SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 957–987

TENSOR-CUR DECOMPOSITIONS FOR TENSOR-BASED DATA*

MICHAEL W. MAHONEY†, MAURO MAGGIONI‡, AND PETROS DRINEAS§

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 from 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 a natural interpretation in terms of the processes generating the data. Assume the data may be modeled as a $(2+1)$-tensor, i.e., an $m \times n \times p$ tensor $\mathcal{A}$ in which the first two modes are similar and the third is qualitatively different. We refer to each of the p different $m \times n$ matrices as "slabs" and each of the mn different p-vectors as "fibers." In this case, the tensor-CUR algorithm computes an approximation to the data tensor $\mathcal{A}$ that is of the form $\mathcal{CUR}$, where $\mathcal{C}$ is an $m \times n \times c$ tensor consisting of a small number c of the slabs, $\mathcal{R}$ is an $r \times p$ matrix consisting of a small number r of the fibers, and $\mathcal{U}$ is an appropriately defined and easily computed $c \times r$ encoding matrix. Both $\mathcal{C}$ and $\mathcal{R}$ may be chosen by randomly sampling either slabs or fibers according to a judiciously chosen and data-dependent probability distribution, and both c and r depend on a rank parameter k, an error parameter ϵ, and a failure probability δ. Under appropriate assumptions, provable bounds on the Frobenius norm of the error tensor $\mathcal{A} - \mathcal{CUR}$ are obtained. 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.

Key words. CUR decomposition, tensor decomposition, hyperspectral imagery, recommendation system

AMS subject classifications. 15A23

DOI. 10.1137/060665336

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 [44, 31, 46, 61, 11]. 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

*Received by the editors July 17, 2006; accepted for publication (in revised form) by L. De Lathauwer January 8, 2007; published electronically September 25, 2008. A preliminary version of this paper appeared in *Proceedings of the 12th Annual ACM SIGKDD Conference*, 2006, pp. 327–336.
http://www.siam.org/journals/simax/30-3/66533.html

†Yahoo Research, Sunnyvale, CA 94089 (mahoney@yahoo-inc.com). Part of this work was performed while this author was at the Department of Mathematics, Yale University, New Haven, CT 06520.

‡Department of Mathematics, Duke University, Durham, NC 27708 (mauro.maggioni@duke.edu). Part of this work was performed while this author was at the Department of Mathematics, Yale University, New Haven, CT 06520. This author's research was partially supported by NSF-DMS grant 0512050.

§Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 (drinep@cs.rpi.edu).

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 ± 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 multimode 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 [32]. On the other hand, the model proposed by Tucker [61], as well as the related "canonical decomposition" [11] or "parallel factors" models [31], have a long history in applied data analysis [39, 40, 41, 44]. 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 SVD to multimode data tensors [44, 45, 46, 48].

A seemingly unrelated line of work has focused on matrix CUR decompositions [19, 22, 23]. 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 [19]. Thus, a matrix CUR 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.

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 from 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. In this section, we provide a brief review of relevant multilinear algebra as well as recent work on matrix CUR decompositions.

2.1. Tensor-based extension of the SVD. We shall use calligraphic letters to denote higher-order or multimode 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 $n_1 \times n_2 \times \cdots \times n_d$ and let $N_\alpha = \prod_{i \neq \alpha} n_i$. Consider the following definitions:

- Given a tensor $\mathcal{A}$ and a particular mode $\alpha \in \{1, \ldots, d\}$, define the matrix $A_{[\alpha]} \in \mathbb{R}^{n_\alpha \times N_\alpha}$, where the columns of the matrix consist of varying the αth coordinate of $\mathcal{A}$ while leaving the rest fixed. We refer to the (usually implicit) construction of $A_{[\alpha]}$ as *matricizing* [36] or *unfolding* [44] $\mathcal{A}$ along mode α and define the *α-rank* of the tensor $\mathcal{A}$ to be the rank of the matrix $A_{[\alpha]}$.
- Given an $n_1 \times n_2 \times \cdots \times n_d$ d-mode tensor $\mathcal{A}$, a particular mode α, and any $n_\alpha \times c_\alpha$ matrix B, define the *α-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

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

  Note that the α-mode tensor-matrix product satisfies $(\mathcal{A} \otimes_\alpha B) \otimes_{\alpha'} C = (\mathcal{A} \otimes_{\alpha'} C) \otimes_\alpha B = \mathcal{A} \otimes_\alpha B \otimes_{\alpha'} C$, assuming that the various individual products are defined.
- Given a tensor $\mathcal{A}$, let us denote the SVD of $A_{[\alpha]}$ by

$$A_{[\alpha]} = U_{[\alpha]} \Sigma_{[\alpha]} V_{[\alpha]}^T, \tag{2}$$

  where, e.g., $U_{[\alpha]}$ is an $n_\alpha \times \operatorname{rank}(A_{[\alpha]})$ matrix and $U_{[\alpha],k_\alpha}$ is an $n_\alpha \times k_\alpha$ matrix consisting of the left singular vectors corresponding to the top k_α singular values of $A_{[\alpha]}$.
- Given a d-mode tensor $\mathcal{A}$, define the (square of its) *Frobenius norm* to be

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}_{i_1 \cdots i_d}^2. \tag{3}$$

- Given a tensor $\mathcal{A}$ and a particular mode α, let us refer to *slabs* as each of the n_α $d-1$-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 αth coordinate to some particular value $i_\alpha \in \{1, \ldots, n_\alpha\}$. Similarly, let us refer to as *fibers* each of the N_α vectors (mode-one tensors) of size n_α constructed by fixing each of the other coordinates to a particular value.

Remark. See [44, 45, 36] and the references therein for a more detailed description of these tensor-related definitions. In particular, note that, although they will not be of interest to our main result, the *higher-order SVD* of $\mathcal{A}$ has been defined as the decomposition of $\mathcal{A}$ of the form $\mathcal{A} = \mathcal{S} \times_1 U_{[1]} \times_2 \cdots \times_d U_{[d]}$, where the $\operatorname{rank}(A_{[1]}) \times \cdots \times \operatorname{rank}(A_{[d]})$ tensor $\mathcal{S}$ is the so-called core tensor, and the *best rank-$(k_1, k_2, \ldots, k_d)$ approximation* to the tensor $\mathcal{A}$ has been defined as $\tilde{\mathcal{A}} = \mathcal{S} \times_1 U_{[1],k_1} \times_2 \cdots \times_d U_{[d],k_d}$. See [23] for a randomized algorithm that computes an approximation to this quantity. The algorithm of [23] is similar to the algorithms presented in this paper, except that it "unfolds" the tensor along every mode and computes an approximation to the top singular vectors of the unfolded matrix by random sampling.

Remark. Tensors are a natural generalization of matrices (see, e.g., [30] for more details) and have been studied in several fields. For example, tensors have been studied in mathematics and computer science for their algebraic properties, their ability to efficiently represent multidimensional functions, and the relationship between their properties and problems in complexity theory [30, 27, 32, 50, 8]. In addition, tensors provide a natural way to represent many large and complex data sets [44, 43, 31, 36, 46, 61, 11, 65].

Remark. The dimensionality of the linear space generated by the α-slabs is the α-rank of $\mathcal{A}$. It is worth emphasizing that 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, that only weak bounds are known relating the α-rank and the tensor rank, and that there do not exist definitions of tensor rank and associated tensor SVD such that the optimality properties of the matrix rank and matrix SVD are preserved [40, 33, 41, 32, 45, 38, 48, 67].

2.2. Matrix CUR decomposition. Recent work in theoretical computer science, numerical linear algebra, and statistical learning theory [19, 23, 59, 60, 7, 29, 28, 66, 22] 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*

$$\tilde{A} = CUR. \tag{4}$$

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 from 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 it does this similarly for the rows.

Finally, and most relevant for the present paper, note that a matrix CUR 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 [19].

THEOREM 1. *There exists a randomized algorithm (see the* LINEARTIMECUR *algorithm of* [19]*) that takes as input an $m \times n$ matrix A and a fixed rank parameter*

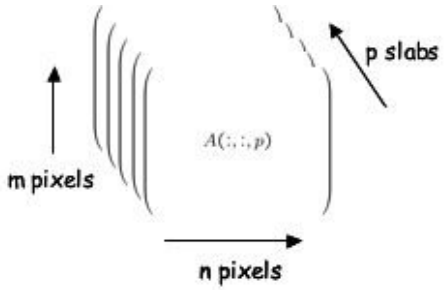

FIG. 1. *Pictorial representation of a* $(2+1)$*-data tensor.*

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 a $c \times r$ 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

$$\|A - CUR\|_F \leq \|A - A_k\|_F + \epsilon \|A\|_F \tag{5}$$

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 [17, 18, 19, 22].

3. A tensor-based extension of the matrix CUR 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 3-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."

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 [19]. More precisely, this algorithm computes the

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 an $r \times n_3$ matrix $\mathcal{R}$.

1. Select c slabs of $\mathcal{A}$ in c independent and identically distributed (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 $(D_C)_{tt} = \frac{1}{\sqrt{c p_{i_t}}}$ if the i_tth slab is chosen in the tth independent trial.
2. Select r fibers of $\mathcal{A}$ 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 $(D_R)_{tt} = \frac{1}{\sqrt{r q_{j_t}}}$ if the j_tth slab is chosen in the tth 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 $\mathcal{U} = D_C \left(D_R W D_C\right)^+ D_R$.

