Bayesian Tensor Decomposition Model for Neuroimages

Sakul Mahat (Texas A&M University)
Advisor: Dr. Cai Li
St. Jude Children's Research Hospital
2024 Summer Internship Report

Abstract

This report presents a Bayesian Tensor Decomposition Model for clustering, specifically applied to the analysis of brain functional connectomes from the Adolescent Brain Cognitive Development (ABCD) study. The project utilizes tensor decomposition techniques to identify subtypes within the dataset through clustering. By employing a Bayesian framework, the model facilitates robust statistical analysis and accounts for uncertainty in parameter estimation. The Gibbs sampling method is implemented to iteratively sample from the posterior distributions of the model parameters. A comprehensive simulation study is conducted to evaluate the performance of the Gibbs sampler, comparing the true and estimated factor matrices and assessing the convergence of the algorithm. The results demonstrate the efficacy of the Bayesian Tensor Decomposition Model in capturing underlying patterns in brain connectivity data and accurately identifying subtypes.

1 Introduction

The Adolescent Brain Cognitive Development (ABCD) study is the largest long-term study of brain development and child health in the United States, funded by the National Institutes of Health (NIH). The study involves 11,880 children aged 9-10 and tracks their biological and behavioral development through adolescence into young adulthood.

The primary goal of this project is to identify subtypes or clusters within the study participants based on their brain functional connectomes. Clustering helps in understanding the heterogeneity in brain development and its association with cognitive and behavioral outcomes.

2 Tensor Representation/ PARAFAC Decomposition

We measure brain functional connectomes for N subjects. The connectome can be represented as a graph $G_i = (V, E_i)$ with V nodes and E_i edges. Each graph G_i is represented by a $V \times V$ connectivity matrix A_i, showing connectivity between nodes k and l. Stacking all N subjects, we now work with a tensor of $\mathcal{A} \in \mathbb{R}^{V \times V \times N}$.

$$\mathcal{A} = \left[\begin{pmatrix} & & \cdots & \\ & & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ & & \cdots & \end{pmatrix}_{V\times V}, \begin{pmatrix} & & \cdots & \\ & & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ & & \cdots & \end{pmatrix}_{V\times V}, \cdots, \begin{pmatrix} & & \cdots & \\ & & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ & & \cdots & \end{pmatrix}_{V\times V} \right]$$

The 3D brain connectome tensor for each subject can be visualized as:

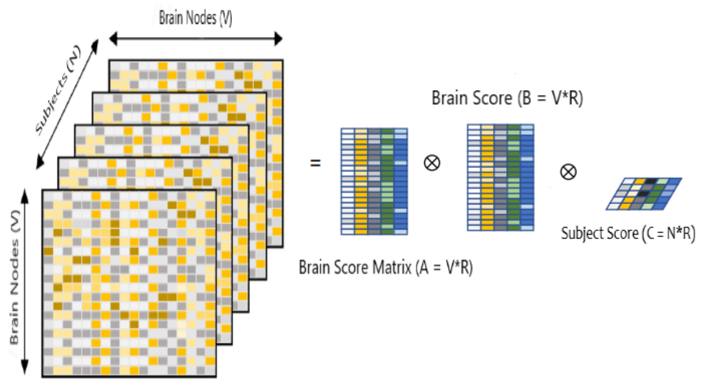

Figure 1: 3D Brain Connectome Tensor Representation

The tensor $\mathcal{A}$ is decomposed using PARAFAC (Parallel Factor Analysis) as follows:

$$\mathcal{A} \approx \sum_{r=1}^{R} \lambda_r (\mathbf{A}[:, r] \otimes \mathbf{B}[:, r] \otimes \mathbf{C}[:, r]) + \mathcal{E}$$

where λ_r is the weight for the r-th rank-one tensor, $\mathbf{A} \in \mathbb{R}^{V\times R}$, $\mathbf{B} \in \mathbb{R}^{V\times R}$, $\mathbf{C} \in \mathbb{R}^{N\times R}$ are the factor matrices, and $\mathcal{E} \in \mathbb{R}^{V\times V\times N}$ is the error tensor.

