Tensor Decomposition for Neurodevelopmental Disorder Prediction

Shah Muhammad Hamdi[1(✉)], Yubao Wu[1], Soukaina Filali Boubrahimi[1], Rafal Angryk[1], Lisa Crystal Krishnamurthy[2,3], and Robin Morris[3,4]

[1] Departement of Computer Science, Georgia State University,
Atlanta, GA 30302, USA
{shamdi1,ywu28,sfilaliboubrahimi1,rangryk}@gsu.edu

[2] Center for Visual and Neurocognitive Rehabilitation, Decatur, GA 30030, USA

[3] Center for Advanced Brain Imaging, Georgia State University and Georgia Institute of Technology, Atlanta, GA 30302, USA
{lkrishnamurthy,robinmorris}@gsu.edu

[4] Department of Psychology, Georgia State University, Atlanta, GA 30302, USA

Abstract. Functional Magnetic Resonance Imaging (fMRI) has been successfully used by the neuroscientists for diagnosis and analysis of neurological and neurodevelopmental disorders. After transforming fMRI data into functional networks, graph classification algorithms have been applied for distinguishing healthy controls from impaired subjects. Recently, classification followed by tensor decomposition has been used as an alternative, since the sparsity of the functional networks is still an open question. In this work, we present five tensor models of fMRI data, considering the time series of the brain regions as the raw form. After decomposing the tensor using CANDECOMP/PARAFAC (CP) and Tucker decomposition, we compared nearest neighbor classification accuracy on the resulting subject factor matrix. We show experimental results using an fMRI dataset from adult subjects with neurodevelopmental reading disabilities and normal controls.

Keywords: fMRI · Tensor decomposition · Reading disabilities

1 Introduction

Brain Informatics, enriched by the advances of neuroimaging technologies such as Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), and Electroencephalography (EEG), pose many challenges to data mining. These imaging modalities are noninvasive methods used to diagnose and investigate neurological and neurodevelopmental disorders. fMRI is a popular brain imaging technique, that records the change in Blood Oxygenation Level Dependent (BOLD) signals in different brain regions over time. Resting-state fMRI-based data analysis has facilitated diagnosis of several neurological and neurodevelopmental diseases such as Alzheimer's, Schizophrenia, Bipolar disorder, Attention-deficit/hyperactivity disorder (ADHD), Autism, and Dyslexia [1,2,5,15].

S. Wang et al. (Eds.): BI 2018, LNAI 11309, pp. 339–348, 2018.
https://doi.org/10.1007/978-3-030-05587-5_32

fMRI data can be represented in various forms, e.g., the sequence of 3D brain volumes over time, multivariate time series, and functional connectivity graphs. Given a training set of fMRI data representations of some human subjects and the associated labels of healthy/diseased, the task of binary classification aims to maximize classification accuracy on test data. Because of the advances in graph mining algorithms, most of the supervised learning studies on fMRI data take functional connectivity graphs (binarized by thresholding) as the inputs and transform the problem into graph classification [1–5]. Graph classification can be addressed by two approaches: structure-based approach and subgraph pattern-based approach. In structure-based approach, node-based features such as degree, PageRank score, and clustering coefficient [1,5] are calculated and each graph is transformed into a vector. In subgraph pattern-based approach [3], discriminative subgraphs are used as features.

When the fMRI data is represented by undirected and unweighted graphs, one big challenge is the correct representation of the graphs. Since the graphs are made by thresholding the functional connectivity matrices (each matrix element denotes the correlation of BOLD time series of two regions of interest or, ROIs), the sparsity of the generated graphs depends on the threshold value. Sparsity affects the performance of both graph classification approaches discussed above. Though most of the data mining papers disregard the edges with negative weights, it is still debated in the neuroscience community whether to keep or discard negative correlations [6].

To address this problem, some recent studies emerged with the idea of tensor-based modeling of fMRI data [7,8]. By stacking the functional connectivity matrices or the multivariate time series of the ROIs of all subjects, a third order tensor can be formed. By decomposing the tensor, we can identify the discriminative representations of subjects, so that subjects with neurological disease and normal controls can be easily separated.

