilde{\mathcal{A}}$. Note that in (13), the $\left\|A_{[\alpha]} - (A_{[\alpha]})_{k_\alpha}\right\|_F$ term is a measure of the extent to which the "unfolded" matrix $A_{[\alpha]}$ is not well-approximated by a rank-$k_\alpha$ matrix, and the $\epsilon \left\|\mathcal{A}\right\|_F$ term is a measure of the loss in approximation quality due to the choice of slabs and fibers (rather than, e.g., the top $k_\alpha$ eigen-slabs and eigen-fibers along the $\alpha$ mode).

Theorem 4. *Let $\mathcal{A}$ be an $n_1 \times n_2 \times \ldots n_d$ tensor, and let $\alpha \in \{1, \ldots, d\}$ be a particular mode, $k_\alpha$ be a rank parameter, $\epsilon > 0$ be an error parameter, and $\delta \in (0, 1)$ be a failure probability. Construct a tensor-CUR approximate decomposition to $\mathcal{A}$ with the output of the* Tensor-CUR *algorithm. Then, with probability at least $1 - \delta$,*

$$\left\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\right\|_F \leq \left\|A_{[\alpha]} - \left(A_{[\alpha]}\right)_{k_\alpha}\right\|_F + \epsilon \left\|\mathcal{A}\right\|_F . \tag{13}$$

*Proof:* Since "unfolding" $\mathcal{A}$ along any mode does not change the value of its Frobenius norm (since it is simply a reordering of indices in a summation) it follows that

$$\left\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\right\|_F = \left\|A_{[\alpha]} - \left(\mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\right)_{[\alpha]}\right\|_F . \tag{14}$$

Note that the Frobenius norm on the left hand side of (14) is a tensor norm and that the Frobenius norm on the right hand side of (14) is a matrix norm. Due to the form of the sampling probabilities used in the Tensor-CUR algorithm, it is this latter quantity that Theorem 5 of [10] bounds. By applying this result [10], the theorem follows.

$\diamond$

With regard to complexity considerations, assume, for simplicity, that the tensor $\mathcal{A}$ is stored externally and assume that $k_i = O(1)$ and that $n_i = n$ for every $i = 1, \ldots, d$. Then the matrices $C_{[i]}$ each occupy only $O(n)$ additional scratch space. In general, $O(n^{d-1})$ additional scratch space will be needed to compute the probabilities of the form used by the Tensor-CUR Algorithm, and this will be comparable to the overall memory requirements if $d$ is large. On the other hand, if the uniform probabilities are approximately optimal for each of the $d$ nodes, then only $O(n)$ additional scratch space and computation time are needed, resulting in a substantial scratch memory and time savings [8].

# 4. APPLICATION TO HYPERSPECTRAL IMAGE DATA

In hyperspectral imagery, an object or scene is imaged at a large number of contiguous wavelengths [33, 34]. Although hyperspectral imagery originated in astronomy and geosensing, it has been employed more recently in numerous other application areas, including agriculture, manufacturing, forensics, and medicine. In many of these applications, target resolution is limited by available spatial resolution. By considering the spectral variation of light intensity, one obtains rich information about the object or scene being imaged that complements traditional spatial information. One also obtains data sets that are very large and contain much redundancy. For example, if a single scene is imaged at 200 frequency bands, where at each frequency a $256 \times 256$ image is generated, then the data cube generated for this single object consists of 13 million values.

When applied to medical samples, a variety of hyperspectral devices have been used to discriminate among, e.g., cell types, tissue patterns, and endogenous and exogenous pigments. Although the increasing power of these methods holds the promise for developing automatic diagnostics, the increased volume and formal dimensionality of the data make the development of more efficient algorithms necessary in order to extract statistically useful and reliable information about the data.

