

Data Splitting and Its Applications

Vahid Nassiri

June 25, 2019

Contents

Contents	i
List of Figures	ix
List of Tables	xvii
1 Introduction	1
Large clustered data	5
Data splitting: a unified approach	9
Different types of splitting	11
Random horizontal splitting	12
Structured horizontal splitting	15
Random vertical splitting	19
Structured vertical splitting	28
Supplementary topics	34
How many sub-samples?	34
Special case: where combination rule fails	36
Software implementation	38

2 Motivating datasets	39
A developmental toxicity study	40
Divorce in Flanders	41
Hearing data	43
Leuven diabetes project	44
Amazon book ratings	45
Clinical Trials in Schizophrenia	45
Cholesterol data	47
Leuven eyes study	47
3 Random horizontal splitting	49
The research question	50
A mixed model solution	51
A data-splitting based remedy	53
How precise are we?	61
4 Structured horizontal splitting	65
Compound-symmetry structure	65
Introduction	65
Developmental Toxicity Study Sets	70
Incomplete Sufficient Statistics	71
The Compound-symmetry Model	72
Split-sample Methods for Clusters of Variable Size .	77
Partitioned-sample Analysis for the Compound-symmetry Model	83
Simulation Study	92
Analysis of Case Study	93

Ramifications and Concluding Remarks	101
AR1 structure	104
Introduction	104
Model Formulation	108
Estimators	109
Complete and Incomplete Sufficient Statistics	114
Clusters Of Variable Size	119
Computational Considerations and Simulation Study	124
Application: Clinical Trials in Schizophrenia	126
Concluding Remarks	129
5 Random vertical splitting	135
Iterative multiple outputation	136
Finite Information Limit Estimators	141
Introduction	141
A compound-symmetry model	146
Finite Information Limit (FIL) Estimators	150
Extensions	157
Case study: Amazon.com book ratings	165
Conclusion	171
6 Structured Vertical Splitting	173
Fast precision estimation	173
Introduction	173
A fast variance estimator based on multiple outputation	177
Simulations	182
Application	187

Page iii

Conclusions	192
7 Supplementary topics	197
Iterative Multiple Imputation	198
Introduction	198
Number of imputed datasets: A review	199
Number of imputed datasets: an alternative proposal	204
Simulation study	211
Applications	220
Discussion and Concluding Remarks	224
Multiple imputation for EFA	227
Introduction	227
Using multiple imputation with factor analysis: a review	229
Using multiple imputation with factor analysis: a proposal	232
Simulations	236
Divorce in Flanders	238
Conclusions	240
8 Software packages	245
R package fastCS	248
R package fastAR1	248
R package miscVSS	249
R package imi	249
R package mifa	250
9 Conclusions	253

10 Appendices	255
Incompleteness in the Compound-symmetry Model	257
Likelihood-based Estimation of the CS Model	259
Score Functions	259
Lack of Closed-form Solution when $K \geq 2$	259
Full Likelihood	261
Pseudo-likelihood for Split Samples	266
General Considerations	266
Pseudo-likelihood for Split Sample	269
Optimal Scalar Weights for CS	270
Cluster-by-cluster Analysis	273
Details About the First Simulation Study	279
Simulation Method	279
Setting 1	280
Setting 2	281
Setting 3	282
Optimal, Approximate Optimal, and Iterated Optimal Weights	283
Details About the Second Simulation Study	286
Simulation Plan	288
Simulation results	293
The Balanced Conditionally Independent Model	300
Algebraic Derivations in the AR(1) Case	304
Some Useful Expressions	304
The Likelihood Estimators in a Given Cluster	306
Hessians, Covariance Matrices, and Optimal Weights	311
Page v	

Proof of Proposition 4.1	316
Optimal weights in case of a general mean structure $X_i^{(k)}\beta$	317
Delta Method for the Mean Estimator	320
Calculating $\hat{\rho}$ and $\hat{\sigma}^2$ in R	321
Details on Additional Simulations	322
Simulations with Proportional and Size-proportional Weights	322
Simulations with Proportional and Size-proportional Weights: ρ near 0/1	323
Simulations With Optimal Weights	328
Simulations on Computation Time	328
Details on PANSS Data Analysis	333
Fieller's method and Delta method	334
Random vertical data splitting for CS	336
Combination rule for p -values in LES dataset analysis (IMI)	339
The surrogate model	340

List of Figures

1.1	Number of items found by Google scholar when searching the exact phrase <i>big data</i> up to different years.	4
1.2	General scheme of the clustered data we consider through this thesis.	6
1.3	Number of parameters in the covariance matrix of jointly normally distributed random effects for different number of responses using only random intercept or random intercept and slope models.	32
3.1	Estimated correlation matrix for each aspect of personality.	60
3.2	120 components of \tilde{D} and their 95% confidence intervals	60
4.1	NTP Data. Scalar weights: proportional and optimal scalar versions for EG and TGDM datasets. The optimal scalar weights are computed for $\rho = d/(\sigma^2 + d) = 0.5$	133

Page vii

5.1	The effective sample sizes for different cluster sizes for CS and AR(1) covariance structures for different correlation values ρ	147
5.2	Comparing the ARE of full data with sub-samples of sizes $n_f = \{5, 10, 50\}$ for $\rho = \{0.2, 0.5, 0.8\}$. The dashed horizontal line shows $ARE = 1$	157
5.3	Comparing the ARE of full data and sub-sampling of size $n_f = 50$, $M = 5$ times with different combinations of n_f and M for $\rho = 0.2$	164
5.4	Comparing the ARE of full data and sub-sampling of size $n_f = 50$, $M = 5$ times with different combinations of n_f and M for $\rho = 0.5$	165
5.5	Comparing the ARE of full data and sub-sampling of size $n_f = 50$, $M = 5$ times with different combinations of n_f and M for $\rho = 0.8$	166
5.6	Comparing the ARE of full data with sub-samples of sizes $n_f = 5, 10, 50$ when the results from $M = 1, 3, 5$ are combined using the weights $w_i = (5/65, 10/65, 50/65)$	168
5.7	Histogram of number of all and unique cluster sizes in Amazon books ratings data.	169
5.8	Amazon books ratings data. Fieller's upper-bound of n_f (log-scale) for different values of ε using $\alpha = 0.05$. The parameters are estimated using a sub-sample of size $n_1 = 5$	170

6.1	Simulations (Gaussian outcomes) comparing sandwich and MO corrections separately	194
6.2	Simulations (Gaussian outcomes): comparing sandwich and MO corrections	195
6.3	Application (hearing data)	196
7.1	Convergence plots for multivariate normal random vectors parameter estimates with CS and AR(1) covariance matrix structures	216
7.2	Convergence plots for the p -values of one sample and paired t -tests for different values of test statistics . .	217
7.3	Averaged selected M with $k_0 = 3$ for data generated from a multivariate normal with mean 0 and covariance matrix with CS and AR(1) structures with $\sigma^2 = 16$ and $\rho = 0.5, 0.8$, as well as a logistic regression model	219
7.4	Convergence plots for p -values of t -test on cholesterol data with $\mu_0 = 200$ and $\mu_0 = 220$ test values, and LES logistic regression, with $\epsilon = 0.05, 0.01$, $M_0 = 2$, and $k_0 = 3$	223
8.1	Top analytical tools for data science	246
1	First simulation study. Setting 1. Split-specific results.	282
2	First simulation study. Setting 1. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.	283

3	First simulation study. Setting 2. Split-specific results.	285
4	First simulation study. Setting 2. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.	286
5	First simulation study. Setting 3. Split-specific results.	287
6	First simulation study. Setting 3. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.	288
7	First simulation study. Setting 1. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.	289
8	First simulation study. Setting 2. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.	290
9	First simulation study. Setting 3. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.	291
10	Second simulation study. Estimates for μ (first row) and its standard error (second row).	294
11	Second simulation study. Estimates for d (first row) and standard errors (second row).	295
12	Second simulation study. Estimates for σ^2 (first row) and standard errors (second row).	296

Page x

13	Second simulation study. Estimated CS parameters (first row) and their standard error (second row) using sample splitting, MI-MLE, and MLE.	302
14	The third degree polynomial in (A.115) for 10 different generated data. The red veertical line shows $\hat{\rho}$	322
15	Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.1$	324
16	Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.5$	325
17	Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.8$	326
18	Simulation study. Boxplots comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.99, 0.95, 0.9, 0.8, 0.5, 0.2, 0.01$	327
19	Simulation study. Comparing proportional, size-proportional and full likelihood results via their empirical density for the 100 replications	327
20	Comparing iterated optimal and size-proportionanl weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.1$	329

27	PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates using sample splitting, combined with proportional and size-proportional weights, and full likelihood (with trial model)	350
28	PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates within each split (with trial model)	351

List of Tables

1.1	Computation time (in seconds) of different methods for different sample sizes.	3
1.2	Unique cluster sizes and their frequency in a developmental toxicity case study for the response DEHP.	17
3.1	D -matrix parameter estimation (120 component \tilde{D})	59
3.2	Comparing full sample and sample splitting results for each factor separately	63
4.1	Developmental Toxicity Study (DEHP). Summary data by dose group.	71
4.2	Cluster-by-cluster analysis- DEHP	95
4.3	Cluster-by-cluster analysis, EG	96
4.4	Cluster-by-cluster analysis, DYME	97
4.5	NTP Data (with dose effect)	100
	Page xv	

4.6 PANSS data. Number of clusters in each trial for each cluster pattern. The pattern consists of the numbers representing the months after starting point for which a PANSS score is available.	127
4.7 PANSS data. Contributing splits in estimating each parameter	130
4.8 PANSS data. Estimating fixed effects and variance components and the standard deviations of these estimates using sample splitting without trials	131
4.9 PANSS data. Estimating fixed effects and variance components and the standard deviations of these estimates using sample splitting with trials	132
5.1 Computation time t (in seconds) for clustered data with unique cluster sizes $n_k \cdot 10^i$, with $n_k = 5, 6, 7, 8, 9, 10$ and $i = 0, 1, 2, 3, 4$, and with 100 clusters from each n_k (600 clusters in total).	143
5.2 Computation time (in seconds) for different combinations of n_f and M and the full data.	167
5.3 Estimating parameter vector (μ, τ, σ^2) and its standard error using $n_f = 5, 50, 100$	171
6.1 Simulations (non-Gaussian outcomes)	187
7.1 Mean, standard deviation (STD) for selected M given different ϵ and k_0 values, and number of times $M > 500$ for different models and different ϵ 's using the Mahalanobis-type distance with $S = \hat{V}$	242

7.2	Mean, standard deviation (STD) for selected M given different ϵ and k_0 values, and number of times $M > 500$ for different models and different ϵ 's for t -test and paired t -test with test value $\mu = 0$	243
7.3	Comparing different methods of doing factor analysis with missing data	243
7.4	Factor loadings using oblimin rotation of DiF data using the estimated covariance matrix from multiply imputed data using $M = 25$ imputations.	244
1	First simulation study. Setting 1. Average of split-specific and combined (weighted) parameters and their precision estimates.	281
2	First simulation study. Setting 2. Average of split-specific and combined (weighted) parameters and their precision estimates.	284
3	First simulation study. Setting 3. Average of split-specific and combined (weighted) parameters and their precision estimates.	284
4	Second simulation study. Mean, standard deviation (S.D.) and MSE for μ among 100 replications for each configuration using different combination weights comparing with full sample MLE.	297
5	Second simulation study. Mean and standard deviation (S.D.) for standard errors of μ estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.	297

Page xvii

6	Second simulation study. Mean, standard deviation (S.D.) and MSE for d estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.	298
7	Second simulation study. Mean and standard deviation (S.D.) for standard errors of d estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.	298
8	Second simulation study. Mean, standard deviation (S.D.) and MSE for σ^2 estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.	299
9	Second simulation study. Mean and standard deviation (S.D.) for standard errors of σ^2 estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.	299
10	Second simulation study. Computation time (in seconds) using closed-form solutions with different implementation forms, compared to PROC MIXED.	300
11	Second simulation study. Mean, standard deviation (S.D.) and MSE for CS parameter estimates in 100 replications for configuration 2 using different combination weights comparing with full sample MLE and MI-MLE.	300

12	Second simulation study. Mean and standard deviation (S.D.) for the standard error of CS parameter estimates in 100 replications for configuration 2 using different combination weights comparing with full sample MLE and MI-MLE.	301
13	Comparing proportional, size-proportional and iterated optimal weights with full likelihood for AR(1) covariance structure.	323
14	Simulation study. Estimating μ and its standard deviation	342
15	Simulation study. Estimating ρ and its standard deviation	343
16	Simulation study. Estimating σ^2 and its standard deviation	344
17	Simulation study. The computation time	345
18	PANSS data. Number of clusters in each trial for each cluster pattern.	346
19	PANSS data. Comparing different error covariance structures using three model comparison criteria for without trial model	347

Chapter 1

Introduction and problem statement

Small sample sizes would make wrong or statistically inaccurate conclusions. Therefore, in designing any statistical study, an important step is to estimate the sample size that would enable the researcher to draw valid conclusions. While a small sample size prevents reliable or meaningful statistical conclusions, large sample sizes has rarely been an issue, in fact a large sample size was a desirable aspect of a study. However, in 21st century, having large sample sizes also, becomes a trouble-maker. By emerging technologies that make the task of data collection easy and fast, the size of samples in many studies becomes larger and larger, and this came to the point that, the power of a typical computer would fail to analyze such huge datasets.

There are several issues related to a large datasets, including: how to store them, how to load them, and how to analyze them. Our interest in this thesis is in the last one. Although the last issue, the same as the two others could be solved using ideas coming from computer science, or simply, increasing the computation power of the hardware we use could solve the issue, we believe statistical methods and ideas can provide solutions that are easy to understand, capable of being implemented in different situations, and also accurate. This becomes more important when we realize increasing the computation power and size of available datasets are highly and positively correlated.

In Table 1.1 we compare the computation when the sample sizes $n = 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7$, for some simple computational tasks including:

- Generating a random sample from standard normal of size n .
- Finding the mean of the generated sample.
- Finding the variance of the generated sample.
- Fitting a linear model to the generated sample as the predictor and $y = 1 + ex + \epsilon$, where $\epsilon \sim N(0, 0.01)$.
- Fitting a MAD (mean absolute deviation) regression to the same data.

As we may see, for some analyses, the computation time increases linearly as the sample size increases, but for others, it would even

Sample size	Generation	Mean	Variance	LS	MAD
10	0.00	0.00	0.00	0.00	0.00
10^2	0.00	0.00	0.00	0.00	0.00
10^3	0.00	0.00	0.00	0.00	0.00
10^4	0.00	0.00	0.00	0.00	0.06
10^5	0.00	0.00	0.00	0.01	3.75
10^6	0.08	0.01	0.00	0.32	764.37
10^7	1.05	0.01	0.06	3.46	60558.83

Table 1.1: Computation time (in seconds) of different methods for different sample sizes.

speeds down with a sub-linear rate. Once the cause of expensive computation time is the sample size, an effective solution could be to split the very large sample into smaller sub-samples and then analyze them separately. As each sub-sample has a smaller size, and analyzing these different sub-samples are independent of each other (can be run in parallel), the overall computation time can be decreased.

Splitting a sample into smaller chunks, then analyzing each part separately, and finally combine the results of these analyses is a known methodology to deal with large samples. Some authors have called this *Software Alchemy*, ?. This idea is also at the heart of MadReduce methodology (?). The split-apply-combine strategy in the R packages `plyr` and `dplyr` also uses this idea (?). All these would show the importance of this approach. Furthermore, Figure 1.1 shows a plot of number of results found using Google scholar when searching the exact phrase "big data" upto different years. We obviously can see the increasing interest in the topic in the recent

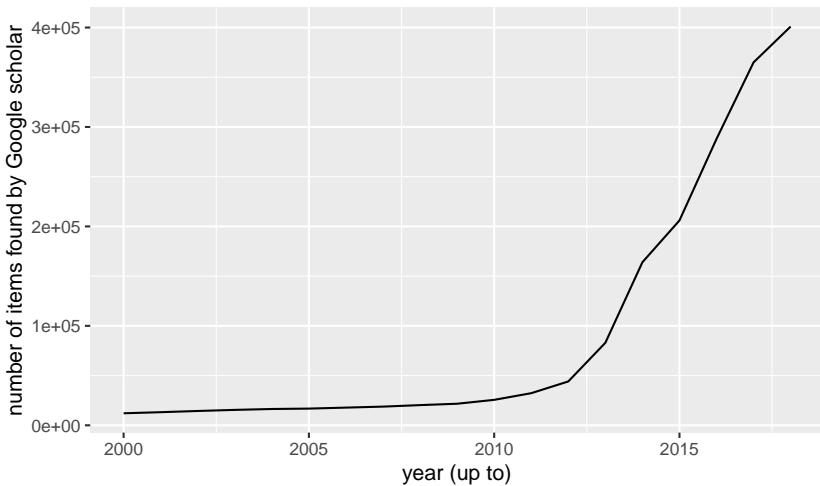


Figure 1.1: Number of items found by Google scholar when searching the exact phrase *big data* up to different years.

years.

However, most of these implementations would consider independent sub-samples, or non-clustered data. Our goal in this thesis to extend the existing methodology to the cases where the subjects would form clusters of correlated data. This type of data is very common in medical studies in particular, but they also can be observed in any longitudinal or multilevel study. Some examples of such data are measurement on offsprings of the same animal. Data collected from family members. Data collected from different countries in a meta-analytic setting. Measuring body temperature of the same person several times, ratings of the same book by different people, measuring hearing ability of elderly people for several frequencies, measuring primary vital signs (body temperature, heart

rate, respiratory rate, and blood pressure) for each person, etc.

As we have seen in Table 1.1, even for non-clustered data and simple models, increasing the sample size could cause very expensive computation times. The time needed to fit a MAD regression with one predictor to a sample of size 10 million is around 18 hours. Now, if we have clustered data that needs more complicated models, things would get even more difficult. In case of clustered data, the sample size can be seen from different angles.

Large clustered data

Consider a study which measures (at most) m outcomes for N clusters, each n_i times. Let y_{rij} be the j th measurement taken on the i th cluster for the r th outcome, $i = 1, \dots, N$, $r = 1, \dots, m$ and $j = 1, \dots, n_{ri}$. As a simple example, take $m = 4$ responses as the vital signs: body temperature, heart rate, respiratory rate, and blood pressure, measured for $n_1 = n_2 = n_3 = n_4 = 100$ hospitalized patients. And assume these measurements are done every 4 hours. So, y_{111} is the body temperature of the first patient measured for the first time).

A general scheme of such clustered data is presented in Figure 1.2. A suitable model for such data should appropriately respect the correlation structures imposed by the fact that several measurements are done on the same subject. While even for moderate sample sizes, fitting such models could already become challenging, we may face a greater challenge when the size of dataset makes it eligible

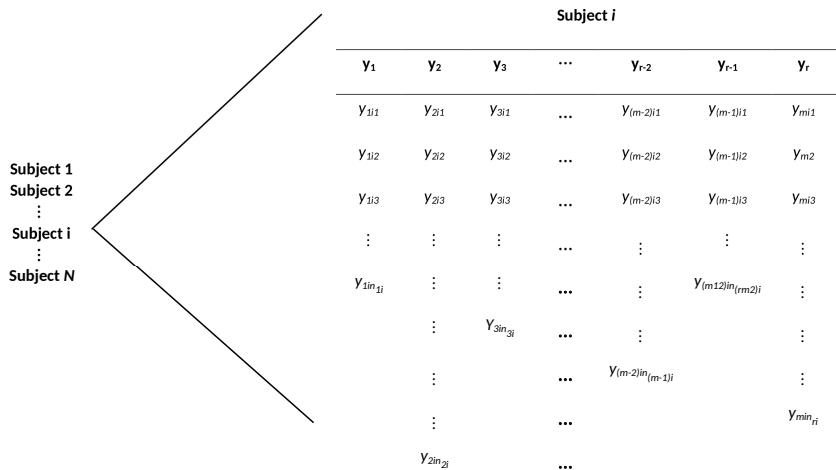


Figure 1.2: General scheme of the clustered data we consider through this thesis.

for the title *big data*.

As it was mentioned, for non-clustered data big would mean a large number of subjects. However, in our case, a dataset could be entitled as *big data* in different situations. In case of the clustered data of interest in this thesis, data are considered to be big if at least one of the following applies:

- When the sample size, N , becomes very large,
- When the cluster sizes become very large:
 - ▷ When the number of measurements per outcome for some clusters, n_{ri} 's, becomes very large,
 - ▷ When the number of outcomes, m , becomes very large.

Note that, *big* would be different for N , n_{ri} , and m . For example, in case of a MAD regression, we have seen a dataset of size $N = 100,000$ could be analyzed in 0.001h, increasing the sample size to $N = 1000,000$ would increase the computation time to 0.2h, which is not fast, but still feasible, but $N = 10,000,000$ would take around 17 hours which could be too much. Therefore, in the case of a simple MAD regression, we may consider $N > 10,000,000$ as *big*.

When it comes to clustered data, however, there is a different story. Some 100's could already be *big* in case of cluster size, n_{ri} (see Amazon book ratings data in Section 2), and even some 10's is *big* when we consider the number of outcomes m (see Hearing data in Section 2).

It goes without saying that, in reality, any combination of the above situations could happen. Our aim in this thesis to provide appropriate methodology to: 1- split the data into smaller chunks, 2- analyze each split, 3- combine these analyses in an efficient way. Our methodology should consider the nature of the data. While the mere task of splitting would already breaks down the difficult (sometimes infeasible) problem into smaller easier and feasible to analyze pieces, sometimes the structure of the data would allow for some *smart* other than random splitting which would then provide more gain. Our aim in this thesis is to cover all of the situations which a dataset of clustered subjects could be called *big*. We will go over our proposal for splitting/analyzing/combining steps for each of these cases.

In developing these methodologies we will make sure to keep two promises:

- While each methodology can deal with one situation, it should be easily possible to combine methodologies to deal with a combined situation.
- The splitting and combining steps of our methodology should be as independent as possible from the analyses step, so one can easily (or with a minor modification) use it for its desired analysis.

Data splitting: a unified approach

Data splitting is a general idea to deal with big, massive, and large data. Large here could be called to any of the cases we have mentioned above, or a combination of them. Our general approach to deal with this situation consists of three main steps which we would call them together Data splitting.

fieuws2007

1. **Splitting:** in this step, the data should be splitted into smaller chunks in a way that analyzing each of them becomes more convenient than the dataset as a whole. This convenience would at least come from having sub-samples of smaller sizes, but it could be possible to perform the splitting in a way that some other nice features (balancedness, for example) that the original data does not own, can be created in all or some of

these sub-samples.

2- Analyzing: in this step, each split of data will be analyzed using the conventional methods. Therefore, using *data splitting* approach, one can use its own favorite methodologies which are only suitable for small to moderate sample sizes. Typically, with *data splitting* approach, the only information one needs to extract and keep from analyzing each split of data are: the estimated parameter vector, and their covariance matrix (if there is a need to provide standard errors of such estimates).

3- Combining: within this unified framework, each parameter in the model could be estimated several times. As our interest is to have one single set of parameter estimates, the last step in this paradigm is to combine all of analysis results from the second step into one single set. This should be done via appropriate combination rules. The combination rule is simply a weighted average of the results from different splits. Therefore, there is the need for proposing appropriate weights. Also, when it comes to combine covariance matrices, the fact that these results are coming from different (possible dependent) sub-samples should be well respected.

In the following sections we will review different splitting approaches, also their appropriate combination rules. WE will also give examples of our motivating datasets where each of these approaches could be beneficial.

Different types of splitting

We have already discussed the MapReduce, or split-apply-combine approaches. The main splitting technique that is implemented using such approaches is to split the data: 1- randomly, 2- at subject level. Throughout this these we would call the splitting at the subject level *horizontal splitting*. The reason for that can be seen in the Figure 1.2. If we illustrate the splits by separating lines, such lines will be horizontal. On the other hand, if the splitting is done for the data within a subject, then the lines should be drawn vertically. This case is specific to the clustered data where we have more than one observation for each subject.

