Variational Approximation for Mixtures of Linear Mixed Models

Linda S. L. Tan and David J. Nott

National University of Singapore

Department of Statistics and Applied Probability

Introduction

- Mixtures of linear mixed models (MLMMs) are useful for clustering grouped data such as gene expression profiles.
- MLMMs can be estimated by likelihood maximization through EM algorithm. A suitable number of components is determined conventionally by comparing different mixture models using penalized log-likelihood criteria such as BIC.
- We propose fitting MLMMs with variational methods which can perform parameter estimation and model selection simultaneously.

Contributions

- We describe a variational approximation for MLMMs where the variational lower bound is in closed form, allowing fast evaluation.
- Develop a variational greedy algorithm for model selection and learning of the mixture components. This approach handles algorithm initialization and returns a plausible number of mixture components automatically.
- In cases of weak identifiability of model parameters, we use hierarchical centering to reparametrize the model and show that there is a gain in efficiency in variational algorithms similar to that in MCMC algorithms.
- Prove that the rate of convergence of variational algorithms by Gaussian approximation is equal to that of the corresponding Gibbs sampler. This suggests that reparametrizations can lead to improved convergence in variational algorithms just as in MCMC algorithms.

Mixtures of linear mixed models

Suppose we observe n multivariate responses $y_i = (y_{i1}, \ldots, y_{in_i})^T$, $i = 1, \ldots, n$. Let the number of mixture components be k and z_i, $i = 1, \ldots, n$ be latent variables indicating which component the ith cluster corresponds to. Conditional on $z_i = j$,

$$y_i = X_i\beta_j + W_i a_i + V_i b_j + \epsilon_i \qquad (1)$$

where X_i, W_i and V_i are design matrices, β_j are fixed effects, a_i and b_j are random effects and ϵ_i are random errors. Note that units from the same cluster are correlated. Given $z_i = j$, $a_i \sim N(0, \sigma_{a_j}^2 I)$, $b_j \sim N(0, \sigma_{b_j}^2 I)$ and $\epsilon_i \sim N\left(0, \Sigma_{ij}\right)$ where $\Sigma_{ij} = \text{blockdiagonal}(\sigma_{j1}^2 I_{\kappa_{i1}}, \ldots, \sigma_{jg}^2 I_{\kappa_{ig}})$ with $\sum_{l=1}^{g} \kappa_{il} = n_i$ for each i. We assume

$$P(z_i = j) = p_{ij} = \frac{\exp(u_i^T \delta_j)}{\sum_{l=1}^{k} \exp(u_i^T \delta_l)} \qquad (2)$$

where $u_i = (u_{i1}, \ldots, u_{id})^T$ is a vector of covariates and $\delta_j = (\delta_{j1}, \ldots, \delta_{jd})^T$ are unknown parameters, $j = 2, \ldots, k$, with $\delta_1 = 0$. This model allows the mixture weights to vary with covariates across clusters. For Bayesian inference, we assume the priors: $\beta_j \sim N(0, \Sigma_{\beta j})$, $\delta = (\delta_2^T, \ldots, \delta_k^T)^T \sim N(0, \Sigma_\delta)$, $\sigma_{a_j}^2 \sim IG(\alpha_{a_j}, \lambda_{a_j})$, $\sigma_{b_j}^2 \sim IG(\alpha_{b_j}, \lambda_{b_j})$ and $\sigma_{jl}^2 \sim IG(\alpha_{jl}, \lambda_{jl})$, where IG denotes inverse gamma,.