## 4.1 Description of Data and Data Generation

The hyperspectral image data set we consider consists of 59 data cubes derived from 59 biopsies (20 normal, 19 benign adenoma, and 20 malignant carcinoma colon biopsies, one per patient). Each data cube consists of 128 grey-scale images at 400X magnification over the frequency range ca. 440 nm to 700 nm, where each image is $495 \times 656$ pixels in size (for a total of ca. 40 million pixels). Each image is generated using a prototype tuned light source by measuring the modulated light transmitted through the sample. For details about the data and its generation, see [33, 34].

Figure 5 illustrates one of the 128 images, i.e., a hyperspectral image at a single frequency, in a typical (very malignant) sample, and Figure 6 illustrates a typical frequency-

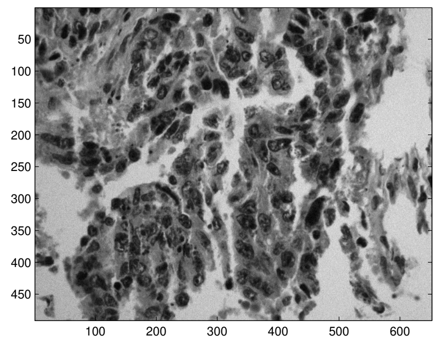

**Figure 5: A very malignant sample at a single frequency in one hyperspectral data cube.**

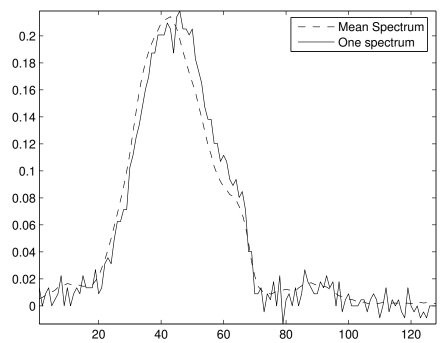


**Figure 6: Average normalized spectrum and a single typical spectrum in one hyperspectral data cube. Vertical axis represents normalized energy-per-frequency in the spectra. Horizontal axis is the slab index.**

resolved pixel and the average spectrum of the ca. 324, 000 frequency-resolved pixels in this data cube. Although not illustrated, both successive images and also pixels from different spatial regions are strongly correlated with one another.

In this imaging application, the tensor $\mathcal{C}$ consists of a small number of dictionary or basis images (which are actual and not eigen-images) with respect to which the remaining images are expressed in an approximately-optimal least-squares manner. Similarly, the matrix $\mathcal{R}$ consists of the spectral variation of a small number of dictionary or basis pixels with respect to which the spectral variation of the remaining pixels are expressed.

In the next two subsections, we will see that the tensor-CUR decomposition can be applied to this hyperspectral image data in order to compress the data and to perform two classification tasks of interest on the data. Slabs will be chosen randomly with a probability proportional to the average normalized spectrum of Figure 6 and fibers will be chosen uniformly at random. The data-dependent motivation for this is that the intensity of transmitted light captures a meaningful notion of information as a function of varying frequency, but not as a function of varying spatial coordinates due to the particular staining technology.

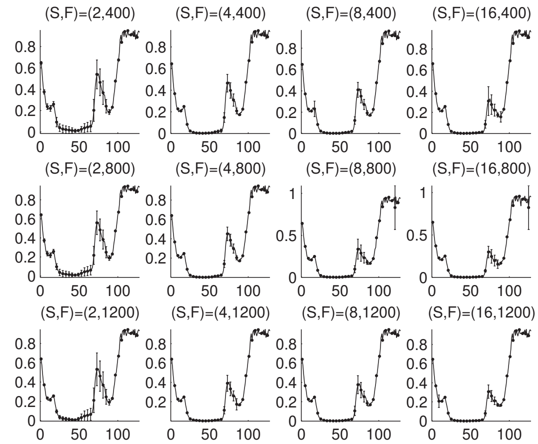


**Figure 8: Reconstruction error. Caption indicates how many slabs (S) and fibers (F) were sampled. Vertical axis is the relative reconstruction error (for the Frobenius norm). Horizontal axis is the slab index. Average and standard deviation are over 4 slab draws and 3 fiber draws.**

