Journal of Statistical Research *https://doi.org/10.47302/jsr.2021550114*
2021, Vol. 55, No. 1, pp. 207-224 *ISSN 0256 - 422 X*

CLUSTERING GENE EXPRESSION TIME SERIES DATA EMBEDDED IN A NON-PARAMETRIC SETUP

MUKTI KHETAN[†]

Department of Mathematics, Indian Institute of Technology Bombay, Mumbai 400 076, India.

Email: mukti.khetan11@gmail.com

SAVITA PAREEK[†]

Department of Mathematics, Indian Institute of Technology Bombay, Mumbai 400 076, India.

Email: savita@math.iitb.ac.in

SIULI MUKHOPADHYAY

Department of Mathematics, Indian Institute of Technology Bombay, Mumbai 400 076, India.

Email: siuli@math.iitb.ac.in

KALYAN DAS[⋆]

Department of Mathematics, Indian Institute of Technology Bombay, Mumbai 400 076, India.

Email: kalyan@math.iitb.ac.in

SUMMARY

A clustering methodology for time series data is proposed. The idea has been cropped up when a subset of gene expression dataset is used to build up the system model by compressing the information through clustering and then by tracing out inherent patterns in the data. A linear mixed model is considered that accommodates time dependent components. The temporal effects are modelled through an autoregressive process that arises in the dispersion of the random component. The joint distribution of coefficients in the time dependent quadratic function and the random effects are embedded within a non-parametric prior (Dirichlet process prior). Such a non-parametric prior induces clustering in the data. Monte Carlo EM (MCEM) based technique has been considered for estimating the parameters. The best cluster is selected through some heterogeneity measures. A rigorous simulation study has been carried out prior to analysis of a gene expression time series data.

Keywords and phrases: Dirichlet Process; Monte Carlo EM algorithm; Mixed effect model; Autoregressive process

⋆ Corresponding author; † Authors contributed equally to this work

1 Introduction

Time-series clustering has been of growing interest because of its abound applications in many areas such as energy, weather, retail, stock, financial, personalized drug design (Pirim et al., 2012) and cancer sub-type identification (de Souto et al., 2008) using gene expression data. The goal in such circumstances is to identify the structure in an unlabelled data set by objectively dividing data points into groups (called 'clusters') such that observations within the same cluster tend to be more similar (according to pre-specified criteria) than those in different clusters (Wu et al., 2008). With the increasing prevalence of time series data, time series clustering has been gaining much attention over the past decade to identify previously unknown trends (Aghabozorgi et al., 2015; Begum et al., 2015; Du et al., 2019; Paparrizos and Gravano, 2015). However, the evaluation of clustering algorithms is inherently challenging because these statistical algorithms are, by design, exploratory in nature.

Clustering gene expression time series is an application that has attracted a lot of interest. Analysis of time series clusters is an essential tool in exploring and understanding gene networks, whilst incorporating knowledge of the time series into the model can improve the method's ability to discern clusters. In time series analysis of gene expression data, the aim is to stratify the genes according to their differential temporal behaviors. Genes with similar expression patterns may reflect functional responses of biological relevance. However, these measurements come with intrinsic noise, which makes their time series clustering a problematic task.

To develop a gene expression model-based time series that accounts for clustering, we propose a random-effects mixture model coupled with a Dirichlet-process (DP) prior. The random-effects would capture the high level of noise in the data that arises from several sources. Under the random-effects model, we use the full data, rather than reducing the data to the means across replicates, which may not be accurate with this level of noise. Moreover, we do not make many assumptions about the underlying biological process, which is still mostly unknown. Novel patterns detected this way are unlikely to be the result of potentially inappropriate assumptions. The advantage of considering a Dirichlet-process prior enables us to estimate the number of clusters directly from the data. Several authors (Green and Richardson, 2001; Medvedovic and Sivaganesan, 2002; Medvedovic et al., 2004; Celeux et al., 2005; Beal and Krishnamurthy, 2006; Fraley and Raftery, 2007; Booth et al., 2008; Rasmussen et al., 2009; McNicholas and Murphy, 2010; Cooke et al., 2011; Kyung, 2015) have considered different approaches for model-based clustering on time series data using mixture of Gaussian distributions.

In this study, we consider a linear mixed model that accommodates subject specific variations and time dependent components. The temporal effects are modeled with a first-order autoregressive process through the random effects dispersion. The joint distribution of some coefficients and the random effects of an entire-time series are embedded within a non-parametric prior (Dirichlet process prior). Such a non-parametric prior induces clustering in the data. Our approach is pseudo-Bayesian, in the sense that the MCEM based technique has been adopted in model estimation under the assumption of the DP process (an infinite mixture of Gaussian components) on the random component and a quadratic trend component in the mixed model. The best cluster is selected following a heterogeneity measure as proposed by Dahl (2009).

We use the gene expression study from Mehra et al. (2006) for illustration purpose. Streptomyces

coelicolor (a bacteria) is the organism of interest in their study. The significance of this bacteria is that it is used for antibiotic production. In this organism, various genes are involved in antibiotic production.

Microarray-based transcription profiling performed over time gives the information required to capture transcription dynamics. It is interesting to find out which of the genes are involved in antibiotic production. Mehra et al. (2006) performed a k-means clustering technique using Spotfire to identify the cluster sequence. However, the k-means algorithm ignores the behaviour of genes over time. Our proposed algorithm using Dirichlet process random-effects estimates the parameters and detects the cluster sequence, average number of clusters taking into account the dependence over time.

Another drawback of the k-means clustering is that the number of cluster, 'c', is fixed and difficult to predict. Also, each observation belongs to the cluster with the nearest mean along with ignoring the time dependency. To overcome these limitations, our proposed model does clustering with the assumption of the distribution of 'c' and utilizes the AR(1) model and quadratic trend simultaneously for modeling the dependence over time.

The structure of the article is as follows: we motivate our model, starting with a dynamic linear model and depict a Bayesian non-parametric mixture framework for clustering. Section 2 deals with the methodology where MCEM based technique has been adopted. Section 3 performs the detailed simulation study on the proposed algorithm. Section 4 covers the analysis of gene expression data and its importance. Section 5 discusses the interpretation of the results and conclusions; the paper ends with Section 5.

2 Model and Methodology

Let y_{it} denote the t^{th} observation from the i^{th} subject where $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, T$. The model is expressed as

$$y_{it} = \beta_0 + \sum_{l=1}^{p} \beta_l x_{lit} + f_i(t) + \gamma_{it} + \varepsilon_{it}. \tag{2.1}$$

The terms in the model are β_0: an intercept; β_l: a fixed effect associated with the predictor variable x_l, $l = 1, 2, \ldots, p$; $f_i(t)$: a quadratic function of time, $f_i(t) = \alpha_{1i}t + \alpha_{2i}t^2$; γ_{it}: an autoregressive term, $\gamma_{it} = \rho\gamma_{i\overline{t-1}} + e_{it}$; ε_{it}: a random error following $N\left(0, \sigma_e^2\right)$.

