TWave: High-Order Analysis of Spatiotemporal Data

Michael Barnathan[1], Vasileios Megalooikonomou[1,2], Christos Faloutsos[3], Feroze B. Mohamed[4], and Scott Faro[4]

[1] Data Engineering Laboratory (DEnLab), Center for Information Science and Technology, Temple University, 1805 N. Broad St., Philadelphia, PA, USA 19122

[2] Computer Engineering and Informatics Dept., University of Patras, Rio 26504, Greece

[3] School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213

[4] Department of Radiology, Temple University School of Medicine, 3401 N. Broad St. Philadelphia, PA 19140

mbarnath@temple.edu, vasilis@ceid.upatras.gr, christos@cs.cmu.edu, {feroze, faros}@temple.edu

Abstract. Recent advances in data acquisition and sharing have made available large quantities of complex data in which features may have complex interrelationships or may not be scalar. For such datasets, the traditional matrix model is no longer appropriate and may fail to capture relationships between features or fail to discover the underlying concepts that features represent. These datasets are better modeled using tensors, which are high-order generalizations of matrices. However, naive tensor algorithms suffer from poor efficiency and may fail to consider spatiotemporal neighborhood relationships in analysis. To surmount these difficulties, we propose TWave, a wavelet and tensor-based methodology for automatic summarization, classification, concept discovery, clustering, and compression of complex datasets. We also derive TWaveCluster, a novel high-order clustering approach based on WaveCluster, and compare our approach against WaveCluster and k-means. The efficiency of our method is competitive with WaveCluster and significantly outperforms k-means. TWave consistently outperformed competitors in both speed and accuracy on a 9.3 GB medical imaging dataset. Our results suggest that a combined wavelet and tensor approach such as TWave may be successfully employed in the analysis of complex high-order datasets.

Keywords: Tensors, matrix models, wavelets, spatiotemporal mining.

1 Introduction

The traditional approach to data representation utilizes a matrix structure, with observations in the rows and features in the columns. Although this model is appropriate for many datasets, it is not always a natural representation because it assumes the existence of a single target variable and lacks a means of modeling dependencies between other features. Additionally, such a structure assumes that observed variables are scalar quantities by definition. This assumption may not be valid in certain domains, such as diffusion tensor imaging, where higher-order features predominate.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 246–253, 2010.

Traditionally, these problems have been solved by reducing the features to scalars and fitting the dataset to a matrix structure. However, as well as potentially losing information, this strategy also employs a questionable approach from a philosophical standpoint: attempting to fit the data to an imprecise model rather than attempting to accurately model the existing structure of the data. Finally, while it may be possible to model dependencies between features by making many runs, each with a different target variable, this yields suboptimal performance and may not be computationally feasible when real-time performance is required or when the dataset is very large.

To address these issues, we propose to model such datasets using *tensors*, which are generalizations of matrices corresponding to *r*-dimensional arrays, where r is known as the *order* of the tensor. Using a combination of wavelet and tensor analysis tools, we propose a framework for summarization, classification, clustering, concept discovery, and compression, which we call TWave. Applying our technique to analysis of the MNIST digit recognition dataset [6] and a large real-world spatiotemporal dataset, we compare the performance of TWave against voxelwise, SVD-based, wavelet-only, and tensor-only techniques and demonstrate that TWave achieves superior results and reduces computation time vs. competing methodologies.

2 Background

2.1 Tensor Tools

Tensors are defined within the context of data mining as multidimensional arrays. The number of indices required to index the tensor is referred to as the *rank* or *order* of the tensor, while each individual dimension is referred to as a *mode*. The number of elements defined on each mode is referred to as the mode's *dimensionality*. The dimensionality of a tensor is written in the same manner as the dimensionality of a matrix; for example, 20x50x10. Tensors represent generalizations of scalars, vectors, and matrices, which are respectively orders 0, 1, and 2.