Variational Approximation

Consider a variational approximation $q(\theta)$ to the joint posterior of all parameters θ. A parametric form is chosen for $q(\theta)$ and we try to minimize the Kullback-Leibler divergence between $q(\theta)$ and the true posterior. This is equivalent to maximizing the lower bound on the log marginal likelihood. We consider a variational approximation of the form

$$q(\theta) = q(\delta)\prod_{j=1}^{k}\{q(\beta_j)q(b_j)q(\sigma_{a_j}^2)q(\sigma_{b_j}^2)\}\prod_{i=1}^{n}\{q(a_i)q(z_i)\}\prod_{j=1}^{k}\prod_{l=1}^{g} q(\sigma_{jl}^2)$$

where $q(\beta_j)$ is $N(\mu_{\beta_j}^q, \Sigma_{\beta_j}^q)$, $q(a_i)$ is $N(\mu_{a_i}^q, \Sigma_{a_i}^q)$, $q(b_j)$ is $N(\mu_{b_j}^q, \Sigma_{b_j}^q)$, $q(\sigma_{a_j}^2)$ is $IG(\alpha_{a_j}^q, \lambda_{a_j}^q)$, $q(\sigma_{b_j}^2)$ is $IG(\alpha_{b_j}^q, \lambda_{b_j}^q)$, $q(\sigma_{jl}^2)$ is $IG(\alpha_{jl}^q, \lambda_{jl}^q)$, $q(\delta)$ is a delta function placing point mass of 1 on μ_δ^q, and $q(z_i = j) = q_{ij}$ where $\sum_{j=1}^{k} q_{ij} = 1$ for each i. Variational posterior for δ is relaxed to a normal distribution at convergence.

We derived a closed form of the variational lower bound and optimized it with respect to each set of variational parameters with others held fixed in a gradient ascent algorithm. This leads to an iterative scheme for obtaining the variational parameters known as **Algorithm 1**.

Partial Centering

When $X_i = W_i$ in (1), we introduce $\eta_i = \beta_j + a_i$ conditional on $z_i = j$ so that (1) is reparametrized as

$$y_i = X_i\eta_i + V_i b_j + \epsilon_i$$

and $\eta_i \sim N(\beta_j, \sigma_{a_j}^2 I_p)$ is centered about β_j. We replace $q(a_i)$ with $q(\eta_i) = N(\mu_{\eta_i}^q, \Sigma_{\eta_i}^q)$ in the variational approximation. Resulting iterative scheme is known as **Algorithm 2**.

Full centering

When $X_i = W_i = V_i$ in (1), we introduce $\rho_i = \nu_j + a_i$ and $\nu_j = \beta_j + b_j$, conditional on $z_i = j$ so that (1) is reparametrized as

$$y_i = X_i\rho_i + \epsilon_i$$

with $\rho_i \sim N(\nu_j, \sigma_{a_j}^2 I_p)$ centered about ν_j and $\nu_j \sim N(\beta_j, \sigma_{b_j}^2 I_p)$ centered about β_j. We replace $q(a_i)$ and $q(b_j)$ in the variational approximation with $q(\rho_i) = N(\mu_{\rho_i}^q, \Sigma_{\rho_i}^q)$ and $q(\nu_j) = N(\mu_{\nu_j}^q, \Sigma_{\nu_j}^q)$. Resulting iterative scheme is known as **Algorithm 3**.

Variational Greedy Algorithm (VGA)

VA refers to Variational Algorithm which can be Algorithm 1, 2 or 3. Let f_k denote the k-component mixture model and C_k the set of k components that form f_k.

1. Compute the one-component mixture model f_1 using VA.
2. Find the optimal way to split each component in the current mixture f_k. Each component goes through a trial where it is randomly partitioned into two and a **partial** VA is applied to the resulting $(k+1)$-component mixture. Only variational parameters of the two split components are updated. M trials are performed for each component and the trial with the highest lower bound yields the optimal way.
3. The components in C_k are then split in descending order according to the lower bound. Each time a component is split, a **partial** VA is applied where variational parameters of components awaiting to be split are kept fixed. A split is “successful” if the estimated log marginal likelihood increases after the split. Stop once an unsuccessful split is encountered.
4. If there are s successful splits in step 3, then a $(k+s)$-component model f_{k+s}^{temp} is obtained and we apply VA on f_{k+s}^{temp} updating all variational parameters this time to obtain f_{k+s}.
5. Repeat steps 2–4 until all splits of the current mixture model are unsuccessful.

Optional merge merges may be carried out after the VGA has converged.