FIG. 2. *The* $(2+1)$-TENSOR-CUR *algorithm.*

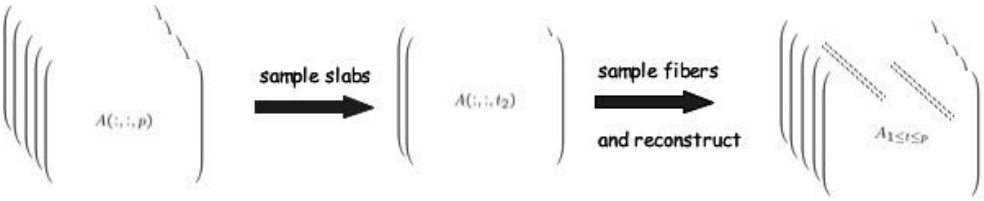

FIG. 3. *Pictorial representation of the action of the tensor-CUR decomposition.*

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 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 an $n_1 \times n_2 \times n_3$ tensor. Thus, by using the $(2+1)$-TENSOR-CUR algorithm, we make the following approximation:

$$\mathcal{A} \approx \tilde{\mathcal{A}} = \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}. \tag{6}$$

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:

$$\mathcal{A}(:,:,i) \approx \sum_{\xi \in C} \mathcal{A}(:,:,\xi) X(\xi, i), \tag{7}$$

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 ith 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 along with the information in a small number of fibers (depicted as the dashed lines). The extent to which (6) or (7) is a good approximation has to do with the selection of slabs and fibers. In sections 4 and 5, we show that (6) holds empirically for our two applications if the slabs and fibers are chosen uniformly and/or nonuniformly with probabilities that depend on the Frobenius norms of slabs and Euclidean norms of fibers, respectively. See the proof of Theorem 2 in section 3.2 and also [17, 18, 19] 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 of the $(2+1)$-Tensor-CUR algorithm and an associated provable quality-of-approximation bound for the Frobenius norm of the error tensor $\mathcal{A} - \mathcal{C} \otimes_3 \mathcal{UR}$.

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_α, 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_α and the number of fibers r_α that are randomly sampled depend on the rank parameter k_α, an error parameter ϵ, and a failure probability δ. 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 2, 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 [17, 18, 19] for details.

In more detail, the approximation $\tilde{\mathcal{A}}$ is computed by performing the following: first, form (implicitly) each of the n_α 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 which to sample the slabs and the fibers; third, choose c_α of the $d-1$-mode slabs in independent random trials, and also choose r_α 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_α 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_α chosen 1-mode fibers; and finally, let $\mathcal{U} \in \mathbb{R}^{c_\alpha \times r_\alpha}$ be an appropriately defined and easily computed (given $\mathcal{C}$ and $\mathcal{R}$) tensor of mode 2 (i.e., matrix). Then we may define

$$\tilde{\mathcal{A}} = \mathcal{C} \otimes_\alpha \mathcal{UR}, \tag{8}$$

where $\mathcal{C} \otimes_\alpha \mathcal{UR}$ is the α-mode tensor-matrix product between $\mathcal{C}$ and $\mathcal{UR}$, 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

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

1. Let $c_\alpha = 4k_\alpha\big(1 + \sqrt{8\log(2/\delta)}\big)^2/\epsilon^4$, $r_\alpha = 4k_\alpha/\delta^2\epsilon^2$, 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_α slabs of $\mathcal{A}$ in c_α i.i.d. trials according to the probability distribution $\{p_i\}_{i=1}^{n_\alpha}$.
   (a) Let $\mathcal{C}$ be the $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ 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_tth slab is chosen in the tth independent trial.
5. Select r_α fibers of $\mathcal{A}$ in r_α 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 Ψ 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_tth slab is chosen in the tth independent trial.
6. Let Φ be the best rank-k approximation to the Moore–Penrose generalized inverse of $(\mathcal{C} \otimes_\alpha D_C)_{[\alpha]}^T (\mathcal{C} \otimes_\alpha D_C)_{[\alpha]}$.
7. Define the $c_\alpha \times r_\alpha$ matrix $\mathcal{U} = \Phi\,(D_R\Psi)^T$.

FIG. 4. *The* TENSOR-CUR *algorithm.*

of the 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 $\tilde{\mathcal{E}} = \mathcal{A} - \tilde{\mathcal{A}}$.

THEOREM 2. *Let $\mathcal{A}$ be an $n_1 \times n_2 \times \cdots \times n_d$ tensor, and let $\alpha \in \{1, \ldots, d\}$ be a particular mode, k_α 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 $1 - \delta$,*

$$\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\|_F \leq \left\|A_{[\alpha]} - \left(A_{[\alpha]}\right)_{k_\alpha}\right\|_F + \epsilon\,\|\mathcal{A}\|_F\,. \tag{9}$$

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

$$\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\|_F = \left\|A_{[\alpha]} - (\mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R})_{[\alpha]}\right\|_F\,. \tag{10}$$

Note that the Frobenius norm on the left-hand side of (10) is a tensor norm and that the Frobenius norm on the right-hand side of (10) 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 [19] bounds. By applying this result [19], the theorem follows. □

3.3. Remarks on tensor-CUR decompositions and data applications.

Remark. In (9), 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_α matrix, and the $\epsilon \|\mathcal{A}\|_F$ 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_α eigenslabs and eigenfibers along the α mode). This latter measure is of the form of an arbitrary (but fixed) precision, scaled by a measure of the size of the tensor $\mathcal{A}$.

Remark. The values for c_α and r_α in general differ, as they do with matrix CUR decompositions. Although this is an artifact of the proof techniques [19], this allows for greater flexibility in data applications. For example, if the noise properties of the slabs and fibers differ, then one may wish to oversample the slabs or fibers in different ways.

Remark. The choice for slabs and fibers in the TENSOR-CUR algorithm takes advantage only of linear and not multilinear structure in the data tensors. Equivalently, the algorithm reduces to the corresponding matrix algorithm. It is an open problem whether one can choose slabs and/or fibers to preserve some nontrivial multilinear tensor structure in the original tensor $\mathcal{A}$.

Remark. A crucial decision in applying these techniques to data will be the proper choice (if any) of the preferred mode α. This depends on the application area from which the data are drawn. The theorems will be true but uninteresting if this choice is not made carefully.

Remark. 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. See [17] for additional discussion of resource issues within the framework of the pass-efficient model of data streaming computation.

Remark. Although sampling with respect to the proper probability distribution is critical for our provable results, one might expect that in many cases the slabs and/or fibers will all be approximately the same length due to the manner in which the data are generated, in which case uniform sampling may be successfully employed. This was seen to be the case for an application of the CUR algorithm of [19] to kernel-based learning [21, 22, 66].

Remark. Alternatively, one might expect that in many cases the data are generated in such a way that information about the Frobenius norm of each of the slabs and/or fibers is easily computed at the data generation step. For example, in the case of a $(2+1)$-imaging application, the Frobenius norm of a slab corresponds to the total absorption at one time step or frequency value. In this case, these approximations to the probabilities could be used in the TENSOR-CUR algorithm.

Remark. Although $c_\alpha = 4k_\alpha\left(1+\sqrt{8\log(2/\delta)}\right)^2/\epsilon^4$ slabs and $r_\alpha = 4k_\alpha/\delta^2\epsilon^2$ fibers suffice to prove the claims of Theorem 2, they can be rather large for even moderate values of k_α, δ, and ϵ. In the applications we consider, choosing many fewer slabs and

fibers suffices, e.g., on the order of tens or hundreds; see sections 4 and 5 for more detail.

4. Application to hyperspectral image data. In hyperspectral imagery, an object or scene is imaged at a large number of contiguous wavelengths [51]. 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 very large data sets that may be represented as a tensor and that contain much redundancy. For example, if a single scene is imaged at 128 frequency bands, where at each frequency a 495×656 image is generated, then the data cube generated for this single object consists of 40 million values and may be represented by a $495 \times 656 \times 128$ tensor $\mathcal{A}$, where $\mathcal{A}_{ijk}$ represents the absorbed or transmitted light intensity at pixel ij at physical frequency k.

In this section, we describe an application of the tensor-CUR decomposition to a problem in hyperspectral medical image analysis. In particular, the tensor-CUR decomposition is used to *compress* the data, and we show that tissue segmentation and nuclei classification quality are not substantially reduced even after substantial data compression. In more detail, in section 4.1, we describe the data and its generation. Then, in section 4.2, we describe the reconstruction of the full data from a small sample of slabs and fibers. In section 4.3, we describe the classification task of tissue segmentation, i.e., classifying the pixels in a single image into different tissue types, as a function of how heavily we downsample on the slabs and fibers. This task is of intermediate interest, since nuclei are the most discriminative structures in the final classification task of interest. Finally, in section 4.4, we describe the classification of data cubes into, e.g., normal and malignant, as a function of downsampling on the slabs and fibers.

4.1. Description of data and data generation. The application of hyperspectral imaging to medicine, and pathology in particular, while not new, is becoming more widespread and powerful. A variety of proprietary spectral splitting devices, including prisms and mirrors [64], interferometers [25, 55], variable interference filter-based monochromometers [53], and tuned liquid crystals [47], mounted on microscopes in combination with CCD cameras and computers, have been used to discriminate among cell types, tissue patterns, and endogenous and exogenous pigments [47]. 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.

The prototype-tuned light source used to generate the data we studied (Plain Sight Systems, Inc.; see [16] for details) can generate a large number of combinations of light frequencies, ranging from about 440 nm to about 700 nm, with a wavelength resolution of up to approximately 6 nm. The light modulated by the prototype is shone via a fiber optic cable directed in a Nikon Biophot microscope and transilluminates hematoxylin and eosin (H & E) stained microarray tissue sections of normal, benign (adenoma), and malignant carcinoma colon biopsies. Hyperspectral photomicrographs, collected in random order at 400X magnification, are obtained with a CCD camera (Sensovation) from 59 different patient biopsies (20 normal, 19 benign ade-

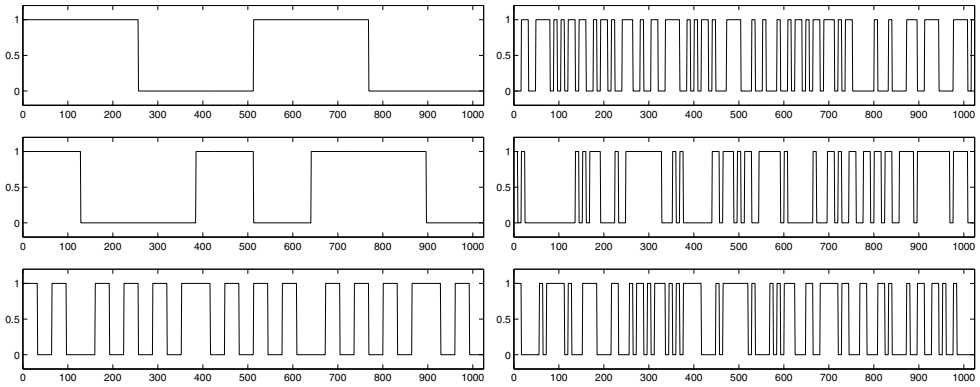

FIG. 5. *Examples of Hadamard patterns (left) and randomized Hadamard patterns (right). The latter are used at the data generation step to improve the signal-to-noise ratio; see the text for details.*

noma, and 20 malignant carcinoma), mounted as a microarray on a single glass slide [14, 2, 3, 5]. From these, 59 hyperspectral grayscale images at 400X magnification are derived. The biopsies are collected randomly on the slide across and within the different groups of biopsies in order to avoid any biases due to instrumentation, e.g., due to temperature or time of collection. This data was collected by G. L. Davis, M.D., as discussed in [51].

