STAT 840: Classification of Longitudinal Trajectories by Co-clustering with Bootstrap Method

Faith Lee and Fangya Mao

April 21, 2018

Abstract

In longitudinal data setting, multiple observations are taken from an individual to allow us to potentially discover a variety of structural patterns and profiles of a dynamic marker. Conventionally, a linear mixed effects (LME) model with a overall common trajectory for all individuals would be utilized in such situations. While individuals exhibit different trajectory paths, individuals with similar characteristics are expected to share similar trajectory paths. Thus, a latent cluster variable accounting for trajectory characteristics can be added into the naive LME model to gain more power so to give a more accurate description of individual trajectories. In this paper, we propose a co-clustering procedure combined with Bootstrap method to subset the study population into clusters. Simulation studies are conducted to investigate the performance of the proposed algorithm and comparison results indicates that a modified LME model considered latent clusters obtained from our proposed algorithm performs better than the naive LME model.

1 Introduction

Disease progression can be assessed with longitudinal data where response measurements are obtained repeatedly over time and studied in conjunction with risk factors or bio-markers that may influence these responses. Longitudinal data analysis can facilitate the timing of therapeutic interventions and may provide an improved design in clinical trials. For example, to evaluate efficiency of medication on disease progression in Huntington's disease, one could take repeated measurements of cognitive and motor impairment [6]. Statistical analyses of longitudinal data requires model specifications that allow for accommodations to the nature of the data. Specifically, model specification should incorporate information from repeated measurements, consider the time intervals between measurements and take into account the correlation between measurements within each individual [10]. A possible and popular choice for model specification is the linear mixed effects (LME) model [11] due to its flexibility in handling a wide range of data types. Laird and Ware (1982) [8] proposed a LME model with

Gaussian distributions for within-individual error and random effects as follows:

$$Y_{ij} = x'_{ij}\beta + z_{ij}b_i + \epsilon_{ij} \tag{1}$$

where Y_i is a vector of n_i observations from the i-th individual, $\epsilon_i \overset{iid}{\sim} N_{n_i}(0, \sigma^2\mathbf{I})$ is a vector of intra-subject error terms, $\mathbf{X}_i$ is a $P \times n_i$ matrix of covariates with the corresponding population-level vector of regression coefficients represented by coefficient vector β (length of P). The term $b_i \overset{iid}{\sim} N_Q(0, \mathbf{\Sigma})$ accommodates for subject-specific random effects with $\mathbf{\Sigma}$ as a $Q \times Q$ covariace matrix. $\mathbf{Z}_i$ is the $Q \times n_i$ covariate matrix of random effects.

Although we account for subject-specific effects in the model (1), patients or individuals with similar medical condition may exhibit same temporal trends in their response measurements than individuals with different medical condition. Regression techniques that consider a clustered structure based on individuals with similar trajectories have shown to provide different estimates of regression estimates as compared to models that neglect this aspect ([14], [9]). A study has shown that the inclusion of cluster effect into the model provides better performance than models that do not include cluster specific effect [4]. Typically, the number of clusters and the clusters in which each individual belongs to are latent variables. The goal of our project is to cluster individuals based on the evolution of their responses (Y_i) and generate a modified linear mixed effects model accounting for clusters and compare it with the naive linear mixed effects model.

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm; simulation studies are conducted to illustrate the proposed approach and investigate its performance, including a comparison to the a naive linear mixed effect model in Section 3 ; Finally, conclusions with discussions about the limitations of the proposed method and future work we are considering are summarized in Chapter 4.

2 Methodology Construction

2.1 Preprocess: LME Model Introduction

In Equation (1), we have a generic formulation of the linear mixed effects model. Since Y_i is apparently time-dependent, thus, the most intuitive idea is to ultilize the intercept and the slope of Y_i with respect to time as the trajectory characteristics considered in clustering procedure. With this principal, an alternative to the representation in (1) is given by:

$$Y_{ij} = \sum_{p=1}^{P} x'_{ip}\gamma_p + \alpha_i + \alpha_0 + (\beta_T + \beta_i)t_{ij} + \epsilon_{ij} \tag{2}$$

where α_0 refers to a fixed intercept effect, γ_p refers to the fixed slope effect for a particular time-fixed covariate x_{ip}; β_T refers to the fixed slope effect corresponding to changes in time;