Results: Clustering of time course data

Spellman *et al.* (1998) identified 800 genes that meet an objective minimum criterion for cell cycle regulation. We consider the 18 α-factor synchronization where the yeast cells were sampled at 7 min intervals for 119 mins and a subset of 612 genes with no missing data across all 18 time points. Our aim is to obtain an optimal clustering of these genes using the VGA.

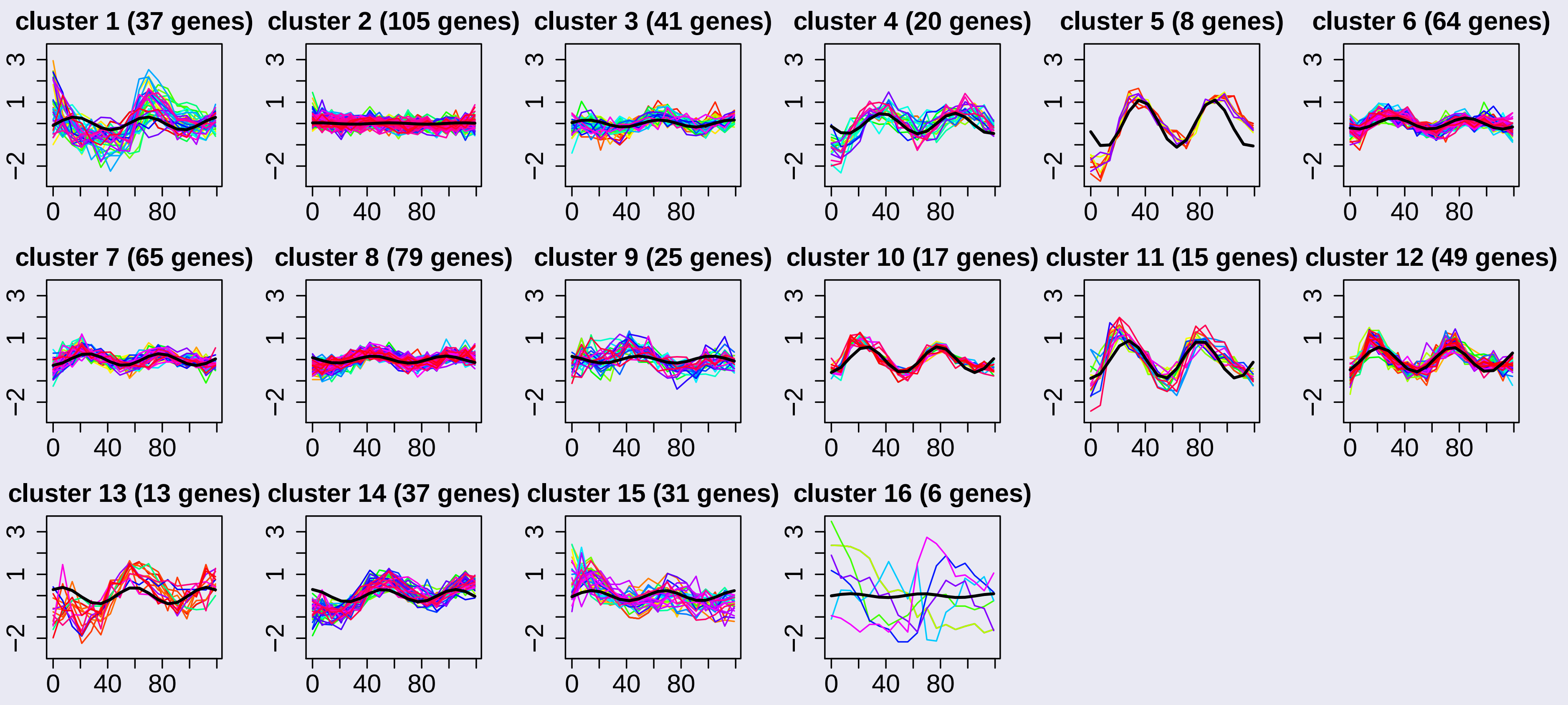
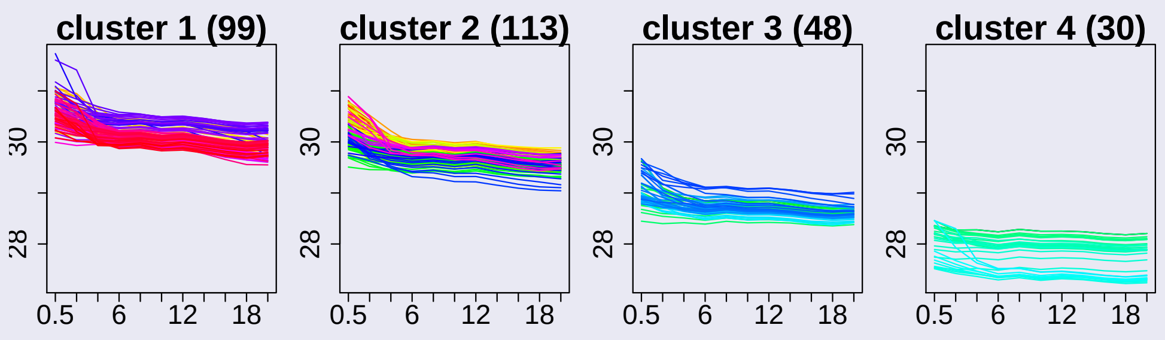