Equation (2.1) can be rewritten in matrix form as,

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i; \quad i = 1, 2, \ldots, n, \tag{2.2}$$

where $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iT})'_{T\times 1}$; $\boldsymbol{X}_i = \left[\mathbf{1}_T \mid \boldsymbol{x}_1 \mid \cdots \mid \boldsymbol{x}_p\right]_{T\times\overline{p+1}}$, for $l = 1, 2, \ldots, p$, $\boldsymbol{x}_l = (x_{l1}, x_{l2}, \ldots, x_{lT})'$; $\boldsymbol{Z}_i = \left[\boldsymbol{Z}_{1i} \mid \boldsymbol{Z}_{2i}\right]_{T\times\overline{T+2}}$, where $\boldsymbol{Z}_{1i} = \left[\boldsymbol{a} \mid \boldsymbol{a} \odot \boldsymbol{a}\right]$, and $\boldsymbol{Z}_{2i} = \boldsymbol{I}_T$, for $\boldsymbol{a} = (1, 2, \ldots, T)'$; $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'_{\overline{p+1}\times 1}$; $\boldsymbol{b}_i = (\boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i)'_{\overline{T+2}\times 1}$, where $\boldsymbol{\alpha}_i = (\alpha_{1i}, \alpha_{2i})'_{2\times 1}$ and $\boldsymbol{\gamma}_i = (\gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{iT})'_{T\times 1}$. Here $\boldsymbol{b}_i \sim G$, where we assume $G \overset{d}{=} DP(\nu, G_0)$, and $G_0 \sim$

$N_{T+2}(\mathbf{0}, \mathbf{\Sigma}_b)$, where DP indicates the Dirichlet process, ν is a positive scaling or precision parameter, G_0 is a base distribution, and

$$\mathbf{\Sigma}_b = \text{diag}(\mathbf{\Sigma}_\alpha, \mathbf{\Sigma}_\gamma), \quad \text{where} \quad \mathbf{\Sigma}_\alpha = \text{diag}(\sigma_1^2, \sigma_2^2), \quad \mathbf{\Sigma}_\gamma = \sigma_g^2 \boldsymbol{R}$$
$$\boldsymbol{R} = \frac{1}{(1-\rho^2)}\left[(\rho^{|j-i|})_{ij}\right]_{T\times T}, \quad \boldsymbol{\varepsilon}_i = \left(\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{iT}\right)', \quad \boldsymbol{\varepsilon}_i \sim N\left(0, \sigma_e^2 \boldsymbol{I}_T\right)$$

and, $\sigma_1^2, \sigma_2^2, \sigma_g^2$ are the variance components involved in the covariance matrix of base measure.

According to Maceachern and Müller (1998), we can represent the $\boldsymbol{b}_i$ in terms of $\boldsymbol{s}$, $\boldsymbol{\phi}$. Here s_i is a latent variable which represents the cluster number of the i$^{\text{th}}$ observation. Suppose k_i denotes the number of clusters among $b_1, b_2, \ldots, b_{i-1}$ and η_{ih} is the number of s_h, $h < i$, such that $s_h = h$, using the above information, the conditional probabilities $s_i|s_1, s_2, \ldots, s_{i-1}, \nu$ can then be explained as the i$^{\text{th}}$ observation, either allocated at the existing h^{th} cluster or the new $(k_i+1)^{\text{th}}$ cluster given $s_1, s_2, \ldots, s_{i-1}$ with probabilities $\eta_{ih}/(\nu+i-1); h = 1, 2, \ldots, k_i$ and $\nu/(\nu+i-1)$ respectively.

Further, we assume that $\boldsymbol{\phi}_h$ represents the distribution of the h^{th} cluster. So, when $s_i = h, \boldsymbol{b}_i = \boldsymbol{\phi}_h \; \forall i = 1, 2, \ldots, n; h = 1, 2, \ldots, k$ (k is the number of distinct s_i). Therefore, the joint distribution of $\boldsymbol{s}$, $\boldsymbol{\phi}$ induces the distribution of $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ through $\boldsymbol{b}_i = \boldsymbol{\phi}_{s_i}$.

The precision parameter ν has an important role in clustering the observations. A larger value of ν specified that more clusters will be formed. The average number of distinct clusters are found from the relation $E(k) = \sum_{i=1}^{n} \nu/(\nu+i-1)$ (Charles E. Antoniak, 1986). Also, determines which $\boldsymbol{y}_i$ should be combined together when considered in terms of the distribution of $\boldsymbol{b}_i$. Under these assumptions, the conditional model takes the form,

$$\boldsymbol{y}_i|\boldsymbol{\phi}, \boldsymbol{s} \sim N_T\left(\boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}, \sigma_e^2 \boldsymbol{I}_T\right), \quad \boldsymbol{\phi}_{si}|\boldsymbol{s} \sim G_0. \tag{2.3}$$

Remark. *If replicates were included at each time point then the model would be rewritten as:*

Let y_{itk} denote the k^{th} replicate at t^{th} time point of i^{th} subject where $i = 1, 2, \ldots, n$; $t = 1, 2, \ldots, T$ and $k = 1, 2, \ldots, r$. The model is expressed in eq. (2.4) *as*

$$y_{itk} = \beta_0 + \sum_{l=1}^{p} \beta_l x_{litk} + f_i(t) + \gamma_{itk} + \varepsilon_{itk}. \tag{2.4}$$

However, $x_{litk} = x_{lit}, \forall k$ and $\gamma_{itk} = \rho\gamma_{i\overline{t-1}k} + e_{itk}$, and ε_{itk}: a random error following $N\left(0, \sigma_e^2\right)$. Equation (2.4) *can be rewritten as before in matrix form as*

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i; \quad i = 1, 2, \ldots, n,$$

where $\boldsymbol{y}_i = (y_{i11}, \ldots, y_{i1r}, \ldots, y_{iT1}, \ldots, y_{iTr})'_{rT\times 1}$ and for $l = 1, 2, \ldots, p$, $\boldsymbol{x}_l = (x_{l1}, x_{l2}, \ldots, x_{lT})'$, therefore we have

$$\boldsymbol{X}_i = \left[\mathbf{1}_T \otimes \mathbf{1}_r \mid \boldsymbol{x}_1 \otimes \mathbf{1}_r \mid \cdots \mid \boldsymbol{x}_p \otimes \mathbf{1}_r\right]_{rT\times\overline{p+1}}.$$

Let $\boldsymbol{a} = (1, 2, \ldots, T)'$. Then

$$\boldsymbol{Z}_i = \left[\boldsymbol{Z}_{1i} \mid \boldsymbol{Z}_{2i}\right]_{rT\times\overline{T+2}}, \quad \boldsymbol{Z}_{1i} = \left[\boldsymbol{a} \otimes \mathbf{1}_r \mid (\boldsymbol{a} \odot \boldsymbol{a}) \otimes \mathbf{1}_r\right], \quad \boldsymbol{Z}_{2i} = \boldsymbol{I}_T \otimes \mathbf{1}_r.$$

The rest of the things will remain unaltered.