Each measurement yields a data cube, which is a set $\{I_i\}_{i=1\ldots128}$ of 128 images, each of which is 495 by 656 pixels in size. (That is, there is one data cube for each of the 59 biopsies.) The intensity of the pixel $I_i(x,y)$ is (ideally) the transmitted light at location (x,y) when the ith light pattern ψ_i shone through the sample. The data is collected by using randomized Hadamard patterns in order to maximize the signal-to-noise ratio. The noise in the measurement of the hyperspectral image can be modeled as independent of the intensity of light shown through the sample. The signal-to-noise ratio of the measurement of each I_i, for a fixed integration time for the measurement of I_i, is maximized when the amount of modulated light shone through the sample is maximized. The instrument allows us to shine patterns $\{\psi_i\}_{i=1,\ldots,S} = \{\psi_i(\nu) = \sum_{j=1}^{N} \epsilon_{ij}\delta_j(\nu)\}$, where $(\epsilon)_{ij}$ is an S by N matrix with entries in $\{0,1\}$, and $(\delta)_j$, an S-dimensional vector, represents (ideally) a Dirac δ-function at physical frequency $\nu_j \sim (700-440)j/N+440$. In our experiment, we set $N = 256$ (the instrument would allow up to $N = 1024$) and $S = 128$. Ideally, $I_i(x,y)$, $i = 1,\ldots,S$, is the value of the inner product (in the frequency variable ν)

$$I_i(x,y) = \langle f(x,y,\nu), \psi_i(\nu)\rangle_\nu = \sum_j f(x,y,\nu_j)\psi_i(\nu_j)\,,$$

where $f(x,y,\nu)$ is the transmittance of the sample at location (x,y) and frequency ν. The choice of the patterns ψ_i is crucial in determining the signal-to-noise ratio of the measurements for a fixed integration time and total intensity of the light source: we use the idea of *multiplexing* and shine a sequence of *randomized Hadamard patterns* $\{\psi_i^H\}_{i=1,\ldots,N}$, obtained from standard Hadamard patterns by randomly shuffling the frequency axis. See Figure 5 for examples of Hadamard and randomized Hadamard patterns.

Thus, each data cube consists of 128 images, each 495 by 656 pixels in size (for a

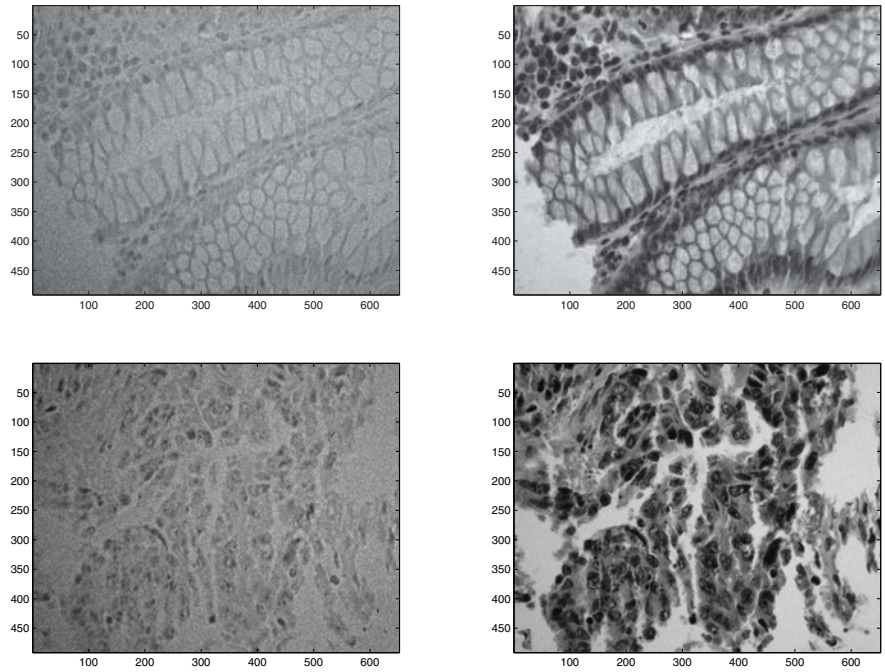

FIG. 6. *Two different spectral slices, i.e., two different images each at a single frequency, from a hyperspectral data cube derived from a normal sample (top) and from a hyperspectral data cube derived from a very malignant sample (bottom).*

total of about 40 million pixels), measuring the modulated light transmitted through the sample. We view this as a $495 \times 656 \times 128$ 3-mode tensor $\mathcal{A}$, where the entry $\mathcal{A}_{ijk}$ is proportional to the light with spectral modulation k transmitted at location (i, j). Each biopsy contains either normal, benign (adenoma), or malignant (cancerous) tissue and is labeled by G. L. Davis, M.D., pathologist. Various algorithms have previously been shown to find and classify automatically normal, abnormal, and malignant small portions of each biopsy [14, 15, 51] using the complete data cube. As we describe in more detail in the next three subsections, we couple the tensor-CUR decomposition described in section 3.1 with ideas from [14, 15, 51] in order to speed up computations, denoise, compress, and preprocess the data, and we show that this causes only a small loss of performance of these algorithms.

In order to gain a feel for the data, consider Figures 6 and 7. Figure 6 illustrates two of the 128 images, i.e., two hyperspectral images at two distinct frequencies, in a normal sample and in a very malignant sample. Similarly, Figure 7 illustrates a typical frequency-resolved pixel in both a normal and a malignant nucleus as well as a single spectrum in the malignant sample and the spectrum averaged over every one of the ca. 324,000 frequency-resolved pixels in the malignant data cube. Note that both successive images and pixels from different spatial regions are strongly correlated with one another.

In this imaging application, the tensor $\mathcal{C}$ in the tensor-CUR decomposition consists of a small number of dictionary or basis images (which are actual and not eigenimages) with respect to which the remaining images are expressed. Similarly, the

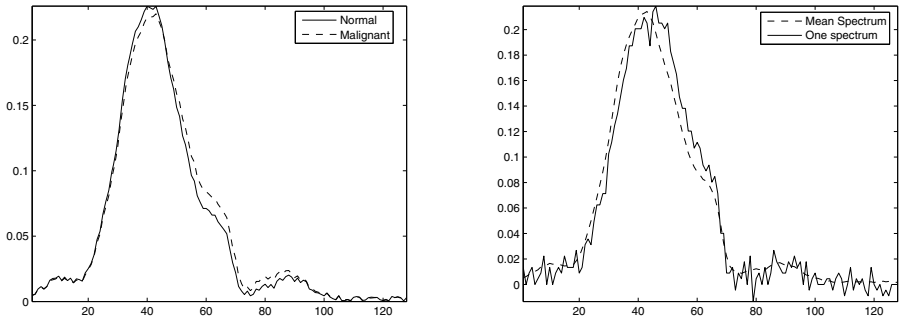

FIG. 7. *Left: Average normalized nuclei spectrum from a normal and a malignant sample. Right: Average normalized spectrum and a single typical spectrum in one hyperspectral data cube. The vertical axis represents normalized energy per frequency in the spectra, and the horizontal axis is the slab index.*

matrix $\mathcal{R}$ consists of the spectral variation of a small number of dictionary or basis pixels with respect to which spectral variation of the remaining pixels is expressed. In the next three 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. That is, the tensor-CUR algorithm will downsample slabs $\mathcal{A}(:,:,\nu_i)$ by sampling a set of images at certain randomly chosen wavelengths $\{\nu_i\}_{i=1}^{128}$ and fibers $\mathcal{A}(x_i, y_i, :)$ by sampling spectra at certain randomly chosen locations $\{(x_i, y_i)\}$. Slabs will be chosen randomly with a probability proportional to the average normalized spectrum of Figure 7, i.e., with probability proportional to $||\mathcal{A}(:,:,\nu)||_F$, 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.

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 the basis provided by the sampled slabs, and we do so using only a small number of pixels in that slab. In Figure 8, we present a representative example of the reconstruction of two spectral slices from a normal biopsy and two spectral slices from a malignant 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 9, we present the approximation error as a function of downsampling to different numbers of slabs and then to different numbers of fibers. As expected, as the number of sampled slabs and fibers increases, the approximation error decreases. The approximation error is very small in the middle range of the frequencies, where the energy per frequency is larger, and hence the sampling probability is larger. Thus, 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). Sampling more than 1200 fibers does not lead to significant

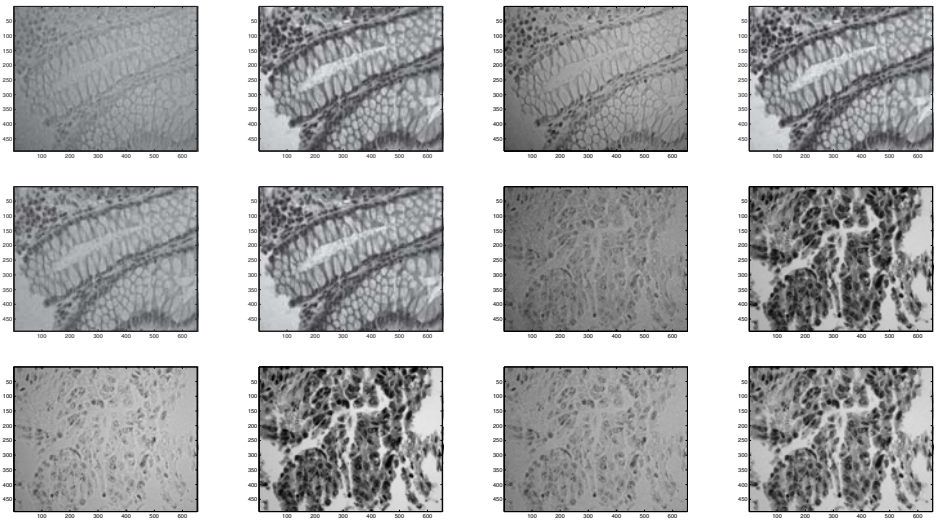

FIG. 8. *Typical reconstruction of the hyperspectral data cubes as a function of sampling. Shown in this figure are two different spectral slabs from a normal biopsy and two from a malignant biopsy, each reconstructed under three different compression ratios. In particular, the three figures in the first column are from slab number* 30 *(out of* 128*) from a normal sample; the second column is from slab number* 60 *from the same normal sample; and the third and fourth columns are slabs number* 30 *and* 60*, respectively, from another biopsy that is malignant. Presented are the original data (in the top row), the data when it is compressed with* 8 *slabs and* 1200 *fibers (in the middle row), and even more compressed data with only* 2 *slabs and* 1200 *fibers (in the bottom row).*

improvements (unless several tens of thousands fibers are sampled).