Important Note: $\mathbf{A}$ and $\mathbf{B}$ are factor matricies that represent the relationship of the first two modes of the data with the latent component r respectively. Since the first two modes are the same (brain nodes), the relationship potrayed by the two factor matricies across the latent components will be the same. Hence, we can write $\mathbf{B}[:, r] = \mathbf{A}[:, r]$. This simplifies the decomposition to:

$$\mathcal{A} \approx \sum_{r=1}^{R} \lambda_r(\mathbf{A}[:, r] \otimes \mathbf{A}[:, r] \otimes \mathbf{C}[:, r]) + \mathcal{E}$$

Each element is approximated as:

$$a_{vwn} \approx \sum_{r=1}^{R} \lambda_r A_{vr} A_{wr} C_{nr} + e_{vwn}$$

3 Objective Function

There are two school of thoughts to go about this analysis:

(i) Frequentist Approach: Minimize the sum of squared errors:

$$\min_{\lambda_r, \mathbf{A}, \mathbf{C}} \|\mathcal{A} - \sum_{r=1}^{R} \lambda_r(\mathbf{A}[:, r] \otimes \mathbf{A}[:, r] \otimes \mathbf{C}[:, r])\|_F^2$$

Use methods like Alternating Least Squares (ALS). Frequentist inference gives us point estimates and inherently does not consider uncertainty in the parameter. Therefore, we pivot towards a Bayesian framework which gives us a probabilistic interpretation, explicitly providing uncertainty about parameters.

(ii) Bayesian Approach: Maximize the posterior distribution (MAP):

$$\theta_{\text{MAP}} = \arg\max_{\theta} P(\theta \mid \mathcal{A}) \propto \arg\max_{\theta} \left(P(\mathcal{A} \mid \theta) \cdot P(\theta)\right)$$

If we assume each elements of the error tensor $\text{e}_{vwn} \sim \mathcal{N}(0, \tau^{-1})$

Then, each observed entry a_{vwn} in the tensor $\mathcal{A}$ as follows:

$$a_{vwn} \sim \mathcal{N}\left(\sum_{r=1}^{R} \lambda_r A_{vr} A_{wr} C_{nr}, \tau^{-1}\right)$$

The likelihood function for the entire tensor $\mathcal{A}$ is then given by:

$$P(\mathcal{A} \mid \theta) = \prod_{v=1}^{V} \prod_{w=1}^{V} \prod_{n=1}^{N} \mathcal{N}\left(a_{vwn} \mid \sum_{r=1}^{R} \lambda_r A_{vr} A_{wr} C_{nr}, \tau^{-1}\right)$$

4 Choice of Priors

(i) For brain nodes $\mathbf{A}[:, r]$, we use a normal prior with precision β_a:

$$a_{vr} \sim \mathcal{N}(0, \beta_a^{-1})$$

(ii) For the elements c_{tr}, we use a normal prior with precision β_c:

$$c_{tr} \sim \mathcal{N}(0, \beta_c^{-1})$$

(iii) For the coefficients λ_r, we use a normal prior with precision β_λ:

$$\lambda_r \sim \mathcal{N}(0, \beta_\lambda^{-1})$$

(iv) For the precision parameters β_a, β_c, and β_λ, we use gamma hyperpriors:

$$\beta_a \sim \text{Gamma}(\alpha_a, \gamma_a)$$

$$\beta_c \sim \text{Gamma}(\alpha_c, \gamma_c)$$

$$\beta_\lambda \sim \text{Gamma}(\alpha_\lambda, \gamma_\lambda)$$

(v) For the precision parameter τ_{vwt} of the error distribution, we use a gamma prior:

$$\tau_{vwt} \sim \text{Gamma}(U, V)$$

5 Bayesian Inference

Given the specified priors, our parameter set θ can be partitioned into:

$$\theta = \{\mathbf{A}, \mathbf{C}, \lambda, \beta_a, \beta_c, \beta_\lambda, \tau_{vwt}\}$$