While random splitting (assigning members of each split at random) is an effective way in many situations, there are cases where assigning members of each split based on a predefined structure has an added value. We call such splitting approach structural splitting as opposed to random splitting. In the following we will give an overview of the four types of splitting, and present examples where each of them could be useful. We will also briefly go over analyzing and combining steps also.

Random horizontal splitting

Random horizontal splitting, as its name suggested tries to assign different subjects to different splits at random. There are several ways to do this task, the way we have used in our implementations is as follows, for a sample of size N ,

1. To split the dataset into M sub-samples, take $m = [N/M]$ as the size of each split to make sub-samples of roughly the same sizes.
2. Generate a random sample of size N from standard normal.
3. Order the generated sample.
4. Divide the ordered indexes into $(M-1)$ parts of size m , and the m th chunk will include all the remaining indexes. These will be the index of subjects that should be assigned to each split.

Note that, M should be carefully selected. Two main points to consider when selecting M are first to make sure the resulting sub-samples are small enough to keep the computation time at an acceptable level, but at the same time one needs to be careful that the sub-samples stay large enough, so the methodologies can still be applied on them in a valid way.

Assume $\hat{\theta}_1, \dots, \hat{\theta}_M$ are the estimated parameters from M sub-samples, the data splitting estimate of this parameter can be computed as follows:

$$\tilde{\theta} = \sum_{m=1}^M \widehat{\theta_m}, \quad (1.1)$$

where $\sum_{m=1}^M w_m = 1$. An appropriate weights could be assigned proportional to the size of each split. We would call such weights proportional weights in this thesis:

$$w_{prop,i} = \frac{m_i}{N}. \quad (1.2)$$

Of course, in case of random splitting, as the split sizes are roughly the same, one can simple assign equal weights to different sub-samples. Such weights are called equal weights in this thesis:

$$w_{equal,i} = \frac{1}{M}. \quad (1.3)$$

The combination rule in case of horizontal splitting is simpler than vertical splitting. This comes from the fact that the sub-samples are independent in this case, therefore, one does not need to be concerned about the correlations between the estimates coming from different sub-samples. For a general weight w_i , consider $\text{Var}(\hat{\theta}_i) - \sigma_i^2$, then the variance of data splitting estimator can be computed as follows:

$$\text{Var}_{horizontal}(\tilde{\theta}) = \sum_{m=1}^M w_m^2 \sigma_m^2. \quad (1.4)$$

As having a smaller variance is a desired property for an estimator, Equation (1.4) would give an idea about an alternative way of assigning weights to our estimates: the weights should be computed in a way that the variance in (1.4) becomes minimized. In this thesis, such weights are called optimal weights. We define the optimal weights as the minimizer of the following objective function:

$$Q = \sum_{m=1}^M w_m^2 \sigma_m^2 - \lambda \left(\sum_{m=1}^M w_m - 1 \right), \quad (1.5)$$

where λ is a Largange multiplier to make sure that the sum of estimated optimal weights is still 1. Solving the optimization problem in (1.5) would lead to the following optimal weights,

$$w_{opt,i} = \frac{1/\sigma_i^2}{\sum_{m=1}^M 1/\sigma_m^2}, \quad (1.6)$$

As one may see, we may give more weights to a sub-sample with smaller variance.

In using optimal weights, one needs to be concerned about the fact that, optimal weights are including parameters from model, which then needs to be replaced by an estimate (we will discuss different ways of doing it in the coming chapters), and these estimates will come with their own variability. So, if the variability these estimates will bring together with them becomes larger than the variability that the optimal weights could decrease, then it is better to use the simpler parameter-free weights.

An example in our motivation datasets that random horizontal splitting can be beneficial is the Divorce in Flanders dataset, where clusters are formed by family members. While each family is at most of size 7, the number of total available families is large enough to make conventional software to fail. By splitting this large sample randomly and at the family level (horizontally), we could be able

to fit the models and answer the research questions.

Structured horizontal splitting

As we have discussed, random horizontal splitting has been used by many researchers in different contexts to deal with big data issue. However, a randomly horizontal split would only decrease the sample size. So all of our gain is due to having smaller sample sizes. We have observed that there are cases where splitting using a pre-defined structure would provide some added benefits on top of just having smaller sub-samples.

It is a well-known fact that a balanced study design works better than an unbalanced one. By balanced we mean, the number of available data for each cluster is the same. In fact, most of the studies are planned to be balanced, but in practice they would rarely end up like that. One can see several reasons for that, for example, the required materials are not enough, some patients die, some animals fail to follow the instructions, etc. Therefore, while having a balanced set of data is desired and even planned, we will usually have an unbalanced data to analyze.

Although most of the data in practice are unbalanced, one can usually find balanced sub-sets of those data. Take our example on a developmental toxicity study. There, the clusters are made of fetuses within litters. Obviously, different mothers would have different number of babies: an inevitably happened unbalanced data. However, it is not like that the number of fetuses within

litter from every single mother is different from the others. Table 1 shows unique number of fetuses in different litters with their frequencies. As one may see, we have a limited number of cluster sizes. Therefore, although the complete data is unbalanced, it is formed by balanced sub-sets. An structured horizontal splitting would assign all clusters of the same size to one sub-sample.

Table 1.2: Unique cluster sizes and their frequency in a developmental toxicity case study for the response DEHP.

Cluster size	9	10	15	8	12	13	14	6	7	3	11	16
Frequency	10	9	3	12	20	18	6	4	5	4	9	4

As it was mentioned, there are several benefits for a balanced dataset, and in the special case of clustered data, we have shown that a closed-form solution exists when the cluster sizes are balanced, whereas for unbalanced data one needs to fit the models using iterative procedures. For very large datasets, the iterative procedures could be prohibitive and slow. While, a closed-form solution can be computed much faster. We have observed tens of thousands time faster computation using structured horizontal splitting based on the cluster sizes. Two cases are studied in details: clustered data with a compound-symmetry (CS) covariance structure, and clustered data with an autoregressive of order 1 (AR1) covariance structure. It worth to note that, even if the closed-form does not exist, with a balanced dataset one would have a faster and more stable convergence compared to unbalanced data.

The two considered covariance structures would cover a wide range of applications: where the correlation between to members of a

cluster is always the same (compound-symmetry), and when it decreases when the distance between the two observations becomes larger (AR1). For these two cases, the existance of the closed-form solution are shown using sufficient complete statistics. Then these closed-form solutions are obtained, also various kinds of weights, including optimal weights are computed for each case. On top of the already discussed weights, as we have clusters of all different sizes in each sub-sample here, we have proposed another type of weights, as follows,

$$w_{size-prop,i} = \frac{m_i n_i}{\sum_{k=1}^M m_k n_k}, \quad (1.7)$$

In some cases that the cluster size is also important (other than the number of subjects in each sub-sample), such weights would perform better. We have shown that AR(1) is one of these cases, while for the CS, proportional weights would work fine.

Note that, the structured horizontal splitting idea is very general and all the introduced methodologies regarding different weights, combination rules, etc., can be used for any suitable structure of splitting (as long as they are horizontal), but in this thesis we only consider the cases where the structure is defined based on size of the clusters.

Random vertical splitting

The horizontal splitting techniques that we have introduced and discussed so far would be beneficial when the number of clusters (N) is large, but the cluster size is still small or moderate, so by breaking down the large N into M smaller sub-samples, we can manage to solve our problem. But there are examples where the size of some or all of the clusters is also very large. And as we have mentioned, large for a cluster size has a much smaller magnitude compared with large for the number of clusters. In such cases, horizontal splitting would not help much, because each cluster in our sub-samples is still large, so by horizontal splitting we just breakdown one problem into M problems.

As an example, take the Amazon book ratings. There, we have the ratings for many books given by different buyers. Therefore, each book could possibly be rated by thousands of people. This will for very large clusters where even the simple models would fail to converge. In such a case, horizontal splitting cannot solve our problem. To solve the issue here, we have to perform the splitting within the subject. For example, if there are 5000 people rating a book, in order to make smaller clusters in our sub-samples, we have to split these 5000 people, e.g., into 100 sub-samples, each of size 50.

As long as the interest is in the parameter estimate, the combination rules in (1.1) is still valid. But the difficulty will appear in deriving the combination rule for the variances. As the data from different

clusters are shared among different sub-samples, in other words, one cluster is present in more than one sub-sample, the estimated parameters from different sub-samples are correlated, and obviously, when combining the variances, these correlations should be taken into account.

One way to deal with the combination rule of variance in case of vertical splitting is to look at our sample splitting based model fitting in the pseudo-likelihood methodology, rather than the likelihood. In fact, once the dataset is analyzed in whole we use the conventional likelihood theory by finding the estimate of our parameters by maximizing the likelihood function. But when the data is splitted into some sub-samples, each of these sub-samples are analyzed using a likelihood theory, but their combination is not the original likelihood function anymore. It is called pseudo-likelihood. Roughly speaking, pseudo-likelihood theory tries to replace the complex or difficult likelihood function with something simpler or easier. But maximizing both likelihood or pseudo-likelihood functions would follow the same goal, i.e., estimate the same set of parameters, but with different precision.

From Cramér-Rao's lower bound we know that the lower bound for variance can be achieved by a maximum likelihood estimator. In other words, any other estimator (including maximum pseudo-likelihood) would come with a variance larger than the one from a maximum likelihood. Therefore, having the easier problem to solve comes at the price of losing a bit of accuracy.

Let us define the pseudo-likelihood in plain English. Assume we have a sample \mathbf{y} and using that we want to estimate the parameters vector θ via the log-likelihood function $\ell(\theta; \mathbf{y})$. Via sample splitting, we split the sample \mathbf{y} into M sub-samples $\mathbf{y}_1, \dots, \mathbf{y}_M$, then feed them to the same likelihood function and estimate the parameter vector θ M times. Under pseudo-likelihood paradigm, one can look at this as estimating a parameter vector $\Theta = (\theta_1, \theta_2, \dots, \theta_M)$ using the sample $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$ via maximizing the pseudo-likelihood function $p\ell(\Theta, \mathbf{Y})$. Note that, $\theta_1 = \theta_2 = \theta_3 = \dots = \theta_M$.

Now that the problem is defined under a pseudo-likelihood theory, all the results obtained there can be carried over to the data splitting as well. If we consider matrix A that combines the estimates, i.e., $\tilde{\theta} = A\hat{\Theta}$, then the most important result we can present is:

$$\sqrt{N}(\hat{\theta} - \theta) = \sqrt{N}(A\hat{\Theta} - A\Theta) \xrightarrow{d} N\left(\mathbf{0}, AI_0^{-1}I_1I_0^{-1}A^T\right), \quad (1.8)$$

where T means the transpose, and I_0 and I_1 are defined as follows,

$$I_0(\Theta) = E\left(\frac{\partial^2 p\ell(\Theta; \mathbf{Y})}{\partial \Theta^T \partial \Theta}\right), \quad I_1(\Theta) = \left[\left(\frac{\partial p\ell(\Theta)}{\partial \theta}\right)^T \frac{\partial p\ell(\Theta)}{\partial \theta}\right]. \quad (1.9)$$

More details on pseudo-likelihood interpretation of the data splitting can be found in ?. However, a main difficulty to find the variance in (1.8) if the need to cluster-wise hessian and gradient of the pseudo-likelihood function. This could easily become expensive. Another
Page 19

difficulty with this approach comes from the fact that, depending on the derivative and hessian means this variance estimator needs to computed case by case. For example, the results obtained for a logistic regression cannot used for another model without the necessary modification. To overcome these difficulties, we have considered an alternative solution proposed by ? in different context. Their approach is called within cluster re-sampling, which was later on extended and renamed to *multiple outputation* by ?.

In order to explain the motivation behind the multiple outputation technique, consider clustered data where the cluster consists of blood pressure of a patient visiting a hospital several times. The objective of the study is the severity of a disease which is highly correlated with blood pressure (more severe the disease, the larger the blood pressure). As the blood pressure is measured every time the patient is visiting the hospital, a larger cluster size, means the patient has visited the hospital more, which itself means the patient had more issues with the blood pressure, so a more severe state of the disease. In other words, in such situations the size of the cluster is also correlated with the response and objective of the study. Such situations are also called *informative cluster size* in the literature. The correlation of cluster size and the response should be respected in the model, otherwise it could cause issues. Multiple outputation has been proposed to deal with this problem.

Within cluster resampling proposed by ? to deal with the so-called informative cluster size, when it is a nuisance parameters, i.e., we are

not interested to study the correlation of the response and cluster size, we just want estimates of other parameters which are not affected by this correlation. ? proposed to take sub-samples of size 1 from each cluster repeatedly to form a sample on non-clustered data every time. Then the conventional methods can be used for each of these non-clustered data. The results then can be combined via combination rules which are similar to (1.1) for the estimates. But for the variance ? proposed the following combination rule:

$$\text{Var}(\tilde{\theta}) = W - \left(1 + \frac{1}{M}\right) B, \quad (1.10)$$

where W and B are within and between sub-samples variances respectively. These quantities are defined as follows,

$$W = \frac{1}{M} \sum_{i=1}^M \sigma_m^2, \quad B = \frac{1}{M-1} \sum_{i=1}^M (\hat{\theta}_i - \tilde{\theta})^2. \quad (1.11)$$

As one may see, the combination rule for the variance in (1.10) is also intuitively interesting also. It states, the variance of the data splitting estimator is an average of the variance in different sub-samples, but when we subtract it from the sample variance of estimated parameter from different sub-samples. Looking at this combination rule, one might also understand the reason behind naming this method multiple outputation by ?. That comes with various similarities of this technique with multiple imputation. Although, the similarities are technically rather than literally.

First let us say a word on multiple imputation itself. Multiple
Page 21

imputation (MI) is an effective way to deal with missing data issue. The traditional missingness remedies would try to either ignore that missing part of the sample, or replace it with a single plausible value. This way, one would treat a missing before and now imputed value the same as a value which is originally observed. In order to respect the fact that imputed values are actually imputed and as well could be any other value, ? proposed to replace each missing value with several values and not just one single value.

Now we may look at similarities of MI and MO. As it was mentioned, multiple imputation creates artificial data and augment it to the incomplete sample to make it complete, on the contrary, but in a similar way, multiple outputation would every time put some part of the same out of the analysis. Therefore, multiple **imputation** increases the sample size by including new artificial data, while multiple **outputation** decreases the sample size by selecting only a subset of them. But in both of the methods, we will have several datasets to analyze rather than the one single original dataset. So, in both methods we estimate each parameter several times. This is also reflected in the variance estimator. Let us remind the combination rule for variance from ?,

$$\text{Var}(\hat{\theta}_{MI}) = W + \left(1 + \frac{1}{M}\right)B. \quad (1.12)$$

As one may see, the only difference of (1.12) and (1.10) is in the fact that MI would add the between datasets variability of the estimated parameter to the averaged variance, while MO would
Page 22

subtract this. And intuitively, that would also make sense, in case of MI we artificially increase the sample size, so we would get a smaller variance which needs a correction (to become larger, as it should be), but in case of MO we decrease the sample size, so a larger variance would be obtained, which is also corrected in the same way as MI just with a different sign.

Because of the discussed similarity and for the vast use of multiple imputation in practice, we have developed part of contributions with the target to be used in data splitting for the case of multiple imputation. As for many cases, the conclusions and results from MI are also valid for MO, and vice versa.

We have extended the idea in within cluster re-sampling and multiple outputation to it them suitable for our purpose: analyzing clustered data with big clusters. The so-called iterative multiple outputation (IMO) procedure is proposed to fulfil this role. IMO is proposed as follows,

1. **Start.** Select an initial number of sub-samples, M_0 , and sub-sampling size m . Take M_0 sub-samples of size m , fit the model to each and obtain $\hat{\theta}_i$ and its variance $\Sigma_{\hat{\theta}_i}$ ($i = 1, \dots, M_0$). Then compute

$$\tilde{\boldsymbol{\theta}}_{M_0} = \sum_{i=1}^{M_0} \hat{\boldsymbol{\theta}}_i, \quad \Sigma_{\tilde{\boldsymbol{\theta}}_{M_0}} = \widehat{W}_{M_0} - \left(\frac{M_0 + 1}{M_0} \right) \widehat{B}_{M_0}. \quad (1.13)$$

2. **Update.** For $m > M_0$,

$$\tilde{\boldsymbol{\theta}}_{m+1} = \frac{m\tilde{\boldsymbol{\theta}}_m + \hat{\boldsymbol{\theta}}_{m+1}}{m+1}, \quad \Sigma_{\tilde{\boldsymbol{\theta}}_{m+1}} = \widehat{W}_{m+1} - \left(\frac{m+1}{m}\right) \hat{B}_{m+1}. \quad (1.14)$$

3. **Distance.** Compute: $d_{m+1} = d(\tilde{\boldsymbol{\theta}}_{m+1}, \tilde{\boldsymbol{\theta}}_m)$ using an appropriate distance.

4. **Stopping rule.** $d_j < \varepsilon$ for $j = m+1, \dots, m+k_0$.

Where for $m = r$ we have,

$$\widehat{W}_r = \frac{\sum_{i=1}^r \sigma_{\hat{\theta}_i}}{r}, \quad \hat{B}_r = \frac{\sum_{i=1}^r (\hat{\theta}_i - \tilde{\theta}_r)(\hat{\theta}_i - \tilde{\theta}_r)'}{r-1}. \quad (1.15)$$

We may remark that, there are cases only $M = 1$ sub-sample would be sufficient to obtain results almost as efficient as analyzing the full dataset. We have studies this matter in detail and called such estimators *finite information limit estimators*. We have also proposed procedures to detect this property.

A very useful dataset to illustrate the usefulness of random vertical splitting is the Amazon's book ratings dataset. This set of data consists of ratings of several books (from 1 to 5) on Amazon website. Each book is rated by different number of people, which could be as small as 1 and as large as 20,000. So we will deal with clustered data of very variable and large sizes.

Structured vertical splitting

So far we have discussed random horizontal and random vertical splitting, as well as structured horizontal splitting. In case of the vertical splitting also, pre-defining some structures to take the sub-samples other than taking them just at random could be beneficial. We have already seen one application of structured splitting in case of horizontal splitting, which enabled us to form balanced sub-samples. Defining a structure is not always for computation efficiency. Sometimes, without such structure, estimating the model parameters is not possible.

We have considered an important case of jointly modeling several outcomes via random effects as a useful example for structured vertical splitting. Such a problem can easily become infeasible when the number of outcomes becomes large (even tens of them). ?? proposed to analyze all possible pairs of all outcomes instead of analyzing them simultaneously. Then use appropriate combination rules to find the parameters of the model of interest. To illustrate the problem, also to show how structural vertical splitting is similar to the approach of ???. Also, to see how it is different from random vertical splitting, we present a simple example.

Imagine we have 3 outcomes to be jointly modeled. A random effects approach would fit a separate model to each outcome, then let them to be correlated by defining a joint distribution for their random effects. This joint distribution is usually a normal distribution where its covariance matrix would capture the dependencies between

different outcomes. Let us elaborate more here, if the three separate models (all linear) for the three outcomes are defined using the following general form:

$$\begin{cases} y_{1ij} = \beta_1 + b_{1i} + \epsilon_{1ij} \\ y_{2ij} = \beta_2 + b_{2i} + \epsilon_{2ij} \\ y_{3ij} = \beta_3 + b_{3i} + \epsilon_{3ij}, \end{cases} \quad (1.16)$$

we let the three random intercepts to be normally distributed as follows,

$$\begin{bmatrix} b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ & D_{22} & D_{23} \\ & & D_{33} \end{bmatrix} \right). \quad (1.17)$$

Therefore, the parameter of interest, $\boldsymbol{\theta}$, is as follows:

$$\boldsymbol{\theta} = (\beta_1, \beta_2, \beta_3, D_{11}, D_{22}, D_{33}, D_{12}, D_{13}, D_{23}). \quad (1.18)$$

As we have 3 outcomes, the number of pairs will be $M = 3 \times (3 - 1)/2 = 3$. However, unlike random vertical splitting where usually all parameters of interest are estimated in each sub-sample, here, because of the special structure we have imposed on the sub-samples, each of them would estimate part of parameter vector.

The parameters estimated in each pair $(\boldsymbol{\theta}^{(s)})$ are as follows:

$$\begin{cases} \boldsymbol{\theta}^{(1)} = \theta_{(y_{1ij}, y_{2ij})} = (\beta_{11}, \beta_{21}, D_{11_1},, D_{22_1}, D_{12_1}) \\ \boldsymbol{\theta}^{(2)} = \theta_{(y_{1ij}, y_{3ij})} = (\beta_{12}, \beta_{32}, D_{11_2},, D_{33_2}, D_{13_2}) \\ \boldsymbol{\theta}^{(3)} = \theta_{(y_{2ij}, y_{3ij})} = (\beta_{23}, \beta_{33}, D_{22_3},, D_{33_3}, D_{23_3}), \end{cases} \quad (1.19)$$

as one may see in (1.19), each $\boldsymbol{\theta}^{(s)}$ ($s = 1, 2, 3$) is a subset of $\boldsymbol{\theta}$ in (1.18). Now, the stacked vector combining all parameters estimated in different sub-samples $\boldsymbol{\Theta}' = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(3)})$ can be constructed as follows:

$$\boldsymbol{\Theta}' = (\beta_{11}, \beta_{21}, D_{11_1},, D_{22_1}, D_{12_1}, \beta_{12}, \beta_{32}, D_{11_2},, D_{33_2}, D_{13_2}, \beta_{23}, \beta_{33}, D_{22_3},, D_{33_3}, D_{23_3}) \quad (1.20)$$

As one may see, the covariance parameters (D_{12}, D_{13}, D_{23}) are only estimated once using their corresponding pair, but the rest of parameters are appearing in two out of three pairs. Therefore, to combine these estimates we cannot just average all estimates as we did in (1.1). The weight matrix to perform the appropriate

combination can be defined as follows,

$$A = \begin{pmatrix} \beta_{11} & \beta_{21} & D_{111} & D_{221} & D_{121} & \beta_{12} & \beta_{32} & D_{112} & D_{332} & D_{132} & \beta_{23} \\ \beta_1 & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \beta_2 & \\ \beta_3 & \\ D_{11} & \\ D_{12} & \\ D_{13} & \\ D_{22} & \\ D_{33} & \\ A = & \\ D_{23} & \end{pmatrix}$$

Obviously, $\sum_i A_i$ (with A_i the i th row of A) shows the number of times each parameter is estimated. Therefore, by defining W with each of its rows computed as $A_i / \sum_i A_i$, and pre-multiplying it by $\hat{\Theta}$ would provide the appropriate averaging. The same goes for the variance combination rule. ?? used a pseuo-likelihood approach to find the combined covariance matrix (the same as in 1.8), we could show using 1.10 with appropriate weights works the same, but much faster.

As one may see, only the covariance matrix of the random effects have 6 parameters to be estimated, if one decides to add random slopes also, then the number of parameters will increase to $(6 \times 7)/2 = 21$. Once the number of responses becomes slightly large, the number of parameters will dramatically increase. This would face

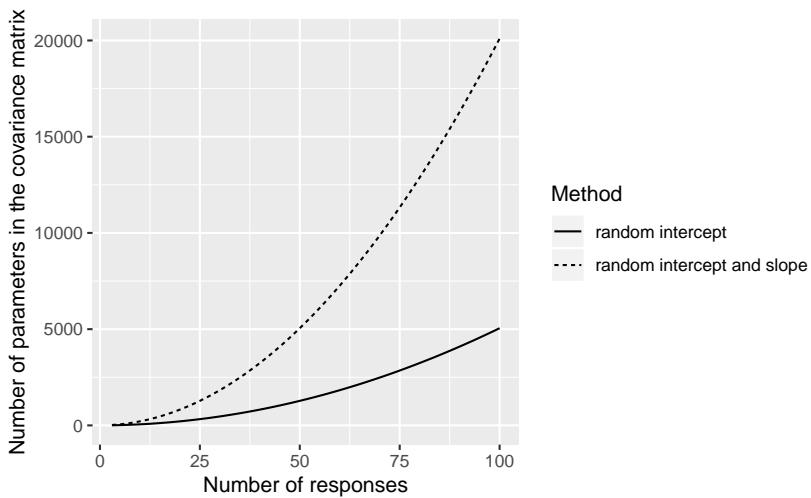


Figure 1.3: Number of parameters in the covariance matrix of jointly normally distributed random effects for different number of responses using only random intercept or random intercept and slope models.

all conventional methods with serious problems, and the problem can be unfeasible. Figure 1.3 shows how the number of parameters only in the covariance matrix of random effects would increase when the number of responses increases. As we may see in this figure, even for less than 100 responses, the number of parameters becomes extremely large.

