Bayesian profile regression with a longitudinal response

Rob Johnson*, Paul DW Kirk*, Simon R White

March 1, 2019

Abstract

Bayesian profile regression is an outcome-guided semi-supervised mixture modelling approach that makes use of a response in order to guide inference toward "relevant" clusterings. Previous applications of profile regression have considered univariate continuous, categorical, and count outcomes. Here we extend Bayesian profile regression to cases where the outcome is longitudinal or multivariate continuous, and provide an updated version of PReMiuM, the R package for profile regression. We consider multivariate normal and Gaussian process response models, and provide proof of principle applications to three simulation studies.

1 Introduction

Clustering is typically formulated as an unsupervised method for identifying heterogeneous subpopulations within a dataset (e.g. Jain et al., 1999). In addition to the perennial problem of how to determine the appropriate number of clusters (e.g. Rousseeuw, 1987; Fraley and Raftery, 2002; Sugar and James, 2003; Tibshirani and Walther, 2005), a key challenge is how to validate a given clustering structure (e.g. Kerr and Churchill, 2001; Brock et al., 2008). A common approach is to make use of left-out information in order to assess if the identified clustering structure provides a *relevant* stratification of the population (e.g. Yeung et al., 2001; Handl et al., 2005). For example, if we had clustered patients on the basis of, say, their blood plasma proteome profiles, then we might use their response to treatment as left-out validation information. That is, as a post-analysis step, we would check to see if patients allocated to the same cluster tended to have similar responses to treatment, while patients in different clusters had different responses. If this were the case, we might conclude that the identified clustering structure was *relevant* for the particular aim of identifying clusters associated with differential treatment responses. Of course, assessments of relevance are necessarily context- and application-specific. The left-out information (which we will refer to here as an *outcome* or *response*) is typically very closely related to the true aim of the clustering analysis, and implicitly defines the criterion by which a given clustering structure is assessed to be relevant or irrelevant.

When clustering high-dimensional datasets — in which there may be many variables that do not define a clustering structure (Law et al., 2003, 2004; Tadesse et al., 2005), or subsets of variables that define a variety of different clustering structures (Cui et al., 2007; Niu et al., 2010; Guan et al., 2010; Li and Shafto, 2011; Niu et al., 2014; Kirk and Richardson, 2018) — it may be desirable to make use of (potentially highly informative) outcome information directly, in order to guide inference toward the most relevant clustering structures. That is, we may wish to use the outcome information during the clustering analysis itself, rather than during post-analysis validation. This is one of the principal motivations for *Bayesian profile regression* (Molitor et al., 2010), a semi-supervised approach for model-based clustering that allows outcome information to be taken into account. Previous applications of profile regression have considered univariate continuous, categorical, and count outcomes (Liverani et al., 2015). Also Bernoulli, binomial, survival? Here we extend Bayesian

profile regression to cases where the outcome is longitudinal or multivariate continuous, and provide an updated version of PReMiuM, the R package for profile regression (Liverani et al., 2015).

2 Profile regression: a semi-supervised clustering model

We suppose that we have data comprising observations on a vector of covariates, $\mathbf{x}$, and outcomes, $\mathbf{y}$. In Molitor et al. (2010), which we follow, the authors permit their model for $\mathbf{y}$ to depend upon additional covariates, $\mathbf{w}$, which they refer to as fixed effects. These fixed effects are covariates that may be predictive of the outcome, but do not directly contribute to the clustering, and which come with associated "global" (i.e. not cluster-specific) parameters, $\boldsymbol{\beta}$, for the model linking $\mathbf{y}$ and $\mathbf{w}$. The general model considered in Molitor et al. (2010) for C clusters is then:

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\beta}, \mathbf{w}) = \sum_{c=1}^{C} \pi_c f_{\mathbf{y}}(\mathbf{y}|\boldsymbol{\theta}_c, \mathbf{w}, \boldsymbol{\beta}) f_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\phi}_c), \tag{1}$$

where the π_c are mixture weights, the $\boldsymbol{\phi}_c$ are cluster-specific parameters for the density $f_{\mathbf{x}}$, and the $\boldsymbol{\theta}_c$ are cluster-specific parameters for the density $f_{\mathbf{y}}$. The original formulation of this model, which we also adopt here (see Section 2.1 below), is in terms of infinite (specifically Dirichlet process) mixture models; however, we note that the model is equally applicable in the case of finite C. Adopting the terminology of Bair and Tibshirani (2004), we refer to this as a *semi-supervised* mixture model, since clustering is guided by an outcome variable. We note that there are several other types of semi-supervised clustering approach described in the literature (e.g. Basu et al., 2004; Crook et al., 2018, who cluster partially labelled data), and we refer the reader to Bair (2013) for a review.

The profile regression model described in Equation 1 implicitly assumes that the covariates, $\mathbf{x}$, and outcomes, $\mathbf{y}$, are linked via a shared clustering structure, but are otherwise conditionally independent (an alternative model, in which $\mathbf{y}$ may depend on $\mathbf{x}$, is described in Shahbaba and Neal, 2009). The inclusion of fixed effects $\mathbf{w}$ in Equation (1) allows for the possibility that there are additional nuisance covariates that might be predictive of the observed $\mathbf{y}$, but which we do not wish to include in the clustering analysis. For example, in an analysis of data from a genome wide association study of lung cancer considered in Papathomas et al. (2012), $\mathbf{x}$ comprises genetic covariates (SNPs), while $\mathbf{w}$ comprises potentially confounding covariates including age, sex, and smoking status.

2.1 Inference of the mixture weights

Inference and choice of prior distribution for the parameters $\boldsymbol{\theta}_c, \boldsymbol{\phi}_c$, and $\boldsymbol{\beta}$ is necessarily context specific, depending on the specific choices for the parametric functions, $f_{\mathbf{y}}$ and $f_{\mathbf{x}}$, and our prior beliefs about their parameters. For the mixture weights, π_c, we consider infinite C and adopt a stick-breaking prior (also known as a GEM or Griffiths-Engen-McCloskey prior; see Pitman, 2002), constructed as follows:

$$u_c \sim \text{Beta}(1, \alpha) \tag{2}$$

$$\pi_1 = u_1, \text{ and } \pi_c = u_c \prod_{r=1}^{c-1} (1 - \pi_r), \text{ for } c \geq 2, \tag{3}$$

where α is a positive parameter, for which we adopt a gamma prior with shape parameter 2 and rate parameter 1. See, for example, Ishwaran and James (2001) and Kalli et al. (2009) for further background, and Liverani et al. (2015) and Hastie et al. (2015) for details of how inference of the π_c's is performed in the Bayesian profile regression model.

3 Clustering guided by a longitudinal outcome

In previous applications of Bayesian profile regression, the outcome $\mathbf{y} = y$ has been assumed to be univariate (Molitor et al., 2010; Papathomas et al., 2012). Here we extend the original model to consider cases in which $\mathbf{y}$ comprises longitudinal continuous data. In Section 3.1, we restrict attention to situations in which we have longitudinal continuous data measured on individuals at a common set of time points. The outcome for the i-th individual is then of the form $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,M}]^\top \in \mathbb{R}^M$, where y_{ij} denotes the value for the response measured on the i-th individual at j-th time point t_j, and M is the number of time points. Crucially, the set of time points, $\{t_1, \ldots, t_M\}$, is assumed to be the same for all individuals, and it is assumed that there are no missing values. In Section 3.2, we consider the (more general) situation in which $\mathbf{y}_i = [y_i(t_{i,1}), \ldots, y_i(t_{i,M_i})]^\top$, where $\{t_{i,1}, \ldots, t_{i,M_i}\}$ denotes the set of M_i time points associated with the i-th individual, and $y_i(t)$ refers to the value of the response for the i-th individual at time t.

3.1 Multivariate normal outcome model

In the case where all individuals have response measurements at a common set of time points, we consider modelling the outcome as multivariate normal, and refer to this as the MVN response model. Such a model may also be appropriate when each individual's longitudinal response is summarised through a collection of statistics (as in, for example, Hathaway and D'agostino, 1993). For flexibility, we make no assumptions about the structure of the covariance matrix in our model, and defer until Section 3.2 discussion of a model in which we explicitly try to capture the time-ordering of the data.

For the MVN response model, the component-specific parameters $\boldsymbol{\theta}_c$ in Equation (1) comprise mean vector, $\boldsymbol{\mu}_c$, and covariance matrix, Σ_c. In the simple case where there are no confounding covariates $\mathbf{w}$, the density $f_{\mathbf{y}}(\mathbf{y}|\boldsymbol{\theta}_c)$ is simply a multivariate normal density with mean $\boldsymbol{\mu}_c$ and covariance matrix Σ_c. When we have confounding covariates, we assume that they act on the response via a constant shift of the mean. In particular, if $\mathbf{w}_i \in \mathbb{R}^R$, then $\boldsymbol{\beta} \in \mathbb{R}^R$, and the MVN outcome model is as follows:

$$\begin{aligned} f_{\mathbf{y}}(\mathbf{y}_i|\boldsymbol{\theta}_c, \mathbf{w}_i, \boldsymbol{\beta}) &= f_{\mathbf{y}}(\mathbf{y}_i|\boldsymbol{\mu}_c, \Sigma_c, \mathbf{w}_i, \boldsymbol{\beta}) \\ &= \frac{1}{\sqrt{(2\pi)^M |\Sigma_c|}} \exp\left(-\frac{1}{2} (\mathbf{y}_i - (\boldsymbol{\mu}_c + (\boldsymbol{\beta}^\top \mathbf{w}_i)\mathbf{1}_M)) \Sigma_c^{-1} (\mathbf{y}_i - (\boldsymbol{\mu}_c + (\boldsymbol{\beta}^\top \mathbf{w}_i)\mathbf{1}_M))^\top \right), \end{aligned} \tag{4}$$

where $\mathbf{1}_M$ denotes a vector of ones of length M. To perform inference, we adopt conjugate normal-inverse Wishart priors $\mu_c|\Sigma_c \sim \mathcal{N}(\mu_0, \Sigma_c/\kappa_0)$, and $\Sigma_c \sim \mathcal{W}^{-1}(R_0, \nu_0)$, where $\mu_0 = \frac{1}{NM} \sum_{j=1}^{N \times M} y_j$, N is the number of participants, $\kappa_0 = 0.01$, $\nu_0 = M$ and

$$R_0 = \left(\frac{\nu_0}{NM} \sum_{j=1}^{N \times M} (y_j - \mu_0)(y_j - \mu_0)^T \right)^{-1}. \tag{5}$$

3.2 Gaussian process outcome model: a Bayesian functional data approach

We consider an outcome model in which we assume that the complete longitudinal trajectory of the i-th individual's response may be expressed as follows:

$$y_i(t) = g_i(t) + \boldsymbol{\beta}^\top \mathbf{w}_i + \epsilon_{i,t}, \tag{6}$$

where $y_i(t)$ is the observed value of the response for the i-th individual at time t; g_i is a continuous function of time; $\epsilon_{i,t} \sim \mathcal{N}(0, \sigma_i^2)$ is i.i.d. Gaussian noise; and $\boldsymbol{\beta}$ and $\mathbf{w}_i$ are as in the MVN case. In

practice, of course, we only observe $y_i(t)$ at a discrete set of time points, $\{t_{i,1}, \ldots, t_{i,M_i}\}$; however, this functional approach proves useful when dealing with individuals who have observations at different time points.