At this point, we observe that the spectra reconstructed after compression are far less noisy than the original spectra. More precisely, a close examination of images such as those presented in Figure 8 reveals a subtle interplay between sampling-induced error and denoising due to the low dimensionality of the sample. This has a denoising and a regularization effect on the spectra, and we can interpret the low-dimensional projection achieved by compression as a denoising mechanism, tuned to each data cube. Note that by giving our tensor-CUR algorithm the flexibility 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. Tissue-type segmentation. In medical applications, one is interested in the classification of an entire data cube, i.e., a medical sample, as normal or malignant. Biological reasons suggest that nuclei are the most discriminative structures for this task. Thus, as an intermediate step, one is interested in classifying the pixels in a single data cube into different tissue types, e.g., nuclei, cytoplasm, or lamina propria, based on the spectral response ("fiber") associated with each pixel. For each of the 59 images, we use the algorithm described in [14, 15, 51] for segmenting the pixels in the image into three sets of regions corresponding to different tissue types. This algorithm is based on the local discriminant basis (LDB) algorithm [13, 56, 57] to find features that best discriminate among the different classes and a nearest neighbor classifier in a discriminant projection found by LDB. Note that for the normal versus malignant classification task of the next subsection (in which we classify entire data cubes), we have access to a label (assumed correct) provided by a pathologist [51],

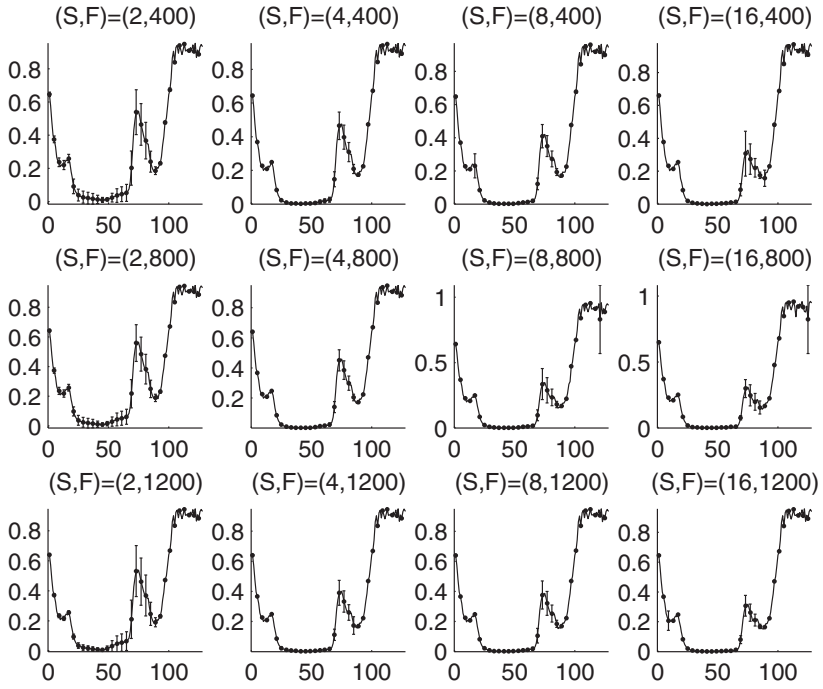

FIG. 9. *Reconstruction error. The caption indicates how many slabs (S) and fibers (F) were sampled. The vertical axis is the relative reconstruction error (for the Frobenius norm). The horizontal axis is the slab index. Average and standard deviation are over* 4 *slab draws and* 3 *fiber draws.*

while no such ground truth is available for this tissue classification in this section (in which we classify pixels in each image).

In Figure 10, we present typical results for running this tissue classification algorithm on two data cubes (one normal and one malignant) with increasing compression ratios. We see that the tissue classification is affected in two different ways. When we sample 16 slabs, the tissue classification, at least qualitatively speaking, improves by becoming less noisy and by generating fewer misclassification errors. See, e.g., the isolated red pixels, which correspond to nuclei, in the images in the leftmost column of Figure 10. As the compression ratio increases further, we observe a slight decreased performance in the tissue classification algorithm. As with the reconstruction problem, in both cases there is little quality loss until the number of fiber samples is less than ca. 1000. In addition, as before, a careful analysis reveals a complex interplay between sampling-induced information loss and sampling-induced denoising. Unfortunately, it is not possible for us to quantify these results, since this would require an individual to mark, by hand and with high precision, the correct tissue segmentation.

4.4. Classification of nuclei and data cubes. If the nuclei identified by the tissue classification described in section 4.3 are then used to classify data cubes, the results can be compared with the true value (assigned by the pathologist). For each nucleus, we consider the mean spectrum, and we use partial least squares (PLS) to build a linear classifier to classify this spectrum. We consider the following two

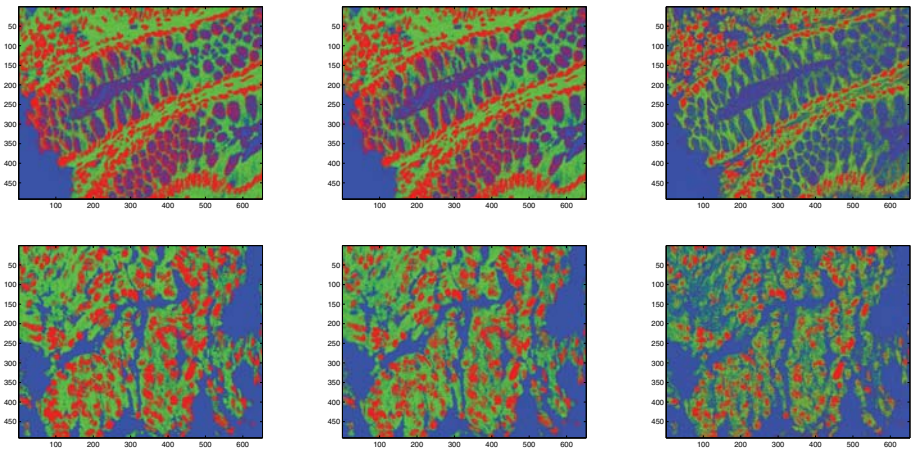

FIG. 10. *Segmentation into three tissue types in a normal biopsy (top row) and a malignant biopsy (bottom row): red for nuclei (the only class that we are interested in for the next classification task), green for cytoplasm, and blue for lamina propria and other regions. From left to right: classification on original data; on compressed data (*16 *slabs and* 1200 *fibers); and on compressed data (*8 *slabs and* 1200 *fibers).*

classification tasks: classify as normal or malignant; and classify as normal, abnormal, or malignant. In addition, we use two cross-validation procedures, described below, for each classification task. See [14, 15, 51] for more details on these procedures.

We define the patches we want to classify as follows. A patch is a subset of a data cube of the form $Q^l_{x_0,y_0} \times S$, where $Q^l_{x_0,y_0}$ is a square of side l pixels long, centered at (x_0, y_0), and S denotes the complete spectral range. A patch is *admissible* if it contains at least $\frac{8}{10}l^2$ nuclei pixels. From now on, we will consider each patch simply as a collection of the nuclei spectra it contains and hence as a cloud in $\mathbb{R}^{128}$. For the results reported here, we have chosen and fixed $l = 64$, which provides a size that roughly corresponds to the size of a single nucleus. The set of $l \times l$ patches we consider consists of 3298 patches chosen by the algorithm by randomly picking a square in the slide and checking if it is admissible. About 60 patches per slide are collected. We denote by $\{N_{i,k}\}_{k \in K_i}$ the set of nuclei spectra in the ith patch P_i.

For each admissible patch P_i collected, we compute the mean of the nuclei spectra $\{N_{i,k}\}_k$, and we normalize it to unit energy. We denote this set of normalized average nuclei spectra by $\mathcal{N}$. (Therefore, $|\mathcal{N}| = 3298$, as above.) The label (e.g., normal or abnormal) attached to the patch is transferred to the corresponding mean nucleus spectrum. We used PLS, keeping $k = 15$ top vectors, and we ran 50 rounds of 25-fold cross-validation to avoid overfitting. We run this cross-validation in two different ways:

- (Weak CV) Extract a random training subset of size $\frac{3}{4}|\mathcal{N}|$ and predict on the remaining subset of size $\frac{1}{4}|\mathcal{N}|$.
- (Strong CV) Extract a random subset of biopsies, of size $\frac{3}{4}$#biopsies, train the algorithm on the corresponding normalized average nuclei in $\mathcal{N}$ extracted from those biopsies, and test the algorithm on the remaining subset of $\mathcal{N}$, corresponding to averaged normalized nuclei extracted from the remaining biopsies.

Thus, in each case, the training and testing sets are subsets of biopsies. Note that the

TABLE 1

Confusion matrix of predictions of normal and malignant nuclei (patches of size 64 *by* 64 *with averaged* 25*-fold cross-validated error) using average (weak CV) error. TN, TM stand for true normal and true malignant, and PN, PM stand for the corresponding predictions. From left to right, the number of random slabs sampled is* $128(\text{all}), 16, 8, 2$.

	PN	PM
TN	90%	10%
TM	10%	90%

	PN	PM
TN	100%	0%
TM	0%	100%

	PN	PM
TN	100%	0%
TM	0%	100%

	PN	PM
TN	100%	0%
TM	0%	100%

TABLE 2

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei patches, as in Table 1*, but errors corresponding to (strong CV).*

	PN	PM
TN	79%	21%
TM	26%	74%

	PN	PM
TN	77%	23%
TM	30%	70%

	PN	PM
TN	79%	22%
TM	29%	71%

	PN	PM
TN	68%	32%
TM	33%	67%

first cross-validation is weaker. Since we expect correlations between (nuclei) spectra in the same data cube, and since in (weak CV) the training set contains, with high probability, nuclei spectra from all the biopsies, training and testing sets cannot be assumed to be completely independent. Most of this lack of independence, we think, is due to normalization issues, sample preparation, lighting, and other data collection conditions, which exhibit variations across biopsies. Since we can consider different biopsies as being independent samples as they were collected in random order and independently of the type (e.g., normal, abnormal, or malignant), the second cross-validation is stronger.