Therefore, there is a main difference between the applications we have proposed for random vertical splitting, and structured vertical splitting. While both of them are proposed to deal with large clusters, the issue with random vertical splitting is a large sample size, while structured vertical splitting deals with a high-dimensional parameter space. In other words, in case of structured random

splitting we are dealing with the so-called curse of dimensionality (?). In fact, exactly for this *big* parameter space, we had to impose a structure on it, so in every sub-sample a sub-set of this parameter vector could be estimated. The smart idea of modeling several outcomes by all their sub-sets of two would break down the parameter space by breaking down the sample itself. That is why, in case of random splitting, every sub-sample usually estimates all parameters, while this is not often the case for structured vertical splitting.

We have two examples in our motivational datasets which are used to illustrate the applications of structured vertical splitting. We have used *Leuven diabetes project* data as a basis to perform simulation studies to evaluate our proposed way of combining variances with what pseudo-likelihood theory would give. Furthermore, the *Hearing data* is used as an example of high-dimensional parameter space where the conventional methods would fail.

Supplementary topics

As it was mentioned, due to several technical similarities between random vertical data splitting (iterative multiple outputation) and multiple imputation, and due to wide acceptance of MI in different research fields, we have developed part of our required methodology in MI language rather than language of IMO. However, the conclusions are valid in both cases. In this section we briefly go over these contributions which will be presented in Chapter **NUMBER OF**
Page 30

CHAPTERS.

How many sub-samples?

The iterative multiple outputation (IMO) procedure that we have proposed for random vertical splitting displays a stopping rule at step 4. This is used to determine the number of sub-samples. Let us briefly explain the idea behind this stopping rule in this section. Obviously, more sub-samples are better in case of obtaining a more precise estimate. But on the other hand, as the sub-samples are with replacement, one can take an infinite number of sub-samples, so there is a need for a rule to stop the IMO procedure.

As we have seen in (1.1), the combination rule of data splitting for parameter estimate is averaging. From laws of large numbers we know that under general regularity conditions, as the sample size increases, the average converges to a finite quantity. We have seen that also via central limit theorem in (1.8). As we average over sub-samples, sample size here means the number of sub-samples. Therefore, we would expect, after some point, increasing the number of sub-samples would not impressively change the final results. But it is important to come up with appropriate criteria and stopping rule to stop the sub-sampling at the right moment. It is important for two main reasons: 1- if we stop early, we would fail to provide precise and correct estimates, 2- if we stop too late our initial intention for saving computation time could be compromised.

We have already explained the connection of multiple imputation
Page 31

and IMO, also how they would suffer from the same issues. In multiple imputation also, a main issue is to determine the number of imputed datasets. Therefore, we have studied this question in detail via simulation studies and theoretical discussion. Various distance function (Mahalanobis, Euclidean, etc.) were examined for different situations and as a result we proposed the iterative multiple imputation procedure. The obtained results are all valid in case of IMO, therefore, we have used the same stopping rule in IMO and in IMI.

In order to illustrate IMI, we have used two datasets, the *Leuven Eye Study* (LES) and the *Cholesterol data*. LES is collected via one of the largest observational studies for glaucoma. However, the dataset includes a lot of missing data. Practically, no fully observed line exists in this dataset, so without MI the analysis was not possible. The second dataset is a famous dataset used many times to illustrate different aspects of multiple imputation.

Special case: where combination rule fails

A fundamental rule which should be followed when using the combination rule in (1.1) is to average parameters which are counterparts of the same parameter in the model for the full sample. In some applications, finding such counter parts is not a straightforward task. An example of such situations is principal component analysis (PCA). Methods like PCA are working based on eigenvalue decomposition of the covariance (or correlation) matrix. Eigenvalues of a matrix Σ can be obtained by solving the following equation,

$$|\Sigma - \lambda I|,$$

where $|.|$ for a matrix denotes the determinant of it and I is the identity matrix of the same order as Σ . The roots of this equation are the eigenvalues. Obviously, there is no natural ordering among roots of a polynomial but as in PCA one can show that (?) these eigenvalues are actually variances of their corresponding principal components, therefore, a PC with a larger eigenvalue explains a larger proportion of variance.

The issue arises when we use data splitting or multiple imputation where the eigenvalue decomposition should be done for several subsamples or several imputed datasets. There is no guarantee that the largest eigenvalue from first datasets corresponds to the largest eigenvalue from the second dataset. The problem of combining results of PCA or exploratory factor analysis at the factor (or principal components) level has been studies in ?.

Our alternative solution here to first estimate the covariance matrix as out parameter of interest, and then perform the eigenvalue decomposition on this single estimated covariance matrix. We have studied this idea in detail and investigated different aspects of it, also compared it with other alternative solutions. Again, the problem is considered in the setting of multiple imputation, but all the findings are also valid for data splitting. This is because, the issue in this case is that the dataset under study is replaced with several datasets. Now these several datasets could come from

data splitting or multiple imputation, that would not change the problem, hence, the proposed solutions.

Software implementation

Our main concern is R, but for comparison reasons, we have also used SAS.

Chapter 2

Motivating datasets

There are two type of motivating datasets in this thesis: 1- the ones that using our proposed methodology is necessary to analyze them, 2- the ones that these methodologies are an option, and probably not the best one. The first type of motivating datasets are large data where analyzing them with conventional methods is not feasible, or would a very long time. As we also need to evaluate our proposed methodology not only via simulation studies, but in real practice also. In our second type of motivating datasets, we consider cases that are suitable but not large enough, so the conventional methods could also be applied on them. This way, we could compare the performance of our proposed alternative methodology with the well-known methods.

A brief summary of the datasets motivating our research is presented in this section.

A developmental toxicity study

This study investigates the dose-response relationship in mice of the potentially hazardous chemical compound di(2-ethylhexyl)phthalate (DEHP), used in vacuum pumps and as plasticizers for numerous plastic devices made of polyvinyl chloride (?). The developmental toxicity study, conducted in timed-pregnant mice during the period of major organogenesis has attracted much interest in the toxicity of DEHP. The doses selected for the study were 0, 0.025, 0.05, 0.1, and 0.15%, corresponding to a DEHP consumption of 0, 44, 91, 191, and 292 mg/kg/day, respectively.

The dams were sacrificed, slightly prior to normal delivery, and the status of uterine implantation sites recorded. A total of 1082 live fetuses were dissected from the uterus, anesthetized, and examined for external, visceral, and skeletal malformations, as well as for body weight. Our focus will be on the continuous weight outcome. Evidently, fetuses are clustered within mothers; hence the implied association needs to be accommodated in the analysis.

As each mother has a different number of fetuses, we have clusters of different sizes. This would provide a suitable scenario to evaluate structured horizontal splitting approach. Because, as the data is of a moderate size, the conventional methods can also be used to fit the models, so we have the chance of comparing the obtained results using such models with our proposed alternative models.

are also studied in this thesis, ethylene glycol (EG), citeprice1987, and diethylene glycol dimethyl ether (DYME), ?.

Divorce in Flanders

Big Five Inventory (BFI) is questionnaire formed of 44 questions to measure personality based on five underlying factors: Neuroticism (N), Extraversion (E), Openness to Experience (O), Conscientiousness (C), and Agreeableness (A), see ?. A Dutch translation of this questionnaire have been developed and validated by ?. The same translated version have been used in Divorce of Flanders project to collect the data.

The Divorce of Flanders dataset is collected at the Centre for Sociological Research, KU Leuven. The data concerned on studying and comparing the personality among three generations in about 4000 Flemish families (?). Each family would at most consist of mother, father, step-mother, step-father, one child, grand-mother, and grand-father. The personality is measured using the so-called Big Five Inventory (BFI) score as it is described above. These scores are built upon 5 factors which in total are measured through 44 items.

Although, using factor analysis is required for such data in order to confirm the existence of the expected underlying factors, but it cannot answer all of our questions. For example, the researcher wants to see how the correlations between different personality aspects and family roles (child, mother, father, etc.) would look

like. The usual factor analysis would lose most of the hierarchical information in the data.

The hierarchical information we have mentioned comes from the fact that, there are family members who answer 44 questions which themselves will form 5 factors. Therefore, We have five factors combined with family roles as our response variables. If we only consider 3 family roles (child, mother, and father), it will form 15 different responses (5 personality aspects and 3 family roles). In other words, in this example, each of our clusters consists of a family with its members (likely 3 or 5), combined with 5 factors, answering 44 questions. And we will have more than 4000 of such families. Conventional software like SAS would immediately fail to fit such models, but as we will see, splitting this large sample into 3 parts would help to obtain the required results. Therefore, this is an example of case studies where the conventional methods fail to work and our proposed alternative methods are necessary to fit the required models.

It worth to mention that we have well studied and validated these data and the Dutch translation of the questionnaire in case of a Flemish sample, see ?, ?. While validating the Dutch translation for the Flemish sample, in order to deal with the missing data problem, we also had to develop methodologies to combine multiple imputation (??). Our contribution to this field had also applications in case of data splitting, this will be discussed in Chapter **the number of the chapter should be added.**

Hearing data

To evaluate the hearing performance of a subject, hearing threshold sound pressure levels (measured in dB) are considered. By definition, a hearing threshold is the lowest signal intensity possible to perceive by a subject at a specific frequency. In a study considered in ? and ?, hearing thresholds measured in 11 different frequencies between 125 Hz and 8000 Hz for left and right ears obtained on 603 male participants from the Baltimore Longitudinal Study of Ageing, BLSA. The number of visits for each subject which varies between 1 to 15 are unequally spaced.

As one may already observed, not only we have 22 outcomes which should be simultaneously modeled, but the study is also longitudinal. Therefore, one may need both random intercepts and random slopes. That means estimating a 44×44 covariance matrix of random effects, as well as the fixed effects. Conventional software like **Proc MIXED** in SAS would immediately fail to estimate such huge model. While our proposed structured vertical splitting can deal with the issue and provide necessary estimates and results.

Leuven diabetes project

In order to study how well diabetes is controlled, three outcomes were measured for each patient at baseline (T_0) and one year later (T_1). These three outcomes of interest were LDL-cholesterol (low-density lipoprotein cholesterol, mg/dl), HbA1c (glycosylated hemoglobin, %), and SBP (systolic blood pressure, mmHg).

Although these three variables are continuous, the interest was in expert-specified cut-off values of these outcomes. The cut-off values are defined as $\{< 100, [100, 115), [115, 130), \geq 130\}$ (mg/dl) for LDL-cholesterol, $\{< 7, [7, 8), \geq 8\}$ % for HbA1c, and $\{\leq 130, (30, 140], (140, 160], > 160\}$ mmHg for SBP. Modelling these 3 ordinal outcomes together provides a suitable platform to study the performance of our proposed methodologies dealing with non-Gaussian data.

Note that, this data is analyzed in ?, and here we only use it as a basis for our simulation studies to evaluate our proposed methodology in case of generalized linear mixed models.

Amazon book ratings

These data consist of the ratings of the books on the Amazon website which are recently made publicly available. The dataset contains 2,330,066 books, which are rated at least 1 and at most 21,398 times. This is an example of clustered data with large highly unbalanced clusters and a large sample size. Therefore, it provides an ideal situation where the conventional methods would fail and we may need alternative methods to deal with the big data issue.

Clinical Trials in Schizophrenia

These data were collected from five double-blind randomized clinical trials to compare the effects of two treatments for chronic schizophrenia: risperidone and conventional antipsychotic agents.

Subjects who received doses of risperidone (4–6 mg/day) or an active control (haloperidol, perphenazine, zuclopentixol) have been included in the analysis. Patients were clustered within country, and longitudinal measurements were made on each subject over time. The number of patients ranges from 9 to 128 per country with a total of 2039.

The positive and negative syndrome scale (PANSS) was used to assess the global condition of a patient. This scale is constructed from 30 items, each taking values between 1 and 7, giving an overall range of 30 to 210. PANSS provides an operationalized, drug-sensitive instrument, which is useful for both typological and dimensional assessment of schizophrenia. Depending on the trial, treatment was administered for a duration of 48 weeks with at most 12 monthly measurements. Because not all subjects received treatment at the same time points and, not the same amount, the final dataset is unbalanced. Our analysis shows the AR(1) structure could be a good candidate for modelling the covariance matrix of the random effects, which makes it a suitable example to illustrate our contribution to structured horizontal splitting based on cluster sizes for AR(1) models.

Cholesterol data

The cholesterol dataset includes cholesterol levels for 28 patients treated at a Pennsylvania medical center. The cholesterol levels have been recorded on day 2, day 4 and day 14 after an attack for
Page 41

each patient. However, there are 9 missing values for day 14. This dataset was analyzed by ?, Chapter 5 and is publicly available in R package **norm2**. We will use this dataset to illustrate our proposed procedure iterative multiple imputation (IMI) in case of Student's *t*-test.

Leuven eyes study

This is an extensive observational study of glaucoma performed at the ophthalmology department of KU Leuven. This study investigates glaucoma which is one of the leading diseases to cause irreversible vision loss. Early diagnosis of glaucoma could greatly help to delay the vision loss. Therefore, studying various diagnostic methods is of a great importance.

The dataset so far consists of 141 variables measured in 614 subjects. Various diagnostic tests and procedures are used on subsets of these patients. As measuring all of these variables and performing all diagnostic tests was not feasible for all of the subjects, this dataset exhibits considerable missingness. In fact, no single fully observed subject can be found in this dataset. Considering the size of dataset and amount of missing values, applying multiple imputations and determining the number of imputations is challenging in this case.

Chapter 3

Random horizontal splitting

Random horizontal splitting is useful for the cases where the dataset becomes *big* because of number of clusters in there. We have already discussed the splitting pattern as well as the combination rules in this case. In this chapter we will use random horizontal splitting to fit a model to a subset of Divorce in Flanders dataset that could not be fitted without splitting. First, we explain the research question and its statistical formulation, then we will see how conventional methods would fail in this case, and finally using random horizontal splitting we can fit the model that we could not fit otherwise.

The research question

Divorce in Flanders dataset presented data from Big Five Inventory (BFI) answered by Flemish families. This BFI is a questionnaire of 44 items which should be answered with scores from 1 to 5. In this chapter. These 44 questions are measuring 5 aspects of personality: openness to the experience, conscientiousness, extraversion, agreeableness, and neuroticism. Our interest is to study the correlation of each of these 5 aspects of personality between 3 main family roles: mother, father, and child. This would possibly reveal pattern is different characteristic qualities and how they are related between parents, and their child.

One may simply computes a correlation coefficient between these variables, but that would ignore the intraclass correlation coming from the fact that some of the sample members are coming from the same family. In other words, we have two sources of correlation here: one as a result of correlation between different personality aspects and family roles, and the other which reflects the fact that the subjects are coming from the same family. We are interested in the former.

To properly take this hierarchical structure into account, we proposed to define 15 responses formed by all combination of 5 personality factors and 3 family roles. Therefore, are are interested between the associations between all these 15 factor-roles. A mixed model can be used to let all these responses of interest be correlated.

A mixed model solution

We proposed to use a linear mixed model to jointly model all 15 responses that are formed by combining 5 personality factors and 3 family roles. We call these 15 variables factor-role hereafter. Consider y_{ijk} as the response for the i -th factor-role for j -th item in the k -th family. Then We need to fit the following mixed model,

$$y_{ijk} = \beta_i + b_{ij} + \epsilon_{ijk}, \quad b_{ij} \sim N(0, d_{ii}), \quad \epsilon_{ijk} \sim N(0, \sigma_1^2) \quad (3.1)$$

b_{ij} 's ($i = 1, \dots, 15, j = 1, \dots, m_j$) are the random effects (latent variables), as we are interested in the association between all factor-roles, they all will be considered to be correlated (an unstructured covariance matrix) following a normal distribution with mean zero and covariance matrix D presented in (3.2). In there the first letter corresponds to the factors (**A** for agreeableness, **C** for conscientiousness, **E** for extraversion, **N** for neuroticism, and **O** for openness,) and the second letter corresponds to family roles (**c** for child, **m** for mother, and **f** for father).

$$D = \left(\begin{array}{cccccccccc} d_{Ac} & d_{Ac,Af} & d_{Ac,Am} & d_{Ac,Cc} & d_{Ac,Cf} & d_{Ac,Cm} & d_{Ac,Ec} & d_{Ac,Ef} & d_{Ac,Em} & d_{Ac,Nc} & d_{Ac,Nf} \\ d_{Af} & d_{Af,Am} & d_{Af,Cc} & d_{Af,Cf} & d_{Af,Cm} & d_{Af,Ec} & d_{Af,Ef} & d_{Af,Em} & d_{Af,Nc} & d_{Af,Nf} \\ d_{Am} & d_{Am,Cc} & d_{Am,Cf} & d_{Am,Cm} & d_{Am,Ec} & d_{Am,Ef} & d_{Am,Em} & d_{Am,Nc} & d_{Am,Nf} \\ d_{Cc} & d_{Cc,Cf} & d_{Cc,Cm} & d_{Cc,Ec} & d_{Cc,Ef} & d_{Cc,Em} & d_{Cc,Nc} & d_{Cc,Nf} \\ d_{Cf} & d_{Cf,Cm} & d_{Cf,Ec} & d_{Cf,Ef} & d_{Cf,Em} & d_{Cf,Nc} & d_{Cf,Nf} \\ d_{Cm} & d_{Cm,Ec} & d_{Cm,Ef} & d_{Cm,Em} & d_{Cm,Nc} & d_{Cm,Nf} \\ d_{Ec} & d_{Ec,Ef} & d_{Ec,Em} & d_{Ec,Nc} & d_{Ec,Nf} \\ d_{Ef} & d_{Ef,Em} & d_{Ef,Nc} & d_{Ef,Nf} \\ d_{Em} & d_{Em,Nc} & d_{Em,Nf} \\ d_{Nc} & d_{Nc,Nf} \\ d_{Nf} \end{array} \right) \quad (3.2)$$

As one may see in 3.2, by estimating the D matrix, all possible associations between personality factors and family roles can be derived. The advantages of using a mixed for this problem can be summarized as follows:

- Specifying the theory-based model in a straightforward fashion (the same as confirmatory factor analysis).
- Properly taking different sources of correlations into account.
- If needed later on, including extra covariates in the model (such as education) is easily possible in an standard way.
- All required associations can be derived and tested in a proper modeling framework.

While considering the above benefits using a mixed model approach is very appealing, the number of parameters in the final model becomes very large. As one may see, in (3.2), in the D matrix alone there are 120 parameters to be estimated. Trying to fit the model in (3.1) in PROC MIXED in SAS would lead to **ERROR: The SAS System stopped processing this step because of insufficient memory.**

A data-splitting based remedy

While the cluster sizes are not large here (at most 3), considering the size of the sample we have used in this example (4460 families) also the number of outcomes of interest (15 factor-roles), one may think of two possible data splitting based solutions: random horizontal splitting, and structured vertical splitting. We may first try the former, and if it does not work, we may try the latter, or a combination of both methods.

In order to apply random horizontal splitting, we split 4460 families into three roughly equal sized sub-samples with the sizes 1480, 1480, and 1500. Therefore, the proportional weights to combine the results from the three sub-samples will be:

$$W = \left(\frac{1480}{4460}, \frac{1480}{4460}, \frac{1500}{4460} \right). \quad (3.3)$$

Now that the sub-samples are assigned and the proper weights are
Page 47

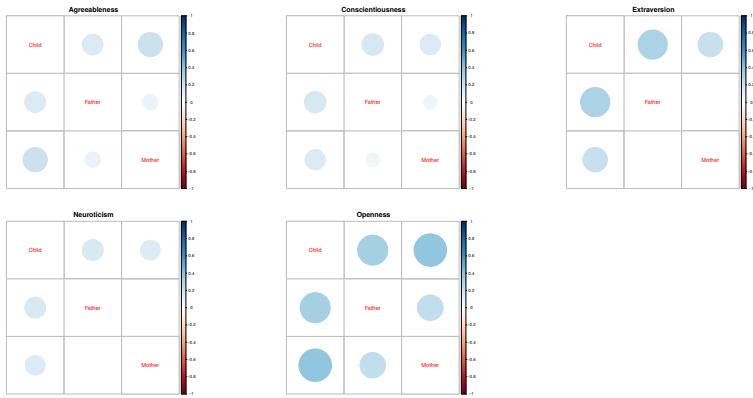


Figure 3.1: Estimated correlation matrix for each aspect of personality.

determined, we may fit the not feasible mixed model in (3.1) to each of these sub-samples, obtain the parameters, and use combination rules with weights in (3.3) to find the final estimates and their standard errors.

The detailed information on \tilde{D} is given in Table 3.2. The estimate for each of the 120 parameters together with standard deviation, 95% confidence interval, z -value and p -value are given in this table. Figure 3.2 shows the 120 parameter estimates in \tilde{D} with their corresponding 95% confidence interval. Just to make drawing conclusions easier, Figure 3.1 shows parts of the correlation matrix corresponds to each personality factor separately.

CovParm	Estimate	Std	95%-Lower	95%-Upper
A.child	0.183	0.013	0.158	0.208
(A.father,A.child)	0.028	0.010	0.008	0.048

A.father	0.172	0.008	0.156	0.187	21.1
(A.mother,A.child)	0.040	0.009	0.022	0.057	4.3
(A.mother,A.father)	0.015	0.007	0.002	0.029	2.1
A.mother	0.176	0.007	0.162	0.191	24.1
(C.child,A.child)	0.081	0.011	0.060	0.103	7.4
(C.child,A.father)	0.020	0.012	-0.004	0.044	1.0
(C.child,A.mother)	0.008	0.011	-0.014	0.030	0.7
C.child	0.318	0.018	0.282	0.354	17.1
(C.father,A.child)	0.011	0.011	-0.011	0.033	0.9
(C.father,A.father)	0.096	0.006	0.084	0.108	15.1
(C.father,A.mother)	0.009	0.007	-0.005	0.024	1.2
(C.father,C.child)	0.045	0.013	0.018	0.071	3.1
C.father	0.211	0.009	0.193	0.229	23.1
(C.mother,A.child)	0.020	0.010	0.000	0.039	2.0
(C.mother,A.father)	0.024	0.007	0.010	0.039	3.2
(C.mother,A.mother)	0.101	0.006	0.090	0.113	17.1
(C.mother,C.child)	0.042	0.012	0.018	0.065	3.4
(C.mother,C.father)	0.015	0.008	-0.001	0.031	1.8
C.mother	0.233	0.009	0.216	0.250	26.1
(E.child,A.child)	0.104	0.012	0.081	0.126	8.9
(E.child,A.father)	0.008	0.013	-0.018	0.033	0.5
(E.child,A.mother)	-0.002	0.012	-0.025	0.020	-0.1
(E.child,C.child)	0.076	0.014	0.049	0.102	5.9
(E.child,C.father)	0.039	0.015	0.011	0.068	2.0
(E.child,C.mother)	0.018	0.013	-0.007	0.043	1.4
E.child	0.375	0.020	0.335	0.414	18.1

(E.father,A.child)	0.020	0.013	-0.007	0.046
(E.father,A.father)	0.107	0.007	0.093	0.122
(E.father,A.mother)	0.004	0.009	-0.014	0.021
(E.father,C.child)	0.021	0.016	-0.010	0.052
(E.father,C.father)	0.132	0.008	0.116	0.147
(E.father,C.mother)	0.012	0.010	-0.008	0.031
(E.father,E.child)	0.116	0.017	0.083	0.150
E.father	0.359	0.013	0.333	0.385
(E.mother,A.child)	0.025	0.012	0.002	0.048
(E.mother,A.father)	0.003	0.009	-0.014	0.020
(E.mother,A.mother)	0.083	0.007	0.070	0.097
(E.mother,C.child)	-0.008	0.014	-0.036	0.020
(E.mother,C.father)	0.010	0.010	-0.008	0.029
(E.mother,C.mother)	0.141	0.008	0.126	0.157
(E.mother,E.child)	0.085	0.015	0.056	0.115
(E.mother,E.father)	-0.000	0.012	-0.023	0.022
E.mother	0.393	0.013	0.367	0.418
(N.child,A.child)	-0.136	0.013	-0.162	-0.110
(N.child,A.father)	-0.010	0.015	-0.038	0.019
(N.child,A.mother)	0.017	0.013	-0.009	0.043
(N.child,C.child)	-0.069	0.015	-0.099	-0.038
(N.child,C.father)	0.007	0.016	-0.025	0.039
(N.child,C.mother)	-0.030	0.014	-0.058	-0.001
(N.child,E.child)	-0.196	0.017	-0.229	-0.162
(N.child,E.father)	-0.034	0.019	-0.072	0.004
(N.child,E.mother)	-0.046	0.017	-0.080	-0.013

(N.child,N.Child)	0.490	0.026	0.439	0.540	18.
(N.father,A.child)	-0.023	0.014	-0.051	0.005	-1.
(N.father,A.father)	-0.152	0.008	-0.168	-0.136	-18
(N.father,A.mother)	-0.004	0.010	-0.023	0.015	-0.
(N.father,C.child)	0.019	0.017	-0.015	0.052	1.0
(N.father,C.father)	-0.124	0.009	-0.141	-0.108	-14
(N.father,C.mother)	-0.017	0.011	-0.038	0.004	-1.
(N.father,E.child)	-0.023	0.018	-0.059	0.014	-1.
(N.father,E.father)	-0.211	0.011	-0.232	-0.190	-19
(N.father,E.mother)	-0.002	0.013	-0.026	0.023	-0.
(N.father, N.child)	0.072	0.021	0.031	0.113	3.
(N.father,N.father)	0.404	0.015	0.375	0.434	26.
(N.mother,A.child)	-0.030	0.013	-0.055	-0.006	-2.
(N.mother,A.father)	0.000	0.010	-0.018	0.019	0.
(N.mother,A.mother)	-0.129	0.008	-0.144	-0.114	-16
(N.mother,C.child)	-0.018	0.015	-0.048	0.011	-1.
(N.mother,C.father)	0.002	0.010	-0.018	0.023	0.2
(N.mother,C.mother)	-0.117	0.008	-0.133	-0.101	-14
(N.mother,E.child)	-0.050	0.016	-0.081	-0.019	-3.
(N.mother,E.father)	0.008	0.013	-0.017	0.033	0.
(N.mother,E.mother)	-0.243	0.011	-0.264	-0.222	-22
(N.mother,N.child)	0.067	0.018	0.031	0.102	3.
(N.mother,N.father)	-0.001	0.014	-0.028	0.026	-0.
N.mother	0.428	0.015	0.399	0.457	29.
(O.child,A.child)	0.028	0.010	0.008	0.048	2.
(O.child,A.father)	-0.005	0.011	-0.027	0.016	-0.