In this work, we apply two well-known tensor decomposition methods, CP and Tucker decomposition, on five different tensor models of fMRI data, and compare the nearest neighbor classification performance on the resulting subject factor matrix. We evaluate the methods on a dataset containing fMRI data of normal adult controls and the subjects with reading disabilities, a neurodevelopmental disorder.

2 Related Work

fMRI data mining research can be divided into three categories: tensor imaging analysis, brain network analysis, and tensor decomposition-based analysis. Tensor imaging analysis deals with the raw fMRI images, i.e., the temporal sequence of brain volumes. Given a set of four-dimensional tensors and their corresponding case/control labels, the classification methods predict the labels of the unlabeled tensors. Vectorizing the tensors causes *curse of dimensionality.* Support Tensor Machine (STM) [9], which is a generalization of the Support Vector Machine (SVM) is proposed to address this issue.

Though the tensor imaging-based classification shows good classification accuracy, this approach could not identify the discriminating features required for interpretation. Wee et al. [1] used functional connectivity networks constructed from fMRI and DTI modalities to distinguish Mildly Cognitive Impaired (MCI) subjects from the healthy controls (MCI is the early stage of Alzheimer's disease). Their approach is an example of a structure-based graph classification approach. Given a functional network, weighted local clustering coefficient of each node is calculated, and the graph is represented by a vector consisting of these local connectivity measures. Then multi-kernel SVM is applied on this vector space. This model gives a ranking of the ROIs in terms of how well they are clustered with respect to other ROIs, which provides a good step towards the interpretability of the discriminating features. Jie et al. [5] presented another structure-based graph classification approach, where they use Weisfeiler-Lehman graph kernel [10] for computing the global connectivity features of each graph, which are used along with a local connectivity feature of weighted local clustering coefficient of each node. Considering the edge weights as the probabilities of the link between two nodes, Kong et al. [4] presented a discriminative subgraph feature selection method based on dynamic programming to compute the probability distribution of the discrimination scores for each subgraph pattern. In some cases of studies of neurological and neurodevelopmental disorders, along with the neuroimaging data, there are additional clinical, serologic and cognitive measures data from each subject that may be available. Cao et al. [3] presented a discriminative subgraph mining algorithm for brain networks which leverages such multiple side views-based data.

Recently, tensor decomposition is used to extract the latent discriminative features of each subject. In [7], tensors are formed by stacking the non-negative connectivity matrices of all subjects. The resulting tensor is decomposed with several constraints such as symmetry of the factor matrix representing the ROI space and orthogonality of the factor matrix representing the subject space in order to maximize the discrimination among the subjects of different classes. In [8], the time-sliced non-negative connectivity matrices are used to make the tensors in order to discover the latent factors of the time windows.

Tensor modeling, i.e., the proper construction of the tensor by stacking different types of data representations, is still a challenging problem. In this work, we present a comparative analysis of different tensor models and show the nearest neighbor classification performance after decomposing the tensors using CP and Tucker decompositions independently.

3 Tensor Decomposition

A multidimensional array is also known as a tensor. An N-th order tensor is the tensor product of N vector spaces, where each vector space has its own coordinate system. Decomposing higher order tensor into lower order tensors is a prominent research problem in mathematics. There are several tensor decomposition algorithms such as CP, Tucker, INDSCAL, PARAFAC2, CANDELINC, DEDICOM, and PARATUCK2 [11]. In this paper, we consider third order tensors for fMRI data and consider CP and Tucker as the methods of tensor decomposition.

3.1 CP Decomposition

CP decomposition factorizes the tensor into a sum of rank one tensors. Given a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, where I, J and K denote the indices of tensor elements in three of its modes, CP decomposition factorizes the tensor in the following way.

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \ \mathbf{o}\mathbf{b}_r \ \mathbf{o} \ \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{1}$$

Here, $\mathbf{o}$ denotes the outer product of the vectors. R is a positive integer and also called the tensor rank. $\mathbf{a}_r, \mathbf{b}_r$, and $\mathbf{c}_r$ are vectors, where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, 2, 3, \ldots R$. After stacking those vectors, we can get the factor matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_R]$, and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_R]$, where $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$.

