1 Model

1.1 Multivariate Skew Normal Mixture Model

[Check capitalization of headings for Biometrics – maybe only first letter is capitalized?]

A primary goal of the Nurture study is to identify clusters of infants characterized by distinct motor development trajectories. To address this aim, we propose a flexible finite mixture model that accommodates relevant features of the data, such as skewness and dependence among the responses. To this end, let $\mathbf{y}_i = (y_{i1}, \ldots, y_{iJ})^T$ be a $J \times 1$ vector of responses (i.e., Baley scores) for subject i $(i = 1, \ldots, n)$. For the analysis of the Nurture data, we propose a finite mixture model of the form

$$f(\mathbf{y}_i) = \sum_{k=1}^{K} \pi_{ik} f(\mathbf{y}_i | \boldsymbol{\theta}_k), \tag{1}$$

where $\boldsymbol{\theta}_k$ is the set of parameters specific to cluster k $(k = 1, ..., K)$ and π_{ik} is a subject-specific mixing weight representing the probability that subject i belongs to cluster k. For now we assume that K is fixed; in Section 4, we discuss model selection strategies for choosing the optimal value of K. We also assume that class membership is fixed throughout the study period, since our focus is to cluster individuals based on their overall developmental patterns over the course of the study. In Section 6, we discuss extensions to allow for class membership to vary over time. **[We could omit these last two sentence – are they really needed? Not sure. Maybe keep for now and think about it.]**

To facilitate posterior inference, we introduce a latent cluster indicator variable z_i taking the value $k \in \{1, ..., K\}$ with probability π_{ik}. Conditional on $z_i = k$, we assume $\mathbf{y}_i$ is distributed as

$$\mathbf{y}_i | (z_i = k) \sim MSN_J(\boldsymbol{\zeta}_{ki}, \boldsymbol{\alpha}_k, \boldsymbol{\Omega}_k), \tag{2}$$

where $MSN_J(\cdot)$ denotes the J-dimensional multivariate skew normal density, $\boldsymbol{\zeta}_{ki}$ is a $J \times 1$ vector of subject- and cluster-specific location parameters, $\boldsymbol{\alpha}_k$ is a $J \times 1$ vector of cluster-specific skewness parameters, and $\boldsymbol{\Omega}_k$ is a $J \times J$ cluster-specific scale matrix that captures dependence among the J responses. The vector $\boldsymbol{\alpha}_k$ has components α_{kj}, $j = 1, ..., J$ **[let's use kij as our index hierarchy. Please review throughout]**, that control the skewness of outcome j in cluster k. When $\boldsymbol{\alpha}_k = \mathbf{0}$, the MSN distribution reduces to the multivariate normal distribution $N_J(\boldsymbol{\zeta}_k, \boldsymbol{\Omega}_k)$, where $\boldsymbol{\Omega}_k$ is a $J \times J$ covariance matrix.

We can extend model (2) to the regression setting by modeling $\boldsymbol{\zeta}_{ki}$ as a function of covariates. Here we adopt a convenient stochastic representation of the MSN density (Azzalini and Dalla Valle, 1996):

$$\mathbf{y}_i | (z_i = k, t_i) = \mathbf{X}_i \boldsymbol{\beta}_k + t_i \boldsymbol{\psi}_k + \boldsymbol{\epsilon}_{ki}, \tag{3}$$

where $\mathbf{X}_i$ is a $J \times Jp$ design matrix that includes potential time-varying covariates (e.g., indicators denoting quarterly visits); $\boldsymbol{\beta}_k = (\beta_{k11}, \ldots, \beta_{k1p}, \ldots, \beta_{kJ1}, \ldots, \beta_{kJp})^T$

is a $Jp \times 1$ vector of cluster- and outcome-specific regression coefficients; $t_i \sim N_{[0,\infty)}(0,1)$ is a subject-specific standard normal random variable truncated below by zero; $\boldsymbol{\psi}_k = (\psi_{k1}, \ldots, \psi_{kJ})^T$ is a $J \times 1$ vector of cluster-specific skewness parameters; and $\boldsymbol{\epsilon}_{ki} \sim N_J(\mathbf{0}, \boldsymbol{\Sigma}_k)$ is a $J \times 1$ vector of error terms. Thus, conditional on t_i and $z_i = k$, $\boldsymbol{y}_i$ is distributed as $N_J(\mathbf{X}_i\boldsymbol{\beta}_k + t_i\boldsymbol{\psi}_k, \boldsymbol{\Sigma}_k)$. Marginally (after integrating over t_i), $\mathbf{y}_i$ is distributed $MSN_J(\boldsymbol{\zeta}_{ik}, \boldsymbol{\alpha}_k, \boldsymbol{\Omega}_k)$, where through back-transformation [**Carter: Please review these expressions carefully and make sure they conform to those in FS and Pyne**]

$$\begin{aligned} \boldsymbol{\zeta}_{ik} &= \mathbf{X}_i\boldsymbol{\beta}_k \\ \boldsymbol{\alpha}_k &= \frac{1}{\sqrt{1 - \boldsymbol{\psi}_k^T\boldsymbol{\Omega}_k^{-1}\boldsymbol{\psi}_k}}\boldsymbol{\Omega}_k^{-1}\boldsymbol{\psi}_k \quad \text{and} \\ \boldsymbol{\Omega}_k &= \boldsymbol{\Sigma}_k + \boldsymbol{\psi}_k\boldsymbol{\psi}_k^T. \end{aligned}$$

Additional details can be found in Fr uwirth-Schnatter and Pyne (2010).