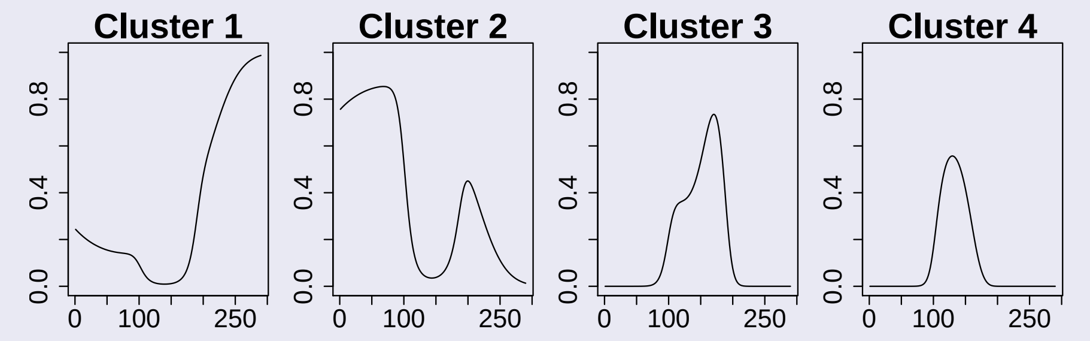
Figure: Clustering results for time course data after applying one merge move to a 17-component mixture produced by VGA using Algorithm 1. The x-axis are time points and y-axis are gene expression levels. Line in black is the posterior mean of the fixed effects.

Results: Clustering of water temperature data

We consider the daily average water temperature readings collected during 9 Sep 2010–10 Aug 2011 at Upper Peirce Reservoir, Singapore. No data were available during the periods 23–28 Dec 2010, 10–23 Feb 2010 and 14 Apr–10 May 2011. Readings were collected at eleven depths from the water surface; 0.5m, 2m, 4m, 6m, 8m, 10m, 12m, 14m, 16m, 18m and at the bottom. Using data from the remaining 290 days, we apply the VGA to obtain a clustering of this data.



(a) Clustering results for water temperature data obtained from VGA using Algorithm 3. The x-axis is depth and y-axis is temperature.



(b) Fitted probabilities from gating function (2). The x-axis are days 1 to 290 and y-axis are probabilities.

VGA with Algorithm 1 took an average of 725 s, while Algorithm 3 took 469 s. Computation time was reduced by 35% using hierarchical centering.

We thank the Singapore Delft Water Alliance for supplying the water temperature data. Linda Tan was partially supported as part of Singapore Delft Water Alliance's tropical reservoir research programme. More details can be found in the online version of this paper at http://arxiv.org/abs/1112.4675.