(O.child,A.mother)	0.002	0.010	-0.018	0.022
(O.child,C.child)	0.026	0.012	0.003	0.050
(O.child,C.father)	0.015	0.013	-0.010	0.039
(O.child,C.mother)	-0.001	0.011	-0.023	0.020
(O.child,E.child)	0.092	0.013	0.066	0.117
(O.child,E.father)	0.032	0.015	0.003	0.061
(O.child,E.mother)	0.039	0.013	0.013	0.065
(O.child,N.child)	-0.024	0.014	-0.052	0.004
(O.child,N.father)	-0.009	0.016	-0.040	0.022
(O.child,N.mother)	-0.035	0.014	-0.063	-0.007
O.child	0.284	0.016	0.252	0.315
(O.father,A.child)	0.002	0.012	-0.021	0.025
(O.father,A.father)	0.041	0.007	0.028	0.054
(O.father,A.mother)	0.025	0.008	0.010	0.041
(O.father,C.child)	-0.035	0.014	-0.063	-0.008
(O.father,C.father)	0.057	0.007	0.043	0.071
(O.father,C.mother)	0.006	0.009	-0.011	0.024
(O.father,E.child)	0.028	0.015	-0.001	0.058
(O.father,E.father)	0.137	0.009	0.120	0.154
(O.father,E.mother)	0.010	0.010	-0.010	0.030
(O.father,N.child)	-0.016	0.017	-0.049	0.018
(O.father,N.father)	-0.065	0.009	-0.083	-0.047
(O.father,N.mother)	-0.014	0.011	-0.036	0.008
(O.father,O.child)	0.095	0.013	0.069	0.120
O.father	0.281	0.011	0.259	0.302
(O.mother,A.child)	0.026	0.011	0.005	0.048

(O.mother,A.father)	-0.005	0.008	-0.021	0.011	-0.64
(O.mother,A.mother)	0.047	0.006	0.034	0.059	7.39
(O.mother,C.child)	-0.006	0.013	-0.032	0.020	-0.40
(O.mother,C.father)	-0.024	0.009	-0.041	-0.006	-2.63
(O.mother,C.mother)	0.054	0.007	0.041	0.068	7.82
(O.mother,E.child)	0.014	0.014	-0.013	0.041	0.99
(O.mother,E.father)	-0.007	0.011	-0.028	0.015	-0.63
(O.mother,E.mother)	0.166	0.009	0.149	0.184	18.80
(O.mother,N.child)	-0.017	0.016	-0.048	0.013	-1.10
(O.mother,N.father)	0.013	0.012	-0.010	0.036	1.11
(O.mother,N.mother)	-0.100	0.009	-0.118	-0.082	-11.00
(O.mother,O.child)	0.117	0.012	0.093	0.141	9.48
(O.mother,O.father)	0.072	0.010	0.053	0.091	7.40
O.mother	0.314	0.011	0.292	0.335	28.80

Table 3.1: D -matrix parameter estimation (120 component \tilde{D})

Interesting observations can be made from the results in Table 3.1. For example, considering only the point estimates of correlations, we may see in case of Agreeableness, the correlation of mother and child is larger than the correlation of father and child. The same is true for Openness to experience. However, for Extraversion, this is larger for father than child. In case of Consciousness and Neuroticism, the correlations are more or less similar. Note that,

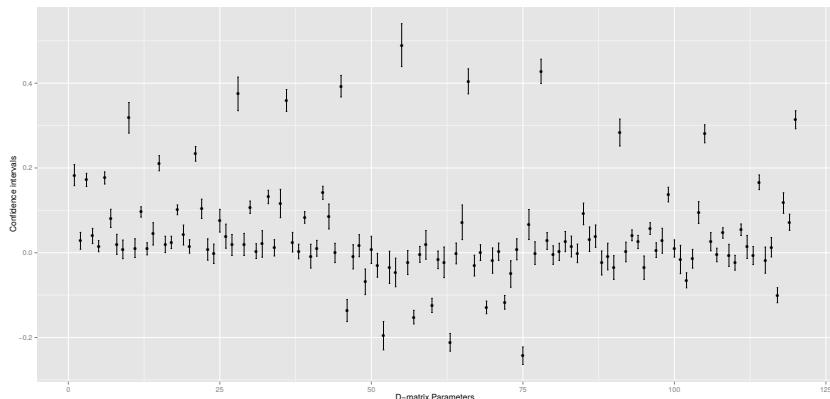


Figure 3.2: 120 components of \tilde{D} and their 95% confidence intervals

the correlations are not that large but still comparisons would be beneficial.

How precise are we?

As it was discussed, while data splitting would help the computation time, and even solve a problem that could not be solved otherwise, this all will come at a price: efficiency loss. Usually, it is not possible to quantify and study this efficiency loss, as the conventional maximum likelihood is not feasible, but one can have an idea about it by using both methods on the same problem but in a smaller scale. In this section we study the estimates using sample splitting in some sub-problems where the likelihood is feasible to compute. This way the performance of sample splitting can be compared with the full sample likelihood.

The sub-models would still include all family roles, but the person-

ality factors are considered separately. Therefore, each time we only have 3 outcomes of interest. The same model will be fitted using the same sample splitting in Section ???. Table 3.2 shows the results for D matrix. As one may see the results of sample splitting and full sample are very similar. This would provide a greater degree of confidence in what we have obtained for all of the 15 responses using random horizontal splitting.

Table 3.2: Comparing full sample and sample splitting results for each factor separately

Factor	Parameter	Estimate		Std	
		Split	Full	Split	Full
Agreeableness	child	0.176	0.186	0.013	0.013
	(father,child)	0.027	0.027	0.010	0.010
	father	0.176	0.170	0.008	0.008
	(mother,child)	0.044	0.042	0.009	0.009
	(mother,father)	0.021	0.015	0.007	0.007
Conscientiousness	mother	0.171	0.175	0.007	0.007
	child	0.275	0.320	0.017	0.018
	(father,child)	0.044	0.046	0.013	0.014
	father	0.202	0.210	0.009	0.009
	(mother,child)	0.042	0.044	0.011	0.012
Extraversion	(mother,father)	0.019	0.013	0.008	0.008
	mother	0.212	0.231	0.008	0.009
	child	0.324	0.374	0.019	0.020
	(father,child)	0.076	0.111	0.016	0.017
	father	0.298	0.355	0.012	0.013
Neuroticism	(mother,child)	0.074	0.083	0.014	0.015
	(mother,father)	0.014	-0.000	0.010	0.012
	mother	0.319	0.389	0.011	0.013
	child	0.400	0.490	0.023	0.026
	(father,child)	0.055	0.064	0.018	0.021
Openness	father	0.321	0.400	0.013	0.015
	(mother,child)	0.055	0.061	0.016	0.018
	(mother,father)	0.011	-0.001	0.012	0.014
	mother	0.346	0.424	0.013	0.015
	child	0.255	0.283	0.016	0.016
	(father,child)	0.070	0.095	0.012	0.013
	father	0.245	0.278	0.010	0.011
	(mother,child)	0.095	0.119	0.012	0.012
	(mother,father)	0.063	0.071	0.009	0.010
	mother	0.269	0.311	0.010	0.011

Chapter 4

Structured horizontal splitting

Clusters with unequal size: Maximum likelihood versus weighted estimation in large samples

Introduction

Much statistical theory is derived under the paradigm of a fixed sample size. However, there are many common practical settings in which this paradigm does not hold. Examples include sequential trials, where the trial may be stopped early at a number of time points during accrual, because of the strength, or lack, of a treatment effect; incomplete data in longitudinal studies or sur-

veys; longitudinal data with random measurement occasions; and censored survival data. ? provide an overview of such situations.

Here, we focus on hierarchical (or clustered) data with unequal cluster sizes.

Clustering is taken in its broadest sense, encompassing longitudinal data, family-based studies, toxicology (?), agricultural experiments, multi-level designs in the social and behavioral sciences, and so on. In longitudinal trials, it is not uncommon to plan for the same number of measurements to be taken per study subject, often at a common set of time points. If all data were collected according to protocol, the cluster size would be fixed. However, even in such studies, cluster sizes are often *de facto* random because of incompleteness in the data. In many random cluster size settings there may be associations between outcomes and cluster size. In part of the literature, this is termed ‘informative cluster size’ and a suite of methods has been proposed to accommodate this situation, many based on inverse probability weighting (????????). Unequal cluster sizes can occur for any outcome type, including continuous, binary, categorical, count, and event time.

Unequal cluster sizes may or may not be governed by a stochastic mechanism. For example, they can be unequal by design choice, without being stochastic; e.g., when a sample is selected in each town proportion to the population size. Litter sizes in pregnant rodents will truly be stochastic. When stochastic, the mechanism is completely random when it depends on neither observed nor

unobserved data; it is random when it depends on observed but, given these, not on unobserved data; other mechanisms are termed non-random. In the literature, mechanisms other than complete random are often termed informative. Although an important issue, we do not focus on informative cluster sizes here. Attention is confined to the case where cluster size is unequal, but independent of both observed and unobserved outcomes. In doing so we distinguish issues that stem purely from the non-constant nature of the cluster size, from those that result from the association between cluster size and outcome. We focus on the differences between the case of a fixed cluster size that is common to all clusters, and that of a fluctuating cluster size, whether for design reasons or randomly. In particular, the joint modelling of outcomes and cluster size is not considered.

As a simple, yet non-trivial, clustering paradigm, we consider the normal compound-symmetry (CS) model, which is a three-parameter multivariate normal model, with a common mean μ , a common variance $\sigma^2 + d$, and a common covariance d . ? studied this case in the context of so-called split-sample methodology: they proposed a particular form of pseudo-likelihood where a sample is subdivided into M subsamples, which are separately analyzed as if they were unrelated, after which the results are averaged using appropriate weights, leading to proper point and precision estimates. Pseudo-likelihood has received considerable attention (? , Ch. 9, 12, 21, 24, 25); (? , Ch. 6, 7).

Assume that there are c_k clusters of size n_k , $k = 1, \dots, K$. For ease of development, we allow for some of the n_k to be equal, which is useful when a subgroup of clusters that is of the same size is chosen to be sub-divided (because there are very many or for other reasons). A natural split is made with respect to the cluster size, i.e., as if every cluster size defines its own stratum.

Evidently, for medium to large sample sizes, full maximum likelihood or Bayesian inferences are statistically optimal and computationally feasible; hence, the work done here might be less relevant. However, with really big data, where the number of independent clusters runs in the millions or beyond, and/or in settings where the number of measurements per cluster becomes very large (e.g., in meta-analysis), maximum likelihood eventually becomes prohibitive in terms of computation time. At the other end of the spectrum, in very small samples (e.g., in small-area epidemiology applications, or when studies are conducted in so-called orphan diseases), maximum likelihood estimates may become unstable, to the point where it is difficult to obtain convergence. This may be due, for example, to relatively flat likelihood functions. The non-iterative nature of our proposal removes such issues. Small samples refers here to a small number of clusters; the clusters themselves may consist of smaller or larger numbers of within-cluster replication. We are not the first to consider these issues. ? considered multiple imputation to bring clusters to the same size before applying maximum likelihood. If done with care, convergence problems are drastically reduced. ? and ? proposed so-called multiple outputation, to repeatedly create

Page 60

independent samples by randomly selecting one member per cluster. To ensure that correlation is taken into account, combination rules reminiscent of multiple imputation are then applied to combine inferences from the samples drawn. These methods are based on repeated sampling and come at computational cost for high-dimensional data (?). Therefore, in this paper, the focus is on entirely non-iterative methods, bringing together the advantages of balanced data and simple averaging methodology. A consequence of our approach is the need for applying weights when combining results from the K strata. We establish how results on incomplete sufficient statistics in the context of weighted averages (??) imply that there may be no optimal set of weights. Given this, we propose pragmatically attractive weights, in terms of efficiency, bias, and computational ease.

The remainder of the paper is organized as follows. Two motivating datasets are described in Section 2. In Section 3 essential background material on incomplete sufficient statistics is presented. The compound-symmetry model is introduced in Section 4, and a review is provided of the relevant incompleteness results from ?, together with implications for likelihood-based estimation. Background from the pseudo-likelihood-based split-sample method is given in Supplementary Material Section S.4. A general split-sample approach to the CS model is provided in Section 5, and a number of specific but practically relevant cases are considered. Details about the specifics for the CS case are presented in Section 6. Section 7 is dedicated to a simulation study, examining situations for which

Table 4.1: Developmental Toxicity Study (DEHP). Summary data by dose group.

dose	# implants	# dams with viable implants	# live fetuses	Titter size	average weight
0 mg/kg/day	30	30	330	13.2	0.9483
44 mg/kg/day	26	26	288	11.1	0.9592
91 mg/kg/day	26	26	277	10.7	0.8977
191 mg/kg/day	24	17	137	8.1	0.8509
292 mg/kg/day	25	9	50	5.6	0.6906

there are no closed forms on the one hand, and studying numerical performance (speed and convergence) on the other. The data, described in Section 2, are analyzed in Section 8. Ramifications and recommendations for practice are offered in Section 9.

Developmental Toxicity Study Sets

Data from the Research Triangle Institute under contract to the National Toxicology Program of the U.S.A. (NTP data), are analyzed. These developmental toxicity studies investigate the effects in mice of three chemicals: di(2-ethylhexyl)phthalate (DEHP) (?) ethylene glycol (EG) ?, and diethylene glycol dimethyl ether (DYME) ?. The studies were conducted in timed-pregnant mice during the period of major organogenesis. The dams were sacrificed, just prior to normal delivery, and the status of uterine implantation sites recorded. The outcome of interest here is fetal weight. Summary data from the DEHP trial are presented in Table 4.1. The design for EG and DYME is similar. It is clear from the table that average litter size is depleted with increasing dose, as is the average weight.

Incomplete Sufficient Statistics

A statistic $k(Y)$ of a random variable Y , with Y belonging to a family P_θ , is complete if, for every measurable function $g(\cdot)$, independent of θ , $E[g\{k(Y)\}] = 0$ for all θ , implies that $P_\theta[g\{k(Y)\} = 0] = 1$ for all θ (?). The Lehman-Scheffé theorem (?) states that, if a statistic is unbiased, complete, and sufficient for a parameter θ , then it leads to the best mean-unbiased estimator for θ , while Basu's theorem (?) has it that statistic that is both boundedly complete and sufficient is independent of any ancillary statistic. As has been shown in the sequential trial context, a lack of completeness does not preclude the existence of estimators with very good properties (?).

? established the incompleteness of the sufficient statistic for a clinical trial with a stopping rule, for the case of normally distributed endpoints. ? generalized this result to the entire exponential family. ? and ? broadened it further to a stochastic rather than a deterministic stopping rule, hence encompassing the case of a completely random sample size. Indeed, it would seem at first sight that this latter case is standard, because the sample size is unrelated to the data, whether observed or not. Yet, even in this case, completeness no longer holds. What is more, incompleteness holds when the cluster size is non-constant for whatever reason.

The Compound-symmetry Model

Let \mathbf{Y} be a vector of length n , with $\mathbf{Y} \sim N(\mu \mathbf{1}_n, \sigma^2 I_n + d J_n)$. In general, both \mathbf{Y} and n are random variables.

Suppose that there is a sample of N independent clusters, among which K different cluster sizes n_k ($k = 1, \dots, K$) are distinguished. Let the multiplicity of cluster size n_k be equal to c_k . Evidently, $N = \sum_{k=1}^K c_k$. Denote the outcome vector for the i th ($i = 1, \dots, c_k$) replicate among the clusters of size n_k by $\mathbf{Y}_i^{(k)}$. We first show incompleteness of the sufficient statistic, then turn to likelihood estimation. For both, we start from the log-likelihood function.

Incompleteness

The data-dependent terms in the log-likelihood can be written as:

$$\sum_{k=1}^K \sum_{i=1}^{c_k} -\frac{1}{2} (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k})' (\sigma^2 I_{n_k} + d J_{n_k})^{-1} (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k}) \quad (4.1)$$

$$= \sum_{k=1}^K \sum_{i=1}^{c_k} -\frac{1}{2} (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k})' \left(I_{n_k} - \frac{d}{\sigma^2 + n_k d} J_{n_k} \right) (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k}) \quad (4.3)$$

$$= \sum_{k=1}^K \sum_{i=1}^{c_k} \frac{\mu}{\sigma^2 + n_k d} \left(\sum_{j=1}^{n_k} Y_{ij}^{(k)} \right) - \frac{1}{2\sigma^2} \left(\sum_{k=1}^K \sum_{i=1}^{c_k} \sum_{j=1}^{n_k} Y_{ij}^{(k)2} \right) \quad (4.3)$$

$$+ \sum_{k=1}^K \sum_{i=1}^{c_k} \frac{d}{2\sigma^2(\sigma^2 + n_k d)} \left(\sum_{j=1}^{n_k} Y_{ij}^{(k)} \right)^2. \quad (4.4)$$

The three terms in (4.4) are qualitatively different. Indeed, the middle one corresponds to a single sufficient statistic, the sum of all squares across clusters, while the first and last split into as many sufficient statistics as there are unique cluster sizes.

? proved a characterization of incompleteness, essentially stating that when the dimension of the sufficient statistic is larger than the dimension of the parameter vector, the sufficient statistic is

no longer complete. More details can be found in Supplementary Materials Section S1. This sharp division also occurs when studying certain properties of the maximum likelihood estimator.

Likelihood-based Estimation of the CS Model

Similar in spirit to (4.4), but now using all terms, the log-likelihood can be written as

$$\ell(\mu, \sigma^2, d) = \sum_{k=1}^K \ell_k(\mu, \sigma^2, d), \quad (4.5)$$

with the cluster size specific log-likelihood term

$$\ell_k(\mu, \sigma^2, d) = -\frac{1}{2} \sum_{i=1}^{c_k} \left\{ \ln \left[\sigma^{2n_k} + n_k \sigma^{2(n_k-1)} d \right] \right. \quad (4.6)$$

$$+ \left. (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k})' \frac{1}{\sigma^2} \left(I_{n_k} - \frac{d}{\sigma^2 + n_k d} J_{n_k} \right) (\mathbf{Y}_i^{(k)} - \mu \mathbf{1}_{n_k}) \right\}. \quad (4.7)$$

Using derivations similar to those in ?, the cluster size specific log-likelihood can be maximized analytically *assuming that there is a separate parameter per cluster size*. By replacing $\ell_k(\mu, \sigma^2, d)$ by $\ell_k(\mu_k, \sigma_k^2, d_k)$, we can consider the kernel of the log-likelihood, in general for K cluster sizes, and allowing for the parameter vector to change with cluster size:

$$\begin{aligned} \ell \left(\{\mu_k\}_k, \{\sigma_k^2\}_k, \{d_k\}_k \right) &\propto -\frac{1}{2} \sum_{k=1}^K \sum_{i=1}^{c_k} \{ \ln |\Sigma_{n_k}| \quad (4.8) \\ &+ \left. (\mathbf{y}_i^{(k)} - \boldsymbol{\mu}_{n_k})' \Sigma_{n_k}^{-1} (\mathbf{y}_i^{(k)} - \boldsymbol{\mu}_{n_k}) \right\} \end{aligned}$$

where $\boldsymbol{\mu}_k = \mu_k \mathbf{1}_{n_k}$, $\Sigma_{n_k} = \sigma_k^2 I_{n_k} + d_k J_{n_k}$. The score functions are presented in Supplementary Materials Section S.2. Solving these score functions (S2.1)-(S2.3) leads to:

$$\hat{\mu}_k = \frac{1}{c_k n_k} \sum_{i=1}^{c_k} \sum_{j=1}^{n_k} Y_{ij}^{(k)}, \quad (4.10)$$

$$\hat{\sigma}_k^2 = \frac{1}{c_k n_k (n_k - 1)} \left(n_k \sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} \mathbf{Z}_i^{(k)} - \sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} J_{n_k} \mathbf{Z}_i^{(k)} \right), \quad (4.11)$$

$$\hat{d}_k = \frac{1}{c_k n_k (n_k - 1)} \left(\sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} J_{n_k} \mathbf{Z}_i^{(k)} - \sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} \mathbf{Z}_i^{(k)} \right), \quad (4.12)$$

where $\mathbf{Z}_i^{(k)} = (\mathbf{Y}_i^{(k)} - \mu_k \mathbf{1}_{n_k})$.

When the cluster size is constant, the compound-symmetry model has closed form ML estimators, given by (4.10)–(4.12). Closed-form estimators for the variance-covariance matrix of the estimator exist as well (?). For the mean, the variance is:

$$\text{var}(\hat{\mu}_k) = \frac{\sigma_k^2 + n_k d_k}{c_k n_k}. \quad (4.13)$$

The expressions for the variance-covariance structure of $(\hat{\sigma}_k^2, \hat{d}_k)$ is:

$$\text{var} \begin{pmatrix} \hat{\sigma}_k^2 \\ \hat{d}_k \end{pmatrix} = \frac{2\sigma_k^4}{c_k n_k (n_k - 1)} \begin{pmatrix} n_k & -1 \\ -1 & \frac{\sigma_k^4 + 2(n_k - 1)d_k\sigma_k^2 + n_k(n_k - 1)d_k^2}{\sigma_k^4} \end{pmatrix}. \quad (4.14)$$

The mean parameter is independent of the variance components.