3.2 Tucker Decomposition

Tucker decomposition is a form of higher order Principal Component Analysis (PCA). A tensor is decomposed into a core tensor, which is multiplied by a matrix along its each mode. Tucker decomposition of a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is given by,

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = [\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{2}$$

Here, $\times_n$ denotes mode-n tensor product. $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the factor matrices. These factor matrices can be thought as the principal components along each mode. The $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ is the core tensor and its elements represent the interaction between those principal components.

Both CP and Tucker decomposition can be solved by Alternating Least Squares (ALS) optimization. After a random initialization of all factor matrices, ALS updates one factor matrix while keeping other two as fixed until convergence. The details of ALS optimization for CP and Tucker decomposition can be found in [11].

4 Modeling the fMRI Data in Tensors

In this section, we describe five tensorization schemata for the fMRI data. In Fig. 1, we visualize five models of the tensor. Among these five models, Tensor Model 3 was previously used in the literature [7], while we designed other four for the purpose of comparison. All the models of the tensors are third order. After tensorizing the data, we use CP and Tucker decomposition for computing the factor matrices. For the healthy/disabled prediction, we use only the factor matrix found in *subjects* mode. Factor matrices in other modes such as *ROIs* and *Timestamps* are out of the scope of this work.

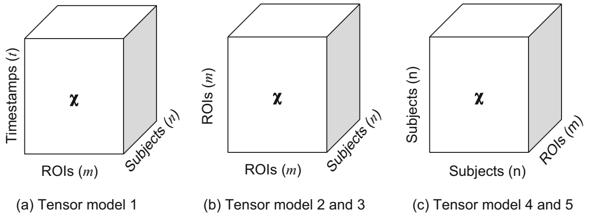


Fig. 1. Dimensions for fMRI data using third order tensor

4.1 Tensor Model 1: Stacked Multivariate Time Series

We have the dataset $D = \{A_1, A_2, \ldots, A_n\}$, where each matrix $A_i \in \mathbb{R}^{m \times t}$ is a multivariate time series, and their corresponding labels of healthy/disabled, which are given by $y_i = \{-1, +1\}$. Here, m denotes the number of ROIs, n denotes the number of subjects, and t denotes the number of time samples. In this tensorization scheme, we simply stack all A_i's together. Therefore, $\mathcal{X} = [A_1; A_2; \ldots, A_n]$ and $\mathcal{X} \in \mathbb{R}^{m \times t \times n}$ (Fig. 1a). After CP decomposition, we get three factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, where $\mathbf{A} \in \mathbb{R}^{m \times R}$, $\mathbf{B} \in \mathbb{R}^{t \times R}$ and $\mathbf{C} \in \mathbb{R}^{n \times R}$. After Tucker decomposition, we get three factor matrices of different number of columns, where $\mathbf{A} \in \mathbb{R}^{m \times P}$, $\mathbf{B} \in \mathbb{R}^{t \times Q}$ and $\mathbf{C} \in \mathbb{R}^{n \times R'}$. Therefore, $\mathbf{C}$ is the factor matrix in the *subject* space, where each row is a vector-based representation of each subject. Then, we split the rows (subjects) into train and test set, concatenate corresponding class label of each training subject, and train a classifier. Finally, we can evaluate the classification performance by predicting the class labels of the test subjects.

4.2 Tensor Model 2: Stacked Functional Connectivity Matrices

For each multivariate time series matrix A_i, we calculate Pearson correlation coefficient between each pair of time series. It gives us functional connectivity matrices $C_1, C_2, \ldots, C_n$, where $C_i \in \mathbb{R}^{m \times m}$. Each matrix C_i is symmetric and can be thought of as an adjacency matrix of an edge-weighted complete graph K_m. By stacking the C_i's one after another, we get a tensor $\mathcal{X} \in \mathbb{R}^{m \times m \times n}$ (Fig. 1b). After CP decomposition we get three factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, where $\mathbf{A} \in \mathbb{R}^{m \times R}$, $\mathbf{B} \in \mathbb{R}^{m \times R}$ and $\mathbf{C} \in \mathbb{R}^{n \times R}$. Since two modes are the same in the third order tensor, after CP decomposition we get two identical factor matrix, i.e., $\mathbf{A} = \mathbf{B}$. The similar case is also found in Tucker decomposition. In this tensor modeling scheme, $\mathbf{C}$ is the necessary subject factor matrix.