2.1 Maximum likelihood estimation

To obtain the estimate of the parameters $(\boldsymbol{\beta}, \sigma_e^2, \sigma_1^2, \sigma_2^2, \sigma_g^2, \rho, \nu)$, we maximize the marginal log likelihood with respect to the parameters. The expression of marginal log likelihood is given below.

$$l = \log\Big(\sum_{\boldsymbol{s}} \int_{\boldsymbol{\phi}} f\big(\boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s}; \boldsymbol{\xi}\big) d\boldsymbol{\phi}\Big), \tag{2.5}$$

where $f(\boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s}; \boldsymbol{\xi})$ is the the joint distribution of $(\boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s})$ and can be written as

$$f(\boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s}; \boldsymbol{\xi}) \propto \Big(\prod_{i=1}^{n} f(\boldsymbol{y}_i|\boldsymbol{\phi}, \boldsymbol{s}) f(\boldsymbol{\phi}_{s_i}|\boldsymbol{s})\Big)\, \mathrm{p}(\boldsymbol{s}|\nu),$$

where $\boldsymbol{\xi} = (\boldsymbol{\beta}, \sigma_e^2, \boldsymbol{\tau}, \nu),\ \boldsymbol{\tau} = (\sigma_1^2, \sigma_2^2, \sigma_g^2, \rho)$ and $\mathrm{p}(\boldsymbol{s}|\nu)$ is obtained by multiplying the conditional probabilities, discussed in Section 2. However, the exact computation of the integral and sum in eq. (2.5) is intractable. The MCEM algorithm (Dempster et al., 1977) is a popular iterative algorithm to solve such a problem. However, there remains a difficulty that the joint posterior distribution of $(\boldsymbol{\phi}, \boldsymbol{s})$ involves high dimensional integrals. Next, we illustrate how an MCEM type algorithm can be used for ML estimation in the Dirichlet process linear mixed model.

2.2 An MCEM-type algorithm

To perform the two steps of the EM algorithm for maximizing the marginal log-likelihood given in eq. (2.5), we need the following result. The proof of the result is given in Appendix A.1.

Proposition 1. *Let $\boldsymbol{y}$ denote the observed data, $\boldsymbol{u}$ denote the unobserved data, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ denote the parameter vector, $(\boldsymbol{y}, \boldsymbol{u})$ is the complete data, l is the marginal log likelihood and l_c is the complete data log likelihood. Then the score function based on marginal log likelihood $(\boldsymbol{S}(\boldsymbol{\theta}))$ is same as conditional expected value of score function based on complete data log likelihood $(\boldsymbol{S}_c(\boldsymbol{\theta}))$ given the observed data, i.e.,*

$$\boldsymbol{S}(\boldsymbol{\theta}) = E_{\boldsymbol{u}|\boldsymbol{y}}\big(\boldsymbol{S}_c(\boldsymbol{\theta})\big).$$

In our problem, the score function of marginal log likelihood is

$$\frac{\partial Q}{\partial \boldsymbol{\xi}} = E_{\boldsymbol{\phi}, \boldsymbol{s}|\boldsymbol{y}, \boldsymbol{\xi}^{(m)}}\Big(\frac{\partial l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s})}{\partial \boldsymbol{\xi}}\Big), \tag{2.6}$$

where $l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s})$ is the complete data log likelihood that can be written as

$$\begin{aligned} l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s}) &= \log\Big(\prod_{i=1}^{n} f(\boldsymbol{y}_i|\boldsymbol{\phi}, \boldsymbol{s}) f(\boldsymbol{\phi}_{s_i}|\boldsymbol{s})\Big) + \log \mathrm{p}(\boldsymbol{s}|\nu) \\ &= \sum_{i=1}^{n} \log f(\boldsymbol{y}_i|\boldsymbol{\phi}, \boldsymbol{s}) + \sum_{i=1}^{n} \log f(\boldsymbol{\phi}_{s_i}|\boldsymbol{s}) + \log \mathrm{p}(\boldsymbol{s}|\nu). \end{aligned}$$

Using eq. (2.3) $l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s})$ can be expressed as,

$$l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{s}) = \sum_{i=1}^{n} \Bigg(-\frac{T}{2}\log\sigma_e^2 - \frac{1}{2\sigma_e^2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i})^{'}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}) - \frac{1}{2}\log|\boldsymbol{\Sigma}_b| - \frac{1}{2}\boldsymbol{\phi}_{s_i}^{'}\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\phi}_{s_i}\Bigg) + \log\Bigg(\frac{\nu^k(\nu-1)!}{(\nu+n-1)!k!}\prod_{h=1}^{k}(\eta_h-1)!\Bigg), \tag{2.7}$$

where η_h is the total number of observations in r^{th} cluster. Now, we calculate the score equations based on the complete data log likelihood by differentiating eq. (2.7) with respect to parameters. Specifically,

$$\boldsymbol{S}_c(\boldsymbol{\xi}) = \Big(\boldsymbol{S}_c(\boldsymbol{\beta}), S_c(\sigma_e^2), S_c(\sigma_1^2), S_c(\sigma_2^2), S_c(\sigma_g^2), S_c(\rho), S_c(\nu)\Big).$$

(i) The expression for $\big(\boldsymbol{S}_c(\boldsymbol{\beta}), S_c(\sigma_e^2)\big)$ is obtained by differentiating $\sum_{i=1}^{n}\big(\log f(\boldsymbol{y}_i|\boldsymbol{\phi}, \boldsymbol{s})\big)$ with respect to $\boldsymbol{\beta}$, σ_e^2 respectively.

$$\boldsymbol{S}_c(\boldsymbol{\beta}) = \sum_{i=1}^{n}\frac{1}{\sigma_e^2}\boldsymbol{X}_i^{'}\big(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\big),$$
$$S_c(\sigma_e^2) = \sum_{i=1}^{n}\Bigg(-\frac{T}{2\sigma_e^2} + \frac{1}{2\sigma_e^4}\big(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\big)^{'}\big(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\big)\Bigg).$$

(ii) Expression of $S_c(\nu)$ is obtained by differentiating $\log \mathrm{p}(\boldsymbol{s}|\nu)$ with respect to ν.

$$S_c(\nu) = \frac{k}{\nu} - \sum_{i=1}^{n}\frac{1}{\nu+i-1}.$$

(iii) The expression for $\big(S_c(\sigma_1^2), S_c(\sigma_2^2), S_c(\sigma_g^2), S_c(\rho)\big)$ is obtained by differentiating $\sum_{i=1}^{n}\big(\log f(\boldsymbol{\phi}_{s_i}|\boldsymbol{s})\big)$ with respect to σ_1^2, σ_2^2, σ_g^2, ρ respectively.

$$S_c(\sigma_j^2) = \sum_{i=1}^{n}\Bigg(-\frac{1}{2\sigma_j^2} + \frac{1}{2\sigma_j^4}\phi_{s_i[j]}^2\Bigg),\ j = 1, 2,$$
$$S_c(\sigma_g^2) = \sum_{i=1}^{n}\Bigg(-\frac{T}{2\sigma_g^2} + \frac{1}{2\sigma_g^4}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\Bigg),\ \boldsymbol{q} = (1, 2),$$
$$S_c(\rho) = \sum_{i=1}^{n}\Bigg(-\frac{1}{2}tr(\boldsymbol{R}^{-1}\boldsymbol{R}^{*}) + \frac{1}{2\sigma_g^2}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{-1}\boldsymbol{R}^{*}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\Bigg),\ \boldsymbol{q} = (1, 2),$$

where $\boldsymbol{\phi}_{s_i} = \big(\phi_{s_i[1]}, \phi_{s_i[2]}, \ldots, \phi_{s_i[T+2]}\big)$, $\boldsymbol{\phi}_{s_i[-j]}$ contains all elements of $\boldsymbol{\phi}_{s_i}$ except the j^{th} element and $\boldsymbol{R}^{*} = \frac{\partial \boldsymbol{R}}{\partial \rho}$.