## 4.2 Reconstruction of Hyperspectral Data

For each slab we did not randomly sample, we use the tensor-CUR decomposition to reconstruct that slab in an approximately-optimal least-squares sense in the basis provided by the sampled slabs, and we do so using only a small number of pixels in that slab. In Figure 7 we present a representative example of the reconstruction of one spectral slice in a normal biopsy. The redundancy in the data is evident by the quality of the reconstruction under very heavy downsampling. For example, it suffices to judiciously choose as few as 8 or even 2 of the original 128 slabs, and to reconstruct the remaining slabs it suffices to choose ca. 1000 (or fewer) of the original ca. 324, 000 fibers.

In Figure 8, we present the approximation error as a function of downsampling to different number of slabs and then to different number of fibers. Clearly, due to the form of the slab sampling probabilities, slabs between ca. 30 and ca. 60 tend to be reproduced much better than those toward the tails of the spectrum. Slabs below ca. 20 and above ca. 70 tend to have a lower signal-to-noise ratio and are less important for the problem of approximate data reconstruction (but not necessarily for other problems).

A close examination of images such as those presented in Figure 7 reveals a subtle interplay between sampling-induced error and denoising due to the low-dimensionality of the sample. Note that by permitting our algorithm to sample different numbers of slabs and fibers, we can, e.g., sample slabs to a level appropriate for structure identification and sample more fibers for denoising purposes.

## 4.3 Classification of Hyperspectral Data

In medical applications, one is interested in the classification of an entire data cube, i.e., a medical sample, as normal or malignant. Since nuclei are the most discriminative structures for this task, as an intermediate step, one is interested in classifying the pixels in a single data cube into different

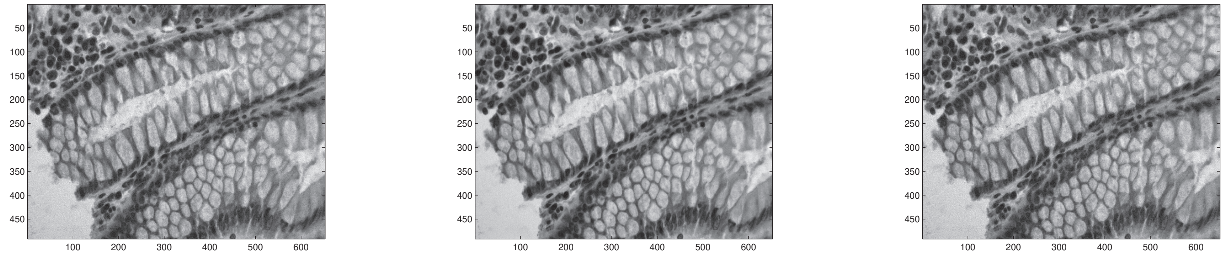

**Figure 7: Typical reconstruction of the hyperspectral data for one spectral slice in a normal biopsy. (Slab 60, when ordered with respect to frequency as in Figure 6, is shown.) From left to right: original data, reconstruction from 8 slabs and 1200 fibers, and reconstruction from 2 slabs and 1200 fibers.**

**Table 1: Confusion matrix of predictions of normal and malignant nuclei (patches of size 64 by 64, with averaged 10-fold cross-validated error). TN, TM stand for True Normal and True Malignant, and PN, PM stand for the corresponding predictions. Classification is based on using all 128 slabs, 16, 8, or 2 slabs, as indicated.**

| all slabs | PN | PM |
|---|---|---|
| TN | 79% | 21% |
| TM | 26% | 74% |

| 8 slabs | PN | PM |
|---|---|---|
| TN | 78% | 22% |
| TM | 29% | 71% |

| 16 slabs | PN | PM |
|---|---|---|
| TN | 77% | 23% |
| TM | 30% | 70% |

| 2 slabs | PN | PM |
|---|---|---|
| TN | 68% | 32% |
| TM | 33% | 67% |

tissue types, e.g., nuclei, cytoplasm, or lamina propria. For details on the classification procedures, see [33, 34]. We simply note that for the normal versus malignant classification task, we have access to a label (assumed correct) provided by a pathologist [33, 34], while no such ground truth is available for the tissue classification.

