From *SIAM News*, Volume 39, Number 8, October 2006

Bridging the Gap Between Numerical Linear Algebra, Theoretical Computer Science, and Data Applications

By Gene H. Golub, Michael W. Mahoney, Petros Drineas, and Lek-Heng Lim

The Workshop on Algorithms for Modern Massive Data Sets (MMDS 2006), sponsored by the National Science Foundation, Yahoo! Research, and Ask.com, was held at Stanford University, June 21–24. The objectives were to explore novel techniques for modeling and analyzing massive, high-dimensional, and nonlinearly structured data sets, and to bring together computer scientists, computational and applied mathematicians, statisticians, and practitioners to promote cross-fertilization of ideas. The program, with 45 talks and 24 poster presentations, drew 232 participants—far exceeding the anticipated 75!

MMDS 2006 grew out of discussions among the four organizers (who are also the authors of this article) about the complementary perspectives brought by the numerical linear algebra (NLA) and the theoretical computer science (TCS) communities to linear algebra and matrix computations. These discussions were motivated by data applications, and, in particular, by technological developments over the last two decades (in both scientific and Internet domains) that permit the automatic generation of very large data sets. Such data are often modeled as matrices: An $m \times n$ real-valued matrix A provides a natural structure for encoding information about m objects, each of which is described by n features. In genetics, for example, microarray expression data can be represented in such a framework, with A_{ij} representing the expression level of the ith gene in the jth experimental condition. Similarly, term–document matrices can be constructed in many Internet applications, with A_{ij} indicating the frequency of the jth term in the ith document. Such data matrices often have structural properties that present challenges and opportunities for researchers in both NLA and TCS.

The workshop was loosely organized around six hour-long tutorials that introduced participants to the major themes of the workshop.

Linear Algebra and Matrix Computations: Complementary Perspectives

Representing the NLA perspective, Di-anne O'Leary of the University of Mary-land gave a tutorial titled "Matrix Factor-izations for Information Retrieval." Historically, matrix factorizations have been of central interest in NLA because they can be used to express a problem in such a way that it can be solved more easily. They have been of interest in statistical data analysis because they can be used to represent structure that may be present in a matrix obtained from object–feature observations. O'Leary began by describing the eigendecomposition of a matrix, which has been used for both purposes. She also described the rank-revealing QR decomposition, the rank-revealing URV decomposition, the semidiscrete decomposition, and nonnegative matrix factorizations, all of which have found application in redundancy reduction, in latent semantic indexing, and elsewhere.

Of course, O'Leary didn't fail to discuss the singular value decomposition, describing it as both the "Swiss Army Knife" and the "Rolls Royce" of matrix decompositions. The SVD decomposes any matrix A as $A = U\Sigma V^T$, where U and V are orthogonal matrices and Σ is a diagonal matrix with nonnegative and nonincreasing elements along the diagonal. The SVD can be substantially more expensive to compute than other commonly used decompositions, but it provides a great deal of useful information about the matrix. It is commonly used in data analysis, for example, via the related method of principal components analysis, because truncating the SVD by keeping just the k largest terms on the diagonal of Σ provides the "best" rank-k approximation to A. Thus, the SVD, its computation, its approximation, its application, and its interpretation were recurrent themes throughout the workshop.

MMDS 2006 organizers (left to right) Lek-Heng Lim, Petros Drineas, Michael Mahoney, and Gene Golub discuss plans for the workshop.

Ravi Kannan of Yale University represented the very different TCS perspective on linear

algebra and matrix computations in the tutorial "Sampling in Large Matrices." Kannan began with a question: How does one pick a *good* random sample of rows of a matrix A *quickly*? Traditionally, *good* might mean that the rows of R are independent and span the row space of A, or that the row span of R satisfies an interpolative approximation condition with respect to the rows of A; similarly, *quickly* might mean in polynomial time in the worst case.