2.2.1 Maximization step

In eq. (2.6) analytical evaluation of conditional expectation is difficult, so we consider Monte Carlo Markov Chain (MCMC) methods to draw random samples from conditional distributions. The expectations are then approximated by Monte Carlo sums (Section 2.2.2). Furthermore, score equations are solved iteratively by one step Newton Raphson method to get the updated estimates of the parameter. For the parameter vector $\boldsymbol{\xi}$, the updated estimates at $(m+1)^{\text{th}}$ iteration is given by

$$\boldsymbol{\xi}^{(m+1)} = \boldsymbol{\xi}^{(m)} + \boldsymbol{I}^{-1(m)} \frac{\partial Q}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}^{(m)}}, \tag{2.8}$$

where information matrix,

$$\boldsymbol{I} = -\frac{\partial^2 Q}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}'} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\bigg(-\frac{\partial^2 l_c(\boldsymbol{\xi};\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{s})}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}'}\bigg).$$

Computation of elements of $\boldsymbol{I}$ matrix is given in Appendix A.2.

2.2.2 Expectation step

Exact computation of the conditional expectation in eq. (2.6) is difficult, so we use Monte Carlo method to approximate the expectation. We adopt the 'no gaps' algorithm given by Maceachern and Müller (1998) to generate the observations from the posterior $\boldsymbol{\phi}, \boldsymbol{s}|\boldsymbol{y}$. In this algorithm, s_i is restricted to cover the set of integers from 1 to k. This algorithm works with both conjugate and non-conjugate prior.

Following Maceachern and Müller (1998), the conditional posterior distribution of s_i can be defined as

$$\Pr\big(s_i = h|\boldsymbol{s}_{-i}, \boldsymbol{\phi}, \boldsymbol{y}\big) \propto \Pr\big(s_i = h|\boldsymbol{s}_{-i}, \boldsymbol{\phi}\big) f\big(\boldsymbol{y}_i|\boldsymbol{\phi}_{s_i}\big), \tag{2.9}$$

where

$$\begin{aligned} \Pr\left(s_i = h|\boldsymbol{s}_{-i}, \boldsymbol{\phi}\right) &\propto \eta_{-ih}, \quad h = 1, \ldots, \bar{k}, \\ \Pr\big(s_i = \bar{k}+1|\boldsymbol{s}_{-i}, \boldsymbol{\phi}\big) &\propto \frac{\nu}{(\bar{k}+1)}, \end{aligned}$$

where $\bar{k}$ is the total number of distinct components in $\boldsymbol{s}_{-i}$ and $\boldsymbol{s}_{-i}$ contain all the elements except s_i^{th} element, moreover $\boldsymbol{s}_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. η_{-ih} is the total number of observation in the h^{th} cluster after removing the i^{th} observation or in other words, η_{-ih} is the number of $s_{i'}$ such that $s_{i'} = h$ for $i \neq i' = 1, 2, \ldots, n$.

The conditional posterior distribution of $\boldsymbol{\phi}$ can be defined as

$$f\left(\boldsymbol{\phi}_h|\boldsymbol{s}, \boldsymbol{y}\right) \propto g_0\left(\boldsymbol{\phi}_h\right); \quad \text{h} = \text{k}+1, \ldots, n, \tag{2.10}$$

$$f\left(\boldsymbol{\phi}_h|\boldsymbol{s}, \boldsymbol{y}\right) \propto \Bigg(\prod_{\substack{i=1 \\ \{s_i = h\}}}^{n} f\left(\boldsymbol{y}_i|\boldsymbol{\phi}_{s_i}\right) \Bigg) g_0\left(\boldsymbol{\phi}_{s_i}\right); \quad \text{h} = 1, 2, \ldots, \text{k}, \tag{2.11}$$

where $g_0(.|\boldsymbol{\tau})$ is the density function of base distribution G_0. As the computation of conditional distributions $(\boldsymbol{\alpha}_h|\boldsymbol{\gamma}_h, \boldsymbol{s}, \boldsymbol{y})$, $(\boldsymbol{\gamma}_h|\boldsymbol{\alpha}_h, \boldsymbol{s}, \boldsymbol{y})$ given in eq. (2.12) is straightforward. Therefore, to generate sample from eq. (2.10), we utilise the Gibbs sampling approach.

$$\boldsymbol{\alpha}_h|\boldsymbol{s}, \boldsymbol{\gamma}_h, \boldsymbol{y} \sim N_2\big(\boldsymbol{\mu}_{\alpha_h}, \boldsymbol{\Omega}_{\alpha_h}\big), \quad \boldsymbol{\gamma}_h|\boldsymbol{\alpha}_h, \boldsymbol{s}, \boldsymbol{y} \sim N_T\big(\boldsymbol{\mu}_{\gamma_h}, \boldsymbol{\Omega}_{\gamma_h}\big), \tag{2.12}$$

where

$$\boldsymbol{\mu}_{\alpha_h} = \boldsymbol{\Omega}_{\alpha_h} \sum_{\substack{i=1 \\ \{s_i=h\}}}^{n} \big(\sigma_e^2\boldsymbol{I}\big)^{-1}\boldsymbol{Z}_{1i}^{'}\big(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_{2i}\boldsymbol{\gamma}_h\big), \quad \boldsymbol{\Omega}_{\alpha_h} = \Bigg[\sum_{\substack{i=1 \\ \{s_i=h\}}}^{n} \big(\sigma_e^2\boldsymbol{I}\big)^{-1}\boldsymbol{Z}_{1i}^{'}\boldsymbol{Z}_{1i} + \boldsymbol{\Sigma}_{\alpha}^{-1}\Bigg]^{-1},$$

$$\boldsymbol{\mu}_{\gamma_h} = \boldsymbol{\Omega}_{\gamma_h} \sum_{\substack{i=1 \\ \{s_i=h\}}}^{n} \big(\sigma_e^2\boldsymbol{I}\big)^{-1}\boldsymbol{Z}_{2i}^{'}\big(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_{1i}\boldsymbol{\alpha}_h\big), \quad \boldsymbol{\Omega}_{\gamma_h} = \Bigg[\sum_{\substack{i=1 \\ \{s_i=h\}}}^{n} \big(\sigma_e^2\boldsymbol{I}\big)^{-1}\boldsymbol{Z}_{2i}^{'}\boldsymbol{Z}_{2i} + \boldsymbol{\Sigma}_{\gamma}^{-1}\Bigg]^{-1}.$$

Ultimately the process to generate samples from joint posterior distribution $\boldsymbol{\phi}, \boldsymbol{s}|\boldsymbol{y}$ is given below:

(i) Let the current state of Markov chain consists of $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$ and $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n)$.

(ii) For $i = 1, 2, \ldots, n$, η_{s_i} is the number of $s_{i'}$ such that $s_{i'} = s_i$, $i^{'} = 1, 2, \ldots, n$ or η_{s_i} denote the size of $s_i{}^{\text{th}}$ cluster.

  (a) if η_{s_i} is not a singleton set then re-sample s_i with probabilities given in eq. (2.9).

  (b) if η_{s_i} is a singleton set then leave s_i unchanged with probability $(k-1)/k$. Else we relabel the cluster sequence $\boldsymbol{s}$ such that $s_i = k$ and then re-sample s_i with probabilities given in eq. (2.9).

(iii) From steps (a) and (b) we get the new sequence say $\boldsymbol{s}^{'} = \big(s_1^{'}, s_2^{'}, \ldots, s_n^{'}\big)$. Next we draw a new value of $\boldsymbol{\phi}$ say $\boldsymbol{\phi}^{'} = (\boldsymbol{\phi}_1^{'}, \boldsymbol{\phi}_2^{'}, \ldots, \boldsymbol{\phi}_n^{'})$ from the posterior distribution given in eq. (2.10) eq. (2.11).