These results can be used when a separate parameter vector is estimated for each of the cluster sizes and, as a special case, when

there is only one cluster size. Four features of use in what follows are: (a) there are closed forms; (b) the sufficient statistic is complete; (c) the estimator is unique minimum variance unbiased; (d) the mean parameter estimator and the variance parameter estimator are independent.

These results are lost when $K \geq 2$. We briefly sketch the lack of closed-form solutions in this case in Supplementary Materials Section S2.2.

The lack of a closed form is well known, but we highlight a few relevant features here. More detail is given in Supplementary Materials Section S3, where we show

$$\widehat{\mu} = \frac{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d} \widehat{\mu}_k}{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d}}. \quad (4.15)$$

Examining (4.15) suggests weighted averages:

$$\widetilde{\mu} = \sum_{k=1}^K a_k \widehat{\mu}_k, \quad \widetilde{\sigma^2} = \sum_{k=1}^K b_k \widehat{\sigma}_k^2, \quad \widetilde{d} = \sum_{k=1}^K g_k \widehat{d}_k. \quad (4.16)$$

This idea is similar to that in ?, who split a sample in sub-samples, analyzed each separately, and then combined the result in an overall estimator. They considered splits in both dependent and independent sub-samples. Dependent samples occur when very long sequences of repeated measures are collected, which are then subdivided for convenience. This approach is not of use here. Independent samples arise when there are many independent replicates, i.e.,

a large number of clusters. They studied the CS case, but only for a single cluster size. The total number of clusters was then split into M parts comprising an equal number of clusters. We modify these ideas to the case of unequal cluster sizes, with a variable number of clusters per split.

Split-sample Methods for Clusters of Variable Size

The derivations are based on general pseudo-likelihood principles, reviewed in Supplementary Materials Section S4. We first make generic the setting at the beginning of Section 4. Let there be a sample of N independent clusters. Partition the sample into K sub-samples, with c_k independent and identically distributed clusters in sub-sample k ; $N = \sum_{k=1}^K c_k$. $\mathbf{Y}_i^{(k)}$ remains the outcome vector for replicate i in sub-sample k .

Subjects in different sub-samples are allowed to have the same distribution, but subjects in the same sub-sample *must* have the same distribution. This covers the running example of CS clusters, partitioned according to cluster size. However, it is possible to further sub-divide such a sub-sample in various sub-samples, all with the same cluster size. This is sensible, for example, in very large databases. An extreme example follows when sub-samples consist of a single independent replicate, useful, for example, in a meta-analysis with large individual studies. This limiting situation can also be considered with CS data, because all clusters (except those of size 1) contribute to all three parameters.

Consider pseudo-likelihood in this general case (see also Eq. (S4.4)). Assume that $\boldsymbol{\theta}^*$ is a vector of length p , and that each $\hat{\boldsymbol{\theta}}_k$ is a separate copy of $\boldsymbol{\theta}^*$. Then it can be shown that the generic combination rules are

$$\tilde{\boldsymbol{\theta}}^* = \sum_{k=1}^K A_k \hat{\boldsymbol{\theta}}_k, \quad (4.17)$$

$$\text{var}(\tilde{\boldsymbol{\theta}}^*) = \sum_{k=1}^K A_k V_k A'_k, \quad (4.18)$$

with $V_k = I_0(\boldsymbol{\theta}_k)^{-1}$. We use the symbol $\tilde{\boldsymbol{\theta}}^*$ to emphasize that this is not necessarily the maximum likelihood estimator even though, in our formalism, $\hat{\boldsymbol{\theta}}_k$ is the maximum likelihood estimator when restricting attention to sub-sample k . Equation (4.18) is appropriate only when the weights A_k are free of the parameters to be estimated. We return to this at the end of the section.

Weighting Schemes Not every choice of the A_k leads to an unbiased estimator. To enforce unbiasedness, consider the requirement

$$\boldsymbol{\theta} = E(\hat{\boldsymbol{\theta}}^*) = \sum_{k=1}^K A_k E(\hat{\boldsymbol{\theta}}_k) = \left(\sum_{k=1}^K A_k \right) \boldsymbol{\theta},$$

whence $I_p = \sum_{k=1}^K A_k$. This requirement is satisfied for (S4.5). This suggests two obvious choices:

Constant weights. Set $A_k = (1/K)I_p$.

Proportional weights. Set $A_k = (c_k/N)I_p$.

Constant weights are the clear choice when all subjects are i.i.d. and partitioning is in sub-samples of equal size. Proportional weights are called for in the i.i.d. case, with sub-samples of varying size.

Consider optimal weights through the objective function

$$Q = \sum_{k=1}^K A_k V_k A'_k - \Lambda \left(\sum_{k=1}^K A_k - I_p \right),$$

where Λ is a matrix of Lagrange multipliers. Taking the first derivative of Q w.r.t. A_k leads to $A_k = \Lambda V_k^{-1}/2$. Because the A_k sum to the identity, $\Lambda = 2 \left(\sum_{m=1}^K V_m^{-1} \right)^{-1}$ and we have the following.

Optimal weights. These take the form:

$$A_k^{\text{opt}} = \left(\sum_{m=1}^K V_m^{-1} \right)^{-1} V_k^{-1}. \quad (4.19)$$

With this choice, (4.17)–(4.18) become:

$$\tilde{\theta}^* = \hat{\theta}^* = \left(\sum_{k=1}^K V_k^{-1} \right)^{-1} \sum_{k=1}^K V_k^{-1} \hat{\theta}_k, \quad (4.20)$$

$$\text{var}(\tilde{\theta}^*) = V = \left(\sum_{k=1}^K V_k^{-1} \right)^{-1}. \quad (4.21)$$

The optimal weights lead to the maximum likelihood estimator. To apply the optimal weights in practice is typically not straightforward. A closed form expression for the V_k does not always exist, and even if it did, as in the CS case, it may depend on the unknown parameters.

The optimal weights can suggest sensible choices and we describe a couple of these. They will be illustrated in the next section for the CS case.

Scalar weights. While optimal weights may be unwieldy, one could consider scalar weights by requiring the A_k to be diagonal. This implies that each component of $\boldsymbol{\theta}^*$, θ_r^* , say, is a linear combination

$$\tilde{\theta}_r^* = \sum_{k=1}^K a_{k,r} \hat{\theta}_{k,r},$$

where then, formally, $A_k = \text{diag}(a_{k,1}, \dots, a_{k,p})$. The optimization route, followed for unrestricted A_k , can then be followed component-wise as well. Because the class of A_k over which to optimize is restricted, the resulting optimum does not necessarily correspond to the maximum likelihood solution. The rationale for choosing this route is computational convenience, and its advantages vary from problem to problem.

Iterated optimal weights. An iterative scheme can be followed:

1. Estimate $\hat{\boldsymbol{\theta}}_k$.
2. Compute an initial estimator for $\boldsymbol{\theta}^*$, $\boldsymbol{\theta}^{*(0)}$, say, using a simple weighting method, e.g., using constant or proportional weights.
3. Using the current parameter estimate, $\boldsymbol{\theta}^{*(t)}$ say, calculate $V_k^{(t+1)}$.

4. Determine:

$$\boldsymbol{\theta}^{*(t+1)} = \left(\sum_{k=1}^K [V_k^{(t+1)}]^{-1} \right)^{-1} \sum_{k=1}^K [V_k^{(t+1)}]^{-1} \widehat{\boldsymbol{\theta}}_k.$$

5. Repeat steps 2–3 until convergence.

This scheme can always be followed and it has the advantage that the data need only be analyzed once, to yield $\widehat{\boldsymbol{\theta}}_k$. From this point on, calculations involve algebraic expressions for the parameters only.

Approximate optimal weighting. Related to the previous method, a non-iterative approximation consists of replacing V_k by $V_k(\widetilde{\boldsymbol{\theta}}_k)$ in (4.20). Here, $\widetilde{\boldsymbol{\theta}}_k$ could be, for example, the sub-sample specific estimator $\widehat{\boldsymbol{\theta}}_k$, or the $\widetilde{\boldsymbol{\theta}}^*$ obtained using a simple scheme, such as constant or proportional weighting. This method avoids all further iteration, once the $\boldsymbol{\theta}_k$ have been determined.

Approximate optimal weighting is a method that could be considered when the use of (4.18) might lead to underestimation of the variability, because the A_k now depend on the parameters estimated from stratum k . To properly account for this extra source of uncertainty. Consider that

$$\frac{\partial}{\partial \boldsymbol{\theta}_k} (A_k \boldsymbol{\theta}_k) = A_k + \left(\frac{\partial A_k}{\partial \theta_{k1}} \boldsymbol{\theta}_k \Big| \dots \Big| \frac{\partial A_k}{\partial \theta_{kp}} \boldsymbol{\theta}_k \right), \quad (4.22)$$

where θ_{kj} , $j = 1, \dots, p$ ranges over the components of $\boldsymbol{\theta}_k$. Writing Page 78

$W_k = V_k^{-1}$ for ease of notation,

$$\frac{\partial A_k}{\partial \theta_{kj}} = W^{-1} \frac{\partial W_k}{\partial \theta_{kj}} (I_p - W^{-1} W_k), \quad (4.23)$$

with I_p the p -dimensional identity matrix. Plugging (4.23) into (4.22), the proper delta-method approximation to the variance is

$$\text{var}(\tilde{\boldsymbol{\theta}}^*) \simeq \sum_{k=1}^K (A_k + B_k) V_k (A_k + B_k)', \quad (4.24)$$

with

$$B_k = (\mathbf{1}'_p \otimes I_p)(I_p \otimes W^{-1}) \text{diag} \left(\frac{\partial W_k}{\partial \theta_{k1}}, \dots, \frac{\partial W_k}{\partial \theta_{kp}} \right) [I_p \otimes (I_p - W^{-1} W_k) \boldsymbol{\theta}],$$

and \otimes signifying Kronecker product.

Partitioned-sample Analysis for the Compound-symmetry Model

For the normal compound-symmetry model, a variety of options exists. We sketch them here, and then consider some in greater detail.

Consider the i.i.d. case, where all clusters are of the same size. Full maximum likelihood then leads to a closed-form solution. ? studied splitting the sample in dependent sub-samples for this case, and showed that splitting leads to efficiency loss for the variance components, but not for the mean. They split the sequences of repeated measures in portions of equal size. Unequally sized splits

could also be considered, although the rationale for this may not be compelling. They did not consider splits in independent sub-samples. We do so here, in Section 6.2, both for sub-samples of equal as well as for unequal size.

With variable cluster size, we know from Section 4.2 that full maximum likelihood does not lead to a closed-form solution. We will study in more detail the natural splitting into sub-samples of constant cluster size.

A special case, for both the i.i.d. and unequal cluster-size settings, is the cluster-by-cluster analysis. We will apply our methodology, outlined in Section 5, to this case, and contrast it with an *ad hoc* moment-based set of estimators.

Variable Cluster Size

Optimal Weights

As we see in Section 6.1.3, scalar and optimal (hence, vectorized) weights do not make a difference for the mean parameter, because of the independence between the mean and the covariance parameters.

We can therefore consider the mean parameter separately from the covariance parameters. Let v_k be the variance of the mean in stratum k , and V_k the corresponding variance-covariance matrix for the variance components. Applying optimal weight (4.19) to the

mean produces

$$\tilde{\mu} = \left(\sum_{k=1}^K \frac{c_k n_k}{\widehat{\sigma}_k^2 + n_k \widehat{d}_k} \right)^{-1} \sum_{k=1}^K \frac{c_k n_k \widehat{\mu}_k}{\widehat{\sigma}_k^2 + n_k \widehat{d}_k}. \quad (4.25)$$

The corresponding estimators for the variance components, specific to a cluster size, are given by (4.11) and (4.12). Using them, and expression (4.14) for the variance, it follows that the optimal weighted estimator satisfies

$$\begin{pmatrix} \widehat{\sigma}^2 \\ \widehat{d} \end{pmatrix} = \left(\sum_{k=1}^K V_k^{-1} \right)^{-1} \sum_{k=1}^K \begin{pmatrix} \frac{Q_k}{2\sigma_k^2} - \frac{d_k(2\widehat{\sigma}_k^2 + n_k \widehat{d}_k)}{2\widehat{\sigma}_k^4(\widehat{\sigma}_k^2 + n_k \widehat{d}_k)^2} R_k \\ \frac{R_k}{2(\sigma_k^2 + n_k d_k)^2} \end{pmatrix}, \quad (4.26)$$

with Q_k and R_k as in (S.9) and (S.10), respectively.

Iterated and Approximate Optimal Weights

Evidently, the principles of iterated and approximate optimal weights can be applied here.

Replacing the variance components in (4.25) by their expectation leads to:

$$\tilde{\mu} = \left(\sum_{k=1}^K \frac{c_k n_k}{\sigma^2 + n_k d} \right)^{-1} \sum_{k=1}^K \frac{c_k n_k \widehat{\mu}_k}{\sigma^2 + n_k d}. \quad (4.27)$$

If we do the same for the mean, on both sides of the equality, we obtain

$$\mu = \left(\sum_{k=1}^K \frac{c_k n_k}{\sigma^2 + n_k d} \right)^{-1} \sum_{k=1}^K \frac{c_k n_k \mu}{\sigma^2 + n_k d}. \quad (4.28)$$

Although (4.25) cannot directly be used, because of circularity,
Page 75

(4.27) and (4.28) are available to us.

Replacing the variance components on the right hand side of (4.26) by their expectations leads to

$$\begin{pmatrix} \tilde{\sigma}^2 \\ \tilde{d} \end{pmatrix} = \left(\sum_{k=1}^K V_k^{-1} \right)^{-1} \sum_{k=1}^K \begin{pmatrix} \frac{Q_k}{2\sigma^2} - \frac{d(2\sigma^2 + n_k d)}{2\sigma^4(\sigma^2 + n_k d)^2} R_k \\ \frac{R_k}{2(\sigma^2 + n_k d)^2} \end{pmatrix}. \quad (4.29)$$

Using their explicit expressions, and using the fact that the expectation must be $(\sigma^2, d)'$, (4.26) leads to the following identity:

$$\begin{pmatrix} \sigma^2 \\ d \end{pmatrix} = V \sum_{k=1}^K \frac{c_k n_k}{2(\sigma^2 + n_k d)} \begin{pmatrix} \frac{\sigma^2 + (n_k - 1)d}{\sigma^2} \\ 1 \end{pmatrix}. \quad (4.30)$$

Expressions (4.25) and (4.26) can be used for approximate weighting, by plugging in, as is done, on the right hand side, the cluster-size specific mean and variance components.

Expressions (4.27) and (4.29) can be used for iterated weighting. The estimator for the mean depends on the variance components, but not vice versa. This dependence is insightful: there is independence between mean and variance components for every cluster-size specific stratum separately. As a consequence, $\tilde{\mu}$ on the one hand, and $\tilde{\sigma}^2$ and \tilde{d} on the other, can be determined separately, provided the latter are done first.

Expressions (4.28) and (4.30) move beyond the previous schemes and exist by virtue of their explicit expressions. In (4.30) an initial consistent estimator for the variance components can be used on

the right hand side. Once the left hand side has been determined, the result can be plugged in again on the right, until convergence. Once done, the final variance component estimates can be used in (4.28) and the process repeated for μ , until convergence.

Scalar Weights

In this case, A_k equals $\text{diag}(a_k, b_k, g_k)$, with the scalars as in (4.16). Obviously, the conditions for unbiased estimators are $\sum_{k=1}^K a_k = \sum_{k=1}^K b_k = \sum_{k=1}^K g_k = 1$.

The stratum-specific estimators are given by (4.10)–(4.12) and their variance-covariance structure by (4.13)–(4.14). The objective function to find the optimum is

$$Q = \sum_{k=1}^K a_k^2 \text{var}(\widehat{\mu}_k) - \lambda \left(\sum_{k=1}^K a_k \right).$$

Logic, similar to the vector case, and using the explicit expressions for the variances, leads to:

$$a_k = \frac{\frac{c_k n_k}{\sigma^2 + n_k d}}{\sum_{m=1}^K \frac{c_m n_m}{\sigma^2 + n_m d}} = \frac{\frac{c_k n_k}{(1-\rho) + n_k \rho}}{\sum_{m=1}^K \frac{c_m n_m}{(1-\rho) + n_m \rho}}, \quad (4.31)$$

$$b_k = \frac{c_k(n_k - 1)}{\sum_{m=1}^K c_m(n_m - 1)}, \quad (4.32)$$

$$g_k = \frac{\frac{c_k n_k}{\frac{\sigma^4}{n_k - 1} + 2\sigma^2 d + n_k d^2}}{\sum_{m=1}^K \frac{c_m n_m}{\frac{\sigma^4}{n_m - 1} + 2\sigma^2 d + n_m d^2}} = \frac{\frac{c_k n_k (n_k - 1)}{(1-\rho)^2 + [2\rho(1-\rho) + n_k \rho^2](n_k - 1)}}{\sum_{m=1}^K \frac{c_m n_m (n_m - 1)}{(1-\rho)^2 + [2\rho(1-\rho) + n_m \rho^2](n_m - 1)}} \quad (4.33)$$

where $\rho = d/(\sigma^2 + d)$. Here, the coefficients depend on the parameter ρ .

ters in different ways. While b_k is independent of the parameters, a_k has denominators linear in σ^2 and d (equivalently, in ρ), and g_k has quadratic functions instead.

These weights, like the optimal ones, depend on the parameters. Evidently, they can be made part of an iterative scheme, as with the vector-valued weights. The added advantages are that matrix computations simplify to scalar computations; for models with relatively few parameters, like the one here, this is a small advantage. More importantly, approximations can be considered for each parameter separately.

Direct calculations show that the variance for the weighted estimator of the mean, using weights (4.31), is equal to that of maximum likelihood. For this parameter, the weighted split-sample estimator is the maximum likelihood estimator, in spite of the use of the scalar weight. This is to be expected, because V_k is block-diagonal and because of independence of the mean estimator from the variance components estimators within a given cluster size. This implies that the optimally weighted estimator and the scalar estimator coincide for the mean. They differ for the variance components.

Approximate Optimal Scalar Weights

To illustrate the logic of this method, consider (4.31)–(4.33) for the case where cluster sizes, for a good majority of the clusters, are

sufficiently large. Taking limits for $n_k \rightarrow +\infty$ produces

$$a_k^{\text{app}} = g_k^{\text{app}} = c_k/N. \quad (4.34)$$

When this approximation is sensible, the very simple proportional weights follow. These approximations are exact, for a_k and g_k , when $\rho = 1$. They deteriorate when ρ becomes smaller. For example, in case $\rho = 0$,

$$a_k(\rho = 0) = \frac{c_k n_k}{\sum_{m=1}^K c_m n_m},$$

$$g_k(\rho = 0) = \frac{c_k n_k (c_k n_k - 1)}{\sum_{m=1}^K c_m n_m (c_m n_m - 1)} \approx \frac{c_k^2 n_k^2}{\sum_{m=1}^K c_m^2 n_m^2}.$$

A reasonable approximation for b_k is

$$b_k^{\text{app}} = \frac{c_k n_k}{\sum_{m=1}^K c_m n_m}, \quad (4.35)$$

which sets it equal to $a_k(\rho = 0)$. The information for σ^2 is thus determined more in terms of the number of measurements than in the number of clusters. Dropping the n_k from this formula is sensible only when cluster sizes are not too different from one another.

Figure 4.1 depicts optimal scalar weights (4.31)–(4.33), alongside the apparently simplistic proportional weights, for two of the five NTP datasets chosen to represent two relatively different empirical cluster size distributions. In both cases, there is a considerable range of cluster sizes, approximately 1 to 20. At the same time, the frequencies of the cluster sizes vary considerably. The values for a_k

and g_k are almost identical to the proportional weights. While a small discrepancy for b_k is noticeable, and understandable in view of (4.35), the proportional weights seem to offer a sensible choice. This issue will be examined further in the data-analytic Section 8.

The Special Case of Common Cluster Size, Splits of (Un)equal Size

When $n_k \equiv n$ is constant, (4.31)–(4.33) reduce to:

$$a_k = b_k = g_k = c_k/N, \quad (4.36)$$

Hence, while $a_k = b_k$ reduce to proportional weights, for g_k there is an impact of the partitioning structure. When, further, c_k is constant, we obtain $a_k = b_k = g_k = c/N = 1/K$, and equal weights follow. The similarities and the subtle differences with the results from Section 6.1.4 are worth pointing out. Expressions (4.34) and (4.36) are identical except for the parameter σ^2 .

Cluster-by-cluster Analysis

The expressions presented earlier in this section, using optimal weights and variations on this theme, can be applied when the partitioning is as extreme as possible: a single cluster per stratum. This sets $c_k \equiv 1$. Evidently, the n_k will then no longer be unique, but that is immaterial; while we make use of the fact that the cluster size is constant within a stratum, it does not need to be different between strata. We examine this case in more detail in Supplementary Materials Section S5.1. In particular, we derive

under what asymptotics such an estimator is consistent.

Simulation Study

A first, limited, simulation study was carried out to examine the behavior of the partitioning method. Details are given in Supplementary Materials Section S6. Three settings were considered: (1) $c_k \cdot n_k$ is kept constant with the factors taking different values; (2) c_k is kept constant; (3) n_k is kept constant. For all of these, k goes from 1 to 4, so that there are four sub-samples. Apart from full likelihood, a series of weights was considered: equal, proportional to c_k , size proportional to $c_k \cdot n_k$, approximate optimal, and iterated optimal.

From the results it is clear that equal weights are not a good choice. For μ and d , proportional weights are excellent, while for σ^2 so are the size proportional weights. Iterated optimal weights perform considerably better than approximate optimal weights, in the sense that the latter, like equal weights, arguably should not be considered for practice. When comparing iterated and approximate optimal weights, the former are more computationally intensive.

However, iterated optimal weights give results very close to proportional weights (for μ and d) and to size proportional weights (for σ^2). Importantly, all of these results are very close to the ones obtained from maximum likelihood.

As a consequence, we have a simple, non-iterative set of weights at our disposal, free of unknown parameters, with excellent performance.

mance.

A second simulation study compares the proposed methods to two alternatives: full maximum likelihood and multiple imputation. Details are reported in Supplementary Materials Section S7. The most striking conclusion is that closed-form solutions are much faster than their alternatives while, at the same time, yielding most precise results. The time gain of our fastest method relative to standard maximum likelihood using PROC MIXED ranges from 5 times to 30,000 times faster.

0

Analysis of Case Study

The data, introduced in Section 2, were analyzed in three ways. In Section 8.1 maximum likelihood estimators are presented, with split sampling, where splitting is by cluster size and using various weighting schemes. In Section 8.2, a dose effect is added to these. Finally, the cluster-by-cluster methodology of Section 6.3 is illustrated in Section 8.3.