If individuals i and j are both in cluster c, then we assume that $g_i(t) = g_j(t) := g^{(c)}(t)$, where $g^{(c)}$ denotes the function associated with the c-th cluster. Moreover, $\sigma_i^2 = \sigma_j^2 := \sigma_c^2$, where σ_c^2 denotes the noise variance associated with the c-th cluster. As is common when dealing with mixture models, we introduce component allocation variables z_i for each i, such that $z_i = c$ if the i-th individual is allocated to the c-th cluster. The conditional distribution of y_i at time t, given that the i-th individual is allocated to the c-th component, is then:

$$y_i(t)|z_i = c \quad \sim \quad \mathcal{N}(g^{(c)}(t) + \boldsymbol{\beta}^\top \mathbf{w}_i, \sigma_c^2). \tag{7}$$

In order to proceed, we could specify a parametric form for the functions $g^{(c)}$ (e.g. we might specify that $g^{(c)}$ is a polynomial of degree d), whose component-specific parameters we would need to infer in order to fit the model. However, as we now describe, here we adopt a Bayesian nonparametric approach, and take a Gaussian process prior for the unknown function $g^{(c)}$.

3.2.1 Gaussian process priors for unknown functions

A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). This definition simply means that a (potentially infinite) collection $v_1, v_2, \ldots$ of random variables defines a Gaussian process if and only if any finite subcollection of the variables is jointly distributed according to a Gaussian distribution.

In GP regression, a GP prior is assumed for the outputs of an unknown function $g^{(c)}$. This means that we assume a priori that $g^{(c)}(t_1), g^{(c)}(t_2), \ldots, g^{(c)}(t_T)$ are jointly distributed according to a T-variate Gaussian distribution for any $t_1, t_2, \ldots, t_T$ and any finite T. To fully specify a GP prior, we require a mean function, $m^{(c)}$, and a covariance function, $k^{(c)}$, which define the mean vectors and covariance matrices of the Gaussian distributions associated with each finite subcollection of the variables. We write $g^{(c)} \sim \mathcal{GP}(m^{(c)}, k^{(c)})$ to indicate that we have assumed a Gaussian process prior with mean function $m^{(c)}$ and covariance function $k^{(c)}$ for the function $g^{(c)}$, so that:

> $g^{(c)} \sim \mathcal{GP}(m^{(c)}, k^{(c)})$ if and only if, for any finite collection $t_1, t_2, \ldots, t_T$ of times, we have $[g^{(c)}(t_1), \ldots, g^{(c)}(t_T)]^\top \sim \mathcal{N}_T(\mathbf{m}^{(c)}, K^{(c)})$, where $\mathbf{m}_j^{(c)} = m^{(c)}(t_j)$, $K_{j,j'}^{(c)} = k^{(c)}(t_j, t_{j'})$, and $\mathcal{N}_T$ denotes a T-variate normal distribution.

There are many possibilities for $m^{(c)}$ and $k^{(c)}$; see, for example, Rasmussen and Williams (2006). In practice, we take the standard default choice of setting $m^{(c)}$ to be the zero function, so that $m^{(c)}(t_j) = 0$ for all t_j, and we take $k^{(c)}$ to be the widely-used *squared exponential* covariance function, so that:

$$k^{(c)}(t_j, t_{j'}) = a_c \exp\left(-\frac{(t_j - t_{j'})^2}{2l_c}\right), \tag{8}$$

where here the hyperparameters a_c and l_c are the *signal variance* and *length-scale* for the squared exponential covariance function k_c.

3.2.2 Marginalising the unknown function

We adopt a GP prior, with component specific hyperparameters a_c and ℓ_c, for each unknown function $g^{(c)}$. Suppose there are n_c individuals associated with component c. For notational convenience, we assume that these are individuals $1, \ldots, n_c$, i.e. we assume that $z_i = c$ for $i = 1, \ldots, n_c$ and

$z_i \neq c$ for all other i. Recalling that $\mathbf{y}_i = [y_i(t_{i,1}), \ldots, y_i(t_{i,M_i})]^\top$, where $\{t_{i,1}, \ldots, t_{i,M_i}\}$ denotes the set of M_i time points associated with the i-th individual, we define the "corrected" vector of observations for the i-th individual to be $\mathbf{q}_i = \mathbf{y}_i - (\boldsymbol{\beta}^\top \mathbf{w}_i)\mathbf{1}_{M_i}$. It then follows from our model for the response (Equations (6) and (7)) that:

$$[\mathbf{q}_1, \ldots, \mathbf{q}_{n_c}]^\top | g^{(c)}, \sigma_c^2 \quad \sim \quad \mathcal{N}_{\mathcal{M}}([\mathbf{g}_1^{(c)}, \ldots, \mathbf{g}_{n_c}^{(c)}]^\top, \sigma_c^2 I_{\mathcal{M}}), \tag{9}$$

where $\mathbf{g}_i^{(c)} = [g^{(c)}(t_{i,1}), \ldots, g^{(c)}(t_{i,M_i})]^\top$ and $\mathcal{M} = \sum_{i=1}^{n_c} M_i$.

According to our GP prior, we have,

$$[\mathbf{g}_1^{(c)}, \ldots, \mathbf{g}_{n_c}^{(c)}]^\top | a_c, \ell_c \sim \mathcal{N}_{\mathcal{M}}(\mathbf{0}, K^{(c)}), \tag{10}$$

where $K^{(c)}$ is an $\mathcal{M} \times \mathcal{M}$ matrix given by:

$$K^{(c)} = \begin{pmatrix} K_{1,1}^{(c)} & \ldots & K_{1,n_c}^{(c)} \\ \vdots & \ddots & \vdots \\ K_{n_c,1}^{(c)} & \ldots & K_{n_c,n_c}^{(c)} \end{pmatrix}, \tag{11}$$

with $K_{i,j}^{(c)}$ defined to be the $M_i \times M_j$ matrix,

$$K_{i,j}^{(c)} = \begin{pmatrix} k^{(c)}(t_{i,1}, t_{j,1}) & \ldots & k^{(c)}(t_{i,1}, t_{j,M_j}) \\ \vdots & \ddots & \vdots \\ k^{(c)}(t_{i,M_i}, t_{j,1}) & \ldots & k^{(c)}(t_{i,M_i}, t_{j,M_j}) \end{pmatrix}. \tag{12}$$

It follows immediately from Equations (9) and (10) that we may marginalise $g^{(c)}$ to give,

$$[\mathbf{q}_1, \ldots, \mathbf{q}_{n_c}]^\top | a_c, \ell_c, \sigma_c \sim \mathcal{N}_{\mathcal{M}}(\mathbf{0}, K^{(c)} + \sigma_c^2 I_{\mathcal{M}}). \tag{13}$$

Recalling that $\mathbf{y}_i = \mathbf{q}_i + (\boldsymbol{\beta}^\top \mathbf{w}_i)\mathbf{1}_{M_i}$, we then have:

$$[\mathbf{y}_1, \ldots, \mathbf{y}_{n_c}]^\top | a_c, \ell_c, \sigma_c \sim \mathcal{N}_{\mathcal{M}}([(\boldsymbol{\beta}^\top \mathbf{w}_1)\mathbf{1}_{M_1}, \ldots, (\boldsymbol{\beta}^\top \mathbf{w}_{n_c})\mathbf{1}_{M_{n_c}}]^\top, K^{(c)} + \sigma_c^2 I_{\mathcal{M}}). \tag{14}$$

3.2.3 Inference of the hyperparameters

The hyperparameters, a and ℓ, and the noise parameter, σ, must all be positive. Following Neal (1999), we deal throughout with the logarithms of these quantities, eliminating the positivity constraint (and thereby making sampling of these quantities more straightforward). For convenience, we adopt independent, standard normal priors for each of $\log(a), \log(\ell)$ and $\log(\sigma)$.

4 Simulation study

In this Section we present three simulation studies. The first two demonstrate our motivation in using outcomes in profile regression to guide clustering: first, where the clustering structure in the covariates is weak and, second, where there are multiple clustering structures in the covariates, of which only one has clinical relevance. The third demonstrates the two methods, comparing them with each other in terms of duration and efficacy in recovering a true underlying structure. The methods are also assessed as the number of observation times is varied, and as the clustering structures overlap.

Efficacy in recovering a true underlying clustering structure is assessed via Rand indices (Rand, 1971). The Rand index is a metric ranging from 0 to 1 that reflects the similarity between two clustering structures. The index is 1 for identical structures. The adjusted Rand index corrects for the extent of agreement that would be expected by chance, given the size of the clusterings (Hubert and Arabie, 1985). Hence, it is possible for the adjusted Rand index to take a negative value.

The adjusted Rand index is the basis of the posterior expectation adjusted Rand (PEAR) criterion for cluster assignment (Fritsch and Ickstadt, 2009). The PEAR index is the criterion used in this Section to assess returned clustering structures based on that used in the simulation of the data, using the R package `mcclust` (Fritsch and Ickstadt, 2009).

4.1 Semi-supervised vs. unsupervised clustering

Here we demonstrate the importance of using outcome data in identifying unseen clustering structures. We define two clusters, each with 50 individuals, and simulate covariate data and 0–4 outcome variables. We use PReMiuM with an MVN response model to identify the underlying clustering structure, which we compare with the true structure.

Our focus is on the difference between inference using no outcome data ($M = 0$), and inference using $M \geq 1$ observation times. Inference using no outcome data corresponds to studies in which, first, clustering is performed on covariate data and, second, cluster memberships are leveraged to explain some outcome. The rationale of profile regression is that, when the purpose of clustering is to group individuals according to their outcomes, using outcome data leads to more meaningful cluster labels.

4.1.1 Design

The covariates are discrete variables generated from a multinomial distribution. We set the number of covariates to 10, denoted $q = 1, ..., Q$. Each covariate q has $R_q = R = 3$ categories, with multinomial parameter $\boldsymbol{\phi}_{c,q}$ generated as follows:

$$\phi_0 \sim \text{Dir}(\alpha_0 \mathbf{1}_R) \tag{15}$$

where ϕ_0 and $\mathbf{1}_R$ are vectors of length R, and $\alpha_0 = 0.01$. We set

$$\boldsymbol{\phi}_{c,q} = v\phi_0 + \frac{1}{R}(1 - v)\mathbf{1}_R \tag{16}$$

where $v = 0.4$.

The outcome is generated from a multivariate normal distribution with (monotonic) means 1 and 4 for the first and second clusters, respectively, and covariance matrices $0.5I_M$, where I_M is the identity matrix of dimension M, the number of observation times.

4.1.2 Results

We run PReMiuM with the MVN response model and $\max(2000, 2000M)$ iterations each for burn in and sampling. The effect of the number of observations on the resulting PEAR indices is shown in Figure 1.

It is possible to recover, to some extent, the clustering structure in the covariates without use of outcome data. However, inclusion of outcome data improves cluster-structure recall (Figure 1).

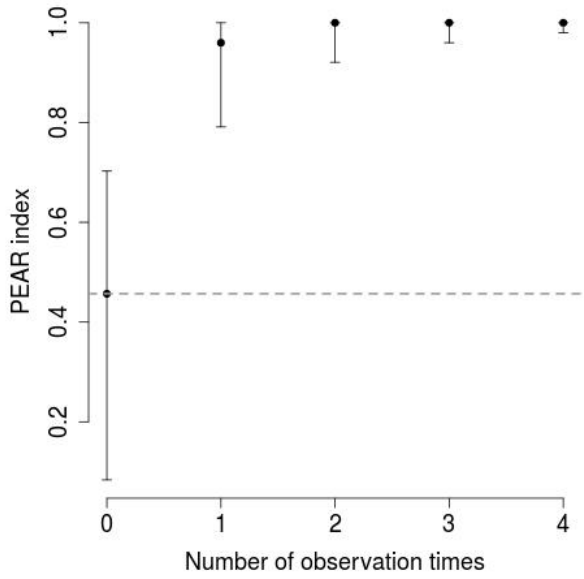

Figure 1: The effect of number of observation times on PEAR index. Points are median PEAR indices for 1000 simulations with error bars showing 0.05 and 0.95 quantiles. In each simulation, data are generated for 100 individuals belonging to one of two clusters. Each individual has 10 covariates and 0 to 4 MVN outcomes. The clustering structure is weak in the covariates and stronger in the outcomes (see main text). The clustering structure is not well recovered without "semi-supervision" from the outcome.

