Multilevel Models for Big Data

Approaches for handling very large datasets

Ömercan Mısırlıoğlu, Yordan Saputra

Department of Statistics
TU Dortmund University

Final Presentation TBA, January (?) 2026

IMPORTANT NOTES

- DONT FORGET TO CITE PROPERLY. rn im just adding links as references but we should have a proper bibliography slide at the end.
- Notes for Omer: the blog Clark, Michael. (2019). goes to talk about speed comparison between lme4 and mgcv (mainly for GAM, but can also be used for linear model). But, I dont think we should explore the GAM part since it is another topic? I want to stay on linear model. what do u think?
- Add more notes here

QUESTIONS FOR FEEDBACK

- do we need to fit our own model using either real or syntethic data? or just explanation is enough?
- should we go in deeper on the dependent / overlapping subsamples?
- is the R packages part needed?
- also, im not sure whether the R packages here is using split sample approach or not. from what i've read, it doesnt seem like it, but this is the only source we have
- Add more questions here

Table of Contents

Introduction – What are multilevel models?

- Hierarchical data are common in observational data, with individuals nested within geographical areas or institutions such as schools or workplaces.
- Multilevel models recognise the existence of such data hierarchies by allowing for residual components at each level in the hierarchy.
- This model would partition the residual variance at different levels of the hierarchy, such as between-group and within-group components, capturing unobserved characteristics that affect outcomes.
- Reference: `https://www.bristol.ac.uk/cmm/learning/multilevel-models/what-why.html`

Introduction – Statistical model

Linear multilevel model with 1 predictor for individual i in group j:

$$Y_{ij} = \underbrace{(\beta_0 + u_{0j})}_{\beta_{0j}} + \underbrace{(\beta_1 + u_{1j})}_{\beta_{1j}} X_{1,ij} + \epsilon_{ij} \tag{1}$$

- Y_{ij}: Dependent variable for individual i in group j
- $X_{1,ij}$: Predictor variable 1 for individual i in group j
- β_{0j}: Intercept for group j, which consists of:
  - β_0: Overall intercept across all groups
  - $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u_0})$: Random effect of intercept for group j
- β_{1j}: Slope for predictor $X_{1,ij}$ for group j, which consists of:
  - β_1: Overall slope for predictor $X_{1,ij}$ across all groups
  - $u_{1j} \sim \mathcal{N}(0, \sigma^2_{u_1})$: Random effect of slope for predictor $X_{1,ij}$ for group j
- $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2_\epsilon)$: Residual error for individual i in group j

Notes for presentation: its possible to add predictor $X_{2,ij}$ or $Z_1 j$ and so on (comment later)

Introduction – Why use multilevel models?

1. **Correct Inferences**: Traditional methods assume independent observations, which are often false. Additionally, ignoring hierarchical structures can lead to underestimated standard errors and overstated statistical significance.
2. **Group Effects Estimation**: Directly quantify between-group variation and identify outlying groups.
3. **Simultaneous Estimation**: Unlike fixed effects models, the separation of observed and unobserved group characteristics is possible, allowing for simultaneous estimation of group-level and individual-level effects.
4. **Generalization Beyond Sample**: Unlike fixed effects models which only describe sampled groups, multilevel models treat groups as random samples from a population, enabling generalizations to unobserved groups.
5. Reference: `https://www.bristol.ac.uk/cmm/learning/multilevel-models/what-why.html`

Introduction — Issues with large multilevel datasets

- Hierarchical data expands dramatically, potentially reaching gigabytes in size (N groups $\times$ n individuals $=$ massive datasets).
- **Large number of groups**: Computational bottleneck from numerical integration over random effects for each group at every optimization step, leading to high computational costs.
- **Large group sizes**: High-dimensional multivariate distributions create numerical issues with large covariance matrix inversion, even in linear models.
- Reference: Clark, Michael. (2019)., Speelman, Dirk. (2018) chapter 2.

Table of Contents

The split-sample approach – Pseudo Likelihood

- Replace a numerically challenging joint density by a simpler function assembled from suitable factors.
- Consider the following log-likelihood function $\ell(\boldsymbol{\theta}) = \sum_i \ell(\boldsymbol{y_i}|\boldsymbol{\theta})$
- Replaces the log-likelihood contribution $\ell(\boldsymbol{y_i}|\boldsymbol{\theta})$ by a weighted sum of log-likelihood contributions for sub-vectors $\boldsymbol{Y_i^{(s)}}$
- More specifically, the pseudo-log-likelihood function:

$$p\ell(\boldsymbol{\psi}) = \sum_i \sum_s \delta_s \; \ell(\boldsymbol{y_i^{(s)}}|\boldsymbol{\psi}) \tag{2}$$

  is maximized instead with respect to $\boldsymbol{\psi}$ (not necessarily identical to $\boldsymbol{\theta}$)
- Although $\hat{\boldsymbol{\psi}}$ is not the MLE estimate, it still has similar properties such as consistency and asymptotic normality.
- Reference: Speelman, Dirk. (2018) chapter 2.