And α_i, β_i is the random intercept effect and random slope effect respectively for the i-th individual. Denote $\boldsymbol{\theta_0} = [\alpha_0, \beta_T]$, $\boldsymbol{\theta}_i = (\alpha_i, \beta_i)$ and assume

$$\boldsymbol{\theta}_i \overset{iid}{\sim} N(\begin{bmatrix} \mu_\alpha \\ \mu_\beta \end{bmatrix}, \begin{bmatrix} \sigma^2_\alpha & \sigma^2_{\alpha\beta} \\ \sigma^2_{\alpha\beta} & \sigma^2_\beta \end{bmatrix}) = N(\boldsymbol{\mu}, \Sigma) \tag{3}$$

and $\epsilon_i \overset{iid}{\sim} N_{n_i}(0, \sigma_i^2 \mathbf{I})$ as before.

In the previous formulation (1), time-dependent measures of covariates were represented through x_{ij}. In this formulation, we consider baseline measures of covariates (x_i) and then represent its time-varying nature through β_T.

Let Λ_i denote the component that is time-independent and Γ_i denote the component that is time-dependent for individual i from Equation (2), that is, $\Lambda_i = \alpha_0 + \alpha_i + X_i^T \gamma$ and $\Gamma_i = \beta_0 + \beta_i$. We adopt a clustering technique, known as co-clustering based on (Λ_i, Γ_i) to estimate the cluster of individual i.

2.2 Co-Clustering via Bootstrap ("BootCluster")

Random effects have been described to be unobserved characteristics coming from individuals and in (3), we denote the distribution of random effects to depend on the parameters $\boldsymbol{\mu}$ and Σ. In order provide an estimation to these parameters, one method of doing so is via bootstrapping, where we repeatedly re-sample individuals in our study and estimate these parameters of interest. Given a data set coming from N individuals observed at a common set of time points $(t_{i1}, ..., t_{im})$, number of B Bootstrap samples are drawn. With each of Bootstrap sample, first of all, a naive LME model (2) is used to fitted to extract the trajectory characteristics $\{(\Lambda_i, \Gamma_i); i = 1, ..., N\}$ (denoted as $(\boldsymbol{\Lambda}, \boldsymbol{\Gamma})$) from the longitudinal data set. The `R` package `nlme` contains the `lme` function that fits linear mixed effects models [12].

Then, the `blockcluster` package [3] can be utilized on $(\boldsymbol{\Lambda}, \boldsymbol{\Gamma})$, i.e., it partitions the "data set" of trajectory characteristics $(\boldsymbol{\Lambda}, \boldsymbol{\Gamma})$ into K blocks. Thus, a sample of clusterings $c^{(1)}, c^{(2)}, \ldots, c^{(B)}$ is the output from the co-clustering method with Bootstrap method. To summarize the sample with a single clustering estimate $\hat{c}$, we use the mode of estimated clusters for each individual as

$$\hat{c}_i = \max(\sum_{b=1}^{B} I_{\{c_i^{(b)} = k\}}, k = 1, \ldots, K). \tag{4}$$

Once the cluster of each individual has been derived from our proposed procedure, we fit an LME model to each cluster.

The pseudo-code for this methodology with more details is summarized as follows:

Algorithm: Co-Clustering via Bootstrap ("BootCluster")

```
Data: Given observations of N subjects $\{(Y_i, X_i); i = 1, ..., N\}$   $(\rightarrow (\mathbf{Y}, \mathbf{X}))$
Result: Obtain $\hat{\mathbf{C}}$ the predicted cluster sample
(Sample with replacement B samples from N individuals $\{(\mathbf{Y}^{(b)}, \mathbf{X}^{(b)}); b = 1, ..., B\}$)
for $b$ in 1:$B$ do
    • Derive $(\hat{\theta}_0^{(b)}, \hat{\gamma}^{(b)}, \hat{\mu}^{(b)}, \hat{\mathbf{\Sigma}}^{(b)})$ by LME model (2);
    • Draw random-effect-matrix $\mathbf{\Theta}^{(b)} \sim N(\hat{\mu}^{(b)}, \hat{\mathbf{\Sigma}}^{(b)})$, where $\mathbf{\Theta}^{(b)} = (\theta_{\mathbf{1}}^{(\mathbf{b})}, ..., \theta_{\mathbf{n}}^{(\mathbf{b})})$
      and then obtain $(\mathbf{\Gamma}^{(b)}, \mathbf{\Lambda}^{(b)})$;
    • Obtain cluster sample $\hat{\mathbf{C}}^{(b)}$ by co-clustering (on a combined $N \times 2$ matrix of
      $(\mathbf{\Gamma}^{(b)}, \mathbf{\Lambda}^{(b)})$).
end
for $i$ = 1, ..., $N$ do
    $\hat{c}_i = \max_k(\sum_{b=1}^{B} \mathrm{I}(c_i^{(b)} = k)$, k = 1, ..., K)
end
```