In this Bayesian inference, we use a Gibbs sampler to iteratively sample each parameter given the other (see Appendix for Update Derivations)

6 Simulation Study

We generate a synthetic tensor $\mathcal{A}$ with known properties. Specifically, we create factor matrices $\mathbf{A}_{\text{True}}$ and $\mathbf{C}_{\text{True}}$ with a sparse block structure. These matrices are used to form the tensor $\mathcal{A}$ using the PARAFAC decomposition, adding Gaussian noise to simulate realistic data.

The Gibbs sampler is applied to the simulated tensor to obtain posterior samples of the factor matrices $\mathbf{A}$ and $\mathbf{C}$. This allows us to estimate the posterior distributions of the model parameters.

We compare the true factor matrices ($\mathbf{A}_{\text{True}}$ and $\mathbf{C}_{\text{True}}$) with the posterior means of the estimated factor matrices ($\mathbf{A}_{\text{Posterior}}$ and $\mathbf{C}_{\text{Posterior}}$). By visualizing these matrices, we can assess whether the posterior estimates capture the underlying sparse block structure of the true matrices.

To assess the convergence of the Gibbs sampler, we calculate the distance between consecutive posterior samples of $\mathbf{A}$ and $\mathbf{C}$. These distances are plotted over iterations to observe if they decrease, indicating convergence of the sampler. Additionally, we also plot the trace plot of the different precision & hyperparameters.

Matrix	**RMSE**
A	1.425
C	0.320

Table 1: RMSE scores for A and C (comparing posterior with true factors)