In light of large data applications, Kannan argued that if one chooses (and rescales appropriately) r rows from an $m \times n$ matrix A to construct an $r \times n$ matrix R, then a simple and reasonable notion of *good* is that $A^TA \approx R^TR$. He further argued that, for extremely large matrices, a reasonable notion of *quickly* is that one can perform the computations in only one or two "passes" over the data (assumed to be stored externally), with only $O(m + n)$ additional space and time. Within this framework, Kannan showed that nontrivial results can be obtained by randomly sampling rows according to a probability distribution proportional to the square of the Euclidean norm of that row. For example, the entire matrix can be approximated quickly with a CUR matrix decomposition: R is chosen first, followed by the construction (in an analogous manner) of an $m \times c$ matrix C consisting of c columns of A and, finally, construction of a $c \times r$ matrix U such that $A \approx CUR$.

In their tutorials, Kannan and O'Leary highlighted complementary perspectives. For example, in computing matrix decompositions like the SVD or QR decomposition, NLA researchers place an emphasis on optimal conditioning, backward error analysis issues, and whether the algorithm takes time that is a large or small constant multiplied by $\min\{mn^2, m^2n\}$. In contrast, a precise statement of the manner in which $A \approx CUR$ involves no reference to conditioning or numerical stability. In addition, this CUR decomposition is constructed by exploiting oversampling and randomness as computational resources. These resources are common in TCS, but they are typically not exploited in NLA.

In another pair of tutorials, S. Muthukrishnan of Rutgers University and Google ("An Algorithmicist's Look at Compressed Sensing Problems") and Dimitris Achlioptas of the University of California at Santa Cruz ("Applications of Random Matrices in Spectral Computations and Machine Learning") examined the relation between vector space operations that people want to perform and what can be computed efficiently.

Muthukrishnan began by describing the sparse approximation problem: Given a dictionary A, i.e., a set of n-dimensional vectors that span $\mathbb{R}^n$, and an n-dimensional signal vector S, a *b-sparse representation* for S is a vector of the form $R_b = \sum' S_i A_{(i)}$, where S_i is the ith element of the signal, $A_{(i)}$ is the ith element of the dictionary, and where the prime on the sum indicates that it contains no more than b terms. A statistician or applied mathematician might ask: Given a signal S that is assumed to be compressible, can I construct an algorithm that computes a vector R such that $||S - R||$ is bounded by some function of b? By contrast, an "algorithmicist" might ask: Given an arbitrary signal S, can I design an algorithm that finds a b-term vector R_b such that $||S - R_b||$ approximates $||S - R_b^{OPT}||$?

At one extreme, the dictionary A is an orthonormal basis for $\mathbb{R}^n$, in which case the algorithmic problem is typically computationally easy—simply a question of keeping the vectors with the b largest coefficients—but finding a dictionary suitable for a particular application is an art. At the other extreme are arbitrary dictionaries A, in which case exact solution of the problem is NP-hard, and hardness results are known for even the approximation problem. Modern versions of these problems that were the main focus of Muthukrishnan's talk interpolate between these two extremes.

Achlioptas began by reviewing the celebrated Johnson–Lindenstrauss lemma: Given an $m \times n$ matrix A, i.e., given m points in an n-dimensional Euclidean space, there exists a function mapping those points into a k-dimensional space, where $k = O(\log(m)/\varepsilon^2)$, such that all pairwise distances between the points are approximately preserved with high probability. Moreover, this function is just a data-independent random projection.

Achlioptas also described an algorithm for computing low-rank matrix approximations that involves randomly sampling and/or quantizing *elements* of the input matrix. Both of those procedures reduce the description length of the input matrix and can thus be used to accelerate iterative computations based on orthogonal iteration and Lanczos iteration. The algorithm involves constructing a data-dependent random projection. The analysis makes clear that noise is not always a problem in data analysis. Instead, the computational properties of some algorithms, such as computing low-rank matrix approximations, can be improved by the addition of carefully constructed "data-dependent or computation-friendly noise" to the data.