3 Results

3.1 Data Generation

In the generation of datasets, we considered two scenarios to investigate the robustness of the algorithm performance. One scenario where cluster-specific trajectories are separated visibly (referred as 'visible') and the other where there is significant overlap between the trajectories across clusters (referred as 'non-visible').

In each scenario, sample size of 200 individuals were assigned to 2 different clusters (the truth) with $P = 2$ baseline covariates (x_1, x_2). Each cluster has a cluster-specific mean trajectory which depends on different slopes and intercepts (represented by fiexed effects $(\alpha_{k0}, \beta_{kT}); k = 1, 2)$. Individual trajectories are assumed to share a common variation across clusters. In other words, there is no subscript k for random effect (α_i, β_i). The trajectories coming generated from these two scenarios are shown in Figure 1. (See Appendix 5.1.1 for more details about the settings)

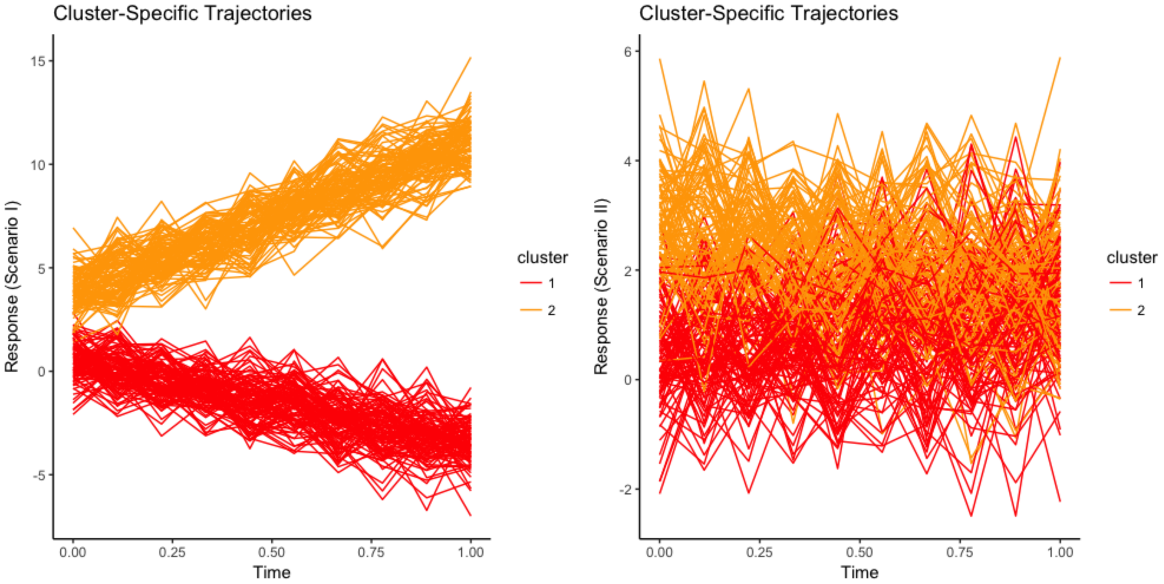

Figure 1: Scenario I: Trajectories with visible clusters (Left); Scenario II: Trajectories with significant overlap across clusters (Right)

3.2 Results and Evaluations

We apply the proposed "BootCluster" algorithm on the simulated data set under Scenario I and II illustrated in Section 3.1. And once individuals have been clustered, a modified LME model using these predicted clusters (*lme.bc*), a naive LME model without clustering (*lme.ori*) and a LME model using the true clusters (*lme.tc*) are fitted.

```
lme.ori <- lme(fixed = response~ x1 + x2 + time, random = ~ 1 + time|id, data
    ↪ = data_set ,control = ctrl, method = "ML")

lme.tc <- lme(fixed = response~ (x1 + x2 + time)*true_cluster, random = ~ 1 +
    ↪ time|id, data = data_set, control = ctrl, method = "ML")

lme.bc <- lme(fixed = response~ (x1 + x2 + time)*B_clust, random = ~ 1 + time|
    ↪ id, data = data_set, control = ctrl, method = "ML")
```