We are interested in measuring any change of performance of the classification algorithm as a function of the compression ratio. The confusion matrices of the classifiers obtained are summarized in Tables 1, 2, 3, and 4 for classifiers of patches of size $l = 64$. These confusion matrices are averages over the performance on the testing set in several cross-validation runs. For the full data, the two-class discrimination between normal and carcinoma nuclei correctly identifies 79% of normal and 74% of malignant nuclei. The three-class discrimination among normal, abnormal (adenoma), and carcinoma nuclei is much more challenging (independently of compression), with identification rates of 33%, 73%, and 40% for normal, abnormal, and carcinoma samples, respectively. We study how this performance changes under compression of the data cubes. As can be seen, in general, high quality results are obtained using samples of 16 and 8 slabs, but quality degrades if only 2 slabs are used. Also, note that the algorithm performs more poorly (about 25% error in the discrimination of the 3 classes of biopsies) on completely new biopsies. This is related to normalization of the data, due both to the process of staining and to the instrument calibration and data collection. Current research is addressing these issues.

In Tables 1 and 2, we classify normal and malignant, and then we run the same classifier on data cubes compressed at different compression ratios; we also show the difference between weak and strong cross-validation. Observe that the performance of the algorithm is very good across compression ratios, except for a significant decrease of performance for a very high compression ratio (sampling of only 2 slabs!). We interpret this as a balancing effect between the possible loss of information due to compression and the denoising and regularization effect due to the dimensionality reduction.

In Tables 3 and 4, we classify normal, abnormal, and malignant, and again we

TABLE 3

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei (patches of size 64 *by* 64 *with averaged* 25*-fold cross-validated error) using average (weak CV) error. TN, TB, TM stand for true normal, true benign (adenoma), and true malignant, and PN, PB, PM stand for the corresponding predictions. From left to right, the number of random slabs sampled is* $128(\text{all}), 16, 8, 2$.

	PN	PB	PM
TN	45%	51%	4%
TB	17%	76%	7%
TM	4%	36%	60%

	PN	PB	PM
TN	96%	4%	0%
TB	1%	98%	0%
TM	0%	3%	97%

	PN	PB	PM
TN	99%	1%	0%
TB	0%	100%	0%
TM	0%	1%	99%

	PN	PB	PM
TN	100%	0%	0%
TB	0%	100%	0%
TM	0%	0%	100%

TABLE 4

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei patches, as in Table 3*, but with (strong CV).*

	PN	PB	PM
TN	33%	61%	6%
TB	22%	73%	5%
TM	9%	51%	40%

	PN	PB	PM
TN	42%	24%	34%
TB	31%	36%	33%
TM	23%	28%	49%

	PN	PB	PM
TN	30%	53%	17%
TB	26%	61%	13%
TM	7%	48%	51%

	PN	PB	PM
TN	30%	45%	25%
TB	29%	53%	16%
TM	12%	35%	53%

run the same classifier on data cubes compressed at different compression ratios; we also show the difference between weak and strong cross-validation. Here we observe a interesting phenomenon: under (weak CV), the algorithm performs much more poorly on the original data than on the compressed data. Hence the compression has a regularization effect that greatly helps the learning phase. This advantage is partly lost when we consider the (strong CV). Of course, the three-class problem is expected to be much harder than the two-class problem, not only because, from a machine-learning perspective, multiclass problems are harder but also because the abnormal samples are often quite similar to normal samples, and even in the field of pathology, the differences are qualitative and often not large.

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" [49]. 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 users who are

most similar to the user after querying the new user about his (or her) rating on a small number of products. For more details, see [54, 10, 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 [20]. 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 [20]. Prior theoretical work on recommendation systems includes Kumar et al. [42], who offer competitive algorithms even with only two samples/customer, assuming a strong clustering of the products; Azar et al. [4], who use spectral methods to recreate very accurately the user-product ratings matrix A, assuming a certain gap requirement and a sample of $\Omega(mn)$ entries of A; Kleinberg and Sandler [37], who develop recommendation algorithms with provable performance guarantees in a probabilistic mixture mode; and (most relevant for our work) Drineas, Kerenidis, and Raghavan [20], who obtain competitive performance by sampling $O(m + n)$ entries of the user-product ratings matrix and reconstructing missing entries with a matrix CUR decomposition. Other applications of linear algebra have used the SVD for dimensionality reduction [9, 58, 26].

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 because utility functions are constructs that encode preference information and because 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 [12, 24, 35, 34] and have been motivated by such observations as that two users with very similar preferences for 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 `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 [26]. 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. We now describe our results for the tensor-CUR decomposition when applied to the Jester dataset in the context of recommendation systems. Let c be an integer between 1 and 14,116 (recall that this is the total number of users that fully rated all 100 jokes in the Jester data), and assume that we sample uniformly at random c of the 14,116 users. For each sampled user, we assume that the corresponding 100×100 slab of the $100 \times 100 \times 14{,}116$ tensor representing the Jester dataset (see the previous section for details) is fully known or, in other words, that we know all pairwise product-product (i.e., joke-joke) comparisons for the c sampled users.

Consider the $14{,}116 - c$ slabs (i.e., users) that we did not sample. For each such *target* slab (i.e., *target* user), we use the tensor-CUR decomposition to reconstruct it as a linear combination of the c sampled slabs. Thus, it suffices to compute c coefficients such that a linear combination of the basis slabs using these coefficients achieves a satisfactory reconstruction of the target slab. However, in order to do such a reconstruction, we need some information from the target slab. This information consists of a small number of product-product preference queries sampled uniformly at random from the target slab. These elements of the target slab will allow us to approximately infer the coefficients to be used in expressing the target slab as a linear combination of the c basis slabs. Once the target slab (i.e., preference matrix) is reconstructed, we can use this reconstruction to make recommendations by picking the N products with the largest row sums. In our model, where the (i, j)th entry of the preference matrix is set to 1 if product i is preferred over product j and to -1 otherwise, such rows correspond to the most desirable products for this user.

To formally evaluate the quality of our recommender system, we use the well-known top-N procedure and compute the precision, recall, and the $F1$ statistic [58]. More specifically, let $\mathcal{T}_N$ be the actual set of the top N products for a certain user, and let $\mathcal{S}_K$ be a set of K products that are *suggested* to this user by a recommender system. Clearly, K can be equal to or larger than N, whereas values of K that are smaller than N are typically not interesting. For some combinations of N and K, we shall measure the following four quantities.

Successful recommendations. The number of elements in the intersection of $\mathcal{T}_N$ and $\mathcal{S}_K$ or, in other words, the number of products that are in the top-N preferred products for a particular user *and* were recommended by an algorithm that made K suggestions.

Recall. The number of successful recommendations divided by the number of suggestions (K) made by the algorithm. This quantity normalizes the number of successful recommendations to take into account the fact that increasing the number of suggestions increases the number of top-N products recommended by the algorithm.

Precision. The number of successful recommendations divided by N. Remember that N essentially determines the number of products that a user is interested in, and hence this quantity normalizes the number of successful recommendations to take into account the fact that increasing N increases the number of top-N products recommended by the algorithm.

F1 statistic. The formal definition is

$$\text{F1 statistic} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

and it is commonly used to reconcile the mutually conflicting nature of the precision and recall statistics. (Notice, for example, that increasing N tends to increase recall but decreases precision [58].)

Prior to presenting the results of our experimental evaluation, we briefly discuss our choices for the four parameters involved in our experiments. First, recall that c denotes the number of basis users that reveal all their pairwise product preferences to the algorithm; we let c be all powers of 2 between 2 and 1024. This choice provides a clear picture of the behavior of tensor-CUR for very small (e.g., $c \leq 32$), medium-sized (e.g., $64 \leq c \leq 256$), and large (e.g., $c = 512, 1024$) basis sets. Second, N is set to be either 5 or 10, implying the algorithm is successful if it recommends one of the top 5 or top 10 products for a certain user. Third, K is set to be equal to N or $2N$, and hence the algorithm is allowed to suggest either 5 or 10 products for the top-5 case and either 10 or 20 products for the top-10 case. Fourth, the number of fibers that the tensor-CUR algorithm samples or, in other words, the number of product-product pairwise comparisons of a target user that are revealed to the algorithm is again set to all powers of 2 between 2 and 1024; the rationale is the same as above. In the first experiment, we will set the number of fibers to $100^2 = 10{,}000$ (all available fibers), in order to illustrate the limiting behavior of tensor-CUR. We emphasize that both the sampling of slabs and the sampling of fibers are done uniformly at random without replacement, and hence sampling 10,000 fibers is equivalent to picking all the fibers.

In our first experiment, we seek to determine an upper bound on the quality of recommendations based on using a small number of basis slabs and all fibers for the remaining users. Clearly, this experiment seeks only to characterize the limiting behavior of tensor-CUR, since having all fibers trivially allows perfect recommendations. Figures 11 and 12 illustrate that almost all users can be very accurately expressed as a linear combination of a small number of basis users, chosen uniformly at random without replacement. In later experiments, given this observation, we will attempt to approximate the coefficients of this linear combination using a small number of fibers.

Figure 11 shows the results for $N = 5$. Notice that using 512 or 1024 slabs and only 5 suggestions results in 4 or more successful recommendations; if the algorithm is allowed to make 10 suggestions, 64 or more slabs are enough to make roughly four successful recommendations. (As a trivial but weak lower bound on quality, by making five suggestions uniformly at random, we expect that we will make ca. .5 predictions correctly, since we are making 5 predictions and there are 100 products.) Notice that the $F1$ statistic shows a change of phase as the number of slabs increases above 256: making more than 5 suggestions is not necessary anymore, since the number of basis slabs suffices to accurately capture the high-ranking products. Given less than 256 basis slabs (e.g., 128 slabs), our results suggest that making 10 suggestions is qualitatively better. The same conclusions essentially apply to Figure 12 as well, which shows the results for $N = 10$. However, we should emphasize that the effect of making 20 versus 10 suggestions, as measured by the $F1$ statistic, is much less obvious in this case. Notice that making 20 suggestions does not result in a significant advantage even for a small number of basis slabs and is clearly worse as the number of basis slabs increases above 128.