Splitting by Cluster Size, No Dose Effect

Tables 4.2–4.4 present (restricted) maximum likelihood estimates (standard errors), together with those from various weighted estimators. The standard CS model is fitted to the fetal weight outcome, ignoring the dose effect. Because there is an effect of dose on litter size, the mean is associated with cluster size. It is therefore inter-

Table 4.2: NTP Data (DEHP). Cluster-by-cluster analysis. Maximum likelihood and weighted split-sample estimates (standard errors): (a) ML: maximum likelihood; (b) REML: restricted maximum likelihood; (c) Prop.: proportional weights; (d) Equal: equal weights; (e) Approx. sc.: like proportional weights, except that for b_k (4.35) is used; (f) Scalar: scalar weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights; (g) Opt.: optimal weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights. Proper: proper variances for optimal weights

Par.	ML	REML	Prop.	Equal	Approx.	sc.	Scalar	Optimal	
								Weighted	[(S5.2)(S5.3)(S5.4)]
μ	0.90718	0.90716	0.90602	0.89558	0.90602	0.92080	0.92080		
σ^2	0.01877	0.01877	0.02122	0.02244	0.01895	0.01871	0.01246		
d	0.01181	0.01195	0.00951	0.01016	0.00951	0.00085	0.00087		
s.e.($\hat{\mu}$)	0.01149	0.01155	0.01076	0.01360	0.01076	0.00766	0.00766	0.00766	
s.e.($\hat{\sigma}^2$)	0.00084	0.00084	0.00128	0.00199	0.00094	0.00092	0.00061	0.00138	
s.e.(\hat{d})	0.00196	0.00199	0.00210	0.00293	0.00210	0.00048	0.00045	0.00340	
Two-stage [(S5.2)(S5.7)(S5.8)]									
μ	0.90718	0.90716	0.90602	0.89558	0.90602	0.92119	0.92119		
σ^2	0.01877	0.01877	0.01868	0.01931	0.01696	0.01679	0.01155		
d	0.01181	0.01195	0.01204	0.01329	0.01204	0.00362	0.00376		
s.e.($\hat{\mu}$)	0.01149	0.01155	0.01169	0.01496	0.01169	0.00901	0.00901	0.00901	
s.e.($\hat{\sigma}^2$)	0.00084	0.00084	0.00092	0.00127	0.00074	0.00072	0.00057	0.02404	
s.e.(\hat{d})	0.00196	0.00199	0.03045	0.02915	0.03045	0.02537	0.00087	0.27337	
Unbiased two-stage [(S5.2)(S5.11)(S5.12)]									
μ	0.90718	0.90716	0.90602	0.89558	0.90602	0.92195	0.92195		
σ^2	0.01877	0.01877	0.02122	0.02244	0.01895	0.01871	0.01244		
d	0.01181	0.01195	0.01390	0.01609	0.01390	0.00448	0.00467		
s.e.($\hat{\mu}$)	0.01149	0.01155	0.01257	0.01679	0.01257	0.00958	0.00958	0.00958	
s.e.($\hat{\sigma}^2$)	0.00084	0.00084	0.00128	0.00199	0.00094	0.00092	0.00061	0.00172	
s.e.(\hat{d})	0.00196	0.00199	0.00291	0.00447	0.00291	0.00101	0.00102	0.00634	

esting to assess the impact of this on the split-sample estimators, when compared to the MLEs.

The ML and REML are very similar, with equal point estimates for μ , nearly equal estimates for σ^2 , and similar estimates for d . The equality for the mean estimator is known for the CS case. The Page 83

Table 4.3: NTP Data (EG). Cluster-by-cluster analysis. Maximum likelihood and weighted split-sample estimates (standard errors): (a) ML: maximum likelihood; (b) REML: restricted maximum likelihood; (c) Prop.: proportional weights; (d) Equal: equal weights; (e) Approx. sc.: like proportional weights, except that for b_k (4.35) is used; (f) Scalar: scalar weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights; (g) Opt.: optimal weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights. Proper: proper variances for optimal weights

Par.	ML	REML	Prop.	Equal	Approx. sc.	Scalar	Simpl.	Optimal Proper
Weighted [(S5.2)(S5.3)(S5.4)]								
μ	0.82952	0.82952	0.83342	0.84653	0.83342	0.84133	0.84133	
σ^2	0.00886	0.00886	0.00885	0.00899	0.00879	0.00878	0.00608	
d	0.01704	0.01724	0.01606	0.01536	0.01606	0.01381	0.01408	
s.e.($\hat{\mu}$)	0.01402	0.01410	0.01393	0.01485	0.01393	0.01346	0.01346	0.01346
s.e.($\hat{\sigma}^2$)	0.00041	0.00041	0.00046	0.00051	0.00044	0.00044	0.00031	0.00328
s.e.(\hat{d})	0.00265	0.00269	0.00264	0.00272	0.00264	0.00230	0.00230	0.00476
Two-stage [(S5.2)(S5.7)(S5.8)]								
μ	0.82952	0.82952	0.83342	0.84653	0.83342	0.84100	0.84100	
σ^2	0.00886	0.00886	0.00803	0.00814	0.00802	0.00802	0.00559	
d	0.01704	0.01724	0.01688	0.01621	0.01688	0.01476	0.01499	
s.e.($\hat{\mu}$)	0.01402	0.01410	0.01423	0.01522	0.01423	0.01379	0.01379	0.01379
s.e.($\hat{\sigma}^2$)	0.00041	0.00041	0.00037	0.00041	0.00037	0.00037	0.00029	0.03410
s.e.(\hat{d})	0.00265	0.00269	0.02814	0.02555	0.02814	0.02632	0.00243	0.05214
Unbiased two-stage [(S5.2)(S5.11)(S5.12)]								
μ	0.82952	0.82952	0.83342	0.84653	0.83342	0.83911	0.83911	
σ^2	0.00886	0.00886	0.00885	0.00899	0.00879	0.00878	0.00608	
d	0.01704	0.01724	0.01857	0.01833	0.01857	0.01657	0.01684	
s.e.($\hat{\mu}$)	0.01402	0.01410	0.01493	0.01665	0.01493	0.01452	0.01452	0.01452
s.e.($\hat{\sigma}^2$)	0.00041	0.00041	0.00046	0.00051	0.00044	0.00044	0.00031	0.00363
s.e.(\hat{d})	0.00265	0.00269	0.00302	0.00333	0.00302	0.00271	0.00271	0.00533

difference in the estimates of σ^2 arises because the denominator used in its calculation is, for ML, the total number of fetuses and, for REML, the same figure less one. For d , the difference is in terms of the cluster sizes (division by n_i or $n_i - 1$), which is more noticeable. All weighted estimators, except with equal weights, lead

Table 4.4: NTP Data (DYME). Cluster-by-cluster analysis. Maximum likelihood and weighted split-sample estimates (standard errors): (a) ML: maximum likelihood; (b) REML: restricted maximum likelihood; (c) Prop.: proportional weights; (d) Equal: equal weights; (e) Approx. sc.: like proportional weights, except that for b_k (4.35) is used; (f) Scalar: scalar weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights; (g) Opt.: optimal weights, with the sub-sample specific weights plugged in for the parameters figuring in the weights. Proper: proper variances for optimal weights

Par.	ML	REML	Prop.	Equal	Approx.	sc.	Scalar	Optimal	
								Weighted	[(S5.2)(S5.3)(S5.4)]
μ	0.84142	0.84141	0.84108	0.84861	0.84108	0.90166	0.90166		
σ^2	0.01031	0.01031	0.01072	0.01071	0.01034	0.01031	0.00700		
d	0.03657	0.03695	0.03102	0.03445	0.03102	0.00745	0.00755		
s.e.($\hat{\mu}$)	0.01926	0.01936	0.01780	0.02502	0.01780	0.01257	0.01257	0.01257	
s.e.($\hat{\sigma}^2$)	0.00044	0.00044	0.00052	0.00079	0.00047	0.00046	0.00033	0.00308	
s.e.(\hat{d})	0.00529	0.00537	0.00570	0.01043	0.00570	0.00159	0.00159	0.00329	
Two-stage [(S5.2)(S5.7)(S5.8)]									
μ	0.84142	0.84141	0.84108	0.84861	0.84108	0.90009	0.90009		
σ^2	0.01031	0.01031	0.00975	0.00964	0.00945	0.00942	0.00650		
d	0.03657	0.03695	0.03199	0.03552	0.03199	0.00836	0.00845		
s.e.($\hat{\mu}$)	0.01926	0.01936	0.01804	0.02535	0.01804	0.01297	0.01297	0.01297	
s.e.($\hat{\sigma}^2$)	0.00044	0.00044	0.00042	0.00059	0.00039	0.00039	0.00030	0.02568	
s.e.(\hat{d})	0.00529	0.00537	0.03433	0.03113	0.03433	0.02215	0.00173	0.04036	
Unbiased two-stage [(S5.2)(S5.11)(S5.12)]									
μ	0.84142	0.84141	0.84108	0.84861	0.84108	0.89672	0.89672		
σ^2	0.01031	0.01031	0.01072	0.01071	0.01034	0.01031	0.00700		
d	0.03657	0.03695	0.03690	0.04514	0.03690	0.01027	0.01037		
s.e.($\hat{\mu}$)	0.01926	0.01936	0.01937	0.02989	0.01937	0.01390	0.01390	0.01390	
s.e.($\hat{\sigma}^2$)	0.00044	0.00044	0.00052	0.00079	0.00047	0.00046	0.00033	0.00353	
s.e.(\hat{d})	0.00529	0.00537	0.00718	0.01542	0.00718	0.00205	0.00205	0.00382	

to very similar point estimates; this is in line with the simulation results. Even for equal weights, the difference is not worrisome. Proportional, equal, and approximate scalar weights are parameter-free and depend at most on the cluster size and/or the number of clusters per size. This explains why these estimators yield standard

errors similar to the likelihood-based ones. Not surprisingly, because of their deviation from optimality, equal weights lead to increased uncertainty.

For the scalar and optimal estimators two issues need to be borne in mind. First, in principle they require knowledge of the true parameters. In the absence of them, plug-in estimates were used. Because of the independence between mean and variance parameters, both methods produce the same results for μ . Also, the estimates for μ are similar to the likelihood-based ones. For σ^2 , this scalar-weight method works better than the optimal, matrix-based one. Because of their matrix nature, optimal weights are less stable when approximated. The standard errors are underestimated because uncertainty, stemming from plugging in the weights is ignored when using the ‘simplified’ precision estimates. When rectified (‘proper’ weights), there is no difference for the mean parameter, because the weights are parameter-free, but there is a strong difference for the variance components. Once the proper standard errors are calculated, it is clear that there is information loss because of using plug-in estimates in the weights, rather than the true ones.

Splitting by Cluster Size, With Dose Effect

While these results illustrate the explicit derivations with a constant mean, the data analysis in Section 8.1 does not do full justice to the actual design of the experiment, because the question of scientific interest is the dose-response relationship. Let x_i be the dose administered to cluster i , taking one out of 4 to 5 values.

The dose levels for the DEHP study are given in Table 4.1. The model then is $\mathbf{Y}_i \sim N(\{\beta_0 + \beta_1 x_i\} \mathbf{1}_n, \sigma^2 I_n + dJ_n)$. Because the mean and covariance parameters are functionally and statistically independent within a sub-sample of constant cluster size, the considerations presented for the constant-mean case will remain valid. The results of fitting this extended model to the DEHP, EG, and DYME compounds, under ML (and REML) on the one hand, and using split-sample methodology (with proportional, equal, and approximate scalar weights) on the other, are presented in Table 4.5. The results are comforting, showing that proportional and approximate scalar weights are a sensible choice. This is consistent with theoretical considerations, the simulations results, and the analysis in Section 8.1.

Cluster-by-cluster Methods

We illustrate the cluster-by-cluster methods here. Results are presented in Tables 4.2–4.4, for DEHP, EG, and DYME, respectively. For brevity, attention here is confined to the case of no dose effect.

We consider three alternatives. In all three, (S5.2) is used for the mean. For the variance components, the pairs (S5.3)-(S5.4), (S5.7)-(S5.8), and (S5.11)-(S5.12) are used, respectively. Because these expressions are derived for a given cluster size, we need to supplement them with a weighting method. For comparison, the same choices are made as reported in Tables 4.2–4.4.

Even though the same estimator *per cluster size* is used for the
Page 87

Table 4.5: NTP Data (with dose effect). Splitting by cluster size. Maximum likelihood and weighted split-sample estimates (standard errors): (a) ML: maximum likelihood; (b) REML: restricted maximum likelihood; (c) Prop.: proportional weights; (d) Equal: equal weights; (e) Approx. sc.: like proportional weights, except that for b_k (4.35) is used.

Par.	ML	REML	Prop.	Equal	Approx. sc.
DEHP					
Interc. β_0	0.96986	0.96987	0.95982	0.95269	0.95982
Dose eff. β_1	-0.00077	-0.00077	-0.00042	-0.00029	-0.00042
σ^2	0.01876	0.01876	0.02122	0.02244	0.01895
d	0.00772	0.00792	0.00538	0.00508	0.00538
s.e.($\widehat{\beta_0}$)	0.01343	0.01357	0.01343	0.01609	0.01343
s.e.($\widehat{\beta_1}$)	0.00012	0.00012	0.00014	0.00018	0.00014
s.e.($\widehat{\sigma^2}$)	0.00084	0.00084	0.00128	0.00199	0.00094
s.e.(\widehat{d})	0.00136	0.00141	0.00137	0.00204	0.00137
EG					
Interc. β_0	0.94228	0.94229	0.94654	0.95320	0.94654
Dose eff. β_1	-0.00009	-0.00009	-0.00010	-0.00010	-0.00010
σ^2	0.00879	0.00879	0.00847	0.00847	0.00833
d	0.00745	0.00765	0.00625	0.00593	0.00625
s.e.($\widehat{\beta_0}$)	0.01453	0.01470	0.01389	0.01406	0.01389
s.e.($\widehat{\beta_1}$)	0.00001	0.00001	0.00001	0.00001	0.00001
s.e.($\widehat{\sigma^2}$)	0.00041	0.00041	0.00044	0.00049	0.00042
s.e.(\widehat{d})	0.00126	0.00130	0.00108	0.00107	0.00108
DYME					
Interc. β_0	1.01875	1.01876	1.02364	1.03680	1.02364
Dose eff. β_1	-0.00102	-0.00102	-0.00099	-0.00100	-0.00099
σ^2	0.01032	0.01032	0.01072	0.01071	0.01034
d	0.00795	0.00813	0.00581	0.00631	0.00581
s.e.($\widehat{\beta_0}$)	0.01356	0.01370	0.01335	0.02000	0.01335
s.e.($\widehat{\beta_1}$)	0.00006	0.00006	0.00006	0.00007	0.00006
s.e.($\widehat{\sigma^2}$)	0.00044	0.00044	0.00052	0.00079	0.00047
s.e.(\widehat{d})	0.00126	0.00130	0.00110	0.00205	0.00110

mean in all three cases, the overall result is different for scalar and optimal weights because these depend on the estimated variance components. A relatively clear message is that proportional and approximate scalar weights show very good performance. This is pleasing, because these weights are parameter-free and hence

easy to apply. As to which of the three versions is better is less clear, this differs somewhat from compound to compound and from parameter to parameter. All three show acceptable behavior. It is interesting to see that in some cases the cluster-by-cluster analysis is closer to ML than the analyses based on splitting per cluster size. Computationally, this approach allows for additional parallel processing, with all clusters analyzed in parallel and the results then combined.

Ramifications and Concluding Remarks

We considered the simple but insightful case of clustered data with a normal compound-symmetry structure and clusters of varying size. Here, there is no closed-form maximum likelihood estimator and maximization must proceed iteratively. Moreover, there is no uniform optimal unbiased estimator and the MLE is only locally optimal.

When considering the collection of estimators obtained from analyzing the data for each cluster size separately, the MLE for the entire dataset is a vector linear combination of them, with the weights depending on the parameters. Based on theoretical results and simulations, as well as on data analysis, we found that equal weights and so-called approximately optimal weights do not perform well. Iterated optimal and proportional weights show excellent performance, and they are simple and parameter-free. One refinement is that for the mean parameter and for the covariance term d weights should be chosen proportional to the number of clusters

of a particular size, c_k , while for the measurement error variance σ^2 proportionality is to the product of the number of clusters of a given size and the cluster size, $c_k \cdot n_k$.

While most of our development is based on the simple, three-parameter compound-symmetry model, in the data analysis we considered a slightly expanded setting, in which the mean takes the form of a regression function. This suggests the use of our results in more elaborate settings, as long as some form of exchangeability prevails. One such setting is the meta-analytic evaluation of surrogate endpoints (?), where two correlated endpoints rather than a single one are considered for each cluster (trial in this case). Admittedly, there may come a point where distinguishing between parameters where it is difficult to determine whether proportional weights or size proportional weights are to be preferred. Based on our simulation results, it may then be sensible to consider proportional weights for all parameters. In the case where clusters take the form of trials, the number of trials may be relatively small, and likely trial sizes are (almost) unique. Our split-sample method then implies that each trial is first analyzed separately, with overall estimates taking the form of linear combinations of trial-specific ones. To provide a formal basis for this, we considered the important special case of a cluster-by-cluster analysis. Such a method is consistent when the number of replicates per cluster (e.g., the number of patients per trial) increases more rapidly than the number of trials. Such an assumption is not realistic in the developmental toxicology setting considered in this paper, but may be sensible in a meta-analysis of

Page 90

clinical trials.

When clusters are very large, it may be attractive to further subdivide them in sub-clusters. Such a splitting method was also considered by ?. Its use in our context would require further investigation.

In the NTP data, the observed cluster size is related to the dose applied. This suggests that it is useful to consider, at the same time, the impact of dose on the outcomes (e.g., fetal weight) as well as on cluster size. This brings us back to the informative cluster sizes mentioned in the Introduction. While work has been done in this area, it is of interest to combine the ideas developed in this paper with a model for cluster size.

Fast, closed-form, and efficient estimators for hierarchical models with AR (1) covariance and unequal cluster sizes

Introduction

It is common for the number of study units in a design, the sample size, to be fixed *a priori*. However, there are many exceptions to this. ? provides an overview of such designs. Some examples are sequential trials, for which the sample size is determined by a stopping rule, missing data, and censored time-to-event data. The focus of this paper is hierarchical data where independent replicates take the form of compound units, generically termed *clusters*.

Clustering can occur in many settings, for example, longitudinal data, toxicology (?), cluster randomized trials and, more generally, multi-level designs. The sizes of the resulting clusters can be fixed by design, or may be random. Random sample sizes can be due to missingness in the data, governed by a stochastic mechanism. And, in some cases, the cluster sizes may be associated with the outcomes, often termed ‘informative cluster sizes’, see for example ??????. However, our interest here is not on informative cluster sizes, but rather unequal cluster sizes that are themselves independent of observed and unobserved outcomes. We concentrate solely on the non-constant nature of the cluster sizes, for which joint modelling of outcomes and cluster size is not required.

? considered the setting of normally distributed clustered data with a compound-symmetry (CS) structure for the dependency, that is, a three-parameter multivariate normal model with a common mean μ , a common variance $\sigma^2 + d$, and a common covariance d . ? showed that, unless the clusters are of the same size, the sufficient statistics are incomplete and the maximum likelihood estimators (MLEs) do not have closed-form solutions. By contrast, if the clusters are all the same size, and hence also within a subset of clusters of the same size, a closed form solution does exist for the MLEs: sufficient statistics are complete and the estimators are minimum variance unbiased. ? studied the CS case and proposed a pseudo-likelihood split-sample approach, as follows. The original sample is divided into subsamples. Maximum likelihood estimation is separately applied to each subsample and the resulting subsample-specific

estimators are averaged using appropriate weights. Appropriate measures of precision for the combined estimators are then obtained. In the current setting it is natural, with this approach, to define the subsamples according to the cluster size (?). When the number of clusters and/or the cluster sizes are very large, standard iterative MLE computation times can become prohibitive. By contrast, the non-iterative nature of the split sample approach can lead to much lower computation times. For the CS setting, ? show that weights proportional to the cluster sizes perform very well for combining the individual estimators.

Although the CS covariance structure is a natural model for settings that exhibit within-cluster symmetry, other settings, such as longitudinal designs, need to be handled. For these we might consider the first-order autoregressive, AR(1), structure, where it is assumed that the correlation between two measurements changes exponentially over time, that is, $\sigma_{ij} = \sigma^2 \rho^{|i-j|}$. This implies that the variance of the measurements is a constant σ^2 and the covariance decreases with increasing time lag. In this paper, we apply the split-sample method to the normal AR(1)-model, which has three parameters, a common mean μ , a common variance σ^2 , and correlation parameter ρ . An important question will be the appropriate choice of weights in such a setting.

As a motivating example, we consider five clinical trials in schizophrenia. These data were collected from five double-blind randomized clinical trials to compare the effects of two treatments for
Page 93

chronic schizophrenia: risperidone and conventional antipsychotic agents. Subjects who received doses of risperidone (4–6 mg/day) or an active control (haloperidol, perphenazine, zuclopentixol) have been included in the analysis.

Patients were clustered within country, and longitudinal measurements were made on each subject over time. The number of patients ranges from 9 to 128 per country with a total of 2039. The positive and negative syndrome scale (PANSS) was used to assess the global condition of a patient. This scale is constructed from 30 items, each taking values between 1 and 7, giving an overall range of 30 to 210. PANSS provides an operationalized, drug-sensitive instrument, which is useful for both typological and dimensional assessment of schizophrenia. Depending on the trial, treatment was administered for a duration of 48 weeks with at most 12 monthly measurements. Because not all subjects received treatment at the same time points and, not the same amount, the final dataset is unbalanced.

This dataset was also analyzed from a surrogate markers point perspective in ?. The focus here is on the treatment effect, accommodating the longitudinal nature of the response.

The rest of paper is organised as follows. In Section 4 the model formulation is given. In Section 4 the estimators for a single constant cluster size are presented. The (in)completeness property is outlined in Section 4, and in Section 4 various weighting schemes for clusters of unequal size are explored. In Section 4, a simulation study is described for the investigation of the performance of the suggested

weights and the data are analysed in Section 4. The closing discussion is presented in Section 4. Some additional background material is available in a separate, web-based appendix¹.

Model Formulation

Suppose that there is a sample of N independent clusters, among which K different cluster sizes n_k ($k = 1, \dots, K$) can be distinguished. Let the multiplicity of cluster size n_k be c_k . The total number of clusters is then $N = \sum_{k=1}^K c_k$. Denote the outcome vector for the i th ($i = 1, \dots, c_k$) replicate among the clusters of size n_k by $\mathbf{Y}_i^{(k)}$.

All models considered in this paper will be versions of the following general linear mixed model:

$$\mathbf{Y}_i^{(k)} | \mathbf{b}_i^{(k)} \sim N(X_i^{(k)}\boldsymbol{\beta} + Z_i^{(k)}\mathbf{b}_i^{(k)}, \Sigma_i^{(k)}), \quad (4.37)$$

$$\mathbf{b}_i^{(k)} \sim N(0, D), \quad (4.38)$$

where $\boldsymbol{\beta}$ is a vector of fixed effects, and $X_i^{(k)}$ and $Z_i^{(k)}$ are design matrices. In what follows, we consider an AR(1) covariance structure, in which case the term $Z_i^{(k)}\mathbf{b}_i^{(k)}$ drops from (4.37), while $\Sigma_i^{(k)} = \sigma^2 C_{n_k}$, with entry (r, s) equal to $\rho^{|r-s|}$. For ease of exposition, the mean structure will often be taken to be $\mu \mathbf{1}_{n_k}$, with $\mathbf{1}_{n_k}$ an n_k column vector of ones.

Note that this is very different from the a so-called balanced conditionally independent model. The contrast between this setting and

¹ Available at <https://ibiotstat.be/online-resources>.

the AR(1) model holds some useful insight. The interested reader can find details about this in the separate Appendix 10.

Estimators

We begin by assuming that there is only one cluster size occurring, that is, $n_k \equiv n$ and the index k will be dropped from notation throughout this section. The resulting expressions are required for our eventual goal, clusters with variable size, which we reach in Section 4.

Again, for the present, we confine attention to clusters of constant size n . (For the purpose of identifiability we assume that there are clusters of size at least two.) Consequently, all dimension-indication subscripts n_k on matrices and vectors can be dropped until we reach Section 4. The AR(1) model of Section 4 can then be written as:

$$\mathbf{Y}_i \sim N\left(X_i\boldsymbol{\beta}, \Sigma = \sigma^2 C\right).$$

Because $C \equiv C(\rho)$, the parameter vector is $\boldsymbol{\theta} = (\boldsymbol{\beta}', \rho, \sigma^2)$. When the mean is constant $\mu_i = X_i\boldsymbol{\beta} = \mu\mathbf{1}$. It is often stated that the MLE for the AR(1) model, with a constant or more elaborate mean structure, requires numerical iteration. This is certainly the case when not all clusters are of the same size. However, in the constant cluster size case considered here, there is a closed-form solution. Our development follows, in part, ?.