In order to compare the performances among these models, we adopt likelihood ratio test (LRT) and information criterion, namely, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) scores. LRT is used to assess the goodness of fit the the statistical mdoel; AIC and BIC are means of assessing the relative quality of models [13]. As a brief summary, Akaike [1] demonstrated that model selection can be determined by the AIC score defined as

$$AIC = 2K - 2\log(\ell(\hat{\theta}|y)) \tag{5}$$

where K is the number of estimable parameters and $\log(\ell(\hat{\theta}|y))$ refers to the log-likelihood of the model estimated, which is LME fit using the true cluster, LME fit using the predicted cluster and the original LME as in (1). BIC is closely related to the AIC and is defined as

$$BIC = K\log(N) - 2\log(\ell(\hat{\theta}|y)) \tag{6}$$

where N is the number of observations. In the LME fit post-clustering, we use maximum likelihood to estimate the parameters. For the AIC, the -2 log likelihood is penalized by 2K. For the BIC it is penalized by log(N)K which is a harder penalization than AIC where large sample sizes are concerned.

Since the number of clusters K is "unknown" in practice, a set of $n\_cluster(= 2, 3, 4, 5)$ are specified to explore the performance of the proposed algorithm. And the corresponding model comparison results are displayed in Table 1. According to the results of LRT conducted between $lme.ori$ and $lme.bc$, we always reject the null $lme.ori$, which suggests that $lme.bc$ has better fitness with the data. And definitely, the "true" model $lme.tc$ has the best performance in terms of AIC, BIC and log-likelihood.

				Scenario I			
	df	AIC	BIC	logLik	Test	L.Ratio	p-value
$n\_cluster = 2$							
lme.ori	8	6796.439	6841.246	-3390.219			
lme.bc	12	6581.992	6649.203	-3278.996	1 vs 2	222.4464	$<$.0001
lme.tc	12	5900.270	5967.481	-2938.135			
$n\_cluster = 3$							
lme.bc	12	6531.135	6598.346	-3253.568	1 vs 2	273.3035	$<$.0001
$n\_cluster = 4$							
lme.bc	12	6603.097	6670.308	-3289.549	1 vs 2	201.3413	$<$.0001
$n\_cluster = 5$							
lme.bc	12	6573.215	6640.426	-3274.608	1 vs 2	231.2237	$<$.0001
				Scenario II			
	df	AIC	BIC	logLik	Test	L.Ratio	p-value
$n\_cluster = 2$							
lme.ori	8	6041.604	6086.411	-3012.802			
lme.bc	12	5990.873	6058.084	-2983.436	1 vs 2	58.73119	$<$.0001
lme.tc	12	5900.270	5967.481	-2938.135			
$n\_cluster = 3$							
lme.bc	12	5987.851	6055.062	-2981.926	1 vs 2	61.75266	$<$.0001
$n\_cluster = 4$							
lme.bc	12	5961.604	6028.815	-2968.802	1 vs 2	87.99948	$<$.0001
$n\_cluster = 5$							
lme.bc	12	5958.097	6025.308	-2967.049	1 vs 2	91.50685	$<$.0001

Table 1: Model comparisons among *lme.ori*, *lme.bc*, and *lme.tc*

Besides, we present trajectories with and without the clustering procedure in Figure 2 in Scenario II as a supplement to show the emprical clustering performance of "BootCluster". According to Figure 2, we can see that although it is "visibly" difficult to separate individuals into groups in Scenario II, the implementation of "BootCluster" Algorithm (with 2 estimated clusters used, i.e., $n\_cluster = 2$) provides comparable results in comparison to the true clusters, except for a few mis-specifications indicated in black. This is an indication of the usefulness of our proposed ‘BootCluster’ algorithm.