4.2 Semi-supervised clustering and variable selection

Here we adapt the simulation study of Section 4.1 to a case in which there are two clustering structures within the covariate data, of which one corresponds to the clustering structure in the response. We use the 'variable selection' feature of PReMiuM, which allows for some covariates to be excluded in the construction of the profile. We show that there is no reason for the meaningful clustering structure to be chosen over the alternate clustering structure in the absence of outcome data. Outcome data are therefore required for guidance towards the clustering structure that is meaningful when more than one structure is present in the data.

4.2.1 Design

Covariate data consist of four variables, simulated as described in Section 4.1. The first two variables correspond to the alternate clustering structure (i.e. they do not align with the response). The last two variables have a clustering structure that aligns with that of the response. We choose $v = 1$ for all covariates, so that each clustering structure has a strong signature when those covariates are viewed alone (see Figure 7).

Response data are simulated as in Section 4.1. Again we vary the number of observation times from 0 to 4. We are interested to know how inclusion of outcome data influences variable selection, and how these together affect PEAR indices.

4.2.2 Results

Inclusion of response variables enables recovery of the relevant underlying clustering structure (Figure 2). The top row of Figure 2 shows a box plot of variable selection for each number of observation times. When the response is excluded, covariates 1–4 are not differentially selected, and correspondence between the inferred and true clustering structures shown in the plot below is low (around 0.4). When four response variables are included, covariates 1 and 2 are deselected in favour of covariates 3 and 4, resulting in identification of the clustering structure that aligns with the response.

Here we have shown that, for a dataset in which there is more than one clustering structure, use of covariate variable selection with outcome data in profile regression improves recovery of the underlying structure. It facilitates the identification of the clustering structure within the covariate data that is relevant to the outcome.

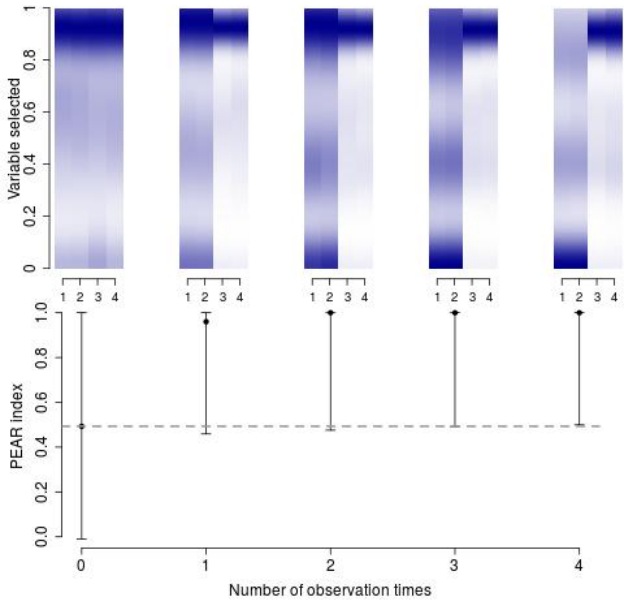

Figure 2: The effect of number of observation times on PEAR index when there are two clustering structures in the covariate data. Top: density plots showing selection for the four covariates. Bars show the weights given to the covariate in each of 1000 simulations, where 0 means unselected and 1 selected. Darker blue indicates higher density. Covariates that correspond to the outcome data are variables 3 and 4. Bottom: points are median PEAR indices for 1000 simulations with error bars showing 0.05 and 0.95 quantiles. The true structure is not well recovered without outcome-guided clustering, as there are two structures to choose between.

4.3 MVN vs. longitudinal

To draw out the differences between the two response models, we use each in turn to simulate data, and use both to infer clusters and profiles. For the sake of comparison, we simulate data that are longitudinal, with all individuals having observations at the same time, so that both methods may be applied to the data generated.

4.3.1 Design

For comparing the two methods, we measure PEAR indices and the time taken by the algorithm, and we assess the effects of a) the data-generating model, b) the number of observation times, and c) the extent to which the true clusters are identified.

We set two clusters with 30 subjects in each. Denoting the number of observation times by M, we perform $5000(M-2)$ iterations each in the burn-in phase and in the sampling phase for the MVN response model. For the GP response model, we use 5000 iterations for each phase.

Covariate data consist of two discrete varaibles. They have no clustering structure. They are simulated anew as often as the outcome data. Outcome data for the first cluster are simulated from a mean vector of $10 \cdot \mathbf{1}_M$ and a covariance matrix that depends on the data-generation method. For MVN generation, it is a matrix of dimension M with diagonal values of 1 and lag-1 values of 0.5. For GP generation, it is given by the covariance function in Equation 8 with parameters $\{a, l, \sigma^2\} = \{-0.5, -0.1, -0.5\}$.

For the second cluster, the outcome has a mean vector that is linear from 10 to $10(1+\xi)$, with M equally spaced points, where ξ is the gradient, ranging from -1 to 0. The covariance matrices are constructed in the same way as those for the first cluster.

4.3.2 Simulated data

For each inference method, there are three variables: the data-generating model, which is MVN or GP; the number of observation times, which range from 3 to 6; and the gradient, ξ, of the second cluster's outcome, which ranges from -1 to 0. For $\xi = 0$, the data-generating models for the two clusters are identical. Some examples of simulated data are given in Figure 3.

4.3.3 Results

Figure 4 summarises the effects of the three variables of interest on the two response models. Both response models perform worse with MVN-generated data, which might be attributed to the choices of covariance matrices presented above. Both perform better when the clusters are more separable, as we would expect.

With the MVN response model, the true clustering structure is better recovered when there are fewer observation times. This effect can be mitigated by increasing the number of iterations of the sampler as more observation times are included (see Figure 8).

For the GP response model, the true clustering structure is well recovered for the GP-generated data, even when the generating models are identical, as these data have an inherent structure that the response model can uncover. This response model has only three parameters to infer per cluster regardless of the number of observation times, in comparison with $\frac{1}{2}M(M-1)$ for the MVN response model, so fewer iterations are required for the GP response model.

Figure 5 summarises the time taken for PReMiuM to complete. The MVN response model is much quicker than the GP model, by two orders of magnitude, despite having more iterations in the burn-in phase and the sampler for datasets with more than three observation times.

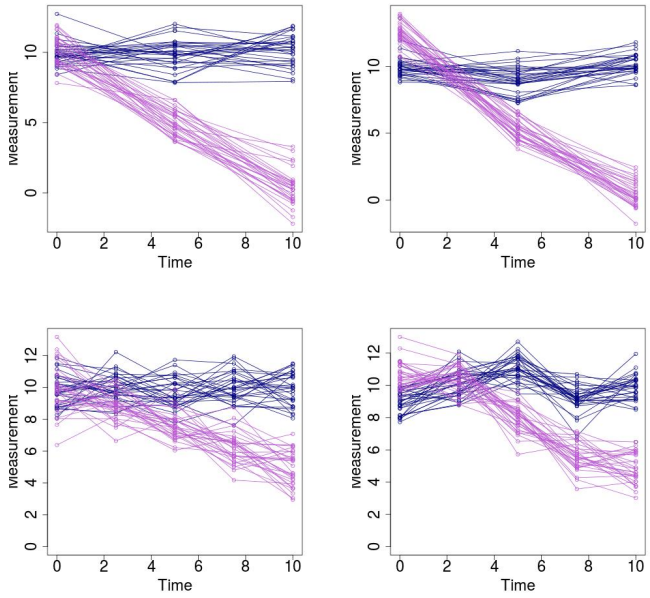

Figure 3: Simulated data. Examples of data generated with the MVN model (left) and the GP model (right). Top: three data points across the time period 0–10. The mean gradient is -1 for the second (purple) cluster. Bottom: five data points across the time period 0–10. The mean gradient is -0.5 for the second (purple) cluster.

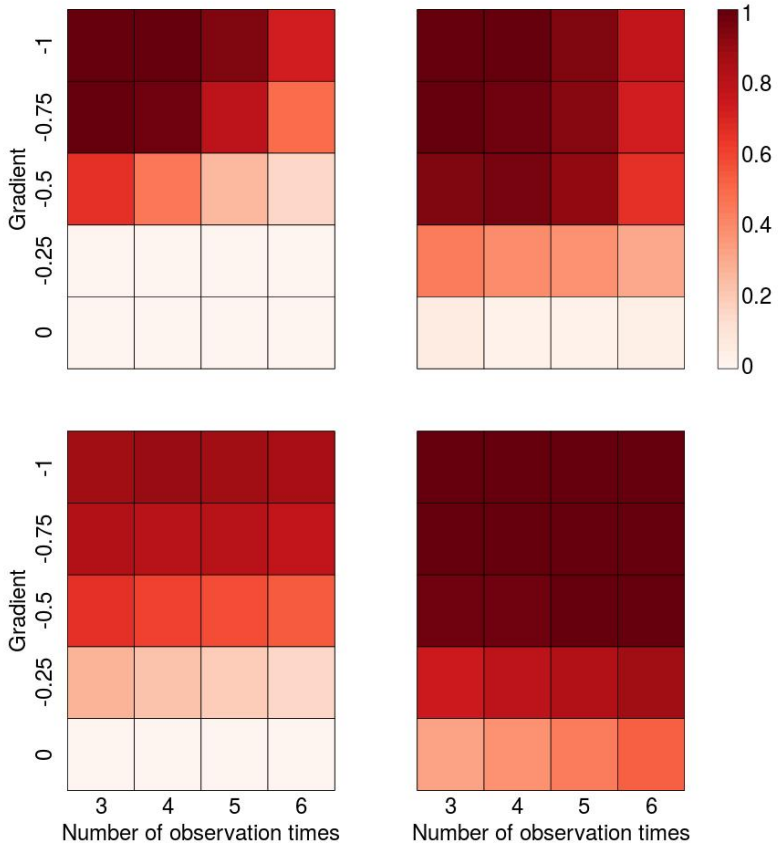

Figure 4: Simulation study results. Heatmaps show PEAR indices following inference with the MVN response model (top) and the GP response model (bottom). Data were generated with an MVN model (left) and a GP model (right). Each square within each heatmap represents a single experimental set-up: the gradient (which indicates how seperable the clusters are) varies on the y axis, and the number of observation times varies on the x axis. The colour corresponds to the average PEAR index of 1000 repetitions, where darker red indicates a higher index. PReMiuM was run with 5000 iterations each for burn in and sampling for the GP response model. For the MVN response model, the number of iterations for the burn in and sampling were each $5000(M-2)$, where M is the number of observation times.

