Tensor-based Clustering for fMRI Shared Response Modeling

Jennifer Hsia
Department of Computer Science
Princeton University
Princeton, United States of America
jhsia@princeton.edu

Harshvardhan K. Babla
Department of Electrical Engineering
Princeton University
Princeton, United States of America
hbabla@princeton.edu

I. Introduction

To effectively compare and analyze data from multiple concordant sources, one can first learn a shared latent space. For example, in fMRI brain imaging, each data point is a video of a person's functioning brain. By learning a shared latent space, one can analyze and predict how subjects as a whole respond to stimuli such as movies, audiobooks, and still images [1].

We focus on multi-subject, multi-dataset fMRI analysis and explore how to better cluster the subjects to develop fast, scalable models to represent multiple subjects' data in the same learned latent space (Fig.1). Formally, the problem can be described as

$$\min_{W,S} 1/s \sum_i \|X_i - W_i S\|_F^2. \tag{1}$$

where X_i represents the matrix of data collected from subject i, W_i is the corresponding gram matrix for the subject. The S matrix describes the evolution of the learned latent vectors with time.

Previous models have employed an orthogonality constraint in learning the latent space. This is not only unnatural, but can also incur a computational penalty that limits scalability [1], [2]. Furthermore, this constraint forces the latent features to have uniform norm and be uncorrelated. The unit norm prevents feature sensitivity to magnitude, while feature independence prevents cross-feature interactions.

In response, we have presented new models that do not *a priori* force an orthogonality constraint, but instead allow the models to naturally learn the appropriate latent structure in the data [3]. The new models use gradient descent paired with specially designed regularizers. This lowers computation complexity, improves scalability, allows for possible cross-feature correlation, and enhances feature magnitude sensitivity.

One of models from [3] uses a centroid regularizer to push all the user gram matrices $W_i^T W_i$ to be closer to each other. Specifically, the algorithm implements gradient descent with the following regularization:

$$(1/s) \sum_{i=1}^{s} \|W_i^T W_i - C\|_F^2 \tag{2}$$

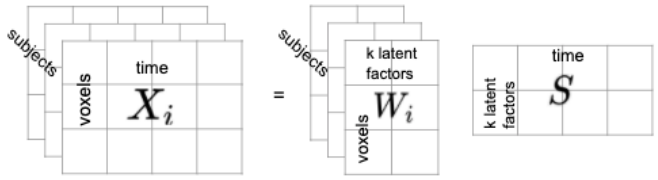

Fig. 1. Matrix factorization of fMRI data.

Our goal is to see if we can perform better by using multiple centroids instead of one (as seen as above). We use tensor factorization and clustering based on the subjects vector to determine the centroids in our regularization. The underlying assumption here is that certain people who share similar traits (e.g., race, gender, socioeconomic background) may react similarly to the same stimulus. If this is true, it makes sense to create centroid regularizations for each group to enforce similarity within the groups instead of among everyone.

II. Tensor Decomposition

Since the data is collected as a tensor, any latent structure in the data would be revealed best by decomposing the tensor into a sum of parts. The CANDECOMP/PARAFAC (Canonical Decompositon/Parallel Factors) decomposition, commonly known as the CP decompositon, factorizes the tensor as a sum of component rank-1 tensors (Fig 2). In our case, we wish to best approximate the third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ by solving the following optimization problem:

$$\min_{\hat{\mathcal{X}}} ||\hat{\mathcal{X}} - \mathcal{X}|| \tag{3}$$

$$\text{where } \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

where R is a positive integer, and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ $\forall r \in [R]$ are normalized. This can be written elementwise as,

$$x_{i,j,k} \approx \sum_{r=1}^{R} a_{i,r} b_{j,r} c_{k,r} \tag{4}$$

Previous work in neuroscience has applied CP decomposition to fMRI data from single subjects [5], arranged as voxels by time by experiment iteration. Other authors [6]–[9] have used CP decomposition to extract the latent elements from

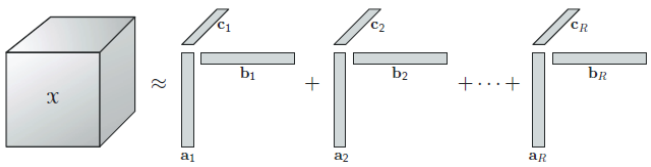

Fig. 2. CP decomposition of an order-3 tensor. Sourced from [4]

time-varying EEG spectra, arranged as a three-dimensional array with modes corresponding to time, frequency, and channel. To our knowledge, no one has used tensor decomposition to extract similarities between subjects using fMRI data.