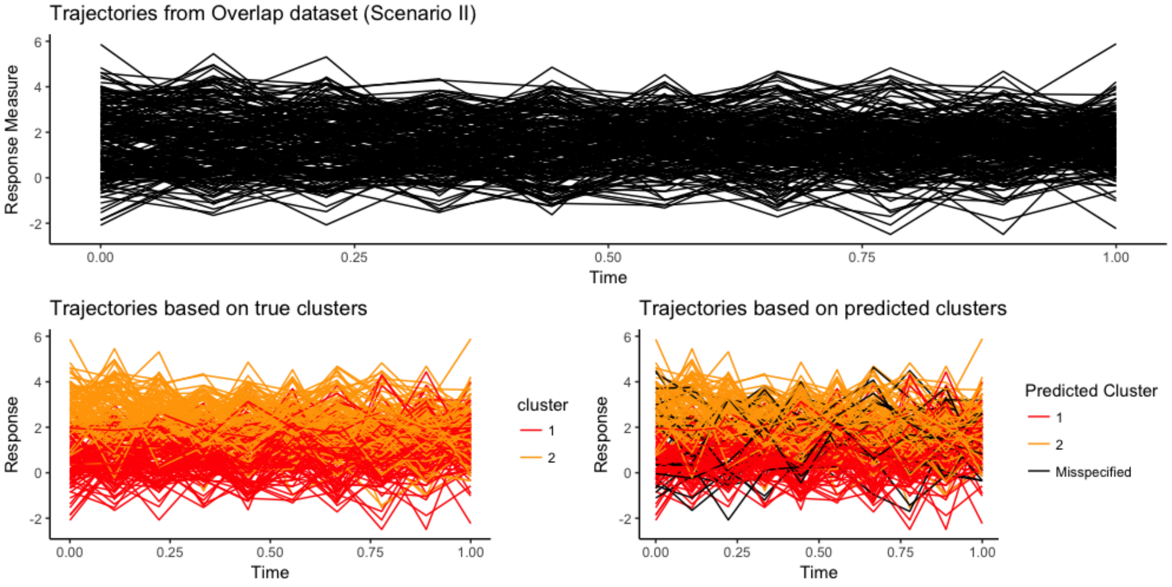

Figure 2: Trajectories from Overlap dataset, no visible detection of clusters (Top); Trajectories based on true cluster (Bottom Left); Trajectories based on predicted clusters using "BootCluster" algorithm (Bottom Right)

Although results shown above seem to be good and reasonable in general, we notice that 3 patterns shown in Table 1 might due to the generation of the data sets or the settings of the built-in functions from the R packages we use. Here are some elaborations on them.

- *under both scenarios lme.bc always returns the same df regardless of the number of latent clusters (i.e., n_cluster) we specify in our function, which means that the study sample is always partitioned into 2 clusters no matter what n_cluster is specified as.*

  The main reason might be the inherent nature of our generated data sets, in which only two clusters specified and each has rather simple mean trajectories only featured by time-independent (intercept) and time-dependent (slope with respect to time) characteristics. And it might also be associated with what kind of the settings we specified in `coclusterContinuous()` function from `blockcluster` package. (See Appendix for more details of the specifications)

- *Although the study sample is always partitioned into 2 clusters no matter what n_cluster is, the performance of lme.bc varies with different n_cluster choices.*

  From the results based on the two datasets, we find that it relates to the "accuracy" of the predicted clusters. For example, we can see the performance of *lme.bc* improves (with smaller AIC, BIC and etc.) when *n_cluster* increases. And we find it corresponds to the increasing "accuracy" rate. The results from Scenario II also support this "finding".

- *the results of lme.tc are same under Scenario I and II in Table 1.*

We like to highlight in both datasets, the baseline covariates are the same but their responses are not. Likelihood is the same as individuals in both datasets are observed at common time points over [0,1] and that their baseline covariates are the same. Data generation essentially differ in the way their trajectories change with time. Under an ML fit, the contribution to the likelihood for both datasets is therefore the same.

4 Discussion

Using cluster analysis as our motivation, individuals with similar trajectory paths could be classified into specific groups where treatment plans could be designed specifically for similar groups of individuals. We propose to cluster individuals based on their time-independent and time-dependent characteristics of the response trajectories ($(\mathbf{\Lambda}, \mathbf{\Gamma})$). The core step of the proposed algorithm is to obtain the $(\mathbf{\Lambda}, \mathbf{\Gamma})$ which are used in the clustering procedure from combination of Bootstrap method and lme model.

And the results shown in Section 3 indicates that separating individuals into clusters and assigning individuals in the same cluster with a modified LME model considering cluster-specific fixed effects provides a better alternative as compared to assigning all individuals with a common linear mixed effects model.