4.3 Tensor Model 3: Stacked Non-negative Functional Connectivity Matrices

Here, the matrices $C_1, C_2, \ldots C_n$ are thresholded by keeping only the non-negative matrix elements. Therefore, C_i's do not represent edge-weighted complete graphs, rather they denote weighted and undirected sparse graphs. The shape of the tensor is the same as Tensor Model 2 and the tensor is given by $\mathcal{X} \in \mathbb{R}^{m \times m \times n}$ (Fig. 1b). The factor matrices that are found after CP and Tucker decomposition in this tensorization scheme is similar to the Tensor Model 2. Factor matrix **C**, which is defined in the *subject* space is the necessary factor matrix.

4.4 Tensor Model 4: Node-Wise Jaccard Kernel on Functional Connectivity Matrices

In this tensor model, we consider each C_i as an edge-weighted complete graph. For each pair of complete graphs, weighted Jaccard is calculated using the vectors represented by each node (vector of each node is found from the weights associated with the adjacent edges of that node). Given two vectors S and T, weighted Jaccard between them is [12]:

$$J(S, T) = \frac{\sum_k min(S_k, T_k)}{\sum_k max(S_k, T_k)} \tag{3}$$

In Fig. 2, we show two example edge-weighted complete graphs. If g_i^j denotes the node j of graph g_i, then the calculation of node-wise Jaccard between these two edge-weighted complete graphs is as follows.

$$\begin{aligned} J(g_1, g_2) &= [J(g_1^A, g_2^A), J(g_1^B, g_2^B), J(g_1^C, g_2^C)] \\ &= [\frac{0.1 - 0.5}{0.2 + 0.3}, \frac{0.1 - 0.6}{0.2 - 0.4}, \frac{-0.5 - 0.6}{0.3 - 0.4}] \\ &= [-0.8, 2.5, 11] \end{aligned}$$

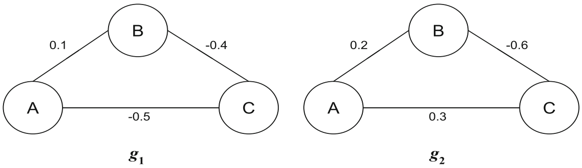


Fig. 2. Two edge-weighted complete graphs.

By calculating the node-wise Jaccard between each pair of complete graphs, we get a tensor $\mathcal{X} \in \mathbb{R}^{n \times n \times m}$ (Fig. 1c). After tensor decomposition, we get two identical factor matrices in the subjects space.

4.5 Tensor Model 5: Node-Wise Jaccard Kernel on Non-negative Functional Connectivity Matrices

In this model, we ignore the negative weighted edges in the complete graphs. Similar to the calculation of node-wise Jaccard described for Tensor Model 4, we get a tensor $\mathcal{X} \in \mathbb{R}^{n \times n \times m}$ (Fig. 1c). We similarly get two identical factor matrices in the subject space after tensor decomposition.

5 Experimental Evaluation

We used *Tensor Toolbox* [13] of MATLAB for computing CP and Tucker decomposition with ALS optimization. In this section, we show the performance of CP and Tucker decomposition on five tensor models.

5.1 Data Collection

In our dataset, the target neurodevelopmental disorder was reading disability. For the study, we used preprocessed resting-state fMRI scans of 27 adult subjects from the local community, 12 labeled as *struggling* readers (below-average reading test scores), and 15 labeled as *typical* (average on reading test).