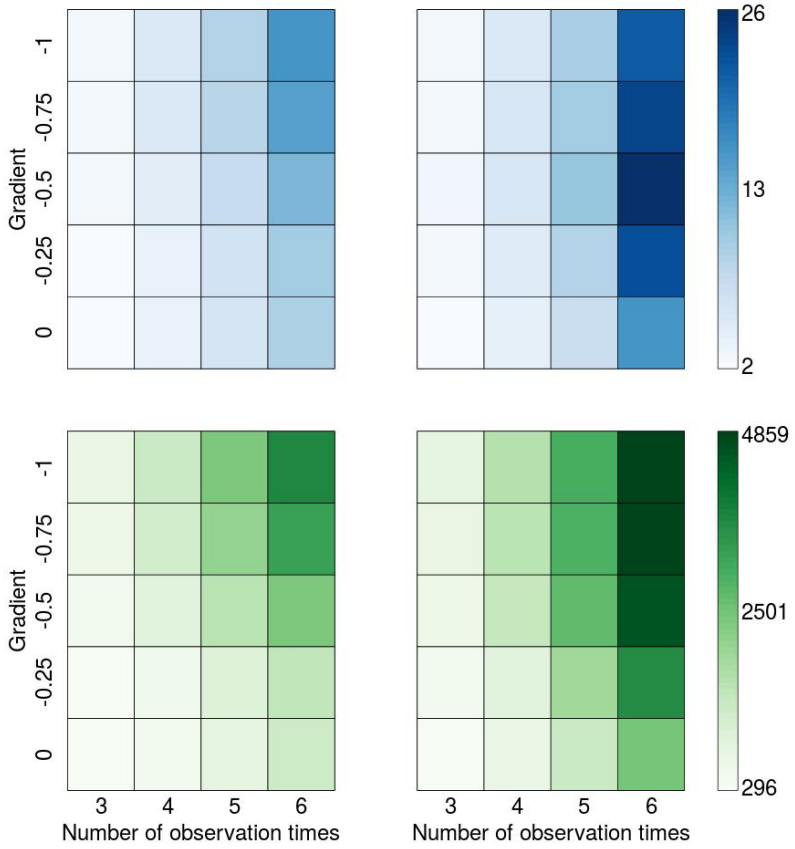

Figure 5: Simulation study timings. Heatmaps show the duration, in seconds, of inference with the MVN response model (top) and the GP response model (bottom). Data were generated with an MVN model (left) and a GP model (right). Each square within each heatmap represents a single experimental set-up: the gradient (which indicates how seperable the clusters are) varies on the y axis, and the number of observation times varies on the x axis. The colour corresponds to the average duration of 1000 repetitions, where darker colour indicates a longer time. PReMiuM was run with 5000 iterations each for burn in and sampling for the GP response model. For the MVN response model, the number of iterations for the burn in and sampling were each $5000(M-2)$, where M is the number of observation times.

References

Bair, E. (2013), 'Semi-supervised clustering methods.', *Wiley interdisciplinary reviews. Computational statistics* **5**(5), 349–361.

Bair, E. and Tibshirani, R. (2004), 'Semi-supervised methods to predict patient survival from gene expression data.', *PLoS biology* **2**(4), E108.

Basu, S., Bilenko, M. and Mooney, R. J. (2004), A probabilistic framework for semi-supervised clustering, *in* 'Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '04, ACM, New York, NY, USA, pp. 59–68.

Brock, G., Pihur, V. and Datta, S. (2008), 'clValid: An R package for cluster validation', *J Stat Softw* **25**.

Crook, O. M., Mulvey, C. M., Kirk, P. D. W., Lilley, K. S. and Gatto, L. (2018), 'A Bayesian Mixture Modelling Approach For Spatial Proteomics', *bioRxiv* .

Cui, Y., Fern, X. Z. and Dy, J. G. (2007), Non-redundant Multi-view Clustering via Orthogonalization, *in* 'Seventh IEEE International Conference on Data Mining (ICDM 2007)', IEEE, pp. 133–142.

Fraley, C. and Raftery, A. E. (2002), 'Model-Based Clustering, Discriminant Analysis, and Density Estimation', *Journal Of The American Statistical Association* **97**(458), 611–631.

Fritsch, A. and Ickstadt, K. (2009), 'Improved criteria for clustering based on the posterior similarity matrix', *Bayesian Analysis* **4**(2), 367–391.

Gelman, A. and Rubin, D. B. (1992), 'Inference from iterative simulation using multiple sequences', *Statistical Science* **7**(4), 457–472.

Guan, Y., Dy, J. G., Niu, D. and Ghahramani, Z. (2010), 'Variational inference for nonparametric multiple clustering', *MultiClust Workshop* .

Handl, J., Knowles, J. D. and Kell, D. B. (2005), 'Computational cluster validation in post-genomic data analysis.', *Bioinformatics (Oxford, England)* **21**(15), 3201–3212.

Hastie, D. I., Liverani, S. and Richardson, S. (2015), 'Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations.', *Statistics and Computing* **25**(5), 1023–1037.

Hathaway, D. K. and D'agostino, R. B. (1993), 'A technique for summarizing longitudinal data', *Statistics in Medicine* **12**(23), 2169–2178.

Hubert, L. and Arabie, P. (1985), 'Comparing partitions', *Journal of Classification* **2**(1), 193–218.

Ishwaran, H. and James, L. F. (2001), 'Gibbs Sampling Methods for Stick-Breaking Priors', *Journal Of The American Statistical Association* **96**(453), 161–173.

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999), 'Data Clustering: A Review', *ACM Comput. Surv.* **31**(3), 264–323.

Kalli, M., Griffin, J. E. and Walker, S. G. (2009), 'Slice sampling mixture models', *Statistics and Computing* **21**(1), 93–105.

Kerr, M. K. and Churchill, G. A. (2001), 'Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments', *Proceedings Of The National Academy Of Sciences Of The United States Of America* **98**(16), 8961–8965.

Kirk, P. D. W. and Richardson, S. (2018), 'Semi-supervised multi-view Bayesian clustering', *arXiv* .

Kirk, P., Griffin, J. E., Savage, R. S., Ghahramani, Z. and Wild, D. L. (2012), 'Bayesian correlated clustering to integrate multiple datasets', *Bioinformatics* **28**(24), 3290–3297.

Law, M. H. C., Figueiredo, M. A. T. and Jain, A. K. (2004), 'Simultaneous feature selection and clustering using mixture models', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(9), 1154–1166.

Law, M. H., Jain, A. K. and Figueiredo, M. (2003), Feature Selection in Mixture-Based Clustering, *in* S. Becker, S. Thrun and K. Obermayer, eds, 'Advances in Neural Information Processing Systems 15', MIT Press, pp. 641–648.

Li, D. and Shafto, P. (2011), Bayesian Hierarchical Cross-Clustering, *in* 'Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics', PMLR, Fort Lauderdale, FL, USA, pp. 443–451.

Liverani, S., Hastie, D. I., Azizi, L., Papathomas, M. and Richardson, S. (2015), 'PReMiuM: An R package for profile regression mixture models using Dirichlet processes.', *Journal of Statistical Software* **64**(7), 1–30.

Molitor, J., Papathomas, M., Jerrett, M. and Richardson, S. (2010), 'Bayesian profile regression with an application to the National Survey of Children's Health.', *Biostatistics* **11**(3), 484–498.

Murphy, K. P. (2007), Conjugate Bayesian analysis of the Gaussian distribution, Technical report.

Neal, R. M. (1999), 'Regression and classification using Gaussian process priors', *Bayesian statistics 6* pp. 475–501.

Neal, R. M. (2000), 'Markov chain sampling methods for Dirichlet process mixture models', *Journal of Computational and Graphical Statistics* **9**(22), 249–265.

Niu, D., Dy, J. G. and Jordan, M. I. (2010), Multiple Non-redundant Spectral Clustering Views, *in* 'Proceedings of the 27th International Conference on International Conference on Machine Learning', Omnipress, USA, pp. 831–838.

Niu, D., Dy, J. G. and Jordan, M. I. (2014), 'Iterative Discovery of Multiple Alternative Clustering Views', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **36**(7), 1340–1353.

Papathomas, M., Molitor, J., Hoggart, C., Hastie, D. and Richardson, S. (2012), 'Exploring data from genetic association studies using Bayesian variable selection and the Dirichlet process: application to searching for gene-gene patterns.', *Genetic epidemiology* **36**(6), 663–674.

Pitman, J. (2002), 'Combinatorial stochastic processes'.

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association* **66**(336), 846–850.

Rasmussen, C. E. and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, The MIT Press.

Rousseeuw, P. J. (1987), 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics* **20 IS -**, 53–65.

Shahbaba, B. and Neal, R. (2009), 'Nonlinear models using Dirichlet process mixtures', *Journal of Machine Learning Research (JMLR)* **10**, 1829–1850.

Sugar, C. A. and James, G. M. (2003), 'Finding the Number of Clusters in a Dataset', *Journal Of The American Statistical Association* **98**(463), 750–763.

Tadesse, M. G., Sha, N. and Vannucci, M. (2005), 'Bayesian Variable Selection in Clustering High-Dimensional Data', *Journal Of The American Statistical Association* **100**(470), 602–617.

Tibshirani, R. and Walther, G. (2005), 'Cluster Validation by Prediction Strength', *Journal Of Computational And Graphical Statistics* **14**(3), 511–528.

Yeung, K. Y., Haynor, D. R. and Ruzzo, W. L. (2001), 'Validating clustering for gene expression data.', *Bioinformatics (Oxford, England)* **17**(4), 309–318.

A Gibbs sampler algorithm

Here we present two Gibbs samplers for MVN and GP profile regression. These are based on algorithms 2 and 8 in Neal (2000). In these algorithms, the end of the burn-in period is determined via assessing sampled valued for convergence through comparison of the distribution of recent samples with that of preceding samples, yielding a value δ (Gelman and Rubin, 1992). Parameters are considered to have converged to the true posterior distributions when $\delta < \delta_{tol}$, where δ_{tol} is some pre-specified threshold.

See text for function definitions and Appendix C for parameter definitions. Note that these are not the algorithms implemented in PReMiuM: see Liverani et al. (2015) for details.

A.1 Algorithm 2 for multivariate normal response model

1. Initialise variables:
   (a) Calculate N, Q, R.
   (b) Set $s = 2$, $b = 1$; $\zeta_{r,q} = 0.5$ for $r = 1, ..., R$ and $q = 1, ..., Q$; C; μ_β; σ^2_β; $\delta = 10$; $\delta_{tol} = 1.01$; `iter`$= 1$; `maxIter`.
   (c) Generate random indicator vector $\mathbf{z} = (z_1, ..., z_N)$, $z_i \in \{1, ..., C\}$
   (d) Sample initial parameters $\phi_{c,q} \sim \text{Dir}(\zeta_{1,q}, ..., \zeta_{R,q})$; $\Sigma_c \sim \mathcal{W}^{-1}$; (R_0, ν_0); $\boldsymbol{\mu}_c|\Sigma_c \sim \mathcal{N}(\mu_0, \Sigma_c/\kappa_0)$; $\frac{\beta_d - \mu_\beta}{\sigma^2_{\beta_d}}|\mu_\beta, \sigma^2_{\beta_d} \sim t_7$
2. **while** $\delta > \delta_{tol}$ || `iter < maxIter`
   (a) **for** $i = 1 : N$
      i. Populate $h_{r,q} = \mathbb{1}_{[x_{i,q}=r]}$
      ii. **for** $c = 1 : (C + 1)$
         A. Calculate $\ell_c = \begin{cases} \prod_{r=1}^{R} \prod_{q=1}^{Q} \phi_{c,q}^{h_{r,q}} \cdot f_{\mathbf{y}}(y_i|\boldsymbol{\mu}_c, \Sigma_c, \mathbf{z}, \boldsymbol{\beta}, \mathbf{w}) & \text{if } c \leq C \\ \prod_{q=1}^{Q} \frac{\Gamma(\sum_{r=1}^{R} \zeta_{r,q})}{\prod_{r=1}^{R} \Gamma(\zeta_r)} \frac{\prod_{r=1}^{R} \Gamma(\zeta_r + h_{r,q})}{\Gamma(\sum_{r=1}^{R}(x_{r,q} + \zeta_{r,q}))} \cdot p(y_i|C+1) & \text{otherwise} \end{cases}$
         B. Calculate "weights" $\eta_c = \begin{cases} \sum_{j \neq i}^{N} \mathbb{1}_{[z_j=c]} & \text{if } c \leq C \\ \alpha & \text{otherwise} \end{cases}$

iii. Sample new z_i from $\eta \cdot \ell$

iv. Sort and relabel $\mathbf{z}$, $\boldsymbol{\phi}$, $\boldsymbol{\mu}$

v. Update C

(b) **for** $c = 1 : C$

i. Populate $h_{r,q}^{(c)} = \sum_j^N \mathbb{1}_{[x_{j,q}=r]} \cdot \mathbb{1}_{[z_j=c]}$

ii. Resample $\phi_{c,q} \sim \text{Dir}(\zeta_{1,q} + h_{1,q}^{(c)}, ..., \zeta_{R,q} + h_{R,q}^{(c)})$

iii. Resample $\Sigma_c|y \sim \mathcal{W}^{-1}\left(R_0 + S_c + \frac{\kappa_0 n_c}{\kappa_0 + n_c}(\bar{y}^{(c)} - \mu_0), \nu_0 + n_c\right)$

iv. Resample $\boldsymbol{\mu}_c|\Sigma_c, \mathbf{y} \sim \mathcal{N}\left(\frac{\kappa_0\mu_0 + n_c\bar{y}^{(c)}}{\kappa_0 + n_c}, \frac{\Sigma_c}{\kappa_0 + n_c}\right)$