An important operation applicable to our analysis is the *tensor product* (also the *outer product*). This product generalizes from the Kronecker product, but results in another tensor rather than a block matrix. Given order r and s tensors $\mathcal{A}$ and $\mathcal{B}$, their tensor product $\mathcal{A} \otimes \mathcal{B}$ is a tensor of order $r + s$:

$$(\mathcal{A} \otimes \mathcal{B})_{i_1,i_2,\ldots,i_r,j_1,j_2,\ldots,j_s} = \mathcal{A}_{i_1,i_2,\ldots,i_r} * \mathcal{B}_{j_1,j_2,\ldots,j_s}$$

Singular value decomposition (SVD) is a unique factorization by which an $m \times n$ matrix is decomposed into two projection matrices and a core matrix, as follows:

$$\mathbf{A} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^{\mathbf{T}}$$

where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{U}$ is an $m \times m$ column-orthonormal projection matrix, $\mathbf{V}$ is an $n \times n$ column-orthonormal projection matrix, and $\mathbf{\Sigma}$ is a diagonal $r \times r$ *core matrix*, where r is the (matrix) rank of matrix $\mathbf{A}$.

SVD is used in Latent Semantic Analysis (LSA), an unsupervised summarization technique [1]. Here $\mathbf{A}$ is treated as a term-document matrix. In this context, singular value decomposition automatically derives a user-specified number of latent *concepts* from the given terms, each representing a linear combination. The projection matrices

U and **V** contain term-to-concept and document-to-concept similarities, respectively. Thus, SVD can be used to provide simple yet powerful automatic data summarization.

The natural extensions of singular value decomposition to tensors are the *Tucker* and *PARAFAC* decompositions [2,3]. Let $\mathcal{A}$ be an order-r tensor. Tucker decomposition is a factorization into a *core tensor* $\mathcal{G}$ and *projection matrices* $\mathbf{U}_i$:

$$\mathcal{A} = \mathcal{G} \times \mathbf{U}_1 \times \mathbf{U}_2 \times \ldots \times \mathbf{U}_r$$

Though the Tucker decomposition provides SVD-like data summarization, evaluating it requires computing $\mathcal{A}$'s covariance matrix. This can come at a memory cost of $\Omega(n^2)$, which, for large datasets such as ours, may be prohibitive. Fortunately, PARAFAC avoids this problem. PARAFAC is a generalization of PCA [2] and forms the basis of our tensor analysis approach. Given a user-specified number of concepts *c*, PARAFAC decomposes an order-*r* tensor $\mathcal{A}$ into a columnwise sum of the tensor product of r projection matrices, denoted $\mathbf{U}^{(1)} \ldots \mathbf{U}^{(r)}$, as follows:

$$\mathcal{A} = \sum_{\mathrm{i}=1}^{\mathrm{c}} \lambda_{\mathrm{i}} \mathbf{U}_{:,i}^{(1)} \otimes \mathbf{U}_{:,i}^{(2)} \otimes \ldots \otimes \mathbf{U}_{:,i}^{(r)}$$

Where the **U** matrices represent projection matrices containing mode-to-concept similarities and λ represents a *c*-element scaling vector, in which each element represents the strength of a concept. The notation $\mathbf{U}_{:,i}$ refers to the *i*th column of **U**.

Both the Tucker and PARAFAC decompositions may be computed using alternating least squares (ALS) [**Error! Reference source not found.**], as shown below:

1. Given an order-r tensor $\mathcal{A}$, declare projection matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(r)}$.
2. Let $i = 1$.
3. Holding all other matrices constant, solve the following equation for $\mathbf{U}^{(i)}$:

$$\mathbf{U}^{(i)} = \mathcal{A}^{(i)} \left(\bigodot_{j=1..r \wedge j\neq i} \mathbf{U}^{(j)} \right) \left(\prod_{j=1..r \wedge j\neq i} \left[\mathbf{U}^{(j)}\right]^T \mathbf{U}^{(j)} \right)^*$$

Where $\odot$ represents the *n*-ary Khatri-Rao product, * represents the Moore-Penrose pseudoinverse, and $\mathcal{A}^{(i)}$ represents $\mathcal{A}$ matricized [4] on mode *i*.

4. Repeat for all i from 1 to r until convergence is attained.

The resulting PARAFAC decomposition is illustrated in Figure 2 below:

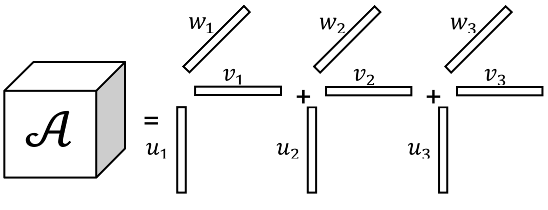

Fig. 2: Illustration of a third-order PARAFAC decomposition.