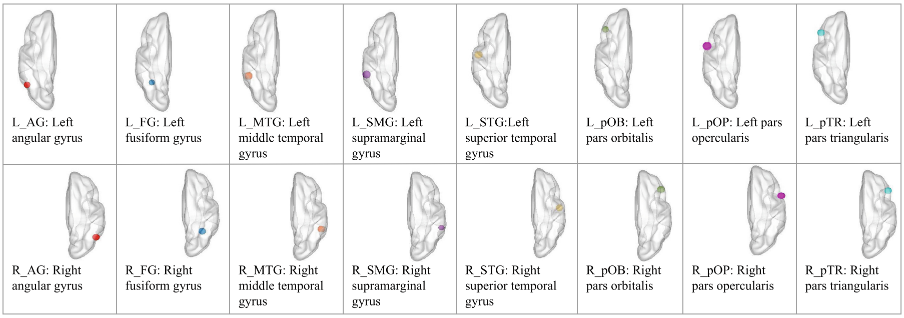
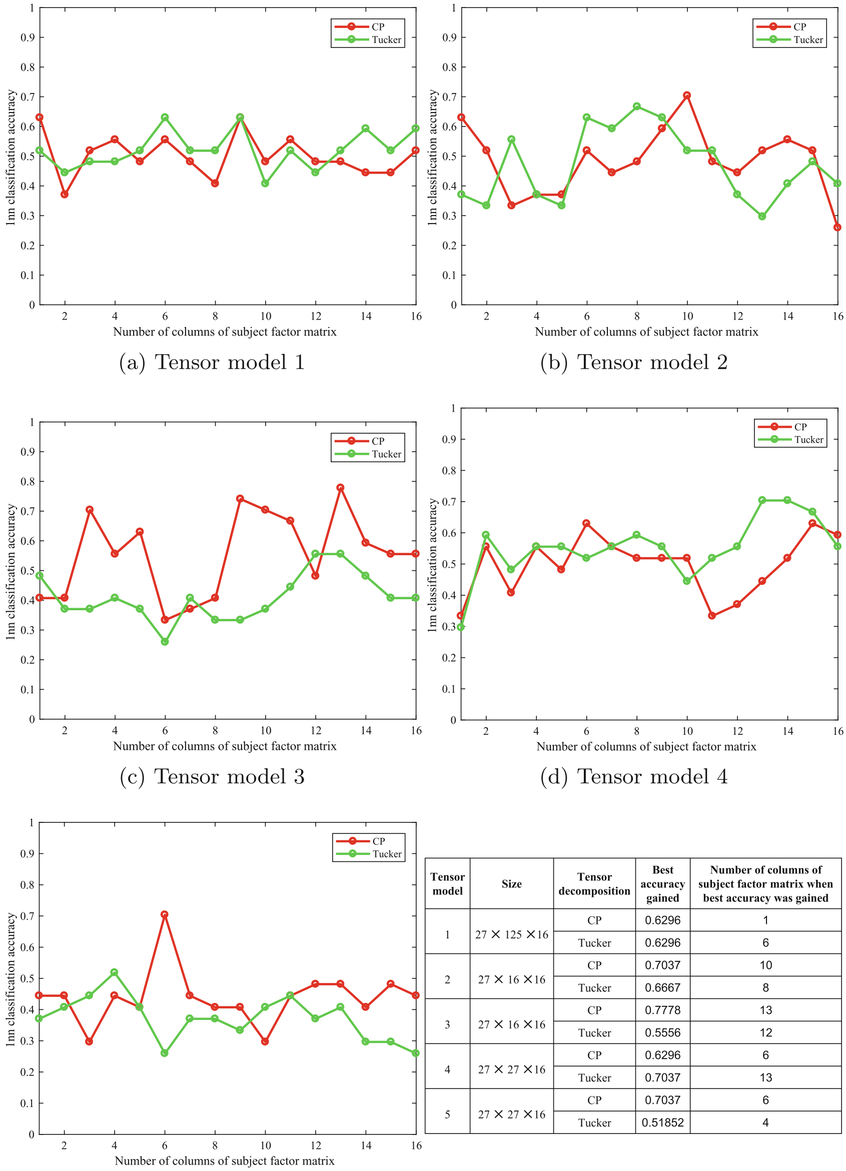