We repeat steps (ii), (iii) until we get a sample of size m_0. Finally, let $(\boldsymbol{\phi}^{(1)}, \boldsymbol{s}^{(1)}), (\boldsymbol{\phi}^{(2)}, \boldsymbol{s}^{(2)}), \ldots, (\boldsymbol{\phi}^{(m_0)}, \boldsymbol{s}^{(m_0)})$ be a sample of size m_0 from joint posterior distribution of $(\boldsymbol{\phi}, \boldsymbol{s})$. Then,

$$\boldsymbol{S}(\boldsymbol{\xi}) = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\boldsymbol{S}_c(\boldsymbol{\xi})\right) \approx \frac{1}{m_0}\sum_{m=1}^{m_0}\boldsymbol{S}_c(\boldsymbol{\xi}; \boldsymbol{\phi}^{(m)}, \boldsymbol{s}^{(m)}),$$

$$\boldsymbol{I} = -\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\xi}} \approx \frac{1}{m_0}\sum_{m=1}^{m_0}\left(-\frac{\partial^2 l_c(\boldsymbol{\xi}; \boldsymbol{y}, \boldsymbol{\phi}^{(m)}, \boldsymbol{s}^{(m)})}{\partial \boldsymbol{\xi}\partial \boldsymbol{\xi}^{'}}\right).$$

For example $\boldsymbol{S}_{\boldsymbol{\beta}}$ can be computed as,

$$\boldsymbol{S}_{\boldsymbol{\beta}} \approx \frac{1}{m_0}\sum_{m=1}^{m_0}\sum_{i=1}^{n}\frac{1}{\sigma_e^2}\boldsymbol{X}_i^{'}\left(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}^{(m)}\right).$$

The above E and $M-$ steps are repeated until convergence is achieved.

2.3 Obtaining the cluster sequence

At each iteration, the Gibbs sampler gives an implicit clustering as $\boldsymbol{\phi}^{'} = (\boldsymbol{\phi}_1^{'}, \boldsymbol{\phi}_2^{'}, \ldots, \boldsymbol{\phi}_n^{'})$, where each $\boldsymbol{\phi}_{s_i}$ indicates a corresponding $\boldsymbol{y}_i$ (step (iii), Section 2.2.2). Following Medvedovic and Sivaganesan (2002) and Dahl (2009), for a given cluster sequence we compute the Heterogeneity measure (HM), i.e.

$$HM(s_1, s_2, \ldots, s_h, \ldots, s_k) = \sum_{h=1}^{k} \frac{2}{\eta_h - 1} \sum_{i<i^{'} \in s_h} \left|\left|(\boldsymbol{y}_i - \boldsymbol{y}_{i^{'}})\right|\right|^2.$$

We select the optimum cluster sequence to be the one with the minimum value of HM.

3 Simulation Study

In this section we portray a simulation study to asses the performance of the proposed model. For the simulation we generated a sample of size one hundred from the model

$$y_{it} = \beta_0 + \beta_1 x_{1it} + f_i(t) + \gamma_{it} + \varepsilon_{it}, \tag{3.1}$$

where for $i = 1, 2, \ldots, 20$, $t = 1, 2, 3$, y_{it} denotes the t^{th} observation from the i^{th} subject. β_0 and β_1 are fixed effects, $f_i(t) = \alpha_{1i}t + \alpha_{2i}t^2$; $\gamma_{it} = 0.427\gamma_{\overline{it-1}} + e_{it}$.

In matrix notations,

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i; \quad i = 1, 2, \ldots, 20, \tag{3.2}$$

where, for $i = 1, 2, \ldots, 20$, $\boldsymbol{\varepsilon}_i \sim N_2(\mathbf{0}, 0.48^2\boldsymbol{I})$, $\boldsymbol{b}_i \stackrel{d}{=} DP(0.567, G_0)$, $G_0 \sim N_5(\mathbf{0}, \boldsymbol{\Sigma}_b)$, $\boldsymbol{\Sigma}_b = \text{diag}(0.57^2, 0.65^2, \boldsymbol{\Sigma}_\gamma)$, and

$$\boldsymbol{\Sigma}_\gamma = \begin{bmatrix} 0.5 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.5 \end{bmatrix}.$$

For the i^{th} subject, the predictor variable x_1 is generated from a gamma distribution $(Gamma(2, 1))$ and we assume $x_{1i1} = x_{1i2} = x_{1i3}$. Further, we take the parameter ρ to be 0.427 and consider it as a known constant throughout. Moreover, $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ matrices are constructed similarly as for Section 2 and true parameter values are in the first column of Table 1.

MCEM algorithm (Section 2.2) is used for parameter estimation, and standard errors are obtained from the information matrix. In order to check the convergence of the algorithm, estimated values of parameters at each M-step iteration are assessed. We have applied our algorithm using E-step samples of size 1000, 2000, respectively. Figure 1(a,b) depicts the plot of parameter values against M-step iteration number, Figure 1(c,d) exhibits fluctuation in the parameter values using box plots for the E-step samples of size 1000, 2000, respectively. Figure 1(a, b) shows that the M-step typically converges after 20 iterations. Height of the box plots that renders the fluctuating range for

Table 1: Simulation results based on one hundred samples

Parameter	True value	Estimate	SE	Bias	Length of CI
β_0	0.80	0.86	0.13	0.08	0.49
β_1	1.30	1.38	0.04	0.06	0.17
σ_e^2	0.23	0.24	0.01	0.02	0.04
σ_1^2	0.32	0.32	0.23	0.01	0.91
σ_2^2	0.42	0.37	0.28	0.14	1.09
σ_g^2	0.43	0.47	0.14	0.08	0.54

parameter values also decreases with an increase in the E-step sample size. Hence, it establishes the empirical evidence of the stability of our estimates.

The simulations were repeated 100 times. We consider the E-step sample size to be 1000 and to generate sample from conditional posterior distribution (eq. (2.10), eq. (2.11)) 500 Gibbs samples were used with a burn-in of 20%. Table 1 shows the average simulation results in terms of parameter estimates, SEs, relative biases, and lengths of confidence intervals. Absolute average relative bias was computed as $|100^{-1}\sum_{w=1}^{100}(\hat{\beta}_{uw}-\beta_u)/\beta_u|$, where $\hat{\beta}_{uw}$ is the u^{th} component of $\hat{\boldsymbol{\beta}}$ for the w^{th} simulation and β_u is the true value.

The results of Table 1 show that all the estimated values and population parameters are close. The standard errors lie in the interval (0.01, 0.28); the absolute value of relative bias varies from 2% to 14%. Also, the average length of confidence interval < 1.1 for all the parameters. The estimated value of the precision parameter ν is 0.62 with sample standard deviation of 0.06. Also, the average number of distinct clusters using estimated ν is 2.8, which is close to the true value 2.6. Moreover, we also calculated optimal number of clusters using Silhouette method, the average of optimal clusters is 2.1. This indicates that our procedure for choosing optimal number of clusters is more accurate.

Further, at the convergence of the EM algorithm, we assess the cluster sequences generated and obtain the optimum cluster sequence using the method discussed in Section 2.3. We have run 100 simulations with true cluster sequence having two groups and noted down the optimal cluster sequence in each simulation. The true cluster sequence has 65% of observations in cluster 1, and the remaining 35% observations are in cluster 2. The estimated cluster sequence has 50%- 65% observations in group 1, and the remaining are in group 2. Out of one hundred optimum cluster sequences, we get 60 percent of sequences with two or three as number of clusters. Moreover, table 2 exhibits the proportion of cluster sequence with 2 or 3 clusters.