3 Proposed Method

3.1 Overview

Our methodology makes use of both wavelets and tensors. Because spatiotemporal data tends to exhibit a high degree of spatial locality, the spatiotemporal modes of the dataset are first preprocessed using an *m*-dimensional discrete wavelet transform (obtained through cascading), where *m* is the number of spatiotemporal modes. For applications other than clustering, we utilize the Daubechies-4 wavelet; clustering itself is optimally paired with a hat-shaped wavelet such as the (2,2) biorthogonal wavelet, as these wavelets boost the strengths of dense clusters of points while suppressing outliers. We then linearize the wavelet coefficients to form a vector representing all spatiotemporal voxels in the dataset, reducing the order of the tensor by *d*-1; this overcomes many of the performance issues associated with a high-order pure tensor approach and allows us to threshold the discovered wavelet coefficients, storing the results in a sparse matrix to achieve a significant compression rate.

PARAFAC is then performed using alternating least squares and the resulting projection matrices are stored and analyzed, either by direct inspection or as input to a classifier. This method provides a general framework for further tensor and wavelet analysis, including concept discovery, compression, clustering, and classification.

3.2 Other Methods

It is also possible to analyze data using wavelets and tensors alone, or by using neither preprocessing method (the voxelwise approach). Singular value decomposition run on the dataset in matrix representation additionally provides a benchmark for comparison of the tensor model and techniques.

We performed voxelwise classification by linearizing each image in the dataset and using the normalized values of an image's voxels as a feature vector in classification. Similarly, we performed wavelet classification by using each image's linearized 3-level wavelet coefficient vector (using the Daubechies-4 wavelet) as a feature vector representing that image. Both approximation and detail coefficients at each resolution were included in this analysis.

3.3 Classification

To perform classification using TWave, we wavelet-transform the dataset, run a tensor decomposition such as PARAFAC or Tucker, and directly use each wavelet coefficient's similarity to each concept as an element in the feature vector. We then perform *k*-nearest neighbor classification, which assigns a class to each image based on the majority class of that image's *k* nearest neighbors (using Euclidean distance).

When classifying on a variable other than the principal variable of the dataset, we subtract the mean of the principal variable from the dataset. We have empirically observed this to boost accuracy.

3.4 TWaveCluster

We extended the WaveCluster algorithm to use the PARAFAC decomposition rather than a connected component algorithm to grow the clusters, calling our algorithm TWaveCluster. Our approach exhibits a number of advantages, including the ability to create a fuzzy clustering (where each voxel's degree of membership in cluster *c* is its similarity to concept *c* in the decomposed tensor), the ability to cluster noncontiguous voxels based on patterns in the projected concept space, and even the ability to discover clusters that extend across modes of the tensor. Our approach also has the advantage of simple cluster validation, as the terms in the λ vector automatically represent cluster variance.

The first few steps of our algorithm are identical to WaveCluster:

- Quantize data, using the counts of each grid cell in place of the original data.
- Apply a wavelet transformation using a hat-shaped wavelet (such as the (2,2) or (4,2) biorthogonal wavelets), retaining the approximation coefficients.
- Threshold cells in the transformed space. Cells with values above a user specified density threshold are considered "significant".

However, the remaining steps in our algorithm differ:

- Model significant cells as a tensor $\mathcal{X} \in \Re^{d_1 \times d_2 \times \ldots \times d_r}$.
- For a user-specified k, run a k-concept PARAFAC-ALS analysis on $\mathcal{X}$: $\mathcal{X} = \sum_{i=1}^{k} \lambda_\mathrm{i} \mathbf{U}_{:,i}^{(1)} \otimes \mathbf{U}_{:,i}^{(2)} \otimes \ldots \otimes \mathbf{U}_{:,i}^{(r)}$.
- For each c from 1 to k, recompose a tensor using only column c of each projection. The resulting tensor $\mathcal{X}_c$ contains voxel similarities to concept *c*:

$$\mathcal{X}_c = \lambda_\mathrm{c} \mathbf{U}_{:,c}^{(1)} \otimes \mathbf{U}_{:,c}^{(2)} \otimes \ldots \otimes \mathbf{U}_{:,c}^{(r)}$$