L_AG: Left angular gyrus
L_FG: Left fusiform gyrus
L_MTG: Left middle temporal gyrus
L_SMG: Left supramarginal gyrus
L_STG:Left superior temporal gyrus
L_pOB: Left pars orbitalis
L_pOP: Left pars opercularis
L_pTR: Left pars triangularis
R_AG: Right angular gyrus
R_FG: Right fusiform gyrus
R_MTG: Right middle temporal gyrus
R_SMG: Right supramarginal gyrus
R_STG: Right superior temporal gyrus
R_pOB: Right pars orbitalis
R_pOP: Right pars opercularis
R_pTR: Right pars triangularis

Fig. 3. Visualization of 16 ROIs in left and right hemisphere

The details of the preprocessing steps and experimental settings of this study can be found in [14]. We used 16 ROIs based on apriori reading research [15]. In Fig. 3, we show the visualization of these ROIs in left and right hemisphere of the brain. The preprocessing steps are done using AFNI, FSL, and FreeSurfer. After preprocessing, the number of voxels in one brain volume is 53 × 63 × 45, while the repetition time (time between capturing two whole brain volume) is 2 s. Finally, using Conn [16] the multivariate time series and functional connectivity matrices of each subject are extracted. The length of each time series of each ROI is 125.

5.2 Evaluation Method

After obtaining the subject factor matrix by tensor decomposition, label information is concatenated and labeled representations of the subjects are fed into a classifier with a predefined train/test splitting strategy. In our experiments,



(a) Tensor model 1

(b) Tensor model 2

(c) Tensor model 3

(d) Tensor model 4

(e) Tensor model 5

Tensor model	Size	Tensor decomposition	Best accuracy gained	Number of columns of subject factor matrix when best accuracy was gained
1	27 × 125 ×16	CP	0.6296	1
		Tucker	0.6296	6
2	27 × 16 ×16	CP	0.7037	10
		Tucker	0.6667	8
3	27 × 16 ×16	CP	0.7778	13
		Tucker	0.5556	12
4	27 × 27 ×16	CP	0.6296	6
		Tucker	0.7037	13
5	27 × 27 ×16	CP	0.7037	6
		Tucker	0.51852	4

(f) Comparison of all tensor models

Fig. 4. Comparison of CP and Tucker decomposition on five tensor models on the basis of nearest neighbor classification on subject factor matrix

we chose nearest neighbor classifier (1nn) with Euclidean distance measure in order to evaluate discriminative subject representations in different subject factor matrices. As the train/test sampling method, we chose leave-one-out sampling because of the small number of samples. Leave-one-out is a special case of K-fold cross-validation while K is the number of samples. At i-th iteration of leave-one-out, all samples except the i-th one are used as training samples and the i-th sample is used as the test sample. Finally, the mean accuracy over all iterations is calculated.

5.3 Performances in Different Tensor Models

We varied the number of columns in the subject factor matrix from 1 to 16, since 16 is the minimum number of elements in three modes of the tensor (note that there are 16 ROIs, 27 subjects, and 125 timestamps). In Tucker decomposition, we fixed the number of columns in other factor matrices (ROIs and timestamps) same as their number of rows. Then, we evaluated the subject representations by feeding the subject factor matrix to the nearest neighbor classifier.

Figure 4a–e show the 1nn classification accuracy on the subject factor matrix found after CP and Tucker decomposition. Since ALS implementation of CP and Tucker decomposition is used, the resulting subject factor matrix depends on the initialization of other factor matrices representing ROIs and timestamps. Therefore, if we increase the number of columns of the subject factor matrix, which is equivalent to increasing the number of features in feature space, the classification accuracy may not increase linearly. The nonlinearity of accuracy with respect to the number of columns of the subject factor matrix is also supported by other studies [7,8]. In Fig. 4f, we summarized our findings by presenting the best accuracies along with the number of columns used in the subject factor matrix. Tensor Model 1 results in poor classification accuracy in both CP and Tucker decomposition, because it does not produce partially symmetric tensors like other models. CP decomposition performed better than Tucker decomposition in terms of best accuracy in all tensor models except Tensor Model 4. Tucker decomposition performed poor, especially when the non-negativity constraint is applied on the construction of the tensors (Tensor Models 3 and 5). Using CP decomposition on the tensor constructed from Tensor Model 3, we observed the best accuracy among all the experimental runs, which is 78%. Therefore, we validate the fact that for maximizing the classification accuracy, the combination of Tensor Model 3 and CP decomposition can be a good choice.