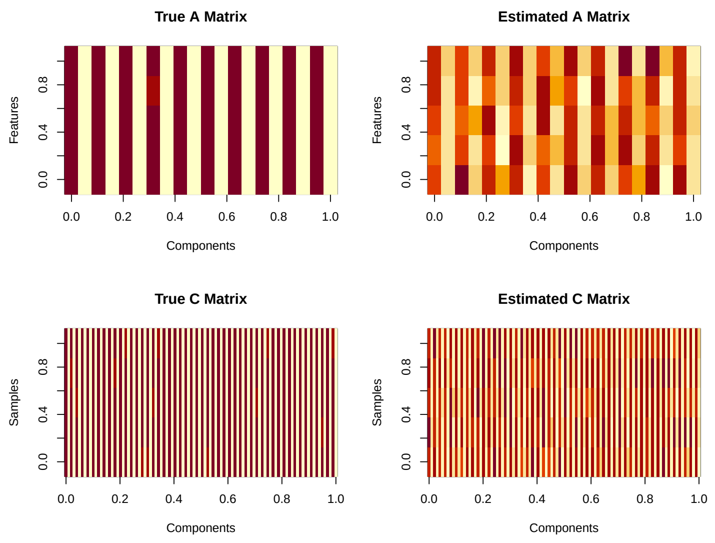

Figure 2: Posterior Assessment of the Factor Matrices

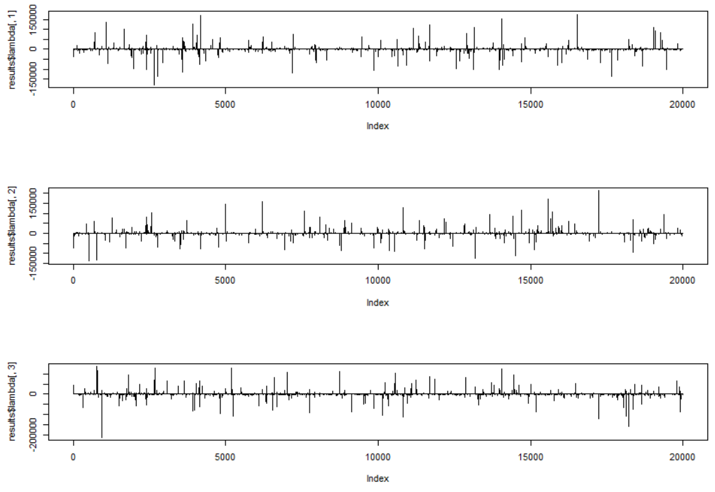

Figure 3: Posterior Assessment of the Lambda

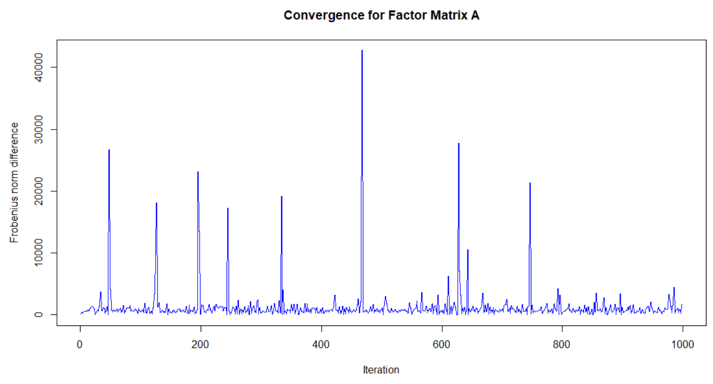

Figure 4: Posterior Assessment of Convergence of A

7 Future Work - Clustering Subjects using Dirichlet Process

After obtaining the factor matrix $\mathbf{C} \in \mathbb{R}^{N \times R}$ from the Bayesian Tensor Decomposition, we proceed to cluster the subjects based on the latent factors captured in $\mathbf{C}$. To achieve this, we employ a Dirichlet Process (DP) mixture model, which allows for a flexible number of clusters without requiring prior knowledge of the number of clusters.

7.1 Dirichlet Process Mixture Model

The DP mixture model assumes that the data points (rows of $\mathbf{C}$) are generated from a mixture of an unknown number of components. Formally, we model each subject's factor vector $\mathbf{c}_n$ as:

$$\mathbf{c}_n \sim \sum_{k=1}^{\infty} \pi_k \cdot \mathcal{N}(\mathbf{c}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where:

- π_k are the mixture weights drawn from a stick-breaking process:

$$\pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j), \quad v_k \sim \text{Beta}(1, \alpha)$$

- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance matrix of the k-th Gaussian component, respectively.
- α is the concentration parameter of the Dirichlet Process, controlling the number of clusters.

7.2 Inference and Clustering

Given the DP mixture model, we infer the cluster assignments for each subject n by sampling from the posterior distribution using methods such as Gibbs sampling or variational inference. The posterior distribution provides the probability that each subject belongs to each cluster.

$$z_n \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi \sim \text{Categorical}(\pi)$$

The clustering results can then be used to identify subtypes of subjects based on their brain functional connectomes, providing insights into the heterogeneity in brain development.

8 References

- Shiying Wang, Todd Constable, Heping Zhang, and Yize Zhao. "Heterogeneity Analysis on Multi-State Brain Functional Connectivity and Adolescent Neurocognition." *Journal of the American Statistical Association*, vol. 119, no. 546, 2024, pp. 851–863. Taylor & Francis. https://doi.org/10.1080/01621459.2024.2311363

- Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I. McCarthy, Kerrin Small, and Jonathan Marchini. "Tensor decomposition for multiple-tissue gene expression experiments." *Nature Genetics*, vol. 48, no. 9, 2016, pp. 1094-1100. https://doi.org/10.1038/ng.3624

9 Appendix

In this appendix, we provide the update equations for each parameter in the Gibbs sampler. The parameter set θ includes $\mathbf{A}$, $\mathbf{C}$, λ, β_a, β_c, β_λ, and τ_{vwt}.