- Assign every voxel the cluster label of its most similar concept:

$$(\forall x \in \mathcal{X})\ \mathcal{L}_x = \arg \max_{1 \le c \le k} (\mathcal{X}_c)_x$$

4 Results

4.1 Dataset

We analyzed each approach on a high-order motor task fMRI dataset consisting of 11 subjects performing 4 simple motor tasks: left finger-to-thumb, left squeeze, right finger-to-thumb, and right squeeze. Classification was also performed on 10,000 randomly-sampled observations from the low-order MNIST digit recognition dataset, split into 5,000 element training and test sets [6]. Acquisition of the fMRI dataset took place using one scanner and one common set of acquisition parameters. Data was acquired from each subject over 120 time points, each 3 seconds long. The period of each task was 30 seconds. Each acquired volume consisted of $79 \times 95 \times 69$ voxels. Thus, the dataset was most easily represented as a 6th order tensor of dimensionality $79 \times 95 \times 69 \times 120 \times 4 \times 11$, of which the first four modes were spatiotemporal and the remaining two were categorical.

4.2 Discovered Concepts

When summarizing the data using a 2-concept TWave analysis, we noticed two outliers among the subject-to-concept similarities, which we found corresponded exactly to the 2 left-handed subjects in the dataset. This pattern was made even more explicit when subtracting the subject means from each subject's set of images, suggesting that the task residuals discriminate better between left and right handed subjects than when task activations are biased by subjects' means. The results of TWave using the Daubechies-4 wavelet and mean subtraction are shown in Figure 3. These results suggest that PARAFAC may be employed as a powerful concept-discovery and feature extraction tool on complex datasets, though we caution that a larger dataset may be necessary to adequately confirm these findings.

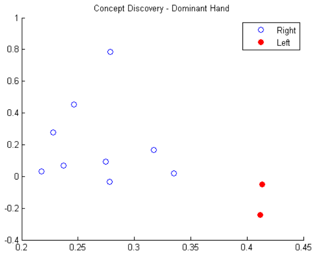

Fig. 3: A 2-concept projection using TWave. The two rightmost points are left-handed.

4.3 Classification

Use of wavelets in particular greatly improved subject classification accuracies, given a complex enough wavelet (to 98% using the Daubechies-4 wavelet but only to 82% for the Haar wavelet). We were able to threshold up to the weakest 98% of wavelet coefficients without any loss of subject or task classification accuracy, greatly improving time and space costs while preserving the discriminative power of the classifier. Further compression is possible in the decomposed tensor through truncation of weak concepts (though computation of these concepts is expensive).

Task classification was more difficult because the intra-subject between-task variance ($\sigma^2 = 179.29$) was less than the between-subject variance ($\sigma^2 = 9066.85$). Initial results yielded only 2% accuracy for voxelwise analysis and 27% accuracy for wavelet-based analysis. However, by subtracting the voxelwise mean of each subject across all tasks, we were able to improve classification substantially. Use of MPCA+LDA [7] as a preprocessing step further improved accuracy. As the sampled MNIST digit recognition dataset is a dense low-order dataset, less difference is seen between low and high-order approaches than in the fMRI dataset, though wavelet preprocessing still did significantly boost accuracy.

4.4 Clustering

We analyzed two subjects on all four spatiotemporal modes of the fMRI tensor using the *k*-means (*k*=4) and TWaveCluster (k=5, density threshold=85th percentile) approaches. Average running times for each method were 53 seconds and 23 seconds, respectively. Discovered clusters are shown in Figure 4. A demarcation can be seen between the frontal and temporal regions of the brain in the TWaveCluster results; this distinction is less clear in *k*-means. The clusters discovered by TWaveCluster show a greater degree of symmetry and homogeneity than the *k*-means clusters, and also yield a clustering in-line with domain expectations.