Of note, the MSN density can be expressed more compactly in terms of the matrix skew normal (MatSN) density (Chen and Gupta 2005). As we will see in Section 3.6, the matrix representation of the MSN distribution admits convenient conjugate prior distributions for the regression parameters and scale matrices, which in turn leads to efficient Gibbs sampling for posterior inference. Let $\mathbf{Y}_k$ be an $n_k \times J$ response matrix with rows $\mathbf{y}_i^T$, $(i = 1, ..., n_k)$, where $n_k = \sum_{i=1}^n 1_{(z_i=k)}$ is the number of observations in cluster k. From equation (3), it follows that $\mathbf{Y}_k$ is distributed as

$$\begin{aligned} \mathbf{Y}_k &\sim MatSN_{n_k \times J}(\mathbf{M}_k, \boldsymbol{\alpha}_k, \mathbf{I}_{n_k}, \boldsymbol{\Omega}_k) \\ \text{vec}(\mathbf{M}_k) &= (\boldsymbol{\zeta}_{k1}^T, ..., \boldsymbol{\zeta}_{kn_k}^T)^T, \end{aligned}$$

where $\boldsymbol{\zeta}_{ki} = \mathbf{X}_i\boldsymbol{\beta}_k$ as in equation (3), $\boldsymbol{\alpha}_k = (\alpha_{k1}, ..., \alpha_{kJ})^T$, $\mathbf{I}_{n_k}$ is the $n_k \times n_k$ identity matrix, and $\boldsymbol{\Omega}_k$ is the $J \times J$ scale matrix defined above in equation (2). From equation (3), it follows that $\mathbf{Y}_k$, conditional on the $n_k \times 1$ vector of random effects $\mathbf{t}_k$, is jointly distributed in matrix form as

$$\mathbf{Y}_k|\mathbf{t}_k \sim MatNorm_{n_k \times J}(\mathbf{M}_k, \mathbf{I}_{n_k}, \boldsymbol{\Sigma}_k),$$

where $MatNorm_{n_k \times J}(\cdot)$ denotes a $n_k \times J$ matrix normal density, $\text{vec}(\mathbf{M}_k) = \mathbf{X}_k\boldsymbol{\beta}_k + \mathbf{t}_k \otimes \boldsymbol{\psi}_k$ is an $n_kJ \times 1$ mean vector, $\mathbf{X}_k$ is an $n_kJ \times Jp$ design matrix, $\boldsymbol{\beta}_k$ is the $(Jp) \times 1$ vector of regression coefficients defined in equation (3), and $\boldsymbol{\Sigma}_k$ is the $J \times J$ conditional covariance of $\boldsymbol{\epsilon}_{ik}$ given in equation (3).

1.2 Multinomial Regression for the Cluster Indicators

To accommodate heterogeneity in the cluster-membership probabilities, we model π_{ik} as a function of coovariates using a multinomial logit model

$$\pi_{ik} = \Pr(z_i = k|\mathbf{w}_i) = \frac{e^{\mathbf{w}_i^T\boldsymbol{\delta}_k}}{\sum_{h=1}^K e^{\mathbf{w}_i^T\boldsymbol{\delta}_h}}, \; k = 1, \ldots, K, \tag{4}$$

where $\mathbf{w}_i$ is an $r \times 1$ vector of subject-level covariates, $\boldsymbol{\delta}_k$ is a $r \times 1$ vector of regression parameters associated with membership in cluster k. For identifiability purposes, we fix the reference category $k = K$ and set $\boldsymbol{\delta}_K = \mathbf{0}$. Under this model, $z_i|\mathbf{w}_i \sim Multinom(1, \boldsymbol{\pi}_i)$, where $\boldsymbol{\pi}_i = (\pi_{i1}, ..., \pi_{1K})$. During MCMC estimation, the cluster labels z_i are updated from their multinomial full conditional distribution and used in the remaining MCMC steps as cluster assignments.