For c clusters of length n , the kernel of the log-likelihood takes the
Page 96

form:

$$\ell \propto -\frac{c}{2} \ln |\Sigma| - \frac{1}{2} \sum_{i=1}^c (\mathbf{y}_i - \boldsymbol{\mu}_i)' \Sigma^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_i). \quad (4.39)$$

The score equation for the mean produces, as usual:

$$\hat{\boldsymbol{\beta}} = \frac{1}{c} \sum_{i=1}^c (X_i' \Sigma^{-1} X_i)^{-1} (X_i' \Sigma^{-1} \mathbf{Y}_i). \quad (4.40)$$

Consider (4.40) for the case of a constant mean. If Σ corresponds to independence or compound-symmetry, the MLE for μ is the ordinary sample average, it does not depend on covariance parameters. For a general design $\boldsymbol{\beta}$ is estimated by the OLS estimator. However, in our AR(1) case, solving the score equations leads to:

$$\hat{\mu} = \frac{1}{c[(n-2)(1-\rho)+2]} \sum_{i=1}^c \left(\sum_{j=1}^n Y_{ij} - \rho \sum_{j=2}^{n-1} Y_{ij} \right). \quad (4.41)$$

Not only does (4.41) depend on ρ (hence the MLE for ρ needs to be plugged in), it differs from the OLS:

$$\tilde{\mu} = \frac{1}{cn} \sum_{i=1}^c \sum_{j=1}^n Y_{ij}. \quad (4.42)$$

It follows easily that, when $\rho = 0$ both estimators are the same, as

it should. Interestingly, when $\rho = \pm 1$:

$$\hat{\mu}(\rho = +1) = \frac{1}{c} \sum_{i=1}^c \frac{Y_{i1} + Y_{in}}{2}, \quad (4.43)$$

$$\hat{\mu}(\rho = -1) = \frac{1}{c(n-1)} \sum_{i=1}^c \left(\sum_{j=1}^n Y_{ij} - \frac{Y_{i1} + Y_{in}}{2} \right). \quad (4.44)$$

Turning to the score equations for the variance components, $\partial\ell/\partial\sigma^2$ leads to

$$\sigma^2 = \frac{1}{cn} \sum_{i=1}^c (\mathbf{y}_i - \boldsymbol{\mu}_i)' C^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_i). \quad (4.45)$$

Through C , the right-hand side depends on ρ . For ρ , we find:

$$\sigma^2 \frac{2\rho}{1-\rho^2} = \frac{1}{c(n-1)} \sum_{i=1}^c (\mathbf{y}_i - \boldsymbol{\mu}_i)' F (\mathbf{y}_i - \boldsymbol{\mu}_i), \quad (4.46)$$

with

$$F = \frac{\partial C^{-1}}{\partial \rho} = \frac{1}{(1-\rho^2)^2} \text{tridiag} \left\{ [2\rho, 4\rho, \dots, 4\rho, 2\rho]'; [-(1+\rho^2), \dots, -(1+\rho^2)] \right\} \quad (4.47)$$

and with $\text{tridiag}(\mathbf{v}_1, \mathbf{v}_2)$ a tri-diagonal matrix with \mathbf{v}_1 along the main diagonal and \mathbf{v}_2 on the adjacent diagonals. Both (4.45) and (4.46) contain a summation that can be rewritten as $\text{tr}(S \cdot Q)$, with

$$S = \sum_{i=1}^c (\mathbf{y}_i - \boldsymbol{\mu}_i)(\mathbf{y}_i - \boldsymbol{\mu}_i)'$$

and Q either C^{-1} or F , as in (4.47), respectively. Using this formula

lation, and some straightforward but tedious algebra, produces:

$$f(\rho) = (n - 1)S_2\rho^3 - (n - 2)R\rho^2 - (nS_2 + S_1)\rho + nR = 0, \quad (4.48)$$

the solution of which is the MLE $\hat{\rho}$. Here,

$$S_1 = \sum_{j=1}^n s_{jj}, \quad S_2 = \sum_{j=2}^{n-1} s_{jj}, \quad R = \sum_{j=1}^{n-1} s_{j,j+1}. \quad (4.49)$$

These can be plugged into (4.41) to obtain $\hat{\mu}$ and into:

$$\hat{\sigma}^2 = \frac{1}{c} \cdot \frac{1}{1 - \hat{\rho}^2} \left(S_1 + \hat{\rho}^2 S_2 - 2\hat{\rho}R \right), \quad (4.50)$$

to obtain the MLE for σ^2 .

It is easy to see that $f(\rho)$ has a single root in $[-1, 1]$. Indeed, $f(-\infty) = -\infty$, $f(+\infty) = +\infty$, $f(-1) > 0$, and $f(1) < 0$. The other two real roots are therefore in $] -\infty, -1]$ and $[1, +\infty [$. The general solution of a third-degree polynomial follows from Cardano's method. The polynomial under study was examined by ? who, using results of ?, derived an expression for the solution inside $[-1, 1]$. Alternatively, the method of ? can be used. It takes the following form. Write the polynomial symbolically as $f(\rho) = a\rho^3 + b\rho^2 + c\rho + d$, define

$$p = \frac{3ac - b^2}{3a^2}, \quad q = \frac{2b^3 - 9abc + 27a^2d}{27a^3},$$

and further

$$C(p, q) = 2\sqrt{-\frac{p}{3}} \cos \left[\frac{1}{3} \arccos \left(\frac{3q}{2p} \sqrt{-\frac{3}{p}} \right) \right].$$

For three real roots $t_0 \leq t_1 \leq t_2$, it follows that $t_0 = C(p, q)$, $t_2 = -C(p, -q)$, and $t_1 = -t_0 - t_2$. Finally, $\hat{\rho} = t_1 - b/(3a)$. While not as simple as the other explicit expressions for estimators, the key point is that it has a closed-form which, in turn, can be used to obtain a closed form solution for the mean and the variance, using (4.41) and (4.50), respectively. Given that it is unambiguously clear which of the three cubic solutions is the right one, no comparisons are needed, which enhances computational efficiency.

We now turn to the second derivatives in view of precision estimation. Denote by \mathcal{I} the information matrix. In the usual fashion: $\mathcal{I}_{\beta\beta} = \sum_{i=1}^c X_i' \Sigma^{-1} X_i$. For a simple common mean μ , this becomes: $\mathcal{I}_{\mu\mu} = c[n - (n-2)\rho]/[\sigma^2(1+\rho)]$. Algebraic derivations, sketched in separate Appendix 10, lead to:

$$\mathcal{I}_{\sigma^2\rho,\sigma^2\rho} = c \begin{pmatrix} \frac{n}{2(\sigma^2)^2} & -\frac{n-1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} \\ -\frac{n-1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} & \frac{(n-1)(1+\rho^2)}{(1-\rho^2)^2} \end{pmatrix}. \quad (4.51)$$

It is convenient to slightly change (4.51) to

$$\tilde{\mathcal{I}}_{\sigma^2\rho,\sigma^2\rho} = c \begin{pmatrix} \frac{n-1}{2(\sigma^2)^2} & -\frac{n-1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} \\ -\frac{n-1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} & \frac{(n-1)(1+\rho^2)}{(1-\rho^2)^2} \end{pmatrix}, \quad (4.52)$$

yielding a very simple inverse:

$$\tilde{\mathcal{I}}_{\sigma^2\rho,\sigma^2\rho}^{-1} = \frac{1}{c(n-1)} \begin{pmatrix} \frac{2(\sigma^2)^2(1+\rho^2)}{1-\rho^2} & 2\sigma^2\rho \\ 2\sigma^2\rho & 1-\rho^2 \end{pmatrix}. \quad (4.53)$$

Complete and Incomplete Sufficient Statistics

The property of central interest in this section is that of *completeness* (? , pp. 285–286). It means that any measurable function of a sufficient statistic, that has the zero expectation for every value of the parameter indexing the parametric model class, is the zero function almost everywhere. As has been shown in the sequential trial context, a lack of completeness does not preclude the existence of estimators with desirable properties (?).

? established the incompleteness of the sufficient statistic for a clinical trial with a stopping rule, for the case of normally distributed endpoints. ? generalized this result to the exponential family. ? and ? broadened it further to a stochastic rather than a deterministic stopping rule. They showed that, while the maximum likelihood estimator for the mean exhibits some small-sample bias and is not uniformly best, it still is consistent and asymptotically normal, in most settings, and it is a very reasonable choice.

? studied in detail the compound-symmetry model, which is essentially our model but with the AR(1) variance-covariance matrix replaced by $\sigma^2 I_{n_k} + dJ_{n_k}$, J_{n_k} being an $n_k \times n_k$ matrix of ones. They showed that, while the CS model yields a complete sufficient statistic when the cluster size is constant, this is no longer the case when at least 2 different cluster sizes occur. Rather than the definition of a complete sufficient statistic, they used a convenient characterization (?), stating that a sufficient statistic \mathbf{k} is complete for a parameter $\boldsymbol{\theta}$ in a exponential family model if and only if $\boldsymbol{\theta}$ cannot be transformed 1-1 to a parameterization $\boldsymbol{\eta}$ with a proper

subset $\boldsymbol{\eta}_1$ such that $\boldsymbol{\eta} = [\boldsymbol{\eta}'_1, \boldsymbol{\eta}_2(\boldsymbol{\eta}_1)']'$. In the type of settings that we consider, the minimal sufficient statistic is complete if it is of the same dimension as the estimator.

In what follows, we will establish completeness for the balanced conditional independence model, with the reverse holding for AR(1) model. So, in contrast to the balanced growth curve model and the compound-symmetry model with constant cluster size, an AR(1) model with constant cluster size does not allow complete sufficient statistics. This leads to some surprising results in the AR(1) case, as well as in a number of related settings of a temporal and/or spatial nature. Some of these have been alluded to in the literature of the interbellum and the early post-war period.

Balanced Conditionally Independent Model

This model of which the estimators are spelt out in Section 10, obviously admits a complete minimal sufficient statistic because the numbers of sufficient statistics (A.83)–(A.86) and estimators match (A.87)–(A.90).

AR(1) Model

The mean estimator (4.41) consists of two sufficient statistics:

$$K_1 = \sum_{i=1}^c \sum_{j=1}^n Y_{ij}, \quad K_2 = \sum_{i=1}^c \sum_{j=2}^{n-1} Y_{ij}, \quad (4.54)$$

with the sufficient statistics for σ^2 and ρ spelt out in (4.49). In other words, the three-component vector $\boldsymbol{\theta} = (\mu, \sigma^2, \rho)'$ has a minimal sufficient statistic (K_1, K_2, S_1, S_2, R) of dimension 5, establishing
Page 102

incompleteness.

Even though the AR(1) model has not been studied before from the perspective of incomplete sufficient statistics, its ramifications have been mentioned in the literature. For example, as described by ?, Papadakis proposed, as early as 1937, a correction to the least-squares estimator for correlated observations arising in such settings as adjoining plots designs (????). The topic was also touched upon by ?, in the context of discriminant analysis combined with analysis of covariance. Clearly, the opportunity for such an *ad hoc* correction arises from the incompleteness. ? and earlier authors discussing Papadakis' method refer to the somewhat unusual dependence of the mean estimator on the variance components. This parallels the property of the MLE for the mean in the AR(1) case, as in (4.41). Indeed, because ρ is estimated from solving a third-degree polynomial with coefficients that are functions of the sums of squares and cross-products matrix, it too is a function of such deviations. Of course, the ρ in our case is more complex than Papadakis' correction, which was more of an *ad hoc* nature, while our estimator is the solution to the likelihood equations. In essence, Papadakis' method builds a covariate from deviations observed from adjacent plots. Especially when the plots are arranged as a linear array, the connection with AR(1) is strong. Both non-iterative and iterative versions were proposed by Papadakis. In the iterative case, the covariate is re-built after every iteration, using the current value of the parameters. In more general settings, the data have a spatial layout.

In all of these cases, dependency on adjacent observations gives rise to tri-diagonal matrices, like C^{-1} in the AR(1) setting.

?, p. 172 noted that the relative efficiency of the estimators with or without the use of covariance is not uniformly larger or smaller than one, but that for sufficiently large sample sizes the difference between them is small. This is entirely consistent with our findings for the AR(1) case. For Papadakis' method, the impact on bias and efficiency is described by ?. We refer to our simulations in Section 4.

Because there is no complete minimal sufficient statistic, the MLE is not *a priori* guaranteed to be optimal. Any claims of optimality need to be demonstrated directly.

Proposition 4.1. *In the AR(1) model with constant mean μ and variance-covariance parameters σ^2 and ρ , and with constant cluster size, the MLE for μ is optimal (in the sense of asymptotically most efficient) and linear in the observations, with weights that depend on the parameters only through ρ .*

Note that this is not the ordinary uniform optimality. In case we demand an estimator that does not depend on the parameters at all, it cannot be uniformly more efficient than the MLE, implying that there is no such uniform estimator. The proof is given in separate Appendix 10. This results offers the opportunity to consider estimators, based on weighting that, while not statistically fully efficient, have computational advantages such as stability (e.g., by Page 104

being entirely non-iterative) and speed.

Proposition 4.2. *The result of Proposition 4.1 easily generalizes to a mean of the form $\mu = X\beta$, when the design is constant among clusters.*

Clusters Of Variable Size

? studied various weighting schemes for clusters of unequal size in the compound-symmetry case. Their work was rooted in the pseudo-likelihood and split-sample methods of ? and ?. We will not reproduce their entire argument here, it suffices to focus on the following two-stage procedure:

1. Consider the MLE estimator for each of the K strata, defined by cluster sizes n_k and with c_k replicates. Denote these estimators generically by $\hat{\theta}_k$, with variance V_k .
2. Combine the $\hat{\theta}_k$ in an overall estimator

$$\tilde{\theta}^* = \sum_{k=1}^K A_k \hat{\theta}_k, \quad (4.55)$$

$$\text{var}(\tilde{\theta}^*) = \sum_{k=1}^K A_k V_k A'_k. \quad (4.56)$$

? showed that the sum of the weight matrices should be the identity matrix, an obvious result, and considered, among others, the optimal expression:

$$A_k^{\text{opt}} = \left(\sum_{m=1}^K V_m^{-1} \right)^{-1} V_k^{-1}. \quad (4.57)$$

In the AR(1) case the mean and the variance components are asymptotically independent, hence we can consider them separately. Of course, the variance components are still dependent among them.

For a general mean structure $\mu_i^{(k)} = X_i^{(k)}\beta$, $V_k = \sum_{i=1}^{c_k} X_i^{(k)}\Sigma_k^{-1}X_i^{(k)'}'$, and the above can be applied. Note that $\Sigma_k = \sigma^2 C_k$ with C_k the AR(1) correlation matrix of dimension n_k .

Using optimal weights the β coefficients can then be estimated by:

$$\tilde{\beta} = \left(\sum_{k=1}^K \sum_{i=1}^{c_k} X_i^{(k)'} C_k^{-1} X_i^{(k)} \right)^{-1} \left(\sum_{k=1}^K \sum_{i=1}^{c_k} X_i^{(k)'} C_k^{-1} Y_i^{(k)} \right). \quad (4.58)$$

In the special case that the mean is constant, all $X_i^{(k)}$ are vectors of ones and then

$$\text{var}(\hat{\mu}_k) = v_k = \frac{\sigma^2(1+\rho)}{c_k} \cdot \frac{1}{[n_k - (n_k - 2)\rho]}. \quad (4.59)$$

The optimal weight is then

$$a_k = \frac{c_k [n_k - (n_k - 2)\rho]}{\sum_{m=1}^K c_m [n_m - (n_m - 2)\rho]}. \quad (4.60)$$

It is insightful to consider (4.60) in a few special cases:

$$a_k(\rho = 0) = \frac{c_k n_k}{\sum_{m=1}^K c_m n_m}, \quad (4.61)$$

$$a_k(\rho = 1) = \frac{c_k}{\sum_{m=1}^K c_m}, \quad (4.62)$$

$$a_k(\rho = -1) = \frac{c_k(n_k - 1)}{\sum_{m=1}^K c_m(n_m - 1)}. \quad (4.63)$$

Note that, even though the matrix C is singular for $\rho = \pm 1$, by taking limits, expressions can be found also for these cases. For every $\rho \neq 1$, it follows that if the n_k are sufficiently large: $a_k \approx a_k(\rho = 0)$. This implies that in a broad range of cases, except when $\rho = 1$ (or very close to it), the weights are proportional to the number of observations in a stratum, i.e., $c_k n_k$. We term these *size-proportional weights*. When $\rho = 1$ (a case where AR(1) and compound-symmetry coincide), the weights are instead *proportional*, that is, proportional to c_k .

How well the approximation works is seen in a few special cases. When $\rho = 0.5$, $a_k \propto c_k(n_k + 2)$; for $\rho = 0.9$ this becomes $a_k \propto c_k(n_k + 18)$; finally for $\rho = 0.99$, we find $a_k \propto c_k(n_k + 198)$. Thus, for larger correlations, the size-proportionality matches clusters of sizes much larger than actually observed. But again, in practice, it is convenient and reasonable to operate under size-proportionality.

When estimating the variance of

$$\tilde{\mu} = \sum_{k=1}^K a_k \mu_k, \quad (4.64)$$

using (4.60), the fact that the weights depend on ρ_k needs to be taken into account. Applying the delta method to (4.64), and using the variance expressions in both (4.59) and (4.53), we find:

$$\text{var}(\tilde{\mu}) = \frac{\sum_{k=1}^K a'_k \sigma_k^2 (1 + \rho_k)}{\left(\sum_{k=1}^K a'_k\right)^2} + \frac{\sum_{k=1}^K \left[c_k(n_k - 2) \sum_{m=1}^K a'_m (\mu_k - \mu_m)\right]^2}{\left(\sum_{k=1}^K a'_k\right)^4} \frac{1 - \rho_k^2}{c_k(n_k - 2)} \quad (4.65)$$

We can plug in the stratum-specific $\hat{\rho}_k$ and $\hat{\sigma}_k^2$, or instead use the overall $\hat{\rho}$ and $\hat{\sigma}^2$. In the latter case, (4.65) becomes:

$$\text{var}(\tilde{\mu}) = \sigma^2(1+\rho) \left\{ \frac{1}{\sum_{k=1}^K a'_k} \right\} + (1-\rho^2) \left\{ \frac{\sum_{k=1}^K \frac{c_k(n_k-2)^2}{(n_k-1)} \left[\sum_{m=1}^K a'_m (\mu_k - \bar{\mu})^2 \right]}{\left(\sum_{k=1}^K a'_k \right)^4} \right\} \quad (4.66)$$

Turning to the variance components, we start from (4.52), and use $V_k^{-1} c_k (n_k - 1) P$ with

$$P = \begin{pmatrix} \frac{1}{2(\sigma^2)^2} & -\frac{1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} \\ -\frac{1}{\sigma^2} \cdot \frac{\rho}{1-\rho^2} & \frac{1+\rho^2}{(1-\rho^2)^2} \end{pmatrix}.$$

Now, clearly, the form of P does not matter because it does not depend on c_k and n_k , that is, it is free of stratum-specific quantities. This leads to:

$$A_k = \frac{c_k(n_k-1)}{\sum_{m=1}^K c_m(n_m-1)} P^{-1} P = \frac{c_k(n_k-1)}{\sum_{m=1}^K c_m(n_m-1)} I_2,$$

with I_2 the identity matrix of dimension 2. There are several implications. First, the two variance components have a diagonal weight matrix, implying that mean, variance, and correlation can be treated separately. Second, the variance and correlation have the same sets of weights. Third, they are identical to the weights for the mean when $\rho = -1$. Fourth, because these in themselves are similar to size-proportional weights, we can simplify calculations considerably, especially in large data sets, as follows:

1. Compute $\hat{\mu}_k$, $\hat{\sigma}_k^2$, and $\hat{\rho}$, using the available closed-form ex-

pressions for the MLE.

2. Construct a weighted average of these using size-proportional weights.

Given that the MLE for unequal cluster sizes does not exist in closed form and hence requires iteration, this two-stage approach is nearly optimal, non-iterative, and hence fast.

Algebraic details on formulas can be found in separate Appendix 10 and 10.

Computational Considerations and Simulation Study

In the compound-symmetry covariance structure case it has been seen that the proportional weights perform very well. Due to a constant correlation d , additional observations within a cluster contribute increasingly less information relative to that already observed. By contrast, with an AR(1) covariance structure, the roles of c_k and n_k are quite different.

A first simulation study was carried out to compare the use of proportional and size-proportional weights with respect to changes in ρ . The number of clusters c_k is considered large, but the sizes n_k small. These have been chosen such that equal weights would become identical to the size-proportional weights. In this way we may see how proportional weights can work even worse in some cases. In addition, optimal weights and full likelihood were considered in the comparison. The results are presented together with those obtained for the compound-symmetry case by ?.

For the simulation we took: $c_1 = 500$, $c_2 = 250$, $c_3 = 250$, $c_4 = 500$, and $n_1 = 5$, $n_2 = 10$, $n_3 = 10$, $n_4 = 5$. Parameters are set as $\mu = 0$, $\sigma = 2$ and $\rho \in \{0.01, 0.2, 0.5, 0.8, 0.9, 0.95, 0.99\}$. The data are generated 100 times and the model is fitted using PROC MIXED in SAS (for a single overall intercept).

The results show that, in contrast to the CS case, with an AR(1) covariance structure the size-proportional weights give acceptable results, implying an important role for the clusters sizes, the n_k 's. Proportional weights perform more poorly than equal weights.

The iterative optimal weights will converge in just one iteration (for both CS and AR(1)), which means that iterative optimal weights are nothing but approximated optimal weights. Instead of using $\hat{\theta}_k$ one could also use $\tilde{\theta}$, obtained by using some proper weighting.

In the CS case the iterative optimal weights mainly converge to proportional weights, but with AR(1), they converge to neither proportional nor size-proportional weights. They rather converge to approximated optimal weights which are obtained by substituting the unknown parameter by its estimate using size-proportional weights.

It is observed that, for $\hat{\mu}$ and $\hat{\sigma}^2$, using $\tilde{\theta}$ in optimal weights does not increase the variance to a noticeable degree, but the effect for $\hat{\rho}$ is dramatic. Though it seems that for a larger ρ this effect is diminished. Finding the proper variances when using $\tilde{\theta}$ to approximate optimal weights could be advantageous.

A second simulation study was conducted to compare computation time for closed form solutions to numerical solutions. Using closed form solutions reduces computation time significantly. Details can be found in the separate Appendix 10.

Application: Clinical Trials in Schizophrenia

The data, introduced in Section 4, are analysed here. The active treatments are: risperidone, haloperidol, perphenazine, and zuclopernthonol. We included for analysis patients with at least one follow-up measurement. Table 4.6 shows the number of patients participating in each trial for all different time patterns in receiving the treatments. As one may see, there are 26 different time patterns, therefore, the final dataset is unbalanced. This makes it suitable for examining the performance of sample splitting according to the cluster size.

For the sake of simplicity, we just take the most frequent cluster pattern for each cluster size. The model used to study the effect of the treatments on the response variable is as follows:

$$Y_{ij} = \mu + \alpha_i + \beta t_{ij} + (\alpha\beta)_{ij} + \epsilon_{ij}, \quad i = 1, \dots, 4, \quad j = 1, \dots, n, \quad \epsilon_{ij} \sim N_n(0, R), \quad (4.67)$$

with $R_{\ell m} = \sigma^2 \rho^{|\ell-m|}$ as elements of R , β as the time effect, α_i as the treatment effect, $(\alpha\beta)_{ij}$ as the time and treatment interaction, and μ as the overall mean. For dummy coding, perphenazine has been taken as the reference treatment level.