In order to compare the proposed method with the model based clustering using 'mclust' function of R. We compute Fowlkes-Mallows (FM) index (Fowlkes and Mallows, 1983), variation of the information (VI) index (Arabie and Boorman, 1973) and the optimal number of clusters, given in Table 3. The FM index is used to determine the similarity between two clusterings. The higher value

Table 2: Proportion of cluster sequence with less than 3 clusters

Remove clusters which has	Proportion
only 1 observation	72
less than 2 observations	91

of this index indicates greater similarity between the cluster sequence obtained from mclust and the true cluster sequence. The VI index is a measure of the distance between two clusterings (partitions of elements). The results presented in Table 3 shows that our proposed algorithm is more reliable as compared to Gaussian finite mixture model.

Table 3: Results of FM, VI index and optimal number of clusters

	mclust	proposed
FM	0.48	0.53
VI	2.01	1.80
no. of clusters	5.19	2.80

Further, to see the performance of the proposed method over kmeans we carried a short simulation study where there is almost zero temporal covariance (i.e., $\rho = 0$) and also choose small values for the covariate coefficient. The two methods are compared using optimal number of clusters. We take true optimal number of clusters as 2.4 and note that the optimal number of clusters using Silhouette method and proposed algorithm are 2.2 and 2.7 respectively. Similarly, we can consider other cases and compare the results.

These behaviours are highly desirable. It encourages using the Dirichlet Process with multivariate normal base distribution for random parameters of a quadratic function and autoregressive model at the estimation stage. These simulation results establish that an MCEM-type algorithm works well for finite samples, even for the complex model structure.

We have used R programming to perform all the simulations. The simulation code with specifications given above takes approximately 7 hours to run one iteration. We have done parallel computing with 6 cores CPU, however with high performance computing system, running time can be further reduced.

4 Gene Data Analysis

In Mehra et al. (2006), a whole genome cDNA microarray Streptomyces coelicolor was studied. The database had $7,825$ genes. From these genes, using a dynamic time warping algorithm 491

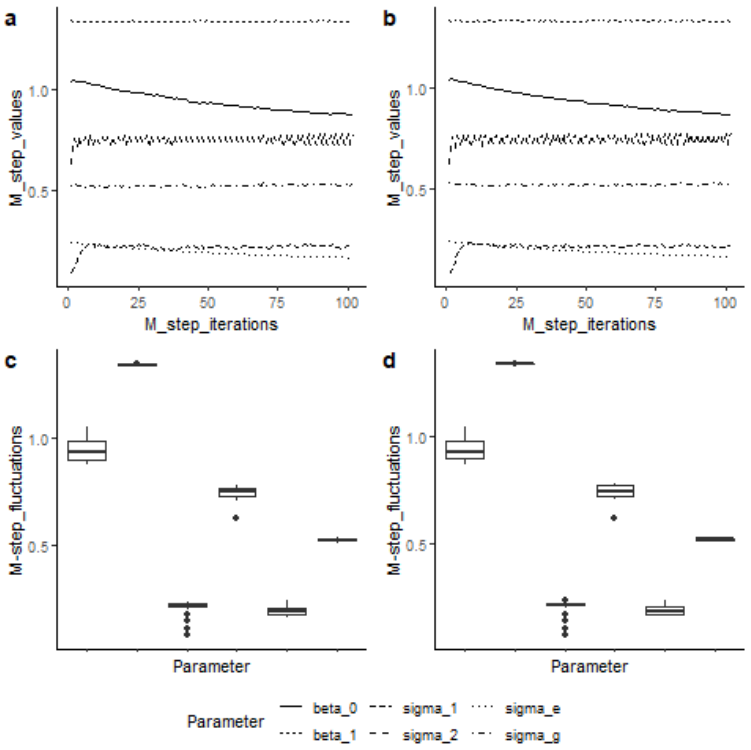

Figure 1: M-step convergence and fluctuations of parameters. The points plotted indicate the fluctuation range using box plot and convergence of parameters using line graph from 100 M-steps. Plots on the left (a, c) correspond to the case when E-step sample size is 1000. Plots on the right (b, d) correspond to the case when E-step sample size is 2000.

kinetically differentially expressed genes were chosen (see Mehra et al., 2006, for details). These kinetically differentially expressed genes were further classified into two classes the wild type and $\Delta absA1$. We chose our 18 genes from one of these classes of genes by fitting a quadratic trend over time and an AR(1) model, because our proposed model has quadratic function of time, AR(1) term in it. Subsequently, 18 genes were chosen which has $R^2 > 0.65$, AIC < 8.

The selected genes were measured over 10, 16, 22, 27 and 37 hours. The level 0 category of a gene belongs to one of the four categories: secondary metabolism, defined family regulators, two-component systems, other regulation.

Our goal is to analyse the data using the proposed algorithm (Section 2). We also find the cluster sequence using the heterogeneity measure based on the distance of observations within a cluster. Optimum cluster sequence will be useful to know which category genes behave similarly.

Table 4: Results based on gene data

	β_0	β_1	β_2	β_3	σ_e^2	σ_1^2	σ_2^2	σ_g^2	ρ	ν
Estimate	0.58	-0.73	-0.46	-1.68	0.26	0.42	0.17	0.29	0.005	0.53
S.E.	0.10	0.14	0.19	0.14	0.01	0.24	0.23	0.11	0.23	0.47

We fit the following Dirichlet process linear mixed model:

$$y_{it} = \beta_0 + \beta_1 \text{level0}_2 + \beta_2 \text{level0}_3 + \beta_3 \text{level0}_4 + \beta_4 \text{level0}_5 + \alpha_{1i} t + \alpha_{2i} t^2 + \gamma_{it} + \varepsilon_{it}, \tag{4.1}$$

where for the real data, $i = 1(1)18$; $t = 1(1)5$. The level 0 category is assumed to be a fixed effect and is represented by an indicator variable, $\gamma_{it} = \rho\gamma_{\overline{it-1}} + e_{it}$; e_{it} and ε_{it} are the independent normal variates with variances σ_g^2 and σ_e^2 respectively.

Table 5: Number of observations in each cluster

Cluster no.	Cluster size	Observation no.
1	6	1, 4, 7, 9, 10, 16
2	3	2, 12, 15
3	1	3
4	8	5, 6, 8, 11, 13, 14, 17, 18

To run the proposed algorithm in the above data set, we assumed the E-step sample size 1000 with a burn-in of 40%. The initial values of fixed effects (β_0, β_1, β_2, β_3) are $(0.71, -0.73, -0.46, -1.68)$, obtained by fitting the simple linear model with fixed effect as level 0 category. The starting values of variance components and precision parameter (σ_e^2, σ_1^2, σ_2^2, σ_g^2, ρ, ν) are $(0.32, 0.20, 0.03, 0.32, 0.62, 0.50)$, chosen in such a way that it allows the proposed algorithm to start with small values. With the above initial values, algorithm converges in 42 iterations. The real data took about 2 days to give the final output. Table 4 reports the estimated values and standard errors of all population parameters. The standard errors of all the parameters except the precision parameter ν lie in the small interval $(0.01, 0.23)$. Table 5 reports the size of cluster and number of observations in each cluster. Further, we observe that cluster 2 has putative marR-family regulatory protein, putative AbaA-like regulatory protein and putative tetR-family transcriptional regulator. All the genes are coding for proteins involved with antibiotic regulations and have been mentioned as putative. Similar interpretation can be made for other clusters as well. We also ran a k-means clustering algorithm with 4 centres. The results showed the same cluster sizes for all the four clusters, however the exact cluster sequences differed.