Our algorithm uses the intuition that the CP decomposition constructs an R-dimensional latent feature space. The vector $\mathbf{a}_r$ is then though of list of each of the subjects' "weights" on the r^{th} feature. Similarly, $\mathbf{b}_r$, and $\mathbf{c}_r$ are the weights for the time-stamps and voxels respectively. We can arrange the rank-one component vectors as factor matrices i.e. $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_R\end{bmatrix}$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. Then following our intuition, the columns of $\mathbf{A}^{\mathrm{T}}$ represent each of the subjects within the latent-space. This allows us to cluster similar subjects, as described in Section III.

We follow the Alternative Least Squares (ALS) algorithm for CP decomposition [4]. The algorithm first fixes $\mathbf{B}$ and $\mathbf{C}$ and solves for $\mathbf{A}$. It then fixes $\mathbf{A}$ and $\mathbf{C}$ and solves for $\mathbf{B}$. Finally, it fixes $\mathbf{A}$ and $\mathbf{B}$ and solves for $\mathbf{C}$. These steps are repeated until the convergence criterion is satisfied. Suppose $\mathbf{B}$ and $\mathbf{C}$ are fixed, then the problem can be simplified as a least-squares optimization of $\mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathrm{T}}$ on the mode-1 matricization of tensor, $\mathbf{X}_{(1)}$ [4], where $M_1 \odot M_2$ represents the Khatri-Rao product.

$$\min_{\hat{\mathbf{A}}} ||\mathbf{X}_{(1)} - \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^{\mathrm{T}}||_{\mathrm{F}} \tag{5}$$

where $\hat{\mathbf{A}} = \mathbf{A} \cdot \mathrm{diag}(\lambda)$. This has the optimal solution [4]:

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^{\mathrm{T}}\mathbf{C} * \mathbf{B}^{\mathrm{T}}\mathbf{B})^{\dagger} \tag{6}$$

where $M_1 * M_2$ denotes the Hadamard matrix product and $M_1^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. Therefore, each iteration of the algorithm simply involves computing the pseudo-inverse of three $R \times R$ matrices, rather than a $JK \times R$ matrix.

Ref. [4] observes that the ALS algorithm can take many iterations to converge, without any guarantee of convergence. Moreover, the accuracy of the algorithm is heavily dependant on the initial starting guess for the factor matrices. Nonetheless, it's success with other applications motivates us to explore it further.

III. Clustering algorithms

As alluded to in Section II, our next step involves clustering the rows of the subject factor-matrix, $\mathbf{A}$. We explore three different clustering methods: K-Means, Birch, and Spectral clustering. Each of these methods separate the subjects into disjoint clusters based on their vectors in the latent space.

The K-means algorithm describes each cluster C_j, by the mean μ_j of samples within the cluster. The algorithm aims to minimize the within-cluster-sum-of-squares:

$$\sum_{i=0}^{N} \min_{C_j \in C} ||x_i - \mu_j||^2 \tag{7}$$

where $C = \{C_j\}$ is the set of clusters and N is the number of samples. This metric makes the assumption that clusters are convex and isotropic, therefore responding poorly to elongated clusters, or manifolds with irregular shapes. Since the metric is not normalized this algorithm suffers from the "curse of dimensionality", where Euclidean distances are over-inflated in high-dimensional spaces. However, the algorithm is guaranteed to converge and usually does so quite quickly [10].

Birch clustering iteratively builds a tree, known as the Clustering Feature Tree (CFT) for the given samples. Every new sample is inserted into the root of the CFT. It is then merged with the "closest" subcluster of the root and merged recursively until it reaches a leaf. After finding the nearest subcluster in the leaf, the properties of this subcluster and the parent subclusters are recursively updated. The closeness metric is the Euclidean distance constrained by a branching factor and a threshold distance. The Birch algorithm is commonly used to identify large data sets with many samples in each cluster. However, as it uses the Euclidean distance, it too doesn't scale well to high-dimensional data [10].