(c) Sample $\xi \sim \text{Beta}(\alpha + 1, N)$

(d) Calculate $\pi_\xi : \frac{\pi_\xi}{1-\pi_\xi} = \frac{s+C-1}{N(b-\log(\xi))}$

(e) Sample $\alpha|\xi, C \sim \pi_\xi G(s + C, b - \log(\xi)) + (1 - \pi_\eta)G(s + C - 1, b - \log(\xi))$

(f) **for** $d = 1 : D$

i. Evaluate $T_1 = \prod_{i=1}^N p(y_i|\boldsymbol{\mu}, \Sigma, \mathbf{z}, \boldsymbol{\beta}, \mathbf{w}) \prod_{d=1}^D p(\beta_d)$

ii. Duplicate $\boldsymbol{\beta}^* = \boldsymbol{\beta}$

iii. Resample $\beta_d^* \sim \mathcal{N}(\beta_d, \sigma_{\beta_d}^2)$

iv. Evaluate $T^* = \prod_{i=1}^N p(y_i|\boldsymbol{\mu}, \Sigma, \mathbf{z}, \boldsymbol{\beta}^*, \mathbf{w}) \prod_{d=1}^D p(\beta_d^*)$

v. Calculate $A = \min\{1, T^*/T_1\}$

vi. Update $\beta_d = \begin{cases} \beta_d^* & \text{with probability } A \\ \beta_d & \text{with probability } 1 - A \end{cases}$

(g) **if** $\delta > \delta_{tol}$

i. Update β step size.

ii. Update δ (using β) following Gelman and Rubin (1992) (CODA).

(h) **if** $\delta < \delta_{tol}$

i. `iter=iter+1`

A.2 Algorithm 8 for Gaussian process response model

1. Initialise variables:

(a) Calculate N, Q, R.

(b) Set $s = 2$, $b = 1$; $\zeta_{r,q} = 0.5$ for $r = 1, ..., R$ and $q = 1, ..., Q$; C; μ_β; σ_β^2; s_β; $\delta = 10$; $\delta_{tol} = 1.01$; U.

(c) Generate random indicator vector $\mathbf{z} = (z_1, ..., z_N)$, $z_i \in \{1, ..., C\}$

(d) Sample initial parameters $\phi_{c,q} \sim \text{Dir}(\zeta_{1,q}, ..., \zeta_{R,q})$; $\frac{\beta_d - \mu_\beta}{\sigma_{\beta_d}^2} \Big| \mu_\beta, \sigma_{\beta_d}^2 \sim t_7$; $\boldsymbol{\theta}_c = \{a_c, l_c, \sigma_c^2\}$

2. **while** $\delta > \delta_{tol}$ || `iter < maxIter`

(a) **for** $i = 1 : N$

i. Populate $h_{r,q} = \mathbb{1}_{[x_{i,q}=r]}$

ii. Sample $\boldsymbol{\theta}_c$, $\boldsymbol{\phi}_c$ for $c = C + 1, ..., C + U$

iii. **for** $c = 1 : (C + U)$

A. Calculate $\ell_c = \prod_{r=1}^R \prod_{q=1}^Q \phi_{c,q}^{h_{r,q}} \cdot p(y_i|t_i, \boldsymbol{\theta}, c, \boldsymbol{\beta}, \mathbf{w})$

B. Calculate "weights" $\eta_c = \begin{cases} \sum_{j \neq i}^{N} \mathbb{1}_{[z_j = c]} & \text{if } c \leq C \\ \alpha / U & \text{otherwise} \end{cases}$

iv. Sample new z_i from $\eta \cdot \ell$

v. Sort and relabel $\mathbf{z}$, $\boldsymbol{\phi}$, $\boldsymbol{\theta}$

vi. Update C

(b) **for** $c = 1 : C$

i. Populate $h_{r,q}^{(c)} = \sum_{j}^{N} \mathbb{1}_{[x_{j,q}=r]} \cdot \mathbb{1}_{[z_j = c]}$

ii. Resample $\phi_{c,q} \sim Dir(\zeta_{1,q} + h_{1,q}^{(c)}, ..., \zeta_{R,q} + h_{R,q}^{(c)})$

iii. Update Σ_c and $\boldsymbol{\mu}_c$

iv. **for** j=1:3

A. Evaluate $T_1 = p\left(y^{(c)} | \boldsymbol{\theta}_c, \boldsymbol{\beta}\right) \cdot p(\hat{\boldsymbol{\theta}}_{c,j})$

B. Set $\hat{\boldsymbol{\theta}}_c^* = \hat{\boldsymbol{\theta}}_c$

C. Sample $\hat{\boldsymbol{\theta}}_{c,j}^* \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_{c,j}, s_{\boldsymbol{\theta}_j})$

D. Evaluate $T^* = p\left(y^{(c)} | \boldsymbol{\theta}_c^*, \boldsymbol{\beta}\right) \cdot p(\hat{\boldsymbol{\theta}}_{c,j}^*)$

E. Calculate $A = \min\{1, T^*/T_1\}$

F. Update $\hat{\theta}_{c,j} = \begin{cases} \hat{\theta}_{c,j}^* & \text{with probability } A \\ \hat{\theta}_{c,j} & \text{with probability } 1 - A \end{cases}$

(c) Sample $\xi \sim \text{Beta}(\alpha + 1, N)$

(d) Calculate $\pi_\xi : \frac{\pi_\xi}{1-\pi_\xi} = \frac{a+C-1}{N(b-\log(\xi))}$

(e) Sample $\alpha | \xi, C \sim \pi_\xi G(a + C, b - \log(\xi)) + (1 - \pi_\xi) G(a + C - 1, b - \log(\xi))$

(f) **for** $d = 1 : D$

i. Evaluate $T_1 = \prod_{c=1}^{C} p\left(y^{(c)} | \boldsymbol{\theta}_c, \boldsymbol{\beta}\right) \cdot p(\beta_d)$

ii. Duplicate $\boldsymbol{\beta}^* = \boldsymbol{\beta}$

iii. Resample $\beta_d^* \sim \mathcal{N}(\beta_d, s_{\beta_d})$

iv. Evaluate $T^* = \prod_{c=1}^{C} p\left(y^{(c)} | \boldsymbol{\theta}_c, \boldsymbol{\beta}^*\right) \cdot p(\beta_d^*)$

v. Calculate $A = \min\{1, T^*/T_1\}$

vi. Update $\beta_d = \begin{cases} \beta_d^* & \text{with probability } A \\ \beta_d & \text{with probability } 1 - A \end{cases}$

(g) **if** $\delta > \delta_{tol}$

i. Update s_β.

ii. Update s_θ.

iii. Update δ (using α) following Gelman and Rubin (1992) (CODA).

(h) **if** $\delta < \delta_{tol}$

i. `iter=iter+1`

B Conditionals for Gibbs sampling

B.1 Conditionals for the multivariate normal response model

The full joint distribution is

$$
\begin{aligned}
&p(\mathbf{x},\mathbf{y},\boldsymbol{\phi},\mathbf{z},\alpha,\boldsymbol{\mu},\Sigma) \\
&\quad = p(x_1,...,x_N,y_1,...,y_N,\phi_1,...,\phi_C,z_1,...,z_N,\alpha,\mu_1,...,\mu_C,\Sigma_1,...,\Sigma_C) \\
&\quad = p(x_1,...,x_N|z_1,...,z_N,\phi_1,...,\phi_C)\int p(z_1,...,z_N|\pi_1,...,\pi_C)p(\pi_1,...,\pi_C|\alpha)d\pi\cdot p(\alpha)\times \\
&\qquad p(\phi_1,...,\phi_C)\cdot p(y_1,...,y_N|z_1,...,z_N,\mu_1,...,\mu_C)\cdot p(\mu_1,...,\mu_C|\Sigma_1,...,\Sigma_C)\cdot p(\Sigma_1,...,\Sigma_C) \\
&\quad = p(\alpha)p(z_1,...,z_N|\alpha)\prod_{i=1}^{N}(p(x_i|z_i,\phi_{z_i})p(y_i|z_i,\mu_i))\prod_{c=1}^{C}(p(\phi_c)p(\mu_c|\Sigma_c)p(\Sigma_c))
\end{aligned}
$$