In Figure 9, we present typical results for the tissue classification task in the exact data cube and in two downsampled and reconstructed data cubes. The two examples presented involve sampling 16 and 8 slabs, respectively, and as with the reconstruction problem, in both cases there is little quality loss until the number of fibers samples is less than ca. 1000. As before, a careful analysis reveals a complex interplay between sampling-induced information loss and sampling-induced denoising. If the nuclei identified by this tissue classification are then used to classify data cubes as normal or malignant, the results can be compared with the true value. Results of the confusion matrix for this classification task are presented in Table 1 [33, 34]. High quality results are obtained using samples of 16 and 8 slabs, but quality degrades if only 2 slabs are used. Similar results are seen when we classify into normal, abnormal, and malignant.

# 5. APPLICATION TO RECOMMENDATION SYSTEM ANALYSIS

In recommendation system analysis, one is typically interested in making purchase recommendations to a user at an electronic commerce web site. Collaborative methods (as opposed to content-based or hybrid) involve recommending to the user items that people with similar tastes or preferences liked in the past. Probably the most well-known example of a collaborative filtering system is that of `Amazon.com`, which is based on rules of the form "users who are interested in item X are also likely to be interested in item Y" [32]. Many collaborative filtering algorithms represent a user as an $n$ dimensional vector, where $n$ is the number of distinct products, and where the components of the vector are a measure of the rating provided by that user for that product. Thus, for a set of $m$ users, the user-product ratings matrix is an $m \times n$ matrix $A$, where $A_{ij}$ is the rating by user $i$ for product $j$ (or is null if the rating is not provided). A recommendation algorithm generates recommendations for a new user based on a few user who are most similar to the user, after querying the new user about his (or her) rating on a small number of products. For details, see [35, 5, 1].

A matrix CUR decomposition has been used to obtain competitive recommendation performance by judiciously sampling $O(m+n)$ entries of the user-product ratings matrix and reconstructing missing entries [11]. In more detail, assuming access to a matrix $C$ consisting of the ratings of every user for a small number of products and a matrix $R$ consisting of the ratings of a small number of users for every product, then under assumptions $CUR$ is a provably good approximation to the user-product matrix $A$ [11]. Other theoretical work includes [27, 2, 23], and other applications of linear algebra have used the SVD for dimensionality reduction [4, 36, 15].

Although the ratings in the user-product matrix $A$ are often interpreted in terms of the utility of product $j$ for user $i$, utility in neoclassical economics is an ordinal and not a cardinal concept. This is since utility functions are constructs that encode preference information and since the same preferences are described when the utility function is subject to a wide class of monotonic transformations. This observation motivates the definition of an $m \times n \times n$ user-product-product $(2+1)$-tensor $\mathcal{A}$, where $\mathcal{A}_{ijk}$ is $+1$ or $-1$ depending on whether product $j$ or product $k$ is preferred by user $i$. Similar preference-based models have appeared [7, 14, 22, 21], and have been motivated by such observations as that two users with very similar preferences over items may have very different rating schemes. When faced with a new user, this preference model depends on obtaining pairwise preference information such as that the user bought product