The split-sample approach – Graphical representation

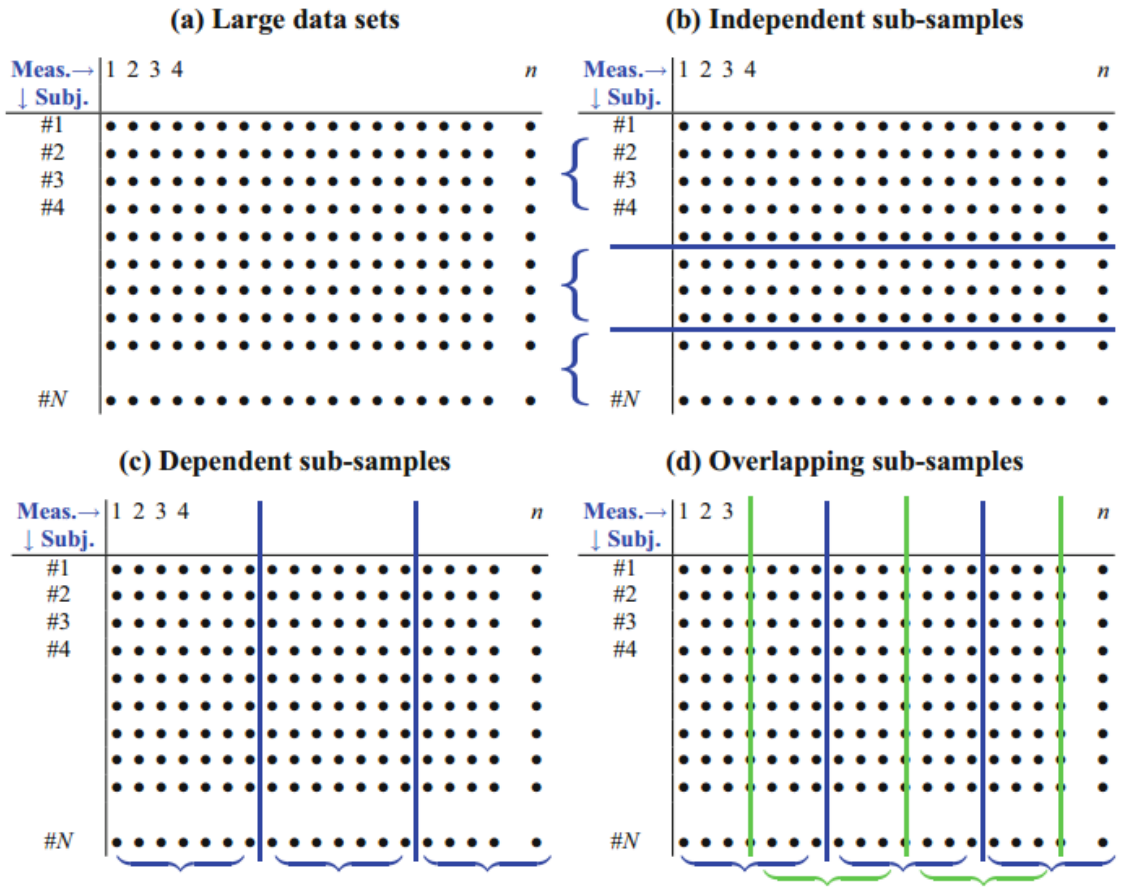

Figure 1: Graphical representation of different ways to split large samples

The split-sample approach — Independent subsamples

- The partition are shown in panel (b) of Figure 1.
- When the number of N groups is large, partition the groups in M independent sets S_m of groups, where $m = 1, \ldots, M$.
- In each subsample, the model is fitted, yielding an estimate $\hat{\boldsymbol{\theta}}_m$ of $\boldsymbol{\theta}$, equivalent to maximizing

$$p\ell(\boldsymbol{\psi}) = \sum_m \sum_{i \in S_m} \ell(\boldsymbol{Y_i}|\boldsymbol{\theta}_m) \tag{3}$$

- All $\boldsymbol{\theta}_m$ are equal to $\boldsymbol{\theta}$, therefore the estimates $\hat{\boldsymbol{\theta}}_m$ can be averaged to obtain an overall estimate $\hat{\boldsymbol{\theta}}$.