Examples from other studies ([2],[7]) have shown that regression coefficients can be different when we take into consideration the clustering of similar units. Although estimates of regression coefficients is not of direct interest in our investigation here, clustering takes into account the correlation between similar units (or individuals) and as shown in our results, LME models with cluster-specific fixed effects provides a better model specification as compared to LME models that neglect the correlation between similar individuals. In our paper, we have ultilized the interent heterogeneity natures of the data set as a way to partition individuals into clusters. We note that there are many clustering techniques other than co-clustering the co-clustering procedure such as K-Means clustering for example. Future work could also assess the different methods of clustering and/or clustering based on other measures of individual characteristics. We like to highlight that we attempted to apply the Bayesian approach to estimate the fixed and random effects using Markov Chain Monte Carlo (MCMC). However, due to the limited time, we includes the incomplete work we have done for now on this algorithm in the Appendix 5.2 and this will also be part of our future work.

We like to highlight some limitations to our approaches. First of all, the datasets here are dense datasets and all individuals are assumed to be observed at equal time points with no censoring or drop-out issues. This is seldom plausible in real-world situations and there may be issues such as censoring of individuals as well in longitudinal studies that should be taken into consideration. Secondly, we assume in our model that responses coming from individuals is a linear function of time. We could explore scenarios where the datasets are sparse and whereby there are more inherent clusters. The rationale behind our methodology comes from the fact that characteristics contributed by individuals can be classified as time-dependent and time-independent. Using these information is clustering of longitudinal data is therefore suitable and we demonstrated through simple analyses that some form of clustering similar individuals provides a better model than assuming common effects for all individuals in the linear mixed effects model.

5 Appendix

5.1 Our R package "BootCluster" documentation

5.1.1 Datasets

- Non-overlap (Scenario I)
- Overlap (Scenario II)

Data sets available in our R package are the ‘Non-overlap’(Scenario I) and the ‘Overlap’ (Scenario II). The R code for the generation of the datasets and Figure 1 is available in the package. To be specific in how the datasets were generated, we randomly assign N = 200 individuals into 2 clusters. Depending on the assignment of clusters, we generate 2 covariates $X_1 \sim Unif(a, b)$ and $X_2 \sim Bin(N, \tau)$ where the values of (a, b, τ) are cluster-dependent. In our dataset, for simplicity we assume that all individuals are observed between [0, 1] at common time points. The fixed intercepts α_0 and fixed slopes β_T are cluster-dependent as well, they are generated from a uniform distribution. For the random intercept and random slope effect, they come from the distribution

$$\boldsymbol{\theta}_i \overset{iid}{\sim} N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix})$$

‘Non-overlap’ dataset was created with ‘visible’ separation of individuals coming from different clusters whereas ‘Overlap’ dataset was created with ‘non-visible separation’ with the main purpose of testing the feasibility of our proposed ‘BootCluster’ methodology. The remaining of our functions and their descriptions are in the R package.

The code for our dataset (‘Non-overlap’) generation are as follows:

```
library(ggplot2)
library(dplyr)
library(gridExtra)
set.seed(840)
N = 200 #Number of individuals
k = 2 #Number of clusters
T <- seq(0,1,length.out = 10)
```

```
n_obs = length(T) #number of observations per individual

#Assignment of clusters
clusters <- c(1:k, sample(c(1:k), N - k , replace = TRUE))

beta.target <- matrix(c(runif(1,0,1), runif(2,0,1), -runif(1,3,4),
                    runif(1,1.8,3), runif(2,0,1), runif(1, 5, 8)), ncol = 4,
                        ↪ nrow = 2, byrow = TRUE)
#Random intercept; random slope

mu_a = 0
mu_b = 0
V_a = .5
V_b = .5

X<-matrix(0, nrow = N, ncol = 3)
X1<-rep(0, N)
X2<-rep(0, N)
yt<-c()
prob_x1_min<-c(0, 1.5, 2.5, 4)
prob_x1_max<-c(1, 3, 5, 7)
prob_x2<-c(0.8, 0.2, 0.6, 0.7)
sigma_targ<- 1
for (i in 1:N){
  a_i = rnorm(n = 1, mean = mu_a, sd = V_a)
  b_i = rnorm(n = 1, mean = mu_b, sd = V_b)
  X1[i] = runif(1, min = prob_x1_min[clusters[i]], max = prob_x1_max[clusters[
      ↪ i]])
  X2[i]<-rbinom(1, 1, prob_x2[clusters[i]])
  X[i,] <-c(beta.target[clusters[i],1], X1[i], X2[i])
  fixed_effects <- beta.target[clusters[i],1] + X1[i]*beta.target[clusters[i
      ↪ ],2] + X2[i]*beta.target[clusters[i],3]
  means = (beta.target[clusters[i],4]+ b_i)*T + fixed_effects
  yt<-c(yt, rnorm(n_obs, mean = means, sd = sigma_targ))
}
```

```
Non_overlap = data.frame(id = rep(1:N, each = n_obs), time = rep(T, N),
                   X1 = rep(X1, each = n_obs), X2= rep(X2, each = n_obs),
                   response = yt, cluster = rep(clusters, each = n_obs))
```