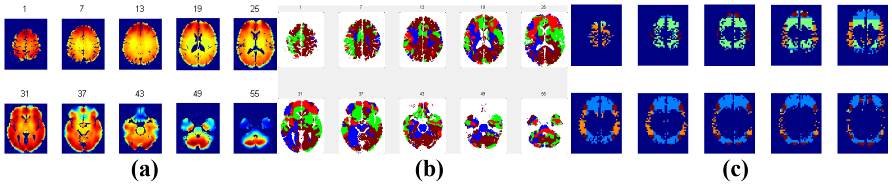

Fig. 4: (a) Activation in a right-handed subject performing a left finger-to-thumb task and the clusters discovered by (b) k-means and (c) TWaveCluster. Only significant voxels are shown.

4.5 Speed and Summary of Results

Runtime was assessed for the voxelwise, wavelet-based, and TWAVE approaches on a dual-processor 2.2 GHz Opteron system with 4 GB of memory. The SVD and pure tensor approaches were measured on an 8 processor (16 core) 2.6 GHz Opteron supercomputer with 128 GB of memory. Despite running on a much more powerful system, the tensor and SVD approaches still took significantly longer to complete than other approaches, as shown in Tables 1 and 2:

Table 1: High-order fMRI dataset runtimes, subject and task classification accuracies, compressed dataset size, and ability to automatically identify left-handed subjects.

	Voxels	Wavelets	SVD	PARAFAC	TWave	TWave+MPCA/LDA
Runtime	95 min	112 min	**3 days**	**8 days**	117 min	130 min
Subjects	52%	98%	80%	88%	96%	100%
Tasks	34%	68%	56%	52%	72%	93%
Size	9.3 GB	181 MB	9.3 GB	9.3 GB	181 MB	181 MB
Lefties?	No	No	No	Yes	Yes	N/A

Table 2: Low-order MNIST digit recognition dataset runtimes and classification accuracies (after random sampling to training set size = 5000, test set size = 5000. *k*=2 in all cases).

	Voxels	Wavelets	SVD	PARAFAC	TWave
Runtime	250 sec	422 sec	**20 min**	**25.3 min**	512 sec
Accuracy	47%	88%	53%	53%	88%

5 Conclusions

From these results, we may conclude that the combination of wavelets and tensor tools in the analysis of fMRI motor task datasets yields better performance in space, time, and accuracy than the voxelwise approach or either technique alone, achieving benefits such as sensitivity to locality while avoiding the prohibitive space and time costs of using only tensors. Additionally, such an approach provides powerful automatic data summarization techniques, as demonstrated through discovery of left-handed subjects in our dataset. Potential avenues for future research include use of different wavelet functions, extension of our methods to streaming and sparse tensor data, and applications to high-order datasets in other fields.

Acknowledgments. We would like to thank Charalampos Tsourakakis for assistance with the literature review and experiments. This work was supported in part by the National Science Foundation under grants IIS-0237921, IIS-0705215, IIS-0705359, and the National Institutes of Health under grant R01 MH68066-05 funded by the National Institute of Mental Health, the National Institute of Neurological Disorders and Stroke, and the National Institute on Aging. All agencies specifically disclaim responsibility for any analyses, interpretations, or conclusions.

References

1. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science, 391-407, (1999)
2. Harshman, R.: Foundations of the PARAFAC Procedure: Models and Conditions for an Explanatory Multimodal Factor Analysis. In: UCLA Working Papers in Phonetics, 1-84 (1970)
3. Carroll, J.D., Chang, J.: Analysis of Individual Differences in Multidimensional Scaling via an n-way Generalization of 'Eckart-Young' Decomposition. Psychometrika, 283-319, (1970)
4. Sands, R., Young, F.W.: Component Models for Three-way Data: An Alternating Least Squares Algorithm with Optimal Scaling Features. Psychometrika,39-67 (1980)
5. Sheikholeslami, G., Chatterjee, S., and Zhang, A: WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. The International Journal on Very Large Data Bases, 289-304 (2000).
6. LeCun, Y., Cortes, C. The MNIST Database of Handwritten Digits, http://yann.lecun.com/exdb/mnist
7. Lu, H., Plataniotis, K. N., Venetsanopoulos, A. N. MPCA: Multilinear Principal Component Analysis of Tensor Objects. IEEE Transactions on Neural Networks 19(1), 18-39 (2008)