In our second experiment, we show that by using a basis of preference information from (say) 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 both for the top-5 and top-10 cases; see Figures 13 and 14, respectively. 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

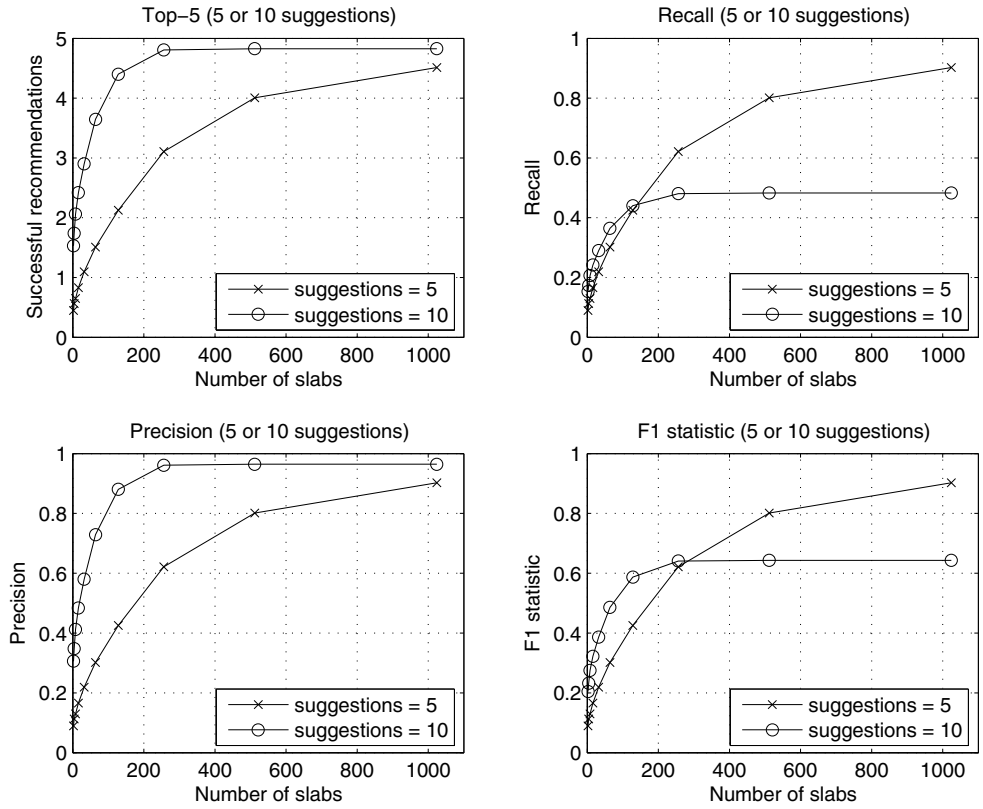

FIG. 11. *Effect of user basis size on top-5 recommendation quality using complete pairwise product-product preference information. The basis users are sampled uniformly at random without replacement.*

of fibers. If the algorithm is allowed to make 10 suggestions, the statistics for top-5 recommendations remain competitive with the upper bounds suggested in Figure 11. However, if the algorithm is allowed only 5 suggestions, the results are markedly worse, especially given a small number of pairwise product-product comparisons. Naturally, the $F1$ statistic illustrates that suggesting 10 products is now always preferable to suggesting 5 products. This observation changes when we evaluate the algorithm on top-10 recommendations, where the $F1$ statistic shows that suggesting 10 or 20 products is essentially the same, and thus suggesting 10 products is the right course of action. Notice that even though the performance of the algorithm is worse than the optimal one of Figure 12, it is clearly well above the random level. We would also like to note the nonmonotonicity near ca. 64 queries; this seems to be a fitting issue. Figures 15 and 16 show the results for top-10 recommendations when the number of basis users is set to 64 and 256, respectively. The results are qualitatively similar, but it is worth noticing that the algorithm making 10 suggestions outperforms the algorithm making 20 suggestions given 256 basis slabs and more than 256 fibers. The results for top-5 recommendations using 64 and 256 basis users are omitted, since they are qualitatively the same as in Figure 11.

In our third and final experiment, we present the distribution of correct top-10 predictions for the 14,116 users by using 64 or 128 basis users and a variable number

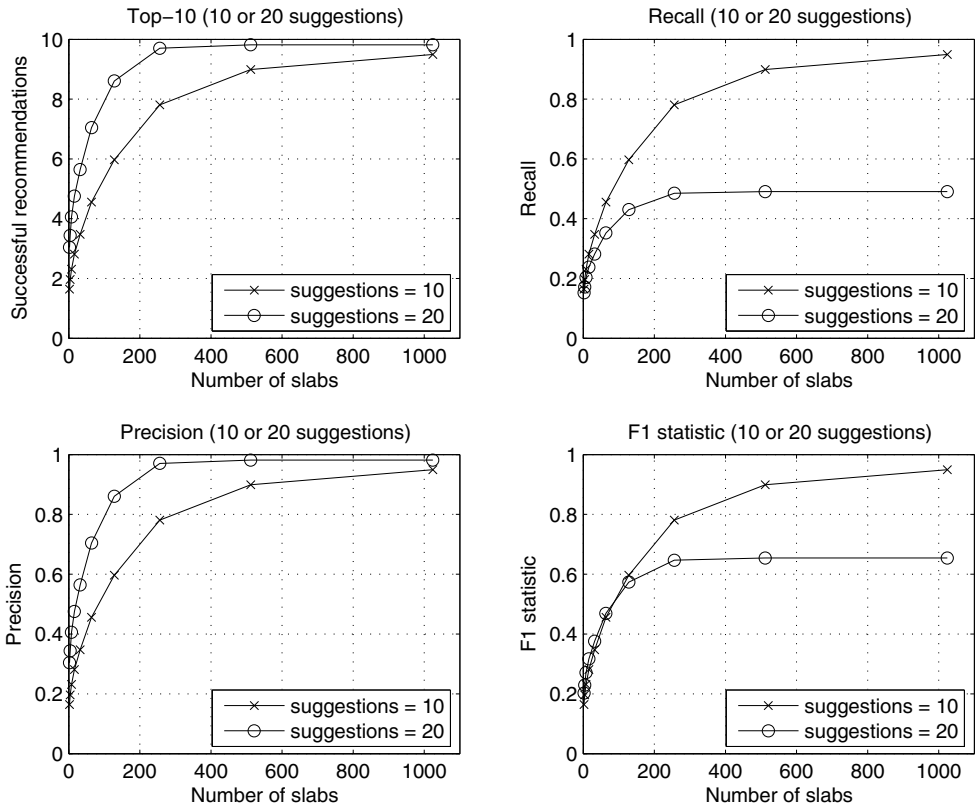

FIG. 12. *Effect of user basis size on top-*10 *recommendation quality using complete pairwise product-product preference information. The basis users are sampled uniformly at random without replacement.*

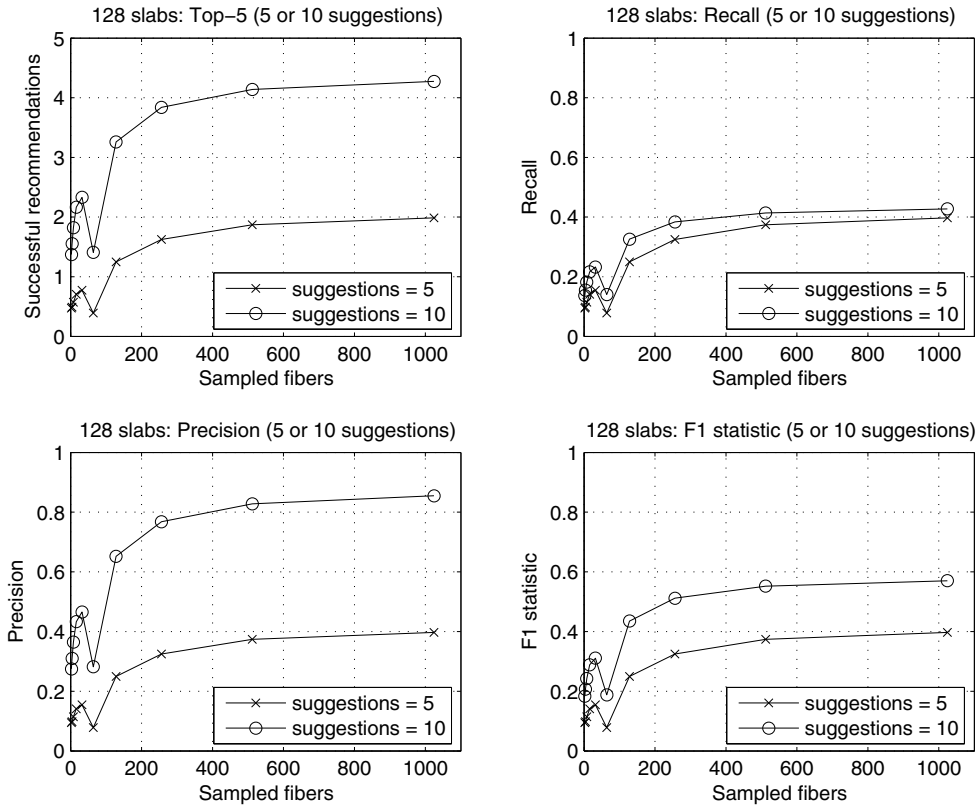

FIG. 13. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-5 recommendation quality given 128 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