Internet and Tensor-based Applications

Internet applications, a major motivation for the workshop, were the subject of "The Changing Face of Web Search," the tutorial given by Prabhakar Raghavan of Yahoo! Research. Raghavan began by pointing out that Web search, because it provides access to distributed information that is heterogeneous in its creation, accuracy, and motivation, has not only created multi-billion-dollar businesses, but has also provided a rich source of mathematical and algorithmic challenges.

Raghavan described Flickr, a Web site where users share and tag each others' photographs, as a new example of how collective knowledge can be used for search. He also discussed auctions for keyword searches on the Internet, focusing mainly on Yahoo! and Google auctions and on the microeconomic, mechanism design, and algorithmic challenges they pose. More generally, Raghavan presented extensive areas for research in the domain of Internet search, services, and software.

Tensor-based data applications were the theme of the final day of the workshop, which began with a tutorial titled "Tensors, Symmetric Tensors, and Nonnegative Tensors in Data Analysis," by Lek-Heng Lim of Stanford University. In applied data analysis, the Tucker model and the related "canonical decomposition" or "parallel factors" model have a long history; these models are based on the extension of linear algebraic ideas from data matrices to multimode data tensors.

As with matrices, an element of a tensor product of vector spaces can be identified with an array subscripted by multiple indices if a basis for each vector space is specified. Unlike the case for matrices, however, Lim reminded the audience that the tensor rank (defined as the minimum number of rank-one tensors into which the original tensor can be decomposed) is base-field dependent and that computing the rank for a general three-mode tensor is an NP-hard problem.

At least as discouraging for those contemplating the use of tensor methods in data analysis is that the problem of computing the "best" rank-

k approximation to a tensor may have no solution. Such hardness and nonexistence results should not be surprising to readers: After all, there are important connections between such concepts as tensor rank and concepts from algebraic complexity theory, such as the number of multiplications required to compute the product of two matrices. Instead, what should be surprising is that interesting computations can be performed efficiently when consideration is restricted from general tensors to matrices. Lim also described connections between an ℓ_1 generalization of the SVD, nonnegative tensor decompositions, and naive Bayes modeling in statistics.

In addition to other talks on the theory of data algorithms, kernel methods, machine learning, other matrix and tensor decomposition techniques, and the topology of data, participants heard about a wide variety of data applications: applications of recently developed matrix decompositions in genetics, hyperspectral imaging, term–document analysis, information retrieval, and recommendation systems; applications of other methods to ranking for Web search and advertising, text mining, learning retrieval functions, and neuroscience; as well as applications of tensor methods to petascale data, link and semantic data, computer vision data, and cellular and bioinformatic data. Interested readers are encouraged to visit the conference Web site (http://mmds.stanford.edu/), where the presentations from nearly every speaker and every poster abstract can be found.

The feedback received, both from established researchers working in focus areas of the workshop and from many students just starting their research careers, was overwhelmingly positive. Clearly, there is a lot of interest in MMDS as a developing research area at the interface between NLA, TCS, and scientific and Internet data applications. Keep an eye out for the next MMDS!

Acknowledgments

The authors are grateful to the numerous individuals from Yahoo! Research and Stanford University who provided assistance prior to and during MMDS 2006. They also thank the American Institute of Mathematics for supporting and hosting the July 2004 workshop on tensor decompositions at which the organizers of MMDS 2006 originally met. Finally, they thank each of the speakers, poster presenters, and other participants, without whom MMDS 2006 would not have been such a success.

Gene Golub is the Fletcher Jones Professor of Computer Science at Stanford University. Michael Mahoney is a senior research scientist at Yahoo! Research. Petros Drineas is an assistant professor of computer science at Rensselaer Polytechnic Institute. Lek-Heng Lim is a graduate student in the Institute for Computational and Mathematical Engineering at Stanford University.