SAND2007-4897C

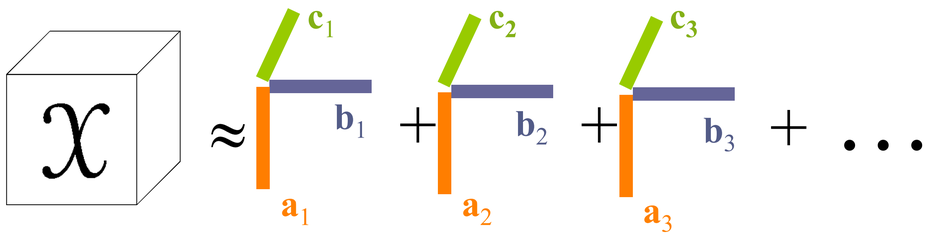
Extracting clusters from large datasets with multiple similarity measures

Sandia Intern Symposium, Livermore, CA - August 2, 2007
Teresa Selee, North Carolina State University, Raleigh NC, Ph. D., Applied Mathematics, est. June 2008
Mentors: Tammy Kolda, Philip Kegelmeyer, Josh Griffin, Mathematics, Informatics, & Decision Sciences Department 8962
Sandia National Laboratories/CA, U. S. Department of Energy

Abstract: Given a group of people, different similarities exist by which to group them, including education, geographic location, social connections, family connections, etc. This idea can be extended to grouping anything, including computer files or papers in academic journals. Our project goal is to devise a model that clusters objects using multiple similarity measures simultaneously. The datasets of interest are too large to be treated by typical procedures, so we have established a new method that exploits the structure of the data to make the computations possible. In order to accomplish this, we store object-feature matrices, each of which is used to form the slices of a tensor. We then employ k-means clustering on compilation feature vectors obtained from a tensor decomposition.

CANDECOMP/PARAFAC (CP) Tensor Decomposition

CANDECOMP (Canonical Decomposition) [4] and PARAFAC (Parallel Factors)[5] are two different names for the same decomposition, first published in 1970, which is a higher-order analogue of the matrix Singular Value Decomposition (SVD) or Principal Component Analysis (PCA). This method, abbreviated as CP, gives a decomposition of a tensor into *R* rank-1 tensors.



For a tensor $\mathcal{X}$ we write : $\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$

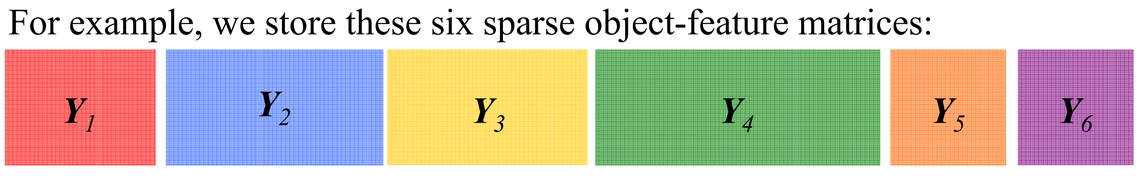
The standard CP Alternating Least Squares (ALS) algorithm begins with a guess at 2 of the matrices, say **A** and **B**, then proceeds as follows:

1. Compute $\mathbf{C} = \mathbf{X}_{(3)} (\mathbf{B} \odot \mathbf{A}) (\mathbf{B}^T\mathbf{B} * \mathbf{A}^T\mathbf{A})^\dagger$
2. Compute $\mathbf{B} = \mathbf{X}_{(2)} (\mathbf{C} \odot \mathbf{A}) (\mathbf{C}^T\mathbf{C} * \mathbf{A}^T\mathbf{A})^\dagger$
3. Compute $\mathbf{A} = \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) (\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B})^\dagger$

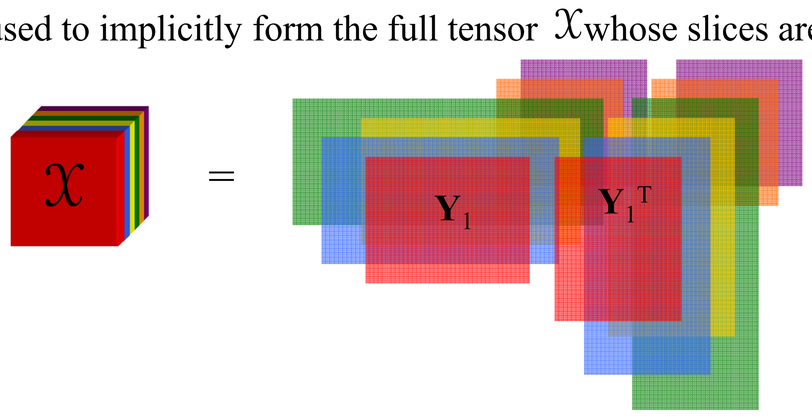
Iterate through steps 1-3 until convergence. The values $\mathbf{X}_{(1)}$, $\mathbf{X}_{(2)}$, and $\mathbf{X}_{(3)}$ are matricized tensors. Matricization of a tensor is a method for converting a tensor to a matrix. For $\mathbf{X}_{(1)}$, the 1st mode (the columns of the tensor) is mapped to the rows, and the 2nd and 3rd modes (the rows and tubes) are mapped to the columns. The Khatri-Rao product is denoted $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \; \mathbf{a}_2 \otimes \mathbf{b}_2 \cdots \mathbf{a}_R \otimes \mathbf{b}_R]$, with Kronecker product $\mathbf{a}_1 \otimes \mathbf{b}_1$. The Hadamard product (elementwise matrix product) is denoted $\mathbf{A} * \mathbf{B}$, and † denotes pseudo-inverse.