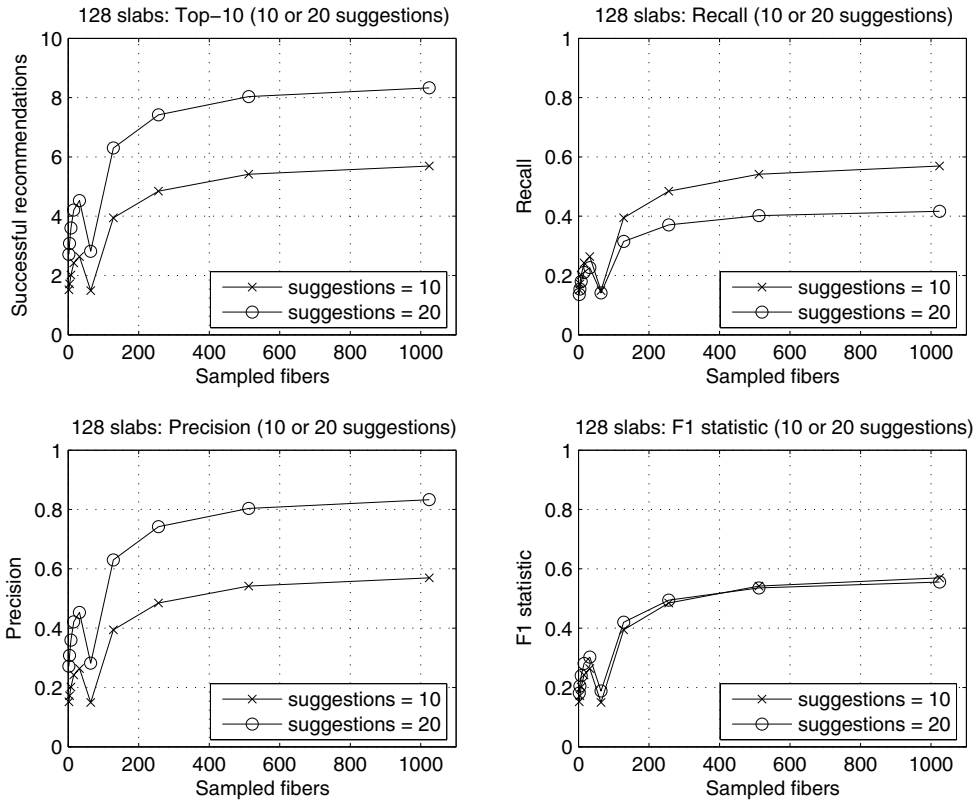

FIG. 14. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-*10 *recommendation quality given* 128 *basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

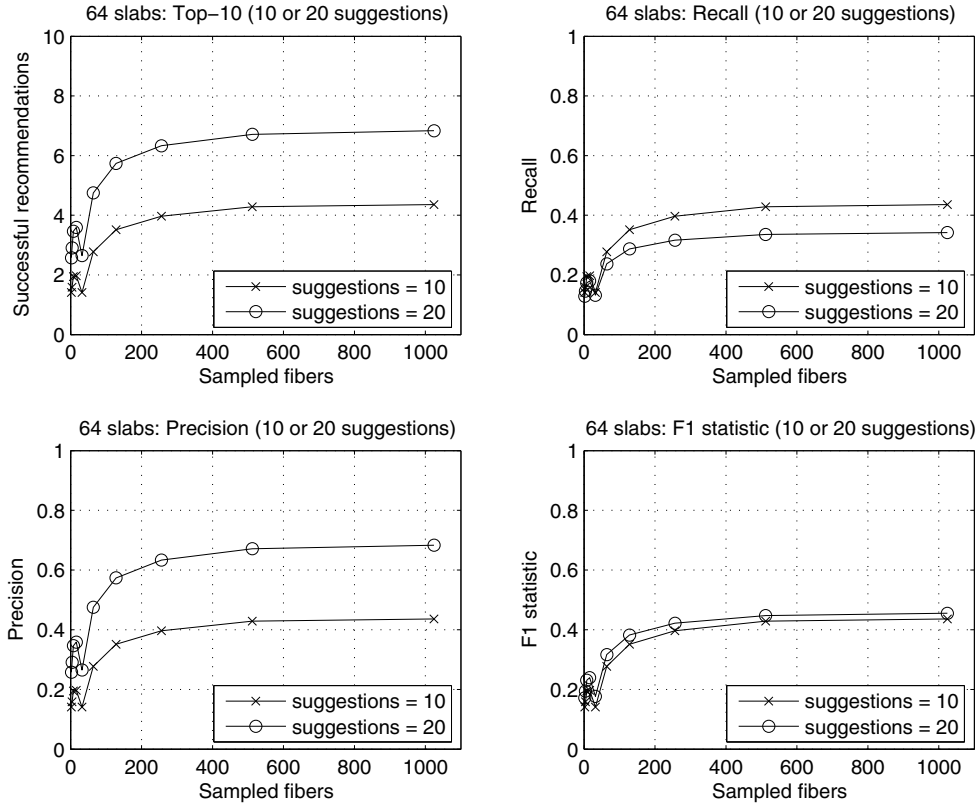

FIG. 15. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-10 recommendation quality given 64 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

of pairwise product-product comparisons; see Figure 17. Clearly, as the number of basis slabs or sampled fibers increases, the curves are shifted to the right, illustrating that a larger number of users receives more accurate recommendations. In this case, we plot results for the algorithm making 10 suggestions. Similar results are seen in all other cases.

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 have developed a tensor-based extension of the matrix CUR decomposition. This tensor-CUR decomposition is of most interest when the data may be modeled by a variable subscripted by three or more indices and when one of those indices/modes is qualitatively different from 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 addition, we have applied the tensor-CUR decomposition to problems in two diverse domains of

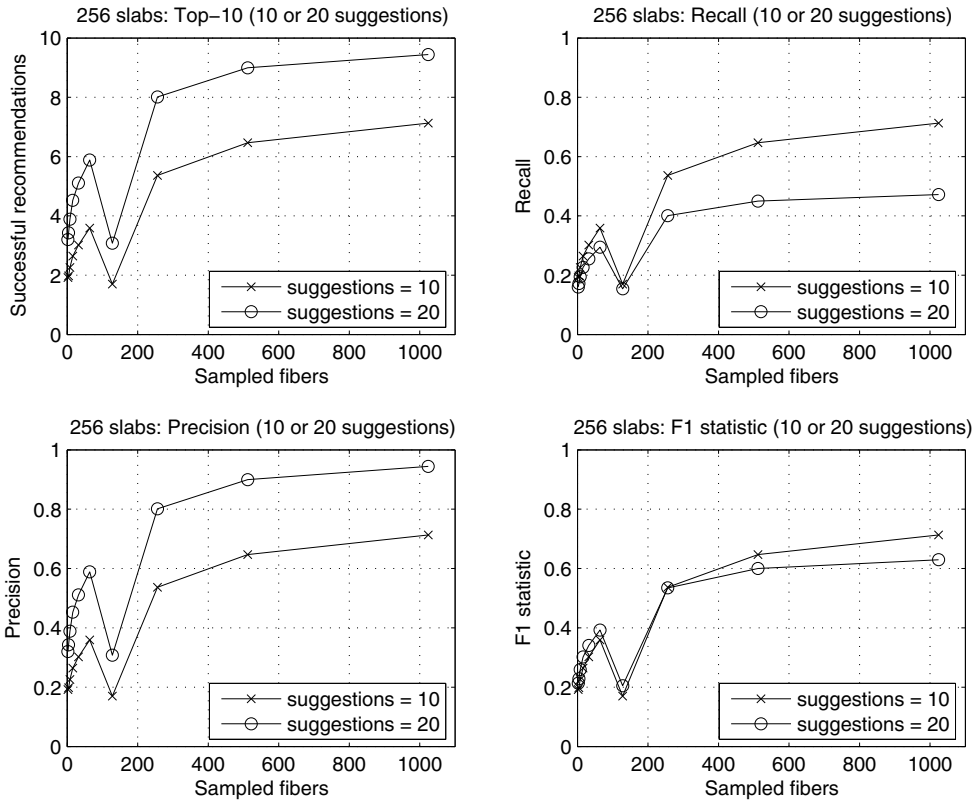

FIG. 16. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-10 recommendation quality given 256 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

data analysis: hyperspectral medical image analysis and consumer recommendation system analysis.

Similarities and differences between the methods discussed in this paper and the image analysis techniques known as "eigenfaces" and "tensor-faces" should be mentioned. The method of eigenfaces computes the eigenvectors of the covariance matrix of a large number of images of faces [62]. Eigenanalysis (and, more generally, SVD analysis) successively computes axes of maximum variation in the data, conditioned on being orthogonal to previously computed axes. Since this orthogonality is not present in natural images of faces, its imposition results in the characteristic "ringing" oscillations generated by eigenanalysis of facial images that in turn leads to difficulty interpreting the eigenfaces after the first few. The methods of the present paper are applicable to a set of time-resolved or frequency-resolved images of a single object. One could apply SVD-type analysis for data compression, i.e., to reduce the dimensionality along the slabs and/or the fibers. On the other hand, it will likely be difficult to interpret the principal components. Our tensor-CUR algorithms provide approximate low-rank tensor decompositions in terms of actual data elements. If orthogonality is not present in the data, e.g., if there are different fibers and/or pixels, then the tensor-CUR decompositions will be in terms of nonorthogonal data elements. Partly in response to ringing artifacts of eigenface analysis, a tensor-based

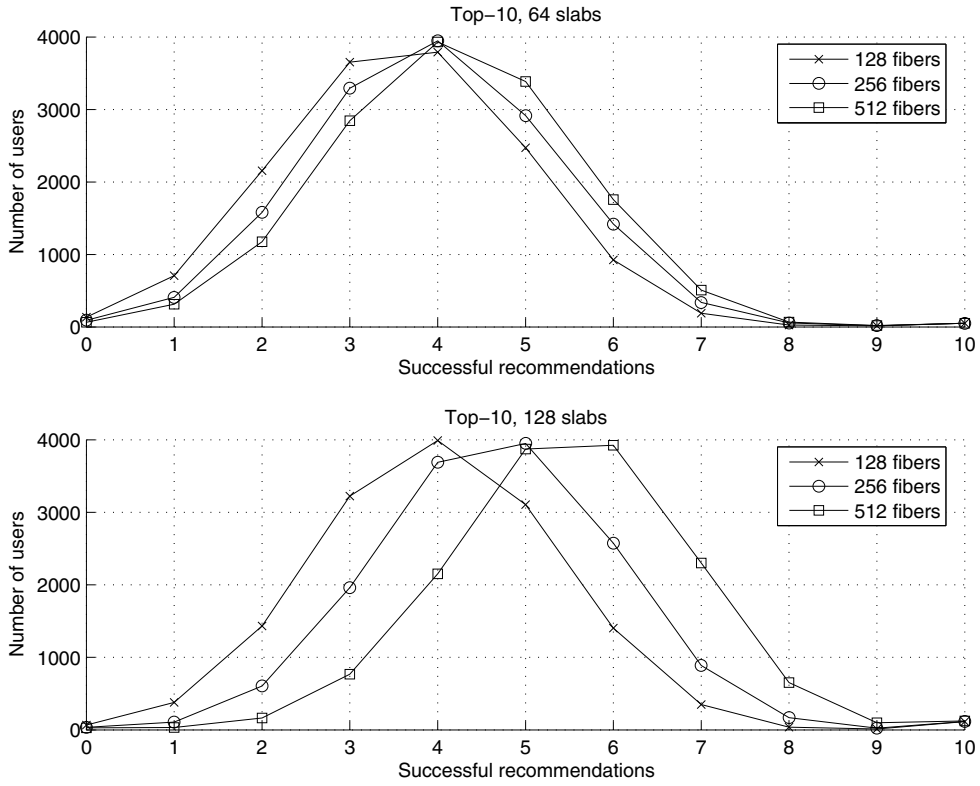

FIG. 17. *Distribution of number of users getting a given number of successful top-*10 *recommendations for a basis consisting of* 64 *or* 128 *users for different numbers of sampled fibers. Both the basis slabs and the fibers are sampled uniformly at random without replacement.*