Moreover, to check the convergence of the algorithm we asses each M-step iteration likewise to section 3. Figure 2 shows that the algorithm converges approximately after 42 iterations.

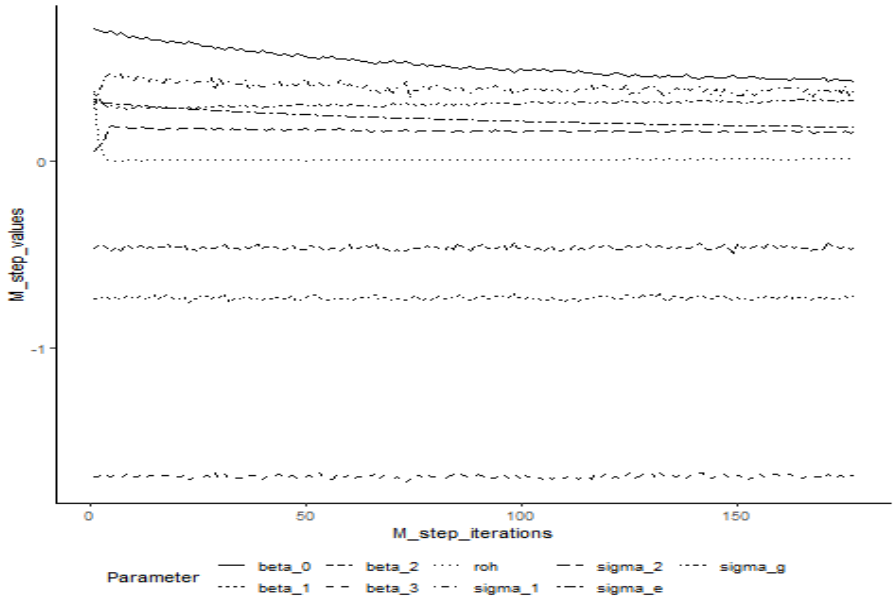

Figure 2: M-step convergence of parameters based on the gene data. It indicates convergence of parameters using line graph from one hundred M-steps.

5 Concluding Remarks

This paper has proposed the Dirichlet process linear mixed model using a non-parametric Bayesian approach with MCMC. The Dirichlet process introduces the flexibility and effectiveness in the linear mixed model set up. An EM type algorithm has been used to reduce the algebraic expressions' complexity in the estimation procedure. Samples have been generated using a Gibbs sampler, which is one of the convenient methods of MCMC to compute the expectation step. In the maximization step, Newton Raphson's method has been used to update the parameters.

From the above analyses, it may be concluded that the proposed estimation algorithm under the linear mixed model with a DP prior is highly effective including the time series data and at the same time it provides good estimates with small standard errors. These results (bias and S.E.) justify the use of the Dirichlet process in the linear mixed model setup. If we increase the number of samples in the expectation step, one may find a noticeable increase in the parameter estimates' efficiencies. However, the computation time increases due to a large number of samples. Hence, considering the properties, i.e., bias, standard errors, and confidence interval of the estimates, we can recommend them for their practical applications in real-life problems.

Acknowledgement

The work of S. Mukhopadhyay was supported by the Science and Research Engineering Board (Department of Science and Technology, Government of India) [Grant Number: RD/0118−DST0000−004]. We would also like to acknowledge Dr. Sarika Mehra, Department of Chemical Engineering, IIT Bombay for providing the gene data. We thank the Editor and two reviewers for their constructive comments, which help us to improve the manuscript.

A Appendix

A.1 Proof of proposition 1

$$
\begin{aligned}
\boldsymbol{S}(\boldsymbol{\theta}) = \frac{\partial l}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \log L &= \frac{\partial}{\partial \boldsymbol{\theta}} \log \int_{\boldsymbol{u}} L_c d\boldsymbol{u} \; ; \quad L, L_c \text{ marginal, complete data likelihood} \\
&= \frac{1}{\int_{\boldsymbol{u}} L_c d\boldsymbol{u}} \int_{\boldsymbol{u}} \frac{\partial}{\partial \boldsymbol{\theta}} L_c d\boldsymbol{u} \\
&= \frac{1}{\int_{\boldsymbol{u}} L_c d\boldsymbol{u}} \int_{\boldsymbol{u}} L_c \frac{\partial}{\partial \boldsymbol{\theta}} \log L_c d\boldsymbol{u} \\
&= \int_{\boldsymbol{u}} \frac{\partial}{\partial \boldsymbol{\theta}} \log L_c \frac{f(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{u}) g(\boldsymbol{u})}{\int_{\boldsymbol{u}} f(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{u}) g(\boldsymbol{u}) d\boldsymbol{u}} \\
&= \int_{\boldsymbol{u}} \left(\frac{\partial l_c}{\partial \boldsymbol{\theta}} \right) f(\boldsymbol{u}|\boldsymbol{\theta}, \boldsymbol{y}) d\boldsymbol{u} \\
&= E_{\boldsymbol{u}|\boldsymbol{y}} \left(\frac{\partial l_c}{\partial \boldsymbol{\theta}} \right) = E_{\boldsymbol{u}|\boldsymbol{y}} \left(\boldsymbol{S}_c(\boldsymbol{\theta}) \right)
\end{aligned}
$$

A.2 Elements of the information matrix

To find the elements of $\boldsymbol{I}$ matrix we use the following general result from the matrix theory (Searle et al., 2006).

If M is a square matrix whose elements are functions of a scalar variable x, then

$$
\frac{\partial \ln |\boldsymbol{M}|}{\partial x} = tr \left(\boldsymbol{M}^{-1} \frac{\partial \boldsymbol{M}}{\partial x} \right) ; \quad \frac{\partial \boldsymbol{M}^{-1}}{\partial x} = -\boldsymbol{M}^{-1} \frac{\partial \boldsymbol{M}}{\partial x} \boldsymbol{M}^{-1},
$$

where tr, the trace, denotes the sum of the diagonal elements of a square matrix.

The Information matrix, $\boldsymbol{I_{\xi}} = \begin{bmatrix} \boldsymbol{I_{\beta\beta}} & \boldsymbol{I_{\beta\sigma_e^2}} & 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{I_{\sigma_e^2\beta}} & I_{\sigma_e^2\sigma_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{\sigma_1^2\sigma_1^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{\sigma_2^2\sigma_2^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{\sigma_g^2\sigma_g^2} & I_{\sigma_g^2\rho} & 0 \\ 0 & 0 & 0 & 0 & I_{\rho\sigma_g^2} & I_{\rho\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{\nu\nu} \end{bmatrix}$, where

$$\boldsymbol{I_{\beta\beta}} = -E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\frac{\partial^2 l_c(\boldsymbol{\xi};\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{s})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\right) = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}}\left(\sum_{i=1}^{n}\frac{1}{\sigma_e^2}\boldsymbol{X}_i^{'}\boldsymbol{X}_i\right),$$

$$\boldsymbol{I_{\beta\sigma_e^2}} = -E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\frac{\partial^2 l_c(\boldsymbol{\xi};\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{s})}{\partial\boldsymbol{\beta}\partial\sigma_e^2}\right) = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}}\left(\sum_{i=1}^{n}\frac{1}{\sigma_e^4}\boldsymbol{X}_i^{'}\left(\boldsymbol{y}_i-\boldsymbol{X}_i\boldsymbol{\beta}-\boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\right)\right),$$