Spectral clustering views the data as a graph. There are several ways to construct this graph, we opt to connect the k nearest neighbors of each data-point with an edge. The spectrum of this graph's Laplacian offers some interesting insights about the graph density, number of clusters, and the min-cut [11]. Particularly, the Fiedler vector (the second eigenvector) assigns each of the nodes to a cluster. Spectral clustering is ideal for non-flat data, clusters in concentric shell shapes, and convex clusters. It struggles as the number of clusters is increased [10].

Since we aren't familiar with the arrangement of our subject vectors in the latent space, we implement all three methods; hoping to learn the structure of our data from their relative successes.

IV. Tensor-based Clustering, Shared Response Modeling Algorithm

Our model for predicting the shared fMRI response of a group of subjects uses gradient descent with the following tensor-based clustering regularization:

$$\sum_{j=1}^{k} (1/s) \sum_{i \in S_j} \|W_i^T W_i - C_j\|_F^2 \tag{8}$$

where k denotes the number of clusters and S_j denotes the indices of the subjects that belong to cluster j. The reason for regularizing the gram matrix $W_i^T W_i$ instead of the subject matrix W_i directly is that the gram matrix can reveal rotationally invariant structure. fMRI measurement inevitably has some small measurement variations and errors that may be

due to rotations. Using the gram matrices to compare subject matrices eliminates confounding factors that may arise from rotation. Formally, if we have a $W_j \approx QW_i$ where Q is an orthogonal matrix, $\|W_j^T W_j - W_i^T W_i\|_F^2 \approx 0$. Whereas, if we use $\|W_j - W_i\|_F^2 \not\approx 0$.

V. Experiment and Results

We tested our cluster-centroid model, the original centroid model [3], and the deterministic model [1] on a dataset which includes the 10 subjects' fMRI responses to the movie Raides of the Lost Ark. We used rank k = 50. SRM is the deterministic model with othogonality constraint. GD-NB is gradient descent without any regularization. GD-Ctr1 is the model from [3] that creates a centroid based on all the subjects. GD-Ctr2 and GD-Ctr3 use gradient descent with two and three centroids respectively based on our tensor factorization and clustering results.

The resulting gram matrices (Fig. 3) reveal some of the differences in the learned latent brain models. From SRM, the subject gram matrices are orthogonal as explicitly required. From GD-NB, or vanilla gradient descent, there is no regularization, hence all the subject gram matrices are noticeably different from each other. From GD-Ctr1 [3], the centroid regularization is applied to all the subjects. For GD-Ctr3, there are three clusters corresponding to three centroids. One cluster contains only the 3rd subject and another cluster contains only the 4th subject. The rest of the subjects are in the last cluster. As a result, the third subject's gram matrix $W_2^T W_2$ looks slighlty different from the others.

We used time-segment matching to evaluate the representative power of the latent brain models. The accuracy results is shown in Fig. 4. First, we train the models on the first timeseries half of the data, then test on the second timerseries half of the data. The training part yields user matrices W_i, which we then use to derive S_i for the test data. During testing, we hold out a subject's stimulus matrix S_i, then create an aggregate matrix of the remaining S_j. We match each time segment with the time segment it has the highest correlation with from the aggregate matrix. If the matched segment is the same segment as the one from the held out set, then the match is successful and accurate. This process is repeated for each subject.