analysis of facial images has been introduced [65]. This analysis involves applying a tensor-based generalization of the SVD to a user-defined set of features derived from a set of images of faces. A randomized variant of this generalization has been presented and analyzed in [23]. This randomized tensor-SVD algorithm bears some similarity to the randomized tensor-CUR algorithms described in this paper. It differs, however, in that there is no preferred mode; instead, the tensor is "unfolded" along every mode, and a projection along each mode is constructed by sampling columns along that mode.

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, or by compressing each individual slab with more traditional methods. 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. Fourth, it

would be worth understanding in greater detail the relationship between the methods we have presented for analyzing tensor data and the well-studied model proposed by Tucker, the "canonical decomposition" model, the "parallel factors" model, and the higher-order SVD model; due to lack of space, a comparison with these models has been omitted. Finally, cross-approximation techniques are powerful and well-developed adaptive methods for low-rank approximation of matrices [6, 63]; it is worth understanding in greater detail the relationship between these methods and matrix CUR decompositions.

Acknowledgment. We thank the authors of [51], in particular Gustave L. Davis of Yale University, for making available the hyperspectral data.

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 Trans. Knowl. Data Eng., 17 (2005), pp. 734–749.

[2] C. Angeletti, N. R. Harvey, V. Khomitch, R. Levenson, and D. L. Rimm, *Detection of malignant cells in cytology specimen using genie hybrid genetic algorithm*, Mod. Pathol., 17 (2004), Suppl 1:350A.

[3] C. Angeletti, R. Jaganth, R. M. Levenson, and D. L. Rimm, *Spectral analysis: A novel method for classification of urine cytology*. Mod. Pathol., 16 (2003), 57A.

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

[5] T. S. Barry, A. M. Gown, H. E. Yaziji, and R. W. Levenson, *Use of spectral imaging analysis for evaluation of multi-color immuno-histochemistry*, Mod. Pathol., 17 (2004), Suppl 1:350A.

[6] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.

[7] 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.

[8] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251.

[9] D. Billsus and M. J. Pazzani, *Learning collaborative information filters*, in Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufman, San Francisco, 1998, pp. 46–54.

[10] 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, Morgan Kaufman, San Francisco, 1998, pp. 43–52.

[11] 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 (1970), pp. 283–319.

[12] 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, 1998, pp. 451–457.

[13] R. R. Coifman, *Multiresolution analysis in non-homogeneous media*, in Wavelets. Time-Frequency Methods and Phase Space, J.-M. Combes, A. Grossmann, and P. Tchamitchian, eds., Springer-Verlag, Berlin, 1989, p. 259.

[14] G. L. Davis, M. Maggioni, R. R. Coifman, D. L. Rimm, and R. M. Levenson, *Spectral/spatial analysis of colon carcinoma*, Mod. Pathol., 16 (2003), 3320:3321A.

[15] G. L. Davis, M. Maggioni, F. J. Warner, F. B. Geshwind, A. C. Coppi, R. A. DeVerse, and R. R. Coifman, *Spectral analysis of normal and malignant microarray tissue sections using a novel micro-optoelectricalmechanical system*, Mod. Pathol., 17 (2004), 1:358A.

[16] R. A. DeVerse, R. R. Coifman, A. C. Coppi, W. G. Fateley, F. Geshwind, R. M. Hammaker, S. Valenti, F. J. Warner, and G. L. Davis, *Application of spatial light modulators for new modalities in spectrometry and imaging*, in Proceedings of the SPIE 4959, 2003, pp. 12–22.

[17] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* I: *Approximating matrix multiplication*, SIAM J. Comput., 36 (2006), pp. 132–157.
[18] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* II: *Computing a low-rank approximation to a matrix*, SIAM J. Comput., 36 (2006), pp. 158–183.
[19] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* III: *Computing a compressed approximate matrix decomposition*, SIAM J. Comput., 36 (2006), pp. 184–206.
[20] P. Drineas, I. Kerenidis, and P. Raghavan, *Competitive recommendation systems*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 82–90.
[21] P. Drineas and M. W. Mahoney, *Approximating a Gram matrix for improved kernel-based learning*, in Proceedings of the 18th Annual Conference on Learning Theory, Springer-Verlag, Berlin, 2005, pp. 323–337.
[22] P. Drineas and M. W. Mahoney, *On the Nyström method for approximating a Gram matrix for improved kernel-based learning*, J. Mach. Learn. Res., 6 (2005), pp. 2153–2175.
[23] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the singular value decomposition*, Linear Algebra Appl., 420 (2007), pp. 553–571.
[24] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, *An efficient boosting algorithm for combining preferences*, J. Mach. Learn. Res., 4 (2003), pp. 933–969.
[25] Y. Garini, N. Katzir, D. Cabib, R. A. Buckwald, D.G. Soenksen, and Z. Malik, *Spectral bio-imaging*, in Fluorescence Imaging Spectroscopy and Microscopy, John Wiley and Sons, New York, 1996, pp. 87–124.
[26] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, *Eigentaste: A constant time collaborative filtering algorithm*, Inform. Retrieval, 4 (2001), pp. 133–151.
[27] T. Gonzalez and J. Ja'Ja', *On the complexity of computing bilinear forms with* $\{0, 1\}$ *constants*, J. Comput. System Sci., 20 (1980), pp. 77–95.
[28] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximum-volume concept in approximation by low-rank matrices*, in Structured Matrices in Mathematics, Computer Science, and Engineering, I., Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 47–51.
[29] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
[30] W. H. Greub, *Multilinear Algebra*, Springer-Verlag, Berlin, 1967.
[31] R. A. Harshman and M. E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H. G. Law, C. W. Snyder, Jr., J. Hattie, and R. P. McDonald, eds., Praeger, New York, 1984, pp. 122–215.
[32] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.
[33] T. D. Howell, *Global properties of tensor rank*, Linear Algebra Appl., 22 (1978), pp. 9–23.
[34] 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, Morgan Kaufman, San Francisco, 2003, pp. 329–336.
[35] R. Jin, L. Si, C. X. Zhai, and J. Callan, *Collaborative filtering with decoupled models for preferences and ratings*, in Proceedings of the 12th ACM International Conference on Information and Knowledge Management, 2003, pp. 309–316.
[36] H. A. L. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[37] J. Kleinberg and M. Sandler, *Using mixture models for collaborative filtering*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 569–578.
[38] T. G. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.
[39] P. M. Kroonenberg and J. De Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.
[40] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[41] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and* N*-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North–Holland, Amsterdam, 1989, pp. 7–18.
[42] 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, 1998, pp. 664–673.

[43] L. De Lathauwer, B. De Moor, and J. Vandewalle, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149.
[44] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[45] 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 J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[46] 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 Appl., 269 (1998), pp. 307–329.
[47] R. M. Levenson and D. Farkas, *Digital spectral imaging for histopathology and cytopathology*, in Proceedings of the SPIE 2983, 1997, pp. 123–135.
[48] 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, 2005.
[49] G. Linden, B. Smith, and J. York, *Amazon.com recommendations: Item-to-item collaborative filtering*, IEEE Internet Comput., 7 (2003), pp. 76–80.
[50] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.
[51] 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*, Technical report YALEU/DCS/TR-1311, Yale University Department of Computer Science, New Haven, CT, 2004.
[52] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, in Proceedings of the 12th Annual ACM SIGKDD Conference, 2006, pp. 327–336.
[53] A. Papadakis, E. Stathopoulos, G. Delides, K. Berberides, G. Nikiforides, and C. Balas, *A novel spectral microscope system: Application in quantitative pathology*, IEEE Trans. Biomed. Eng., 50 (2003), pp. 207–217.
[54] P. Resnick and H. R. Varian, *Recommender systems*, Comm. ACM, 40 (1997), pp. 56–58.
[55] C. Rothman, I. Bar-Am, and Z. Malik, *Spectral imaging for quantitative histology and cytogenetics*, Histology and Histopathology, 13 (1998), pp. 921–926.
[56] N. Saito and R. R. Coifman, *Local discriminant bases and their applications*, J. Math. Imaging Vision, 5 (1995), pp. 337–358.
[57] N. Saito, R. R. Coifman, F. B. Geshwind, and F. Warner, *Discriminant feature extraction using empirical probability density estimation and a local basis library*, Pattern Recognition, 35 (2002), pp. 2841–2852, 2002.
[58] 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, 2000.
[59] G. W. Stewart, *Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix*, Numer. Math., 83 (1999), pp. 313–323.
[60] G. W. Stewart, *Error analysis of the quasi-Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 493–506.
[61] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[62] M. Turk and A. Pentland, *Eigenfaces for recognition*, J. Cogn. Neurosci., 3 (1991), pp. 71–96.
[63] E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic-skeleton method*, Computing, 64 (2000), pp. 367–380.
[64] S. G. Vari, G. Muller, J. M. Lerner, and R. D. Naber, *Telepathology and imaging spectroscopy as a new modality in histopathology*, Stud. Health Technol. Inform., 68 (1999), pp. 211–216.
[65] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear analysis of image ensembles: TensorFaces*, in Proceedings of the 7th European Conference on Computer Vision, Springer-Verlag, Berlin, 2002, pp. 447–460.
[66] 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, MIT Press, Cambridge, MA, 2001, pp. 682–688.
[67] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.