The joint posterior distribution, $p(\boldsymbol{\phi},\mathbf{z},\alpha,\boldsymbol{\mu},\Sigma|\mathbf{x},\mathbf{y})$, is sampled through sequential sampling of the conditional distributions for each parameter to infer. The conditionals are:

$$p(\boldsymbol{\phi}|\mathbf{z},\alpha,\boldsymbol{\mu},\Sigma,\mathbf{x},\mathbf{y}) \propto p(\phi_1,...,\phi_C)\cdot p(x_1,...,x_N|z_1,...,z_N,\phi_1,...,\phi_C) \tag{17}$$

$$p(\phi_c|\mathbf{z},\mathbf{x}) \propto p(\phi_c)\cdot\prod_{i:z_i=c}p(x_i|z_i,\phi_{z_i}) \tag{18}$$

$$
\begin{aligned}
p(\mathbf{z}|\boldsymbol{\phi},\alpha,\boldsymbol{\mu},\Sigma,\mathbf{x},\mathbf{y}) \propto & p(z_1,...,z_N|\alpha)\cdot p(x_1,...,x_N|z_1,...,z_N,\phi_1,...,\phi_C)\times \\
& p(y_1,...,y_N|z_1,...,z_N,\mu_1,...,\mu_C)
\end{aligned} \tag{19}
$$

$$p(z_i|\phi_{z_i},z_{-i},\alpha,\mu_{z_i},x_i,y_i) \propto p(z_i|z_{-i},\alpha)\cdot p(x_i|z_i,\phi_{z_i})\cdot p(y_i|z_i,\mu_{z_i}) \tag{20}$$

$$p(\alpha|\boldsymbol{\phi},\mathbf{z},\boldsymbol{\mu},\Sigma,\mathbf{x},\mathbf{y}) = p(\alpha|\mathbf{z}) \propto p(\alpha)\cdot p(z_1,...,z_N|\alpha) \tag{21}$$

$$p(\boldsymbol{\mu}|\phi,\alpha,\mathbf{z},\Sigma,\mathbf{x},\mathbf{y}) \propto p(\mu_1,...,\mu_C|\Sigma_1,...,\Sigma_C)\cdot p(y_1,...,y_N|z_1,...,z_N,\mu_1,...,\mu_C) \tag{22}$$

$$p(\mu_c|\mathbf{z},\Sigma_c,y^{(c)}) \propto p(\mu_c|\Sigma_c)\cdot p(y^{(c)}|\mathbf{z},\mu_c) \tag{23}$$

$$p(\Sigma|\boldsymbol{\phi},\alpha,\mathbf{z},\boldsymbol{\mu},\mathbf{x},\mathbf{y}) \propto p(\Sigma_1,...,\Sigma_C)\cdot p(\mu_1,...,\mu_C|\Sigma_1,...,\Sigma_C) \tag{24}$$

$$p(\Sigma_c|\mu_c) \propto p(\Sigma_c)\cdot p(\mu_c|\Sigma_c) \tag{25}$$

B.2 Conditionals for the Gaussian process response model

When g is marginalised, the full joint distribution is

$$\begin{aligned}
&p(\mathbf{x},\mathbf{y},t,\mathbf{w},\boldsymbol{\phi},\mathbf{z},\alpha,\boldsymbol{\theta},\boldsymbol{\beta}) \\
&\quad= p(x_1,...,x_N,y_1,...,y_N,t_1,...,t_N,w_1,...,w_N,\phi_1,...,\phi_C,z_1,...,z_N,\alpha,\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_C,\boldsymbol{\beta}) \\
&\quad= p(\mathbf{x}|\mathbf{z},\boldsymbol{\phi})\int p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)d\boldsymbol{\pi}\cdot p(\boldsymbol{\phi})\cdot p(\alpha)\cdot p(\mathbf{w})\cdot p(\boldsymbol{\beta})\cdot p(t)\cdot p(\boldsymbol{\theta})\times \\
&\qquad\int p(\mathbf{y}|t,\mathbf{w},\mathbf{z},f_t,\boldsymbol{\theta},\boldsymbol{\beta})p(f_t|t,\boldsymbol{\theta})df_t \\
&\quad= p(\mathbf{x}|\mathbf{z},\boldsymbol{\phi})\cdot p(\mathbf{z}|\alpha)\cdot p(\boldsymbol{\phi})\cdot p(\alpha)\cdot p(\mathbf{w})\cdot p(\boldsymbol{\beta})\cdot p(t)\cdot p(\boldsymbol{\theta})\cdot p(\mathbf{y}|t,\mathbf{w},\mathbf{z},\boldsymbol{\theta},\boldsymbol{\beta}) \\
&\quad= \prod_{i=1}^{N}p(x_i|z_i,\phi_i)\cdot p(z_1,...,z_N|\alpha)\cdot\prod_{c=1}^{C}p(\phi_c)\cdot p(\alpha)\cdot\prod_{i=1}^{N}p(w_i)\cdot p(\boldsymbol{\beta})\times \\
&\qquad\prod_{i=1}^{N}p(t_i)\cdot\prod_{c=1}^{C}p(\boldsymbol{\theta}_c)\cdot p(\mathbf{y}|t,\mathbf{w},\mathbf{z},\boldsymbol{\theta},\boldsymbol{\beta})
\end{aligned}$$

The joint posterior distribution, $p(\boldsymbol{\phi},\mathbf{z},\alpha,\boldsymbol{\theta},\boldsymbol{\beta}|\mathbf{x},\mathbf{y},t,\mathbf{w})$, is sampled through sequential sampling of the conditional distributions for each parameter to infer. The conditionals are:

$$p(\boldsymbol{\phi}|\mathbf{z},\alpha,\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{x},\mathbf{y},t,\mathbf{w})\propto p(\phi_1,...,\phi_C)\cdot p(x_1,...,x_N|z_1,...,z_N,\phi_1,...,\phi_C) \tag{26}$$

$$p(\phi_c|\mathbf{z},x^{(c)})\propto p(\phi_c)\cdot\prod_{i:z_i=c}p(x_i|z_i,\phi_{z_i}) \tag{27}$$

$$\begin{aligned}
p(\mathbf{z}|\boldsymbol{\phi},\alpha,\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{x},\mathbf{y},t,\mathbf{w})\propto& p(z_1,...,z_N|\alpha)\cdot p(x_1,...,x_N|z_1,...,z_N,\phi_1,...,\phi_C) \\
&p(y_1,...,y_N|t_1,...,t_N,w_1,...,w_N,z_1,...,z_N,\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_C,\boldsymbol{\beta})
\end{aligned} \tag{28}$$

$$p(z_i|\phi_{z_i},z_{-i},\alpha,\boldsymbol{\theta}_{z_i},\boldsymbol{\beta},x_i,y_i,t_i,w_i)\propto p(z_i|z_{-i},\alpha)\cdot p(x_i|z_i,\phi_{z_i})\cdot p(y_i|t_i,w_i,z_i,\boldsymbol{\theta}_{z_i},\boldsymbol{\beta}) \tag{29}$$

$$p(\alpha|\boldsymbol{\phi},\mathbf{z},\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{x},\mathbf{y},t,\mathbf{w})=p(\alpha|\mathbf{z})\propto p(\alpha)\cdot p(z_1,...,z_N|\alpha) \tag{30}$$

$$\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{\phi},\mathbf{z},\alpha,\boldsymbol{\beta},\mathbf{x},\mathbf{y},t,\mathbf{w})\propto& p(\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_C)\times \\
&p(y_1,...,y_N|t_1,...,t_N,w_1,...,w_N,z_1,...,z_N,\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_C,\boldsymbol{\beta})
\end{aligned} \tag{31}$$

$$p(\boldsymbol{\theta}_c|\phi_c,\mathbf{z},\alpha,\boldsymbol{\beta},x^{(c)},y^{(c)},t^{(c)},w^{(c)})\propto p(\boldsymbol{\theta}_c)\cdot p(y^{(c)}|t^{(c)},w^{(c)},\mathbf{z},\boldsymbol{\theta}_c,\boldsymbol{\beta}) \tag{32}$$

$$\begin{aligned}
p(\boldsymbol{\beta}|\boldsymbol{\phi},\mathbf{z},\alpha,&\boldsymbol{\theta},\mathbf{x},\mathbf{y},t,\mathbf{w})=p(\boldsymbol{\beta}|\mathbf{z},\boldsymbol{\theta},\mathbf{y},t,\mathbf{w}) \\
&\propto p(\boldsymbol{\beta})\cdot p(y_1,...,y_N|t_1,...,t_N,w_1,...,w_N,z_1,...,z_N,\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_C,\boldsymbol{\beta})
\end{aligned} \tag{33}$$

C Definitions of parameters

Table 1: Parameters used in this document.

Label	Dimension(s)	Meaning
	Algorithms 2 & 8	
δ_{tol}	1	Convergence threshold
δ	1	Convergence measure
`iter`	1	Iteration counter
`maxIter`	1	Maximum number for `iter`
U	1	Number of phantom clusters
ℓ	$C+U$	Likelihood
η	$C+U$	Weights
T	2	Marginals
A	1	Marginal ratio
	Dimensions	
N	1	Number of subjects
Q	1	Number of covariates
R	1	Number of categories per covariate
D	1	Number of fixed effects
C	1	Number of clusters
M	N	Number of observation times
$\mathcal{M}$	C	Number of observation times per cluster
	Covariate model	
$\mathbf{x}$	$N \times R$	Covariate data
h	$R \times Q$	Tabled covariate data
$\boldsymbol{\phi}$	$(C+U) \times R \times Q$	Cluster-specific parameters for $\mathbf{x}$
ζ	$R \times Q$	Prior for covariates
	DP parameters	
α	1	Concentration parameter
G_0	1	Base distribution
$\boldsymbol{\pi}$	C	Cluster weights
a	1	Shape for α
b	1	Rate for α
π_η	1	Auxiliary variable for α
$\mathbf{z}$	N	Cluster allocations
n_c	1	Number of individuals in cluster c
	Fixed effects	
$\mathbf{w}$	$N \times D$	Fixed effects
$\boldsymbol{\beta}$	D	Fixed effect weights
μ_B	D	Prior mean for B
σ_B^2	D	Prior variance for B
s_B^2	3	B step size
	MVN parameters	
μ_0	1	Prior mean
κ_0	1	
ν_0	1	
R_0	$M \times M$	Inverse scale matrix
	GP parameters	
σ^2	C	Noise variance

a	C	Signal variance
l	C	Length-scale
$m^{(c)}$		GP mean function
$k^{(c)}$		Covariance function
m_0	$\mathcal{M}$	Mean function evaluated at time points
$K^{(c)}$	$\mathcal{M} \times \mathcal{M}$	Cluster-specific covariance function evaluated at time points
	Response model	
t	$M_i : i = 1, ..., N$	Number of observation times
y	$M_i : i = 1, ..., N$	Outcome
λ	$N \times M_i$	Individual mean
$\boldsymbol{\theta}_c$	$C + U$	Cluster-specific parameters for $\mathbf{y}$
$\boldsymbol{\mu}_c$	$C + U$	Cluster mean
Σ_c	$C + U$	Cluster covariance matrix

D Supplementary figures

E Implementation in PReMiuM

Here, we outline the inclusion of the response models in the software package PReMiuM. We treat each response model separately, considering the MVN response model in Section E.1 and the GP response model in E.2.

The R page for PReMiuM, https://github.com/robj411/Longitudinal-PReMiuM, contains both the package and detailed documentation.

E.1 Multivariate normal implementation

E.1.1 Input data

The data input to PReMiuM from the R interface are structured as before, with the `outcome` column now a sequence of columns `outcome1`, `outcome2`, `outcome3`, etc., with the additional requirement that these column names are supplied in the function call.

E.1.2 C++ back end

Within the C++ code for PReMiuM, functions have been added to evaluate likelihoods for the multivariate outcomes, and to sample new parameters following the distributions in Equation 34.