IMSCAND (Implicit Slice Canonical Decomposition)

- In our work, the tensor has a special form: each slice is the product of a sparse matrix ($\mathbf{Y}_i$) and its transpose ($\mathbf{Y}_i^T$).
- We don't store full slices (which are dense), just the sparse $\mathbf{Y}_i$ matrices.
- For example, we store these six sparse object-feature matrices:



which are used to implicitly form the full tensor $\mathcal{X}$ whose slices are similarity matrices.



We developed IMSCAND, a decomposition of these special tensors.

Similarities between CP & IMSCAND:

- The decompositions are identical.
- The same number of iterations are required to compute them.

Differences between CP & IMSCAND:

- IMSCAND stores sparse matrices, which are implicitly multiplied by their transpose to form the slices of the tensor. All computations are done on the sparse matrices directly.
- CP stores fully formed slices, which can be dense.
- IMSCAND requires three new formulas for calculating a tensor times a vector, depending on the mode of multiplication.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

The IMSCAND Algorithm

The main difference in the calculations is that $\mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})$, $\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})$, and $\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$ are computed differently. These matrices are all computed columnwise. We have an $N \times N \times P$ tensor, with each $\mathbf{Y}_k$ having dimension $N \times M_k$, The values for $\mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})$ are computed elementwise, so the *(i, j)* entry is:

$$\widehat{c}_{ij} = \sum_{m=1}^{M_i} \left(Y_i^T b_j\right)_m \left(Y_i^T a_j\right)_m$$

For $\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})$ we compute the j^{th} column as: $\widehat{b}_j = \sum_{p=1}^{P} c_{pj} \left(Y_p \left(Y_p^T a_j\right)\right)$

The computation for $\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$ is identical to the computation for $\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})$ by switching a_j with b_j. These codes are written using Matlab and the Tensor Toolbox [1, 2, 3].

Application: clustering SIAM Journal data

- Dataset: 5 years of publications from 11 journals and 1 conference proceedings published by the Society for Industrial and Applied Mathematics (SIAM).
- Explicit links: when one paper cites another.
- Implicit links: connections between papers by author, title words, abstract words, and keywords.

We construct a tensor in which each slice is formed from the product of a sparse matrix and its transpose, and gives a different similarity measure. The slices are:

$\mathbf{X}_1$ = similarity between words in the abstract
$\mathbf{X}_2$ = similarity between names of authors
$\mathbf{X}_3$ = similarity between author-specified keywords
$\mathbf{X}_4$ = similarity between titles
$\mathbf{X}_5$ = co-citation information
$\mathbf{X}_6$ = co-reference information

The first four slices are formed from feature-document matrices for the specified similarity. An element in the slice is nonzero if there is a similarity between the two documents. For the fifth slice, the *(i, j)* element indicates the number of papers citing both papers *i* and *j*. For the sixth slice, the *(i, j)* element indicates the number of papers that both *i* and *j* cite.

SIAM Journal results

Future Work : CARGIO

CARGIO is a multi-year project whose goal is to determine project clusters from a set of files on a computer hard drive. We are currently considering seven different properties of the data, including names of the files, text within the files, relationships to other files, and time between accessing files. The properties either produce sparse matrices, or sparse matrix products that can be used in the IMSCAND algorithm.

Bibliography

[1] B.W. Bader and T.G. Kolda, *Efficient MATLAB computations with sparse and factored tensors*, Technical Report SAND2006-7592, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006.
[2] B.W. Bader and T.G. Kolda, *Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping*, ACM Transactions on Mathematical Software, 32(4), Dec 2006.
[3] B.W. Bader and T.G. Kolda, *MATLAB Tensor Toolbox Version 2.2,* http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/, Jan 2007.
[4] 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.
[5] R.A. Harshman, *Foundations of the Parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics, 16, 1970, pp. 1-84.