1.3 Conditional MSN Imputation

To accommodate missing at random (MAR) responses, we propose a convenient imputation algorithm that can be implemented "online" as part of the Gibbs sampler. In Section 6, we discuss extension to allow for non-ignorable missingness. Suppose $\mathbf{y}_i$ has $q_i \in (1, \ldots, J)$ observed values, denoted $\mathbf{y}_i^{obs}$, and $J - q_i$ intermittent missing values, denoted $\mathbf{y}_i^{miss}$. We can use of the stochastic representation given in equation (3) to impute $\mathbf{y}_i^{miss}$ from its conditional multivariate normal distribution given (z_i, t_i, Y_i^{obs}):

$$\begin{aligned} \mathbf{y}_i^{miss}|(z_i = k, t_i, \mathbf{y}_i^{obs}) &\sim N_{J-q}(\boldsymbol{\mu}_{ik}^{miss}, \boldsymbol{\Sigma}_k^{miss}), \text{ where} \\ \boldsymbol{\mu}_{ik}^{miss} &= \\ \boldsymbol{\Sigma}_k^{miss} &= \end{aligned} \tag{5}$$

Carter – work on the above – you will need to define notation and refer readers back to equation (3) as needed. We can discuss next week if needed These results follow from conventional multivariate normal theory. An attractive feature of this imputation algorithm is that it provides more opportunities to explore the parameter space than multiple imputation [**based on summary stats right?**] and avoids multiplicative run-time scaling in m, the number of imputations **Give refs**. In Section 4, we conduct simulation studies to demonstrate that imputing the missing MSN responses improves inferences over complete case analysis.

1.4 Bayesian Inference

1.4.1 Prior Specification

We adopt a fully Bayesian inferential approach and assign prior distributions to all model parameters. Conveniently, all parameters admit conditionally conjugate priors, which greatly improves posterior computation via a data-augmented Gibbs sampler. For [**Give priors for each parameter. Be clear about the conditionally joint prior for $\boldsymbol{\beta}_k$ and $\boldsymbol{\Sigma}_k$. Where appropriate, explain advantages**]

1.4.2 Posterior Inference

The above prior specification induces closed-form full conditionals that can be efficiently updated as part of a Gibbs sampler outlined below. Additional details,

including derivations can be found in the Web Appendix. [**Think about the best way to organize this section. Maybe see my Bayesian Analysis paper for guidance? We can discuss next week.**]

Pólya–Gamma Data Augmentation for z_i. The sampler begins by updating the latent cluster indicators z_i $(i = 1, \ldots, n)$ from its multinomial logit full conditional. To facilitate sampling, we adopt an efficient data-augmentation approach introduced by Polson *et al.* (2013), which expresses the inverse-logit function as a mixture Pólya–Gamma densities. [**See my Bayesian Analysis paper for guidance on this part**].

[**I stopped here**]

$$p(\boldsymbol{\delta}_k|\mathbf{z}, \boldsymbol{\delta}_{k'\neq k}) \propto p(\boldsymbol{\delta}_k) \prod_{i=1}^{n} \pi_{ik}^{U_{ik}} (1-\pi_{ik})^{1-U_{ik}}$$

where $p(\boldsymbol{\delta}_k)$ denotes the prior distribution of $\boldsymbol{\delta}_k$, $U_{ik} = 1_{z_i=k}$ is an indicator that subject i belongs to cluster k, and π_{ik} is defined as in Section 3.4. We can rewrite π_{ik} as follows

$$\pi_{ik} = P(U_{ik} = 1) = \frac{e^{\mathbf{w}_i^T \boldsymbol{\delta}_k - c_{ik}}}{1 + e^{\mathbf{w}_i^T \boldsymbol{\delta}_k - c_{ik}}} = \frac{e^{\eta_{ik}}}{1 + e^{\eta_{ik}}}$$

where $c_{ik} = \log \sum_{k'\neq k} e^{\mathbf{w}_i^T \boldsymbol{\delta}_{k'}}$ and $\eta_{ik} = \mathbf{w}_i^T \boldsymbol{\delta}_k - c_{ik}$. We note that the sum $\sum_{k'\neq k} e^{\mathbf{w}_i^T \boldsymbol{\delta}_{k'}}$ includes the reference category, but since we fix $\boldsymbol{\delta}_K = \mathbf{0}$, we have $e^{\mathbf{w}_i^T \boldsymbol{\delta}_K} = 1$, and hence

$$c_{ik} = \log \sum_{k'\neq k} e^{\mathbf{w}_i^T \boldsymbol{\delta}_{k'}} = \log \left(1 + \sum_{k' \notin \{k, K\}} e^{\mathbf{w}_i^T \boldsymbol{\delta}_{k'}} \right)$$

We can use the quantities to re-express the full conditionals for $\boldsymbol{\delta}_k$ as

$$p(\boldsymbol{\delta}_k|\mathbf{Z}, \boldsymbol{\delta}_{k'\neq k}) \propto p(\boldsymbol{\delta}_k) \prod_{i=1}^{n} \left(\frac{e^{\eta_{ik}}}{1 + e^{\eta_{ik}}} \right)^{U_{ik}} \left(\frac{1}{1 + e^{\eta_{ik}}} \right)^{1-U_{ik}} = p(\boldsymbol{\delta}_k) \prod_{i=1}^{n} \frac{(e^{\eta_{ik}})^{U_{ik}}}{1 + e^{\eta_{ik}}}$$

which we note is essentially a logistic regression likelihood. We thus apply this Pólya–Gamma data augmentation scheme to update each $\boldsymbol{\delta}_k$ $(k = 1, ..., K-1)$ one at a time based on the binary indicators U_{ik}.

- Emphasize that PG data augmentation for the multinomial model results in a PG mixture of experts model, which is a computationally efficient way to model edge weights.