The posterior distributions from which we sample are (Murphy, 2007)

$$\Sigma_c|\mathbf{y} \sim \mathcal{W}^{-1}\left(R_c, \nu_0 + n_c\right), \qquad \mu_c|\Sigma_c, \mathbf{y} \sim \mathcal{N}\left(\frac{\kappa_0\mu_0 + n_c\bar{y}^{(c)}}{\kappa_0 + n_c}, \frac{\Sigma_c}{\kappa_0 + n_c}\right). \tag{34}$$

Here,

$$n_c = \sum_{i=1}^{N} \mathbb{1}_{[z_i=c]} \tag{35}$$

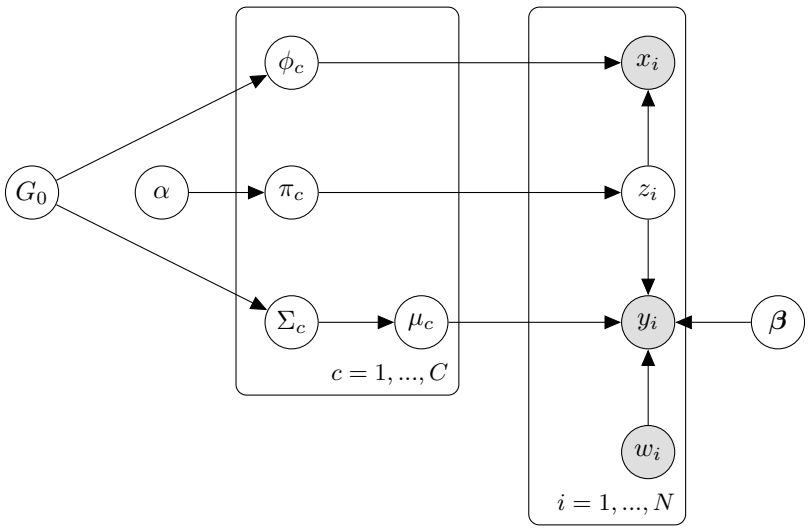

Figure 6: Plate diagram for the MVN response model as in Figure **??** with variable selection. There are Q γ parameters, one for each covariate, that indicate whether or not the variable is cluster dependent. Latent variables are shown in white and observed variables in grey.

is the number of subjects in cluster c,

$$\bar{y}^{(c)} = \frac{1}{n_c} \sum_{i=1}^{N} \left(\left(y_i - \boldsymbol{\beta}^T \mathbf{w}_i \right) \cdot \mathbb{1}_{[z_i = c]} \right) \tag{36}$$

is the mean vector of outcomes for subjects in cluster c,

$$C_c = \sum_{i=1}^{N} (y_i - \bar{y}^{(c)})(y_i - \bar{y}^{(c)})^T \mathbb{1}_{[z_i = c]} \tag{37}$$

is the scatter matrix for cluster c, and

$$R_c = R_0 + C_c + \frac{\kappa_0 n_c}{\kappa_0 + n_c} (\bar{y}^{(c)} - \mu_0)(\bar{y}^{(c)} - \mu_0)^T \tag{38}$$

is the updated scale matrix for cluster c.

The marginal likelihood is

$$p(y|c, \mu_c) = \frac{1}{\pi^{n_c M/2}} \frac{\Gamma_M((\nu_0 + n_c)/2)}{\Gamma_M(\nu_0/2)} \frac{|R_0|^{\nu_0/2}}{|R_c|^{(\nu_0 + n_c)/2}} \left(\frac{\kappa_0}{\kappa_0 + n_c} \right)^{M/2}. \tag{39}$$

E.1.3 Outputs

Additional output files are written for the parameters that are sampled at each iteration of the algorithm: μ and Σ. These files are used to construct the visualisation of the posterior parameter distributions, namely boxplots of mean and standard deviation for each variable within each cluster (see Figure 9 for an example).

These files also form the basis for prediction of a new individual's outcome(s) based on their covariate data. These plots have a similar form; see Figure 10 for an example. The other graphical output, the covariate profile, is unchanged with the exception of the omission of the 'risk' plot (see Figure 13).

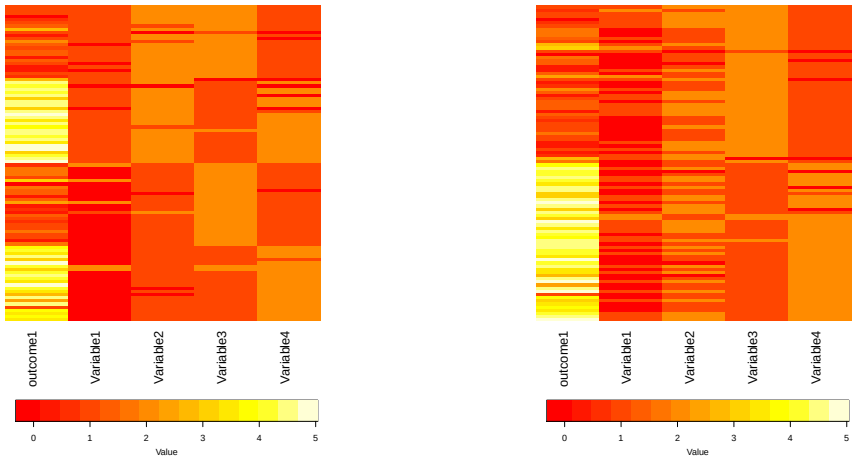

Figure 7: Simulated outcome and covariate data for the second simulation study (Section 4.2). Each row represents five measurements of one person. Of the four variables, the first two do not align with the outcome (left). The last two variables align with the outcome (right).

E.2 Gaussian process implementation

E.2.1 Input data

The R interface requires an additional data frame to be supplied, and a column to be added to the original '`data`' data frame. The additional data frame consists of a column of all the measurements, a column of the time points at which the measurements were taken, and a column of IDs, identifying from whom the measurements were taken. The ID column must correspond to the additional column in the original data frame so that covariate data can be correctly matched with outcome data.

E.2.2 C++ back end

Within the C++ code for PReMiuM, functions have been added to evaluate likelihoods for the longitudinal outcomes, and a Metropolis-within-Gibbs step that proposes new parameter values, and evaluates them through comparison of their conditional distributions, which have the form $p(\boldsymbol{\theta}_c) \cdot p(\mathbf{y}^{(c)}|t^{(c)}, \mathbf{w}^{(c)}, \boldsymbol{\theta}_c, \boldsymbol{\beta})$ (see algorithm 8 in Appendix A for a full example).

We infer the cluster-specific hyperparameters that define the mean and covariance functions. For their prior distributions, we use three independent log-normal distributions (Kirk et al., 2012):

$$\hat{\theta}_i \sim \mathcal{N}(\mu_{\theta_i}, \sigma^2_{\theta_i}); \tag{40}$$

$$\theta_i = \exp(\hat{\theta}_i). \tag{41}$$

The hyperparameters are resampled via Metropolis-within-Gibbs steps. The step sizes s_{θ_i} are updated in the same way as the fixed-effect coefficients in PReMiuM. All s_{θ_i} are initially set to 1 and we aim for an acceptance rate of 0.44 (Liverani et al., 2015).

In evaluating the hyperparameters $\boldsymbol{\theta}$, we marginalise $g^{(c)}(t)$ by conditioning on all the data points in the cluster (see Appendix B for the full conditionals). Equation **??** assumes $g^{(c)}(t)$ is sampled, and relies on conditional independence between measurements in a cluster given cluster

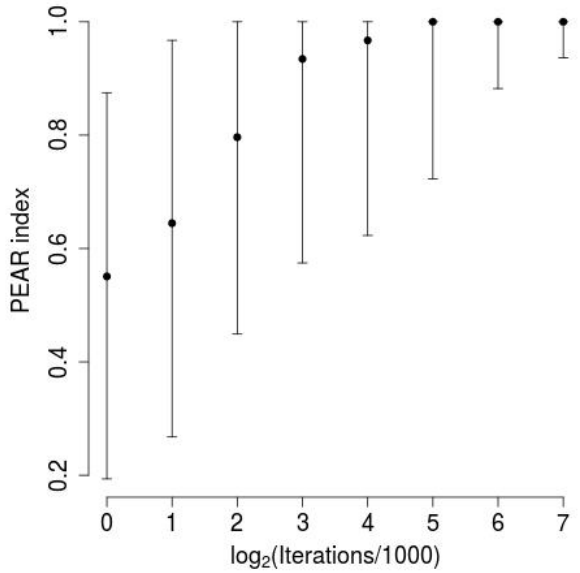

Figure 8: Efficacy of MVN model in recovering a true clustering structure as the number of iterations in PReMiuM changes. This plot corresponds to a single square in Figure 4: the top-right square of Figure 4A (top-left sub-figure). Each point represents the median of 1000 instances of inference of the clustering structure given data generated as described in Section 4.3, and error bars show 0.05 and 0.95 quantiles. The data-generating model is MVN, the response model is MVN, the number of time points is six, and the gradient of the second cluster's response mean is -1 (i.e. the responses of the two clusters are most clearly separable). Inference is trialled with 1000, 2000, 4000, 8000, 16000, 32000, 64000 and 128000 iterations in each of the burn-in and sampling phases. As the number of iterations increases, the true underlying clustering structure is better recovered, as evidenced by the increase in PEAR index.

membership. With marginalisation of $g^{(c)}(t)$, the likelihood becomes

$$f_{\mathbf{y}}(y_i)|\boldsymbol{\theta}, c, \boldsymbol{\beta}, w = p(y_i|\boldsymbol{\theta}_c, \boldsymbol{\beta}, w) \tag{42}$$

$$= \int p\left(y_i \left| g^{(c)}, \boldsymbol{\theta}_c\right.\right) p\left(g^{(c)} \left| y_{-i}^{(c)}, \boldsymbol{\theta}_c\right.\right) df_t^{(c)} \tag{43}$$

$$= \frac{\int p\left(y_i, y_{-i}^{(c)} \left| g^{(c)}, \boldsymbol{\theta}_c\right.\right) p\left(g^{(c)} \left| \boldsymbol{\theta}_c\right.\right) df_t^{(c)}}{\int p\left(y_{-i}^{(c)} \left| g^{(c)}, \boldsymbol{\theta}_c\right.\right) p\left(g^{(c)} \left| \boldsymbol{\theta}_c\right.\right) df_t^{(c)}} \tag{44}$$

where

$$\int p\left(y^{(c')} \left| g^{(c)}, \boldsymbol{\theta}_c\right.\right) p\left(g^{(c)} \left| \boldsymbol{\theta}_c\right.\right) df_t^{(c)} = \frac{1}{\sqrt{(2\pi)^{\mathcal{M}}|K_{c'}|}} \exp\left\{-\frac{1}{2}(y^{(c')} - \lambda^{(c')})K_{c'}^{-1}(y^{(c')} - \lambda^{(c')})^{-T}\right\} \tag{45}$$

and

$$\lambda^{(c')} = m_0^{(c')} + \boldsymbol{\beta}^T \mathbf{w}^{(c')}. \tag{46}$$

E.2.3 Outputs

An additional output file is written for the $\boldsymbol{\theta}$ parameters, which are sampled at each iteration of the algorithm. These values are used to reconstruct the posterior parameter distributions over $g^{(c)}$,

which are visualised as a function of time with 90% credible intervals shown on either side. The standard deviation is a sum of the standard deviation about the mean throughout all samples and the mean standard deviation of all samples. These plots are realised with and without the data in the background. See Figure 11 for examples.

The same procedure also forms the basis for prediction of a new individual's outcome based on their covariate data. These plots have a similar form; see Figure 12 for an example. The other graphical output, the covariate profile, is unchanged with the exception of the omission of the 'risk' plot (see Figure 13).

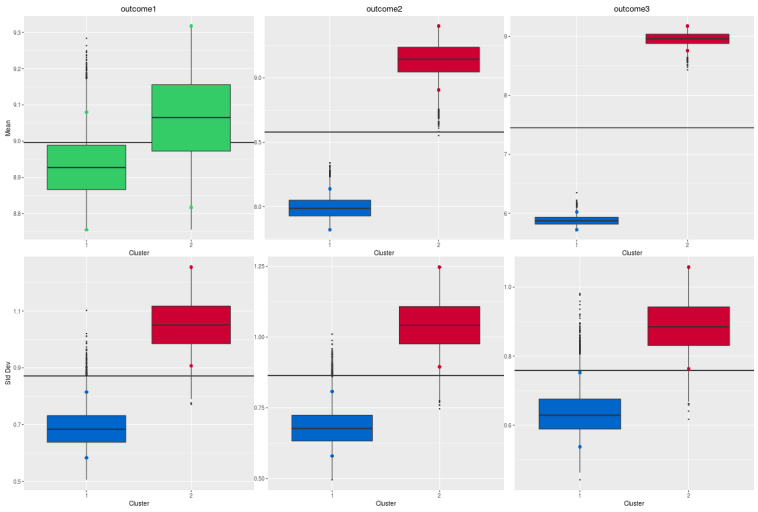

Figure 9: PReMiuM output for MVN response model.