Unfortunately, we don't see an obvious differences in the centroid models of different cluster numbers. However, we suspect a main reason is that there are only 10 subject, so clustering in this case may not be very informative. This is more evident when we examine the clustering results:

```
n_clusters:  2
K_Means:  [1 1 1 0 1 1 1 1 1 1]
Birch:    [0 0 0 1 0 0 0 0 0 0]
Spectral: [0 0 0 0 0 0 0 0 0 1]
```

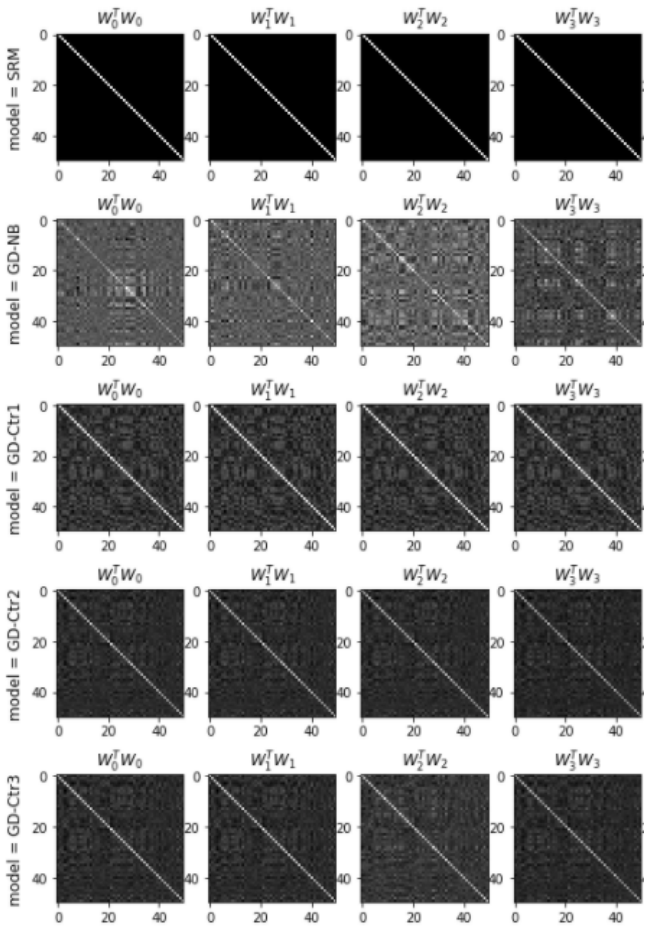

Fig. 3. Gram matrices for the first five subjects of each model.

```
n_clusters:  3
K_Means:  [1 1 2 0 1 1 1 1 1 1]
Birch:    [0 0 2 1 0 0 0 0 0 0]
Spectral: [0 0 2 1 0 0 0 0 0 0]
```

Notice that every time we increase the number of clusters, the new cluster only has one member. We examine the results resulting from number of clusters from 2 to 10, and realize that this phenomenon of one-member clusters hold, as shown in Table 1 below. Since this is consistent across all three algorithms, we suspect that the vectors are sufficiently close together, forming one cluster all-together. As a result, it is not clear whether this tensor-based clustering for shared response modeling is beneficial or not.

VI. Conclusion

We present tensor-based clustering for shared response modeling. However, since the number of subjects is too small, the clustering effect is not pronounced. As a result, we can not confidently claim the reliability and the generalizability of the results. One future step is to test on fMRI datasets with more subjects so that we can potentially find clusters with more members in them. It may also help to use large fMRI datasets

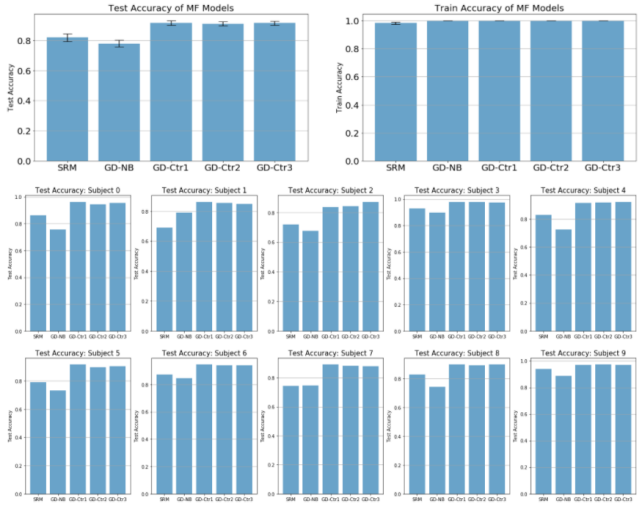

Fig. 4. Results for the time-segment matching experiment.Top: Test and train accuracy averaged over all subjects. Below: Accuracy for each subject.

where we know biological information (e.g., race, gender) so we can test our assumption more accurately.