6 Conclusion

In this study, we have performed a comparative analysis of the tensor decomposition techniques for fMRI data. We evaluated five different tensor models, and showed the nearest neighbor classification accuracy on the subject factor matrix generated by CP and Tucker decomposition.

In the future, we aim to study the interpretability of the features (columns) of the factor matrices. Leveraging other factor matrices of ROIs and timestamps, and using them along with the subject factor matrix can result in more robust subject-based representations.

References

1. Wee, C.-Y., et al.: Identification of MCI individuals using structural and functional connectivity networks. Neuroimage **59**(3), 2045–2056 (2012)
2. Cao, B., et al.: Identification of discriminative subgraph patterns in fMRI brain networks in bipolar affective disorder. In: Guo, Y., Friston, K., Aldo, F., Hill, S., Peng, H. (eds.) BIH 2015. LNCS (LNAI), vol. 9250, pp. 105–114. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23344-4_11
3. Cao, B., Kong, X., Zhang, J., Philip, S.Y., Ragin, A.B.: Mining brain networks using multiple side views for neurological disorder identification. In: 2015 IEEE International Conference on Data Mining (ICDM), pp. 709–714. IEEE (2015)
4. Kong, X., Yu, P.S.: Semi-supervised feature selection for graph classification. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 793–802. ACM (2010)
5. Jie, B., Zhang, D., Gao, W., Wang, Q., Wee, C.-Y., Shen, D.: Integration of network topological and connectivity properties for neuroimaging classification. IEEE Trans. Biomed. Eng. **61**, 576–589 (2014)
6. Zhu, Y., Cribben, I.: Graphical models for functional connectivity networks: best methods and the autocorrelation issue. bioRxiv, p. 128488 (2017)
7. Cao, B., et al.: t-BNE: tensor-based brain network embedding. In: SIAM (2017)
8. Cao, B., Lu, C.-T., Wei, X., Yu, P.S., Leow, A.D.: Semi-supervised tensor factorization for brain network analysis. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016, Part I. LNCS (LNAI), vol. 9851, pp. 17–32. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46128-1_2
9. Tao, D., Li, X., Hu, W., Maybank, S., Wu, X.: Supervised tensor learning. In: 5th IEEE International Conference on Data Mining. IEEE (2005)
10. Shervashidze, N., Schweitzer, P., Leeuwen, E.J.V., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. J. Mach. Learn. Res. **12**, 2539–2561 (2011)
11. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
12. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the 6th New Zealand Computer Science Research Student Conference (NZCSRSC 2008), Christchurch, New Zealand, p. 4956 (2008)
13. Bader, B.W., Kolda, T.G., et al.: Matlab tensor toolbox version 2.6, February 2015. http://www.sandia.gov/tgkolda/TensorToolbox/index-2.6.html
14. Krishnamurthy, V., et al.: Retrospective correction of physiological noise: impact on sensitivity, specificity, and reproducibility of resting-state functional connectivity in a reading network model. Brain Connect. **8**, 94–105 (2017)
15. Martin, A., Schurz, M., Kronbichler, M., Richlan, F.: Reading in the brain of children and adults: A meta-analysis of 40 functional magnetic resonance imaging studies. Hum. Brain Map. **36**(5), 1963–1981 (2015)
16. Whitfield-Gabrieli, S., Nieto-Castanon, A.: CONN: a functional connectivity toolbox for correlated and anticorrelated brain networks. Brain Connect. **2**(3), 125–141 (2012)