$$I_{\sigma_e^2\sigma_e^2} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\sum_{i=1}^{n}\left(-\frac{T}{2\sigma_e^4}+\frac{1}{\sigma_e^6}\left(\boldsymbol{y}_i-\boldsymbol{X}_i\boldsymbol{\beta}-\boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\right)^{'}\left(\boldsymbol{y}_i-\boldsymbol{X}_i\boldsymbol{\beta}-\boldsymbol{Z}_i\boldsymbol{\phi}_{s_i}\right)\right)\right),$$

$$I_{\nu\nu} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\frac{k}{\nu^2}-\sum_{i=1}^{n}\frac{1}{(\nu+i-1)^2}\right),$$

$$I_{\sigma_j^2\sigma_j^2} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\sum_{i=1}^{n}\left(-\frac{1}{2\sigma_j^4}+\frac{1}{\sigma_j^6}\phi_{s_i[j]}^2\right)\right), \quad j=1,2,$$

$$I_{\sigma_g^2\sigma_g^2} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\sum_{i=1}^{n}\left(-\frac{T}{2\sigma_g^4}+\frac{1}{\sigma_g^6}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\right)\right), \quad \boldsymbol{q}=(1,2),$$

$$I_{\sigma_g^2\rho} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\sum_{i=1}^{n}\frac{1}{\sigma_g^4}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{-1}\boldsymbol{R}^{*}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\right), \quad \boldsymbol{q}=(1,2),$$

$$I_{\rho\rho} = E_{\boldsymbol{\phi},\boldsymbol{s}|\boldsymbol{y},\boldsymbol{\xi}^{(m)}}\left(\sum_{i=1}^{n}\left(\frac{1}{2}tr(\boldsymbol{R}^{-1}\boldsymbol{R}^{**}-\boldsymbol{R}^{0}\boldsymbol{R}^{*})-\frac{1}{2\sigma_g^2}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{-1}\boldsymbol{R}^{**}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\right.\right.$$
$$\left.\left.+\frac{1}{\sigma_g^2}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}^{'}\boldsymbol{R}^{0}\boldsymbol{R}^{*}\boldsymbol{R}^{-1}\boldsymbol{\phi}_{s_i[-\boldsymbol{q}]}\right)\right), \quad \boldsymbol{q}=(1,2),$$

where $\boldsymbol{\phi}_{s_i} = (\phi_{s_i[1]}, \phi_{s_i[2]}, \ldots, \phi_{s_i[T+2]})$, $\boldsymbol{\phi}_{s_i[-j]}$ contains all elements of $\boldsymbol{\phi}_{s_i}$ except the j^{th} element, and

$$\boldsymbol{R}^{*} = \frac{\partial\boldsymbol{R}}{\partial\rho}, \quad \boldsymbol{R}^{**} = \frac{\partial^2\boldsymbol{R}}{\partial\rho^2}, \quad \boldsymbol{R}^{0} = \boldsymbol{R}^{-1}\boldsymbol{R}^{*}\boldsymbol{R}^{-1}.$$

References

Aghabozorgi, S., Seyed Shirkhorshidi, A., and Ying Wah, T. (2015), "Time-series clustering - A decade review," *Information Systems*, 53, 16–38.

Arabie, P. and Boorman, S. A. (1973), "Multidimensional scaling of measures of distance between partitions," *Journal of Mathematical Psychology*, 10, 148–203.

Beal, M. J. and Krishnamurthy, P. (2006), "Gene expression time course clustering with countably infinite hidden Markov models," *CoRR*, abs/1206.6, 23–30.

Begum, N., Ulanova, L., Wang, J., and Keogh, E. (2015), "Accelerating dynamic time warping clustering with a novel admissible pruning strategy," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 49–58.

Booth, J. G., Casella, G., and Hobert, J. P. (2008), "Clustering using objective functions and stochastic search," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 70, 119–139.

Celeux, G., Martin, O., and Lavergne, C. (2005), "Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments," *Statistical Modelling*, 5, 243–267.

Charles E. Antoniak (1986), "Mixture of Dirichlet Processes with applications to Bayesian nonparametric problems," *Annals of Statistics*, 14, 590–606.

Cooke, E. J., Savage, R. S., Kirk, P. D., Darkins, R., and Wild, D. L. (2011), "Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements," *BMC Bioinformatics*, 12.

Dahl, D. B. (2009), "Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model," *Bayesian Analysis*, 4, 243–264.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., and Schliep, A. (2008), "Clustering cancer gene expression data: A comparative study," *BMC Bioinformatics*, 9, 1–14.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 39, 1–38.

Du, M., Ding, S., Xue, Y., and Shi, Z. (2019), "A novel density peaks clustering with sensitivity of local density and density-adaptive metric," *Knowledge and Information Systems*, 59, 285–309.

Fowlkes, E. B. and Mallows, C. L. (1983), "A Method for Comparing Two Hierarchical Clusterings," *Journal of the American Statistical Association*, 78, 553–569.

Fraley, C. and Raftery, A. E. (2007), "Bayesian regularization for normal mixture estimation and model-based clustering," *Journal of Classification*, 24, 155–181.

Green, P. J. and Richardson, S. (2001), "Modelling Heterogenity With and Without the Dirichlet Process," *Scandinavian Journal of Statistics*, 28, 355–375.

Kyung, M. (2015), "Dirichlet Process Mixtures of Linear Mixed Regressions," *Communications for Statistical Applications and Methods*, 22, 625–637.

Maceachern, S. N. and Müller, P. (1998), "Estimating mixture of dirichlet process models," *Journal of Computational and Graphical Statistics*, 7, 223–238.

McNicholas, P. D. and Murphy, T. B. (2010), "Model-based clustering of microarray expression data via latent Gaussian mixture models," *Bioinformatics*, 26, 2705–2712.

Medvedovic, M. and Sivaganesan, S. (2002), "Bayesian infinite mixture model based clustering of gene expression profiles," *Bioinformatics*, 18, 1194–1206.

Medvedovic, M., Yeung, K. Y., and Bumgarner, R. E. (2004), "Bayesian mixture model based clustering of replicated microarray data," *Bioinformatics*, 20, 1222–1232.

Mehra, S., Lian, W., Jayapal, K. P., Charaniya, S. P., Sherman, D. H., and Hu, W. S. (2006), "A framework to analyze multiple time series data: A case study with Streptomyces coelicolor," *Journal of Industrial Microbiology and Biotechnology*, 33, 159–172.

Paparrizos, J. and Gravano, L. (2015), "K-shape: Efficient and accurate clustering of time series," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2015-May, 1855–1870.

Pirim, H., Ekşioglu, B., Perkins, A. D., and Yüceer, Ç. (2012), "Clustering of high throughput gene expression data," *Computers and Operations Research*, 39, 3046–3061.

Rasmussen, C. E., De La Cruz, B. J., Ghahramani, Z., and Wild, D. L. (2009), "Modeling and visualizing uncertainty in gene expression clusters using dirichlet process mixtures," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6, 615–628.

Searle, S. R., Casella, G., and McCulloch, C. E. (2006), *Variance compoents*, vol. 3, Wiley Interscience.

Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., and Steinberg, D. (2008), "Top 10 algorithms in data mining," *Knowledge and Information Systems*, 14, 1–37.

Received: February 28, 2021

Accepted: April 22, 2021