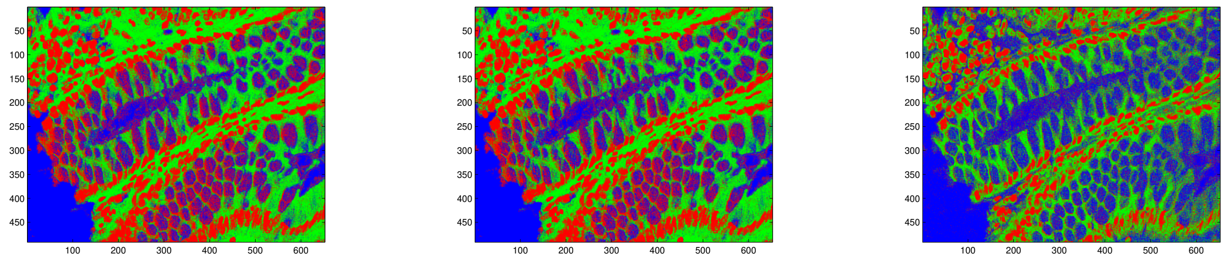

**Figure 9: Segmentation into 3 tissue types in a normal biopsy: red for nuclei (the only class that we are interested in for the next classification task), green for cytoplasm, and blue for lamina propria and other regions. From left to right: classification on original data; on compressed data (16 slabs and 1200 fibers); and on compressed data (8 slabs and 1200 fibers).**

A when he could have bought product B, or that the user clicked on link A when he could have clicked on link B.

## 5.1 Description of Data and the Model

Under this preference model for recommendation system analysis, the tensor $\mathcal{C}$ consists of a small number of dictionary or basis elements from a small number of users, where each element corresponds to the full $n \times n$ pairwise preference matrix for a single user. Similarly, the matrix $\mathcal{R}$ consists of a dictionary or basis set of preference information from every user about a small number of product-product pairs.

In the next subsection, we will see that the tensor-CUR decomposition can be applied to recommendation system data under this model to reconstruct missing entries in the user-product-product preference tensor in order to make high-quality recommendations. Since most recommendation system databases do not provide data in this preference-based format, the data set we will consider will be derived from the ratings in the well-studied Jester data [15]. As an initial application, we consider the $m = 14,116$ (out of ca. $73,421$) users who rated all of the $n = 100$ products (i.e., jokes). From this $m \times n$ user-product ratings matrix, we define an $m \times n \times n$ user-product-product preference tensor by performing the following for each user: convert the $n$ dimensional rating vector into an $n \times n$ preference matrix in which the $ij$ entry is $+1$ or $-1$ depending on whether or not the user prefers product $i$ to product $j$. (Although this results in ordered and fully-consistent preferences, this is not required by our decomposition.) In this application, in the absence of a better model, both slabs and fibers will be chosen uniformly at random.

## 5.2 Recommendation Quality Results

For each slab (i.e., user) we did not randomly sample, we use the tensor-CUR decomposition to reconstruct that slab in an approximately-optimal least-squares sense in the basis provided by the sampled slabs, and we do so using only a small number of product-product preference queries from that slab. Then we will use this reconstruction to make recommendations by approximating the reconstructed matrix of preferences, and picking the 10 products with the largest row sums. We will make 10 recommendations, and we will evaluate the quality of our recommendation with the Top-$N$ procedure [36], i.e., by the number of products out of the exact top 10 that we correctly recommend.

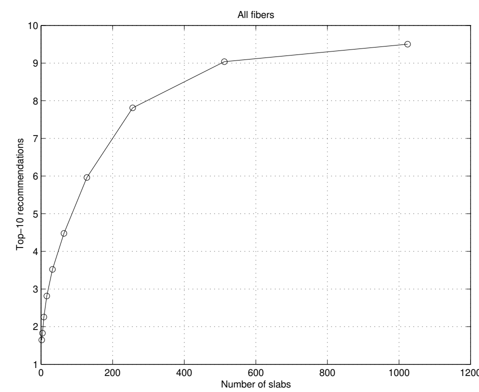


**Figure 10: Average number of successful recommendations out of the top 10 for a basis consisting of a varaible number of users but using complete pairwise product-product preference information.**

In order to determine an upper bound on the quality of recommendations based on using a small number of basis slabs, consider Figure 10, which shows the average number of recommendations out of the top 10 that can be captured using a small number of basis slabs. In this figure, we use full fiber information, and thus we are considering the exact least-squares fit of a new slab on the space spanned by the basis slabs. For example, using 128 basis slabs, we can hope to predict up to 4.5, 6, or 8 of the top 10 items by sampling 64, 128, or 256 fibers, respectively. As a lower bound on quality, we expect that we will make ca. 1 prediction correctly since we are making 10 predictions and there are 100 products.