5.2 Algorithm II: Co-Clustering via MCMC (INCOMPLETE)

Other than obtaining estimates of the random effects by bootstrapping which is a frequentist approach, we can also adopt a Bayesian approach for co-clustering procedure by using Markov Chain Monte Carlo (MCMC). According to Section 2.1, we have

$$Y_i|\boldsymbol{\theta}_i, \boldsymbol{\theta}_0, \boldsymbol{\gamma}, \sigma_i^2 \overset{iid}{\sim} N(d_i'\boldsymbol{\theta_i} + d_i'\boldsymbol{\theta_0} + \mathbf{x}_i'\boldsymbol{\gamma}, \sigma_i^2\mathbf{I}) \text{ and } \boldsymbol{\theta}_i \overset{iid}{\sim} N(\boldsymbol{\mu}, \Sigma) \tag{7}$$

where $Y_i = [Y_{i1}, ..., Y_{in_i}]$, $\boldsymbol{\theta_0} = [\alpha_0, \beta_T]$, $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_p]$, $\mathbf{x}_i = [x_{i1}, ..., x_{ip}]$ and $d_i = \begin{bmatrix} 1 & t_{ij} \end{bmatrix}$. Using the Law of Total Expectation and the Law of Total Variance, the marginal distribution of $\mathbf{Y}$ is also a multivariate normal distribution

$$\begin{aligned} \mu_{\mathbf{Y}} &= E(E(Y_i|\boldsymbol{\theta}_i)) \\ &= \mathbf{d}'\boldsymbol{\mu} + \mathbf{d}'\boldsymbol{\theta}_0 + \mathbf{X}'\boldsymbol{\gamma} \\ &= \begin{bmatrix} \mathbf{d}' & \mathbf{d}' & \mathbf{X}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\theta}_0 \\ \boldsymbol{\gamma} \end{bmatrix} \\ &= \mathbf{w}'\boldsymbol{\lambda} \end{aligned} \tag{8}$$

and

$$\begin{aligned} \text{Var}_{\mathbf{Y}} &= \text{E}(\text{Var}(Y_i|\boldsymbol{\theta}_i)) + \text{Var}(\text{E}(Y_i|\boldsymbol{\theta}_i)) \\ &= \text{E}(\sigma_i^2\mathbf{I}) + \text{Var}(\mathbf{d}'\boldsymbol{\Theta} + \mathbf{d}'\boldsymbol{\theta_0} + \mathbf{X}'\boldsymbol{\gamma}) \\ &= \sigma^2\mathbf{I} + \mathbf{d}\Sigma'\mathbf{d}' \end{aligned} \tag{9}$$

and hence the marginal distribution of $\mathbf{Y}$ has a tractable form $\mathbf{Y} \sim N(\mathbf{w}'\lambda, \mathbf{d}\Sigma'\mathbf{d}' + \sigma^2\mathbf{I})$. We used the two-level hierarchical modeling since we are interested in estimating both $\boldsymbol{\Theta}$ and $\boldsymbol{\theta}_0$.

We implement a Bayesian framework using Gibbs sampling. Gibbs sampling simulates a new value for each parameter (or block of parameters) in turn from its conditional distribution under the assumption that current values for these parameters are true. We assign the following prior distributions to our parameters as in [5]:

$$\begin{aligned} \Sigma &\sim IW_q(v_0^{-1}R_0, v_0), \text{ q} = 2 \\ \sigma^2 &\sim I\Gamma(v_{00}/2, \delta_{00}/2) \\ \beta_0, \beta_T, \gamma_p &\sim N_{p+2}(\boldsymbol{\Omega}_0, B_0), \text{ p} = 1, ..., \text{P} \end{aligned} \tag{10}$$