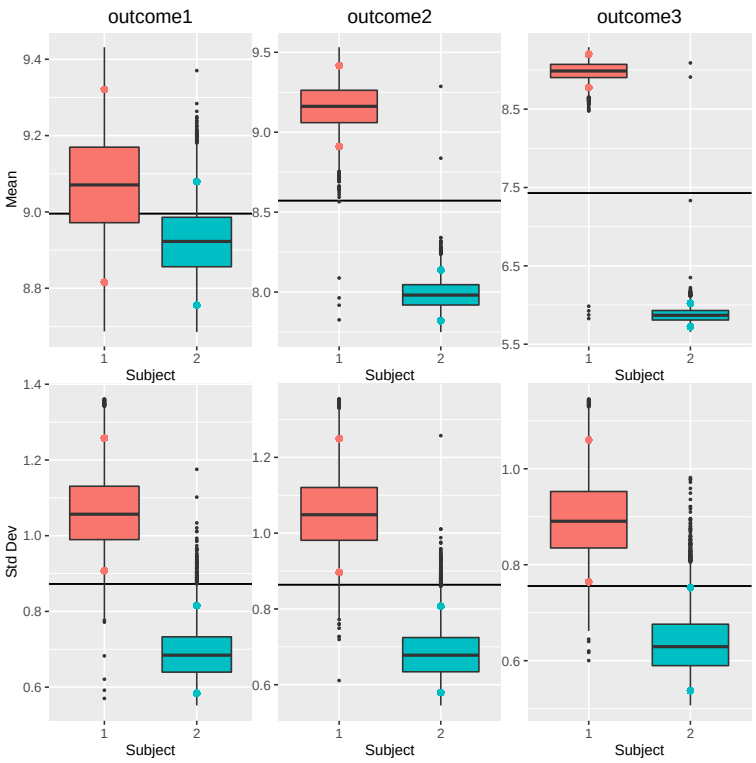

Figure 10: PReMiuM predictions for MVN response for two subjects.

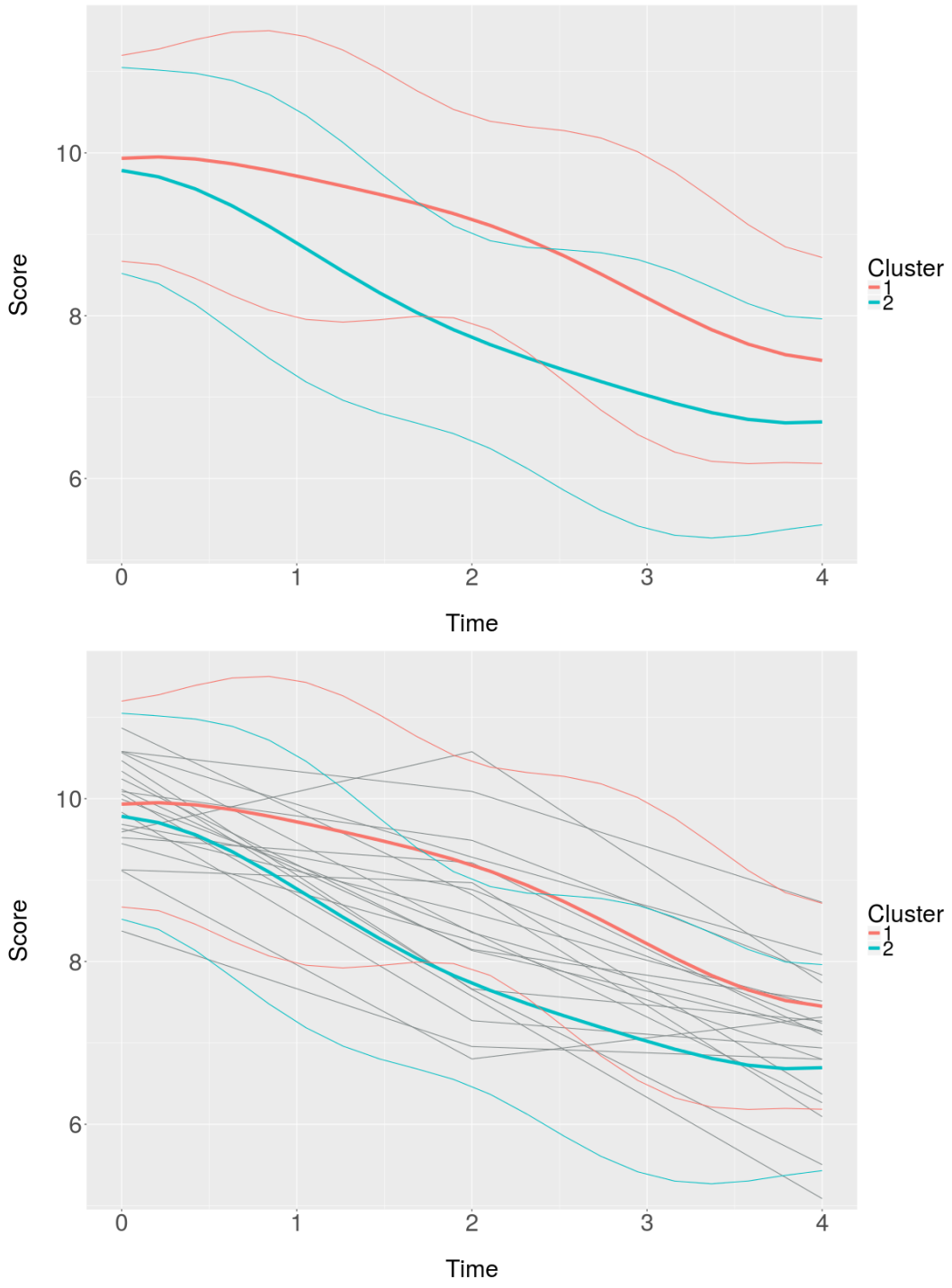

Figure 11: PReMiuM output for GP response model, without data (above), and with data (below). Bold lines show the posterior mean function. Thinner lines either side show 90% credible intervals.

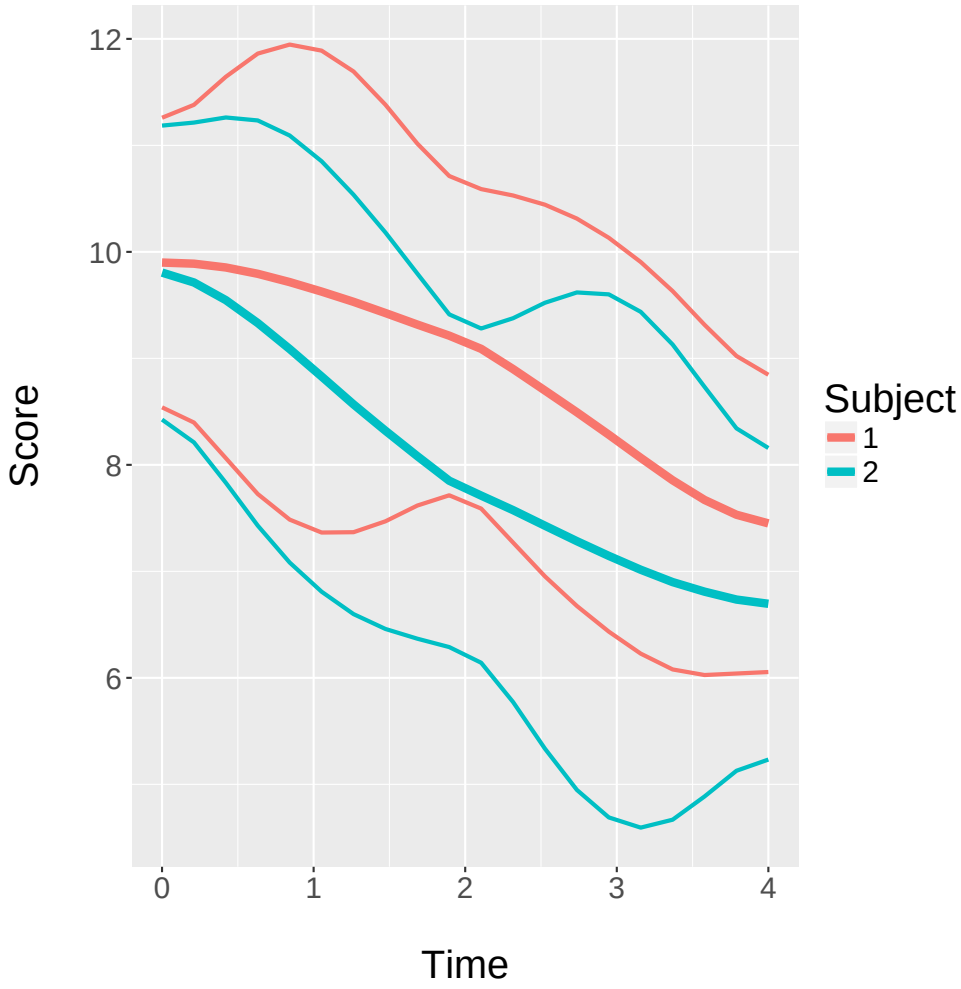

Figure 12: PReMiuM predictions for GP response for two subjects. Bold lines show the posterior mean function. Thinner lines either side show 90% credible intervals.

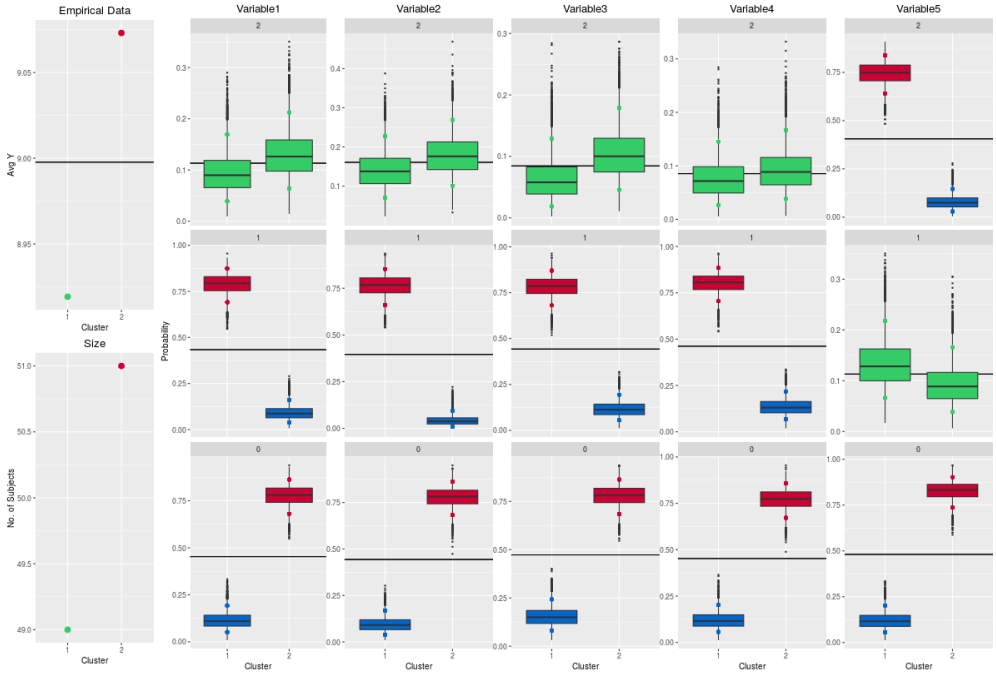

Figure 13: PReMiuM output for covariate data.