In Figure 11, we show that by using a basis of preference information from 128 users and performing a small number of product-product preference queries on a new user, we can make a large number of high-quality recommendations. Similar results are seen with 64 and 256 basis slabs. Since we are sampling a small number of fibers in this case, we are performing an approximate least-squares fit using just the information about a new user contained in a small number of fibers. The number of top-10 recommendations is competitive with the best possible using the small basis and is well-above the random level. Note the nonmonotonicity near

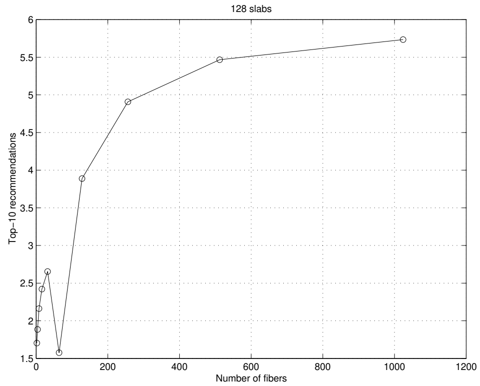


**Figure 11: Average number of successful recommendations out of the top 10 for a basis consisting of 128 users versus the number of pairwise product-product preference queries.**

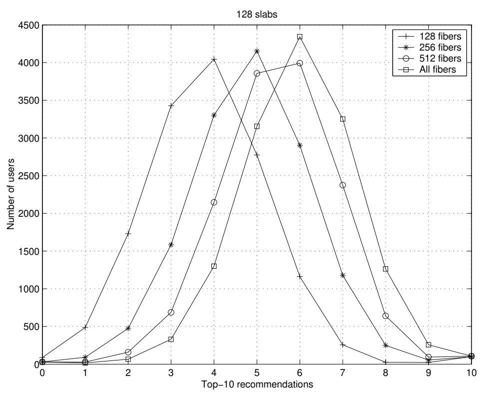


**Figure 12: Distribution of number of users making a given number of successful top 10 recommendations for a basis consisting of 128 users.**

ca. 64 queries; this may be a fitting issue and is the subject of further exploration. Finally, in Figure 12, we present the distribution of correct predictions for the 14,116 users by using 128 slabs and a variable number of fiber queries.

In evaluating performance, we distinguish between prediction and reconstruction. In the former, we want to know how much user $i$ will like product $j$ (in a ratings model) or whether user $i$ will prefer product $j$ or product $k$ (in a preference model). In the latter, which is of interest to us, we want to give a list of, e.g., the top 10 products for user $i$. We use tensor reconstruction as an intermediate step to making high-quality recommendations.

# 6. CONCLUSION

We conclude with several related extensions of the present work. First, it would be worth examining how these methods can be coupled with more traditional methods of image analysis and recommendation system analysis. This could be performed either by choosing slabs and fibers and then analyzing each slab or fiber with more traditional methods, or by using structural insights from more traditional methods to construct the sample of slabs and fibers. Second, it would be worth determining whether the sample of slabs and/or fibers could be chosen to preserve some interesting multilinear structure in the data tensors that is damaged by the sampling techniques we have used. Third, it would be worth determining the extent to which it would be possible to combine fibers from several data cubes into a "dictionary" that could be used, along with a few slabs in a new data cube, to describe the entire new data cube.

**Acknowledgments** We thank the authors of [33, 34], and in particular Gustave L. Davis of Yale University, for making available the hyperspectral data. M. Maggioni is partially supported by NSF-DMS grant 0512050.

# 7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[2] Y. Azar, A. Fiat, A.R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 619–626, 2001.

[3] M.W. Berry, S.A. Pulatova, and G.W. Stewart. Computing sparse reduced-rank approximations to sparse matrices. Technical Report UMIACS TR-2004-32 CMSC TR-4589, University of Maryland, College Park, MD, 2004.