where IW refers to the Inverse Wishart distribution and $I\Gamma$ refers to the Inverse Gamma distribution. The rationale in implementing Gibbs sampling is that firstly, we have Gaussian structure for our model and secondly, conditional conjugacy of the above prior specifications enables an analytical and tractable closed form of the full conditional distribution as in Algorithm II in (4).

```
Algorithm: Co-Clustering via MCMC (Gibbs)

1. Initialize $\gamma_1^{(0)}, ..., \gamma_p^{(0)}, \beta_T^{(0)}, \Sigma^{(0)}, (1/\sigma^2)^{(0)}, \theta_i^{(0)}$ ;
2. for $m$ = 1, ..., $M$ do
    (a) Sample $\gamma_1^{(m)}, ..., \gamma_p^{(m)}, \beta_T^{(m)} | \Sigma^{(m-1)}, (1/\sigma^2)^{(m-1)}$;
    (b) Sample $\theta_i^{(m)} | \gamma_1^{(m-1)}, ..., \gamma_p^{(m-1)}, \beta_T^{(m-1)}, \Sigma^{(m-1)}, (1/\sigma^2)^{(m-1)}$;
    (c) Sample $\Sigma^{-1(m)} | \theta_i^{(m-1)}, \gamma_1^{(m-1)}, ..., \gamma_p^{(m-1)}, \beta_T^{(m-1)}, (1/\sigma^2)^{(m-1)}$ ;
    (d) Sample $(1/\sigma^2)^{(m)} | \theta_i^{(m-1)}, \gamma_1^{(m-1)}, ..., \gamma_p^{(m-1)}, \beta_T^{(m-1)}, \Sigma^{-1(m-1)}$
end
for $i$ = 1, ..., $N$ do
    $\hat{c}_i = \max_k(\sum_{m=1}^{M} \mathrm{I}(c_i^{(m)} = k)$, k = 1, ..., K)
end
```

References

[1] Akaike, H. *Information theory as an extension of the maximum likelihood principle.* Akademiai Kiado, Budapest, 1973.

[2] Bardenheier, B. H., Shefer, A., Barker, L., Winston, C. A., and Sionean, C. K. Public health application comparing multilevel analysis with logistic regression: Immunization coverage among long-term care facility residents. *Annals of Epidemiology 15*, 10 (nov 2005), 749–755.

[3] Bhatia, P., Iovleff, S., and Govaert, G. blockcluster: An r package for model-based co-clustering. *Journal of Statistical Software, Articles 76*, 9 (2017), 1–24.

[4] Bouwmeester, W., Twisk, J. W., Kappen, T. H., van Klei, W. A., Moons, K. G., and Vergouwe, Y. Prediction models for clustered data: comparison of a random intercept and standard regression model. *BMC Medical Research Methodology 13*, 1 (feb 2013).

[5] Chib, S., and Carlin, B. P. On mcmc sampling in hierarchical longitudinal models. *Statistics and Computing 9* (1999), 17–26.

[6] Garcia, T. P., and Marder, K. Statistical approaches to longitudinal data analysis in neurodegenerative diseases: Huntington's disease as a model. *Current Neurology and Neuroscience Reports 17*, 2 (2017).

[7] Guo, G., and Zhao, H. Multilevel modeling for binary data. *Annual Review of Sociology 26*, 1 (aug 2000), 441–462.

[8] Laird, N. M., and Ware, J. H. Random-effects models for longitudinal data. *Biometrics 38*, 4 (1982), 963–974.

[9] Moerbeek, M. A comparison between traditional methods and multilevel regression for the analysis of multicenter intervention studies. *Journal of Clinical Epidemiology 56*, 4 (2003), 341–350.

[10] Molenberghs, G., and Verbeke, G. A review on linear mixed models for longitudinal data, possibly subject to dropout. *Statistical Modelling 1*, 4 (2001), 235–269.

[11] Müller, S., Scealy, J. L., and Welsh, A. H. Model selection in linear mixed models. *Statistical Science 28*, 2 (may 2013), 135–167.

[12] Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., and R Core Team. nlme: Linear and nonlinear mixed effects models. R package version 3.1-137.

[13] Snipes, M., and Taylor, D. C. Model selection and akaike information criteria: An example from wine ratings and prices. *Wine Economics and Policy 3*, 1 (2014), 3–9.

[14] Sullivan, L. M., Dukes, K. A., and Losina, E. An introduction to hierarchical linear modelling. *Statistics in Medicine 18*, 7 (apr 1999), 855–888.