REFERENCES

[1] P.-H. C. Chen et al., "A reduced-dimension fmri shared response model," in *Advances in NIPS 28*. Curran Associates, Inc., 2015, pp. 460–468. [Online]. Available: http://papers.nips.cc/paper/5855-a-reduced-dimension-fmri-shared-response-model.pdf

[2] A. A. Joshi et al., "Synchronization of resting fmri time-series across subjects," *NeuroImage*, vol. 172, pp. 740–752, 2018.

[3] J. Hsia and P. J. Ramadge, "Large-scale shared response model for latent feature analysis," *Women in Machine Learning Workshop collocated with NeurIPS*, 2020.

[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.

[5] A. H. Andersen and W. S. Rayens, "Structure-seeking multilinear methods for the analysis of fmri data," *NeuroImage*, vol. 22, no. 2, pp. 728 – 739, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811904001181

[6] E. Martinez-Montes, P. A. Vales-Sosa, F. Miwakeichi, R. I. Goldman, and M. S. Cohen, "Concurrent eeg/fmri analysis by multiway partial least squares," *NeuroImage*, vol. 22, no. 3, pp. 1023 – 1034, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811904001946

[7] F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, "Decomposing eeg data into space–time–frequency components using parallel factor analysis," *NeuroImage*, vol. 22, no. 3, pp. 1035 – 1045, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811904001958

[8] M. Mørup, L. K. Hansen, and S. M. Arnfred, "Erpwavelab: A toolbox for multi-channel analysis of time–frequency transformed event related potentials," *Journal of Neuroscience Methods*, vol. 161, no. 2, pp. 361 – 368, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016502700600567X

[9] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, "Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg," *NeuroImage*, vol. 29, no. 3, pp. 938 – 947, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811905005896

[10] SciKit Learn. Clustering algorithms. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html

[11] W. Fleshman. Spectral clustering. [Online]. Available: https://towardsdatascience.com/spectral-clustering-aba2640c0d5b

APPENDIX A: CLUSTERING RESULTS

Number of Clusters	Clustering Results									
2	K-Means :	1	1	1	0	1	1	1	1	1
	Birch :	0	0	0	1	0	0	0	0	0
	Spectral :	0	0	0	0	0	0	0	0	1
3	K-Means :	1	1	2	0	1	1	1	1	1
	Birch :	0	0	2	1	0	0	0	0	0
	Spectral :	0	0	2	1	0	0	0	0	0
4	K-Means :	0	0	2	1	0	0	0	3	0	0
	Birch :	0	0	2	3	0	0	0	0	0	1
	Spectral :	0	0	0	1	0	0	0	0	0	2
5	K-Means :	1	1	2	0	1	1	1	4	1	3
	Birch :	0	0	2	3	0	0	0	4	0	1
	Spectral :	3	0	4	1	0	0	0	2	0	3
6	K-Means :	1	1	2	0	1	5	1	4	1	3
	Birch :	0	0	5	3	0	2	0	4	0	1
	Spectral :	1	4	2	3	4	0	4	3	4	1
7	K-Means :	1	1	2	4	1	5	6	3	1	0
	Birch :	0	0	5	3	0	6	2	4	0	1
	Spectral :	3	5	2	4	1	0	5	6	5	3
8	K-Means :	1	1	4	2	0	5	6	3	1	7
	Birch :	7	4	5	3	7	6	2	1	7	0
	Spectral :	2	5	1	6	3	0	4	6	4	2
9	K-Means :	1	6	2	0	7	5	8	3	1	4
	Birch :	0	1	2	3	0	4	5	6	0	7
	Spectral :	2	5	1	7	4	0	6	7	2	3
10	K-Means :	9	6	2	0	7	5	8	3	1	4
	Birch :	0	1	2	3	0	4	5	6	0	7
	Spectral :	7	5	1	9	4	0	6	9	2	3

TABLE I
CLUSTERING RESULTS FROM ALL THREE ALGORITHMS. THE CLUSTERS ARE LABELLED ARBITRARILY.