[4] D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceedings of the 15th International Conference on Machine Learning*, pages 46–54, 1998.

[5] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[6] J.D. Carroll and J.J. Chang. Analysis of individual differences in multidimensional scaling via an $N$-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[7] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. In *Annual Advances in Neural Information Processing Systems 10: Proceedings of the 1997 Conference*, pages 451–457, 1998.

[8] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *To appear in: SIAM Journal on Computing.*

[9] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *To appear in: SIAM Journal on Computing.*

[10] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *To appear in: SIAM Journal on Computing.*

[11] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 82–90, 2002.

[12] P. Drineas and M.W. Mahoney. A randomized algorithm for a tensor-based generalization of the Singular Value Decomposition. *To appear in: Linear Algebra and its Applications.*

[13] P. Drineas and M.W. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[14] Y. Freund, R. Iyer, R.E. Schapire, and Y Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[15] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4:133–151, 2001.

[16] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.

[17] S.A. Goreinov and E.E. Tyrtyshnikov. The maximum-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–51, 2001.

[18] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, 261:1–21, 1997.

[19] R.A. Harshman and M.E. Lundy. The PARAFAC model for three-way factor analysis and multidimensional scaling. In H.G. Law, C.W. Snyder Jr., J. Hattie, and R.P. McDonald, editors, *Research Methods for Multimode Data Analysis*, pages 122–215. Praeger, 1984.

[20] J. Håstad. Tensor rank is $NP$-complete. *Journal of Algorithms*, 11(2):644–654, 1990.

[21] R. Jin, L. Si, and C.X. Zhai. Preference-based graphic models for collaborative filtering. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 329–336, 2003.

[22] R. Jin, L. Si, C.X. Zhai, and J. Callan. Collaborative filtering with decoupled models for preferences and ratings. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 309–316, 2003.

[23] J. Kleinberg and M. Sandler. Using mixture models for collaborative filtering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 569–578, 2004.

[24] P.M. Kroonenberg and J. De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.

[25] J.B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and Its Applications*, 18:95–138, 1977.

[26] J.B. Kruskal. Rank, decomposition, and uniqueness for 3-way and $N$-way arrays. In R. Coppi and S. Bolasco, editors, *Multiway Data Analysis*, pages 7–18. Elesvier Science Publishers, 1989.

[27] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: a probabilistic analysis. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 664–673, 1998.

[28] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[29] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1,R_2,\ldots,R_N)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[30] D. Leibovici and R. Sabatier. A singular value decomposition of a $k$-way array for a principal component analysis of multiway data, PTA-$k$. *Linear Algebra and Its Applications*, 269:307–329, 1998.

[31] L.-H. Lim and G.H. Golub. Tensors for numerical multilinear algebra: ranks and basic decompositions. Technical Report 05-02, Stanford University SCCM, Stanford, CA, April 2005.

[32] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, 2003.

[33] M. Maggioni, G.L. Davis, F.J. Warner, F.B. Geshwind, A.C. Coppi, R.A. DeVerse, and R.R. Coifman. Algorithms from signal and data processing applied to hyperspectral analysis: Discriminating normal and malignant microarray colon tissue sections using a novel digital mirror device system. *Manuscript*, 2006.

[34] M. Maggioni, G.L. Davis, F.J. Warner, F.B. Geshwind, A.C. Coppi, R.A. DeVerse, and R.R. Coifman. Hyperspectral microscopic analysis of normal, benign and carcinoma microarray tissue sections. *Manuscript*, 2006.

[35] P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[36] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system - a case study. In *Proceedings of the WebKDD 2000 Workshop*, pages 000–000, 2000.

[37] G.W. Stewart. Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix. *Numerische Mathematik*, 83:313–323, 1999.

[38] G.W. Stewart. Error analysis of the quasi-Gram-Schmidt algorithm. Technical Report UMIACS TR-2004-17 CMSC TR-4572, University of Maryland, College Park, MD, 2004.

[39] L.R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[40] C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Annual Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 682–688, 2001.