9.1 Updating A

Here we only update the upper triangular elements.

$$a_{vr} \mid \mathbf{A}_{-vr}, \mathbf{C}, \lambda, \beta_a, \tau, \mathcal{A} \sim \mathcal{N}(\mu_{a_{vr}}, \sigma^2_{a_{vr}})$$

$$\mu_{a_{vr}} = \frac{\sum_{ijk} \tau \left(a_{ijk} - \sum_{r' \neq r} \lambda_{r'} A_{ir'} A_{jr'} C_{kr'} \right) \lambda_r C_{kr}}{\sum_{ijk} \tau \lambda_r^2 C_{kr}^2 + \beta_a}$$

$$\sigma^2_{a_{vr}} = \frac{1}{\sum_{ijk} \tau \lambda_r^2 C_{kr}^2 + \beta_a}$$

9.2 Updating C

For each c_{tr}:

$$c_{tr} \mid \mathbf{A}, \mathbf{C}_{-tr}, \lambda, \beta_c, \tau, \mathcal{A} \sim \mathcal{N}(\mu_{c_{tr}}, \sigma^2_{c_{tr}})$$

$$\mu_{c_{tr}} = \frac{\sum_{ijk} \tau \left(a_{ijk} - \sum_{r' \neq r} \lambda_{r'} A_{ir} A_{jr} C_{kr'} \right) \lambda_r A_{ir} A_{jr}}{\sum_{ijk} \tau \lambda_r^2 A_{ir}^2 A_{jr}^2 + \beta_c}$$

$$\sigma^2_{c_{tr}} = \frac{1}{\sum_{ijk} \tau \lambda_r^2 A_{ir}^2 A_{jr}^2 + \beta_c}$$

9.3 Updating λ_r

For each λ_r:

$$\lambda_r \mid \mathbf{A}, \mathbf{C}, \lambda_{-r}, \beta_\lambda, \tau, \mathcal{A} \sim \mathcal{N}(\mu_{\lambda_r}, \sigma^2_{\lambda_r})$$

$$\mu_{\lambda_r} = \frac{\sum_{ijk} \tau \left(a_{ijk} - \sum_{r' \neq r} \lambda_{r'} A_{ir} A_{jr} C_{kr} \right) A_{ir} A_{jr} C_{kr}}{\sum_{ijk} \tau A_{ir}^2 A_{jr}^2 C_{kr}^2 + \beta_\lambda}$$

$$\sigma^2_{\lambda_r} = \frac{1}{\sum_{ijk} \tau A_{ir}^2 A_{jr}^2 C_{kr}^2 + \beta_\lambda}$$

9.4 Updating β_a

$$\beta_a \mid \mathbf{A}, \alpha_a, \gamma_a \sim \mathrm{Gamma}(\alpha'_a, \gamma'_a)$$
$$\alpha'_a = \alpha_a + \frac{VR}{2}$$
$$\gamma'_a = \gamma_a + \frac{1}{2}\sum_{vr} a_{vr}^2$$

9.5 Updating β_c

$$\beta_c \mid \mathbf{C}, \alpha_c, \gamma_c \sim \mathrm{Gamma}(\alpha'_c, \gamma'_c)$$
$$\alpha'_c = \alpha_c + \frac{NT}{2}$$
$$\gamma'_c = \gamma_c + \frac{1}{2}\sum_{tr} c_{tr}^2$$

9.6 Updating β_λ

$$\beta_\lambda \mid \lambda, \alpha_\lambda, \gamma_\lambda \sim \mathrm{Gamma}(\alpha'_\lambda, \gamma'_\lambda)$$
$$\alpha'_\lambda = \alpha_\lambda + \frac{R}{2}$$
$$\gamma'_\lambda = \gamma_\lambda + \frac{1}{2}\sum_{r} \lambda_r^2$$

9.7 Updating τ_{vwt}

$$\tau_{vwt} \mid \mathcal{A}, \mathbf{A}, \mathbf{C}, \lambda, U, V \sim \mathrm{Gamma}(U', V')$$
$$U' = U + \frac{1}{2}$$
$$V' = V + \frac{1}{2}\left(a_{vwt} - \sum_r \lambda_r A_{vr} A_{wr} C_{tr}\right)^2$$