The split-sample approach — Dependent subsamples

- The partition are shown in panel (c) of Figure 1.
- In case of large n, the data is partitioned in M (not independent) sets S_m of groups, where $m = 1, \ldots, M$.
- Fitting the model on each subsample, equivalent to maximizing

$$p\ell(\psi) = \sum_m \sum_i \ell(\boldsymbol{Y}_i^{(m)} | \boldsymbol{\theta}_m) \tag{4}$$

  where $\boldsymbol{Y}_i^{(m)}$ is the observations in $\boldsymbol{Y}_i$ belonging to subsample S_m.
- All $\boldsymbol{\theta}_m$ are not necessarily equal to $\boldsymbol{\theta}$, therefore the combination of all $\hat{\boldsymbol{\theta}}_m$ into a single estimator $\hat{\boldsymbol{\theta}}$ depends on the precise model and data structure.

The split-sample approach — Overlapping subsamples

- The partition are shown in panel (d) of Figure 1.
- Suitable for longitudinal data with very large n.
- Similar to dependent subsamples, but association between longitudinal observations is accounted for by letting the subsamples overlap.
- Denoting the parameters in pair $\{\boldsymbol{Y}_i^{(p)}, \boldsymbol{Y}_i^{(q)}\}$ by $\boldsymbol{\theta_{p,q}}$, fitting the models on all pairs is equivalent to maximizing

$$p\ell(\psi) = \sum_{p<q} \sum_i \ell(\boldsymbol{Y}_i^{(p)}, \boldsymbol{Y}_i^{(q)} | \boldsymbol{\theta_{p,q}}) \tag{5}$$

- Similarly, the combination of all $\hat{\theta}_{p,q}$ into a single estimator $\hat{\theta}$ depends on the precise model and data structure.

Table of Contents

R Packages – `lme4`

- `lme4` is an R package for fitting linear and generalized linear mixed-effects (multilevel) models using 'Eigen' C++ library and S4 classes.
- It's computationally efficient, enabling it to handle very large sample sizes for simpler mixed models and to process hundreds of thousands of observations with random effects on a typical laptop.
- Modeling functions:
  - `lmer()`: fits linear multilevel model using restricted maximum likelihood (REML) or maximum likelihood estimation.
  - `glmer()`: fits generalized linear multilevel model, accommodating non-normal response distributions.
- Reference:
  `https://cran.r-project.org/web/packages/lme4/lme4.pdf`

R Packages – mgcv

- mgcv is an R package for fitting generalized additive models (GAMs) and generalized additive mixed models (GAMMs) using penalized regression splines.
- Modeling functions:
  - gam(): fits generalized additive multilevel models using penalized regression splines with smooth terms that can incorporate multilevel structure through random effect splines.
  - bam(): a computationally efficient version of gam() optimized for very large datasets.
- Reference:
  https://cran.r-project.org/web/packages/mgcv/mgcv.pdf

R Packages – Why use `bam()`?

- The underlying model between `gam()` function and `lme4` is the same, with differences only in the way parameters are estimated.
- `bam()` employs parallelized computation on model matrix subsets and optional data discretization to extract minimal necessary information, enabling efficient estimation of large multilevel models.
- With large enough datasets, this discretization has negligible impact on parameter estimates (differing only at high decimal precision), but leads to dramatic speed improvements.

R Packages – When to use `bam()`?

- In general, `lme4` is preferred for most multilevel datasets due to its straightforward syntax and robust estimation methods.
- `bam()` is particularly useful when:
  - The model structure is complex and computationally demanding for `lme4`
  - Smooth (nonlinear) terms are required
  - The dataset is very large and memory limitations arise
  - Parallel computing resources can be effectively utilized

Table of Contents

Summary

- Multilevel models are powerful tools for analyzing hierarchical data structures, allowing for accurate inferences and simultaneous estimation of group and individual effects.
- However, large multilevel datasets pose significant computational challenges, including memory constraints and slow estimation times.
- The split-sample approach offers a practical solution by partitioning data into manageable subsamples, enabling efficient model fitting while retaining key statistical properties.
- R packages like `lme4` and `mgcv` provide robust tools for fitting multilevel models, with `bam()` in `mgcv` being particularly suited for very large datasets due to its computational efficiency.
- Choosing the right approach and tools depends on the specific characteristics of the dataset and the research questions at hand.