Table 4.7 shows the treatment levels which appear in the different
Page 111

Table 4.6: PANSS data. Number of clusters in each trial for each cluster pattern. The pattern consists of the numbers representing the months after starting point for which a PANSS score is available.

n	Pattern	Trial					Total
		FIN-1	FRA-3	INT-2	INT-3	INT-7	
2	(0, 1)	17	8	71	43	3	142
	(0, 2)	0	0	2	0	1	3
	(0, 4)	0	0	1	0	0	1
3	(0, 1, 2)	8	4	83	41	7	143
	(0, 2, 4)	0	0	2	0	0	2
	(0, 1, 4)	1	0	3	1	0	5
	(0, 1, 2, 4)	11	0	85	66	5	167
4	(0, 2, 4, 6)	0	0	1	0	1	2
	(0, 2, 4, 8)	0	0	1	0	0	1
	(0, 1, 2, 6)	0	0	3	0	0	3
	(0, 1, 2, 3)	0	4	1	0	0	5
	(0, 1, 3, 6)	0	1	0	0	0	1
	(0, 2, 6, 8)	0	0	0	0	1	1
	(0, 1, 2, 4, 6)	58	0	85	35	6	184
5	(0, 1, 2, 4, 8)	0	0	8	0	1	9
	(0, 1, 4, 6, 8)	0	0	6	0	0	6
	(0, 1, 2, 6, 8)	0	0	8	0	0	8
	(0, 2, 4, 6, 8)	0	0	3	0	2	5
	(0, 2, 4, 8, 12)	0	0	1	0	0	1
	(0, 1, 2, 3, 4)	0	44	0	0	0	44
	(0, 1, 3, 4, 5)	0	1	0	0	0	1
6	(0, 1, 2, 4, 6, 8)	0	0	986	240	74	1300
	(0, 1, 4, 6, 8, 10)	0	0	1	0	0	1
	(0, 1, 2, 6, 8, 12)	0	0	1	0	0	1
	(0, 1, 2, 4, 6, 10)	0	0	1	0	0	1
	(0, 1, 2, 4, 5, 6)	0	0	2	0	0	2

splits. Not all the treatments are present in each split. In other words, not all the splits are contributing to the estimation of every parameter. This fact should be taken into account for constructing
Page 112

the weights. For example, for estimating levomepromazine effect, just the first two splits are contributing, therefore, we have ($c_1 = 142, n_1 = 2$) and ($c_2 = 143, n_2 = 3$), which give proportional weights as (0.498, 0.502), and the size-proportional weights as (0.398, 0.602).

Table 4.8 shows the parameter estimates using sample splitting with proportional and size-proportional weights, compared to the full sample data. Note that, while the point estimates, for example for Zuclopenthixol, differ even in signs, this has to be seen against the background of the precision estimates; their confidence intervals largely overlap.

As mentioned previously, these data are assembled from 5 trials. It might be useful to include the trial and its interaction with the variables already in the model (4.67) to control for the trial effect:

$$Y_{ijk} = \mu + \tau_i + \alpha_j + \beta t_{ij} + (\tau\alpha)_{ij} + (\tau\beta)_{ik} + (\alpha\beta)_{jk} + (\tau\alpha\beta)_{ijk} + \epsilon_{ijk}, \\ i = 1, \dots, 5, \quad j = 1, \dots, 4, \quad k = 1, \dots, n, \quad \epsilon_{ijk} \sim N_n(0, R), \quad (4.68)$$

with $R_{\ell m} = \sigma^2 \rho^{|\ell-m|}$ as elements of R , β as the time effect, α_j as the treatment effect, τ_i as the trial effect, $(\tau\alpha)_{ij}$ as the trial and treatment interaction, $(\tau\beta)_{jk}$ as the trial and time interaction, $(\alpha\beta)_{jk}$ as the treatment and time interaction, $(\tau\alpha\beta)_{ijk}$ as the three-way trial, treatment and time interaction, and μ as the overall mean.

Table 4.9 shows the estimates for the parameters of interest in this model.

Table 4.7: PANSS data. Contributing splits in estimating each parameter. A checkmark signifies that a split contributes, a hyphen the reverse.

Parameter	Split 1	Split 2	Split 3	Split 4	Split 5
Intercept	✓	✓	✓	✓	✓
time	✓	✓	✓	✓	✓
haloperidol	✓	✓	✓	✓	✓
levomepromazine	✓	✓	-	-	-
risperidone	✓	✓	✓	✓	✓
zuclopentixol	✓	✓	✓	✓	-
t*haloperidol	✓	✓	✓	✓	✓
t*levomepromazine	✓	✓	-	-	-
t*risperidone	✓	✓	✓	✓	✓
t*zuclopentixol	✓	✓	✓	✓	-
correlation ρ	✓	✓	✓	✓	✓
variance σ^2	✓	✓	✓	✓	✓

Justification of the chosen model and further details as confidence limits of the tabulated estimates can be found in separate Appendix 10.

Concluding Remarks

As an extension to the normal-compound symmetry model, discussed in ?, the normal AR(1) model was studied in the light of computationally effective estimation for clustered data with unequal cluster sizes.

For constant cluster size there are closed-form solutions but no complete minimal sufficient statistics. However the MLE is shown to be optimal, with weights depending on ρ for the mean. Returning to unequal cluster sizes, there are, in general, no closed

Table 4.8: PANSS data. Estimating fixed effects and variance components and the standard deviations of these estimates using sample splitting (combined with proportional (Prop.) and size-proportional (Size.Prop.) weights) and full likelihood. The model used in here is without trial effect (4.67).

Effect	Par.	Prop.	Size Prop.	
Intercept	μ	89.218 (3.036)	88.167 (2.956)	88
Haloperidol	α_1	-1.916 (3.254)	-1.868 (3.191)	-0
Levomepromazine	α_2	11.823 (14.155)	8.402 (14.366)	32
Risperidone	α_3	-1.474 (3.079)	-0.812 (3.000)	-0
Zuclopentixol	α_4	-1.926 (7.245)	0.146 (7.216)	2
time	β	-3.047 (1.057)	-2.890 (0.613)	-2
time \times haloperidol	$(\alpha\beta)_1$	2.146 (1.108)	1.568 (0.652)	1
time \times levomepromazine	$(\alpha\beta)_2$	6.466 (9.006)	6.924 (8.668)	3
time \times risperidone	$(\alpha\beta)_3$	1.831 (1.070)	1.243 (0.621)	0
time \times zuclopentixol	$(\alpha\beta)_4$	1.551 (3.609)	1.103 (2.655)	0
Correlation	ρ	0.805 (0.006)	0.818 (0.005)	0
Variance	σ^2	419.782 (10.202)	412.850 (10.018)	429.

form solutions. But again estimators have been obtained using a two-stage procedure. Estimators are calculated separately within each stratum (typically defined by cluster size) and combined in an overall estimator. Both theoretical and simulation results show excellent performance of the size-proportional weights, that is through weighting according to the number of measurements in a cluster ($c_k \cdot n_k$), rather than the number of clusters c_k in a subsample, that is, proportional weights. By contrast, the latter are a good choice for the compound-symmetry structure. Under AR(1) they are worse than equal weights. Approximate optimal weights can also be used, but this leads to an estimate of ρ with a large sample variance. In practice, it is convenient and appropriate to use size-proportional

Table 4.9: PANSS data. Estimating fixed effects and variance components and the standard deviations of these estimates using sample splitting (combined with proportional (Prop.) and size-proportional (Size.Prop.) weights) and full likelihood. The model used in here is with trial effect (4.68).

Effect	Par.	Prop.	Size Prop.
Intercept	μ	89.217 (3.016)	88.165 (2.949)
Haloperidol	α_1	2.249 (5.239)	1.491 (5.053)
Levomepromazine	α_2	-9.213 (22.578)	-12.044 (21.761)
Risperidone	α_3	2.353 (4.542)	2.956 (4.216)
Zuclopenthixol	α_4	-2.135 (11.617)	-0.877 (11.509)
time	β	-3.047 (1.049)	-2.890 (0.610)
time \times haloperidol	$(\alpha\beta)_1$	2.170 (1.835)	1.294 (1.056)
time \times levomepromazine	$(\alpha\beta)_2$	16.104 (15.080)	17.287 (13.763)
time \times risperidone	$(\alpha\beta)_3$	1.766 (1.716)	0.794 (0.947)
time \times zuclopenthixol	$(\alpha\beta)_4$	5.218 (5.746)	2.041 (4.188)
Correlation	ρ	0.804 (0.006)	0.818 (0.005)
Variance	σ^2	416.190 (10.139)	410.819 (10.006)

weights; these are parameter free and simple to use. Simulations show, that in certain large settings, computation time can be 1000 times faster than with standard maximum likelihood.

There are missing observations in the PANSS data set. One might therefore consider possible dependencies between cluster size and the outcomes themselves. To handle such informative cluster sizes it might be of interest to extend the current methodology of this paper to a joint model including cluster size. This is a topic for further research.

For non-normal data, no corresponding closed-form formulations are possible. While gains will be less, there might still be computational

advantages, in terms of time and stability, in analyzing the data in cluster-size dependent strata, followed by weighting the so-obtained estimates.

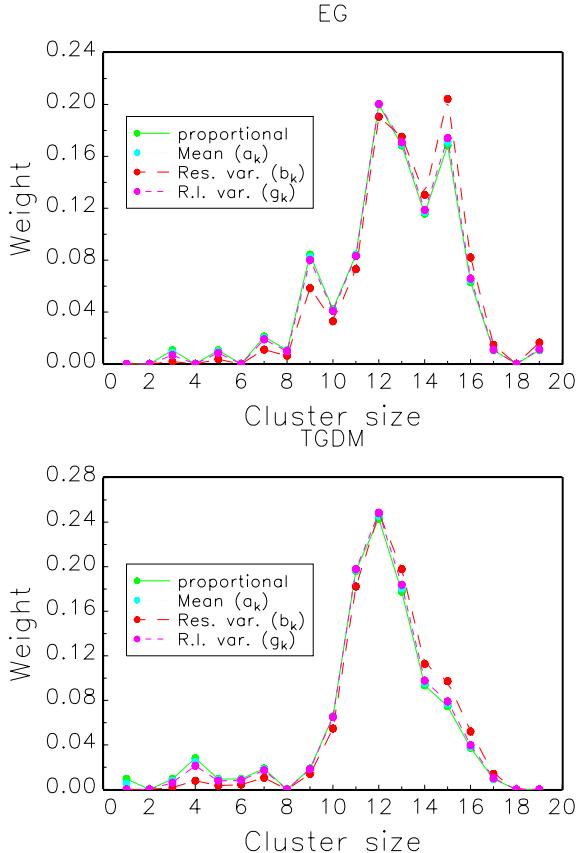


Figure 4.1: NTP Data. Scalar weights: proportional and optimal scalar versions for EG and TGDM datasets. The optimal scalar weights are computed for $\rho = d/(\sigma^2 + d) = 0.5$.

Chapter 5

Random vertical splitting

One of the situations that could make clustered big data is where the cluster sizes are becoming large. For such cases, data splitting at cluster level would not be able to solve the issue. Therefore, one may need to perform the splitting for the observations within each subject. There are several ways to perform this splitting. ? considered non-overlapping vertical clusters. Here we proposed the *iterative multiple oututation* (IMO) as general framework to perform random vertical splitting which can be applied to a wide range of applications. In this chapter, first we introduce the IMO procedure, followed by a multiple-imputation based discussion on determining the number of required sub-samples. Then a special case of it will be studied in detail where only a few sub-samples (sometimes even one) would be sufficient. We call such estimators the final information limit estimators.

Iterative multiple outputation

Iterative multiple outputation (IMO) procedure is introduced as follows,

1. **Start.** Select an initial number of sub-samples, M_0 , and sub-sampling size m . Take M_0 sub-samples of size m , fit the model to each and obtain $\hat{\theta}_i$ and its variance $\Sigma_{\hat{\theta}_i}$ ($i = 1, \dots, M_0$). Then compute

$$\tilde{\boldsymbol{\theta}}_{M_0} = \sum_{i=1}^{M_0} \hat{\boldsymbol{\theta}}_i, \quad \Sigma_{\tilde{\boldsymbol{\theta}}_{M_0}} = \widehat{W}_{M_0} - \left(\frac{M_0 + 1}{M_0} \right) \widehat{B}_{M_0}. \quad (5.1)$$

2. **Update.** For $m > M_0$,

$$\tilde{\boldsymbol{\theta}}_{m+1} = \frac{m\tilde{\boldsymbol{\theta}}_m + \hat{\boldsymbol{\theta}}_{m+1}}{m+1}, \quad \Sigma_{\tilde{\boldsymbol{\theta}}_{m+1}} = \widehat{W}_{m+1} - \left(\frac{m+1}{m} \right) \widehat{B}_{m+1}. \quad (5.2)$$

3. **Distance.** Compute: $d_{m+1} = d(\tilde{\boldsymbol{\theta}}_{m+1}, \tilde{\boldsymbol{\theta}}_m)$ using an appropriate distance.
4. **Stopping rule.** $d_j < \varepsilon$ for $j = m+1, \dots, m+k_0$.

Where for $m = r$ we have,

$$\widehat{W}_r = \frac{\sum_{i=1}^r \sigma_{\hat{\theta}_i}}{r}, \quad \widehat{B}_r = \frac{\sum_{i=1}^r (\hat{\theta}_i - \tilde{\theta}_r)(\hat{\theta}_i - \tilde{\theta}_r)'}{r-1}. \quad (5.3)$$

One important step in this procedure is the Stopping rule step where we need to decide when to stop sub-sampling. It is important
Page 120

because early termination of the procedure would lead to imprecise results, and continuing too long would be against the initial motivation behind using such a procedure which was saving time.

As it was discussed, instead of studying the number of sub-samples in an IMO procedure, we have introduced the iterative multiple imputation (IMI) procedure and studied the stopping rule for the number of imputed datasets. The same conclusions as there can be made for the case of IMO. More details on this topic can be found in Section ???. In brief, we proposed to use Mahalanobis distance (?) between estimated parameters and their covariance matrices between two successive steps.

$$d_{m+1}^{\text{Mah}} = \sqrt{(\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m)^T S^{-1} (\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m)}, \quad (5.4)$$

For S in (3.1) we proposed to use the combined covariance matrix at step $m + 1$. The combination rule is the same as in (1.10). More details on the reasons behin this choice can be found in Section ??.

Furthermore, as one may see the procedure will terminate if the stopping rule stays valid for k_0 successive steps. That is to prevent an early termination, because the convergence here is not monotone. In an extensive simulation study we have observed $k_0 = 3$ is a reasonable choice (see Section ??).

While the sub-sampling would create smaller datasets each time to help the computation time, the task of sub-sampling itself could become cumbersome for very large number of clusters. For such

situations we proposed to use the idea of structured horizontal data splitting to take the sub-samples. We have discussed in Chapter 4 and it is proposed in ?? to split the clustered data based on the cluster sizes such that every split contains clusters of equal sizes. These authors have discussed how this can enhance computations. Here, we have observed that the same idea can be applied to accelerate the sub-sampling step. If the sub-sampling takes place cluster by cluster, it could become very expensive when the number of clusters becomes large. In such case, even using parallel computations would not help much, unless an enormous amount of computation power is at hand.

When all clusters are of equal size, the sub-set selection can be done for all of them at once. Therefore, a practical solution is first to split the data based on the cluster sizes, then to take the sub-samples from each of these sub-sets. Considering the fact that the unique cluster sizes are not usually too variable, this task can easily be distributed among different machines. To implement this idea, we proposed the following steps,

Step1. Construct a frequency table of the cluster ID's. For example:

ID1	ID2	ID3	ID4	ID5	ID6	ID7
5	7	3	3	5	7	5

Step2. Form a vector consists of repeating each cluster size. For example:

Step3. Define the unique cluster sizes. For example,

$$(5, 7, 3)$$

Step4. Take all clusters which their size is smaller than or equal to m (the predefined sub-sampling size). Exclude such cluster sizes from unique cluster sizes. For example, if $m = 4$ then all cluster of size 4 will always be fully present in all of the sub-samples. The remaining of unique cluster sizes will be 5 and 7.

Step5. For each of the remaining unique cluster sizes (where $n_k > m$), repeat the following steps:

- (a) Select all clusters with unique size n_k , call c_k the number of such clusters. For example, $n_k = 5$, so $ID1$, $ID5$ and $ID7$ will be selected.
- (b) Generate $n_k \times c_k$ random numbers (e.g., from normal distribution) and put them in a matrix of size $n_k \times c_k$. For example, for $n_k = 5$ with $c_k = 3$ one should generate 15 random numbers formed as a 4×3 matrix:

$$\begin{array}{ccc} 0.955 & 0.434 & -0.196 \\ -0.861 & 1.319 & -0.245 \\ 2.594 & -0.274 & 0.272 \\ 0.465 & -0.355 & -0.595 \\ 1.131 & -0.311 & -1.034 \end{array}$$

(c) Order the columns of the matrix in the previous step,

For example:

$$\begin{matrix} 2 & 4 & 5 \\ 4 & 5 & 4 \\ 1 & 3 & 2 \\ 5 & 1 & 1 \\ 3 & 2 & 3 \end{matrix}$$

(d) Select the first m observations in the clusters of size n_k .

For example, for $m = 4$:

$$\begin{matrix} 2 & 4 & 5 \\ 4 & 5 & 4 \\ 1 & 3 & 2 \\ 5 & 1 & 1 \end{matrix}$$

We have implemented this procedure in R, but it can as well be implemented in any other programming language.

Finite Information Limit Estimators: Is the Entire Dataset Needed for Analysis?

Introduction

Sample size has always been an issue in statistical analysis. While for many years small sample sizes were a central issue, in recent years, large sample sizes too might confront the statistical analyst with serious problems. When large to huge sample sizes concur with complex models, prohibitive computation times, convergence

issues, and the mere impossibility to analyze the data with the preferred inferential technique (e.g., full maximum likelihood) can ensue. While in the simplest designs all measurements are independent, many designs lead to hierarchical (a.k.a., clustered, correlated, dependent) data ??. By cluster here we mean a set of measurements collected for one single unit (e.g., subject, household, trial, etc.). Therefore, the number of repetitions refers to number of measurements available per cluster. Examples include: patients measured repeatedly over time, several patients attending the same general practitioner or clinic, several links clicked by a specific user, genetic data available for a person, etc. Accounting for the correlation remains a challenge today, when the total number of clusters is large and/or there are a large number of repetitions per cluster. Sometimes, computations become prohibitive. Indeed, Table 5.1 shows the computation time for estimating parameters of a multivariate normal vector of varying length, with constant mean μ and compound-symmetry covariance matrix (CS; which considers constant variance and equal correlation between every pair of variables; see Section 5). The model is fitted using the MIXED procedure in SAS 9.4 for Windows. As one may see, by increasing the cluster size, the computation time increases dramatically, and after some point, computation is no longer feasible.

There are several approaches to deal with large clustered data sets, such as splitting the large sample either at the subject level or at measurement level ?. They proposed combination rules based on pseudo-likelihood theory. On the other hand, ?? suggested

Table 5.1: Computation time t (in seconds) for clustered data with unique cluster sizes $n_k \cdot 10^i$, with $n_k = 5, 6, 7, 8, 9, 10$ and $i = 0, 1, 2, 3, 4$, and with 100 clusters from each n_k (600 clusters in total).

n_k	$n_k \times 10^0$	$n_k \times 10^1$	$n_k \times 10^2$	$n_k \times 10^3$
t	1.76	2.47	1016.96	ERROR*

* The SAS System's message: "ERROR: The SAS System stopped processing this step because of insufficient memory."

splitting a large sample along unique cluster sizes, such that each sub-sample becomes balanced, i.e., formed of clusters of equal sizes, leading to closed-form (weighted) and efficient solutions ?. A permutation-based solution was proposed and applied by ?. A semi-parallel approach was adopted by ?? to deal with large clustered data stemming from genome-wide association studies. Further, ? employed Bayesian approaches to tackle large clustered datasets.

In this paper, we focus on the special case where the number of clusters may or may not be large, but where definitely each or at least some clusters are very to extremely large. The nature of the problem is that the variance-covariance matrix of a cluster usually is not of a diagonal form, such as $\sigma^2 I$, but instead a variance-covariance matrix Σ with non-zero off-diagonal elements is assumed to capture correlation. A typical example is compound-symmetry.

Consider Y_{ij} as the outcome of interest, measured on independent clusters $i = 1, \dots, N$ with $j = 1, \dots, n_i$ measurements per cluster. Outcomes from cluster i are grouped into \mathbf{Y}_i . ? studied the so-called effective sample size for such data. It is defined as the number

of independent measurements that leads to the same amount of information as the clustered data under investigation. It can be formulated at the level of an individual cluster or for the dataset as a whole. For a cluster of size n_i and non-negative correlation, it lies somewhere between 1 (if correlation is 1) and n_i (if correlation is 0). Strictly speaking, for negative correlation, it will exceed n_i , but such cases are rare in practice and require separate investigation altogether. Intuitively, this is because cluster replications share information and hence additional members of a clusters do not provide as much information as would come from an independent replicate. The effective sample size depends, apart from n_i , on the magnitude of the correlation as well as on the specific form of the correlation matrix (e.g., CS, autoregressive, etc.)

Based on ?, we will show that, for specific correlation structures, a very large fraction of the information in a cluster can be captured by considering only a small subset of it. CS being a special case, the general class of structures with this property will be termed to possess the Finite Information Limit (FIL) property. It is specific to an estimator, rather than to a data generating model as a whole. For example, the FIL may apply to a mean estimator but not a contrast parameter in a CS model. The computational prospects are considerable, for example, in a case where analyzing a 1% subsample leads to estimators nearly as efficient as their conventional counterparts based on the entire dataset. Note that the results here are extensions from a proceedings contribution ?.

In the current paper more general cases, such as sub-sampling more

then once, are discussed and investigated. Also, the Amazon books ratings are analyzed as an illustrative application.

CS, as a generic example, is examined in Section 5. The FIL property is defined in Section 5 and extended in Section 5. A simple procedure to detect FIL and to determine a fraction for sub-sampling is proposed in that section, too. Simulation studies are presented in Sections 5 and 5. The FIL property is studied in a data of Amazon website book ratings in Section 5. Finally, Section 5 concludes the paper.

A compound-symmetry model

Consider the data setting of the previous section with further $\mathbf{Y}_i \sim N(\mu \mathbf{1}_{n_i}, \Sigma_i)$ and $\Sigma_i = \sigma^2 I_{n_i} + \tau J_{n_i}$, where $\mathbf{1}_{n_i}$ is an all-ones vector of length n_i , I_{n_i} is an identity matrix of order n_i , and J_{n_i} is an all-ones square matrix of order n_i . In other words, it features constant variance and equal correlation between every pair of variables. The CS covariance structure is also called exchangeable or equicorrelated in the literature. Wherever possible without ambiguity, the index i will be dropped from notation and we will focus on a ‘generic cluster’.

The FIL property is rooted in the effective sample size (ESS) concept ?. The ESS is the number of independent observations needed to reach the same amount of information for a particular estimator as in the original, clustered dataset. For the CS structure, ? have shown that the ESS \tilde{n} for estimating a common mean μ from a

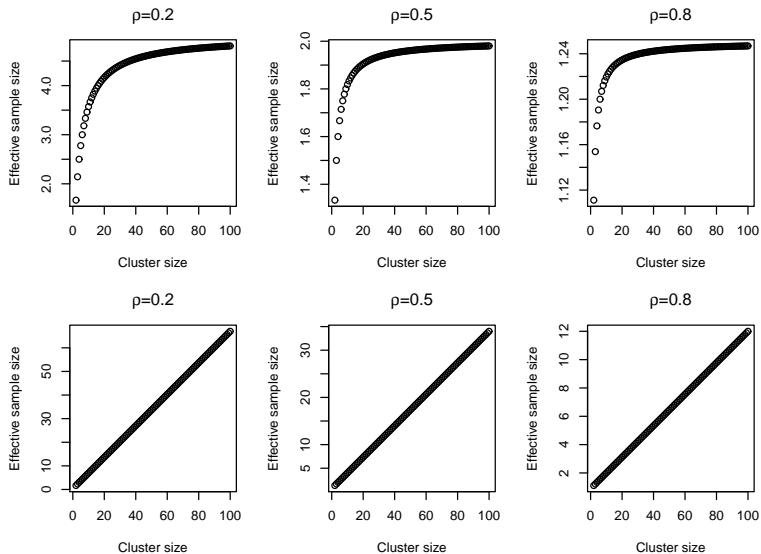


Figure 5.1: The effective sample sizes (vertical axis) for different cluster sizes for CS (first row) and AR(1) (second row) covariance structures for different correlation values ρ .

cluster of size n is:

$$\tilde{n} = \frac{n}{1 + \rho(n - 1)}, \quad \text{where } \rho = \frac{\tau}{\sigma^2 + \tau}. \quad (5.5)$$

The ESS for the entire dataset, \tilde{N} say, follows from summing the corresponding \tilde{n}_i over all clusters.

Figure 5.1 (first row) shows the effective sample sizes in the CS case for $\rho = \{0.2, 0.5, 0.8\}$ and $n = \{1, 2, \dots, 100\}$. As one may see, the effective sample size in the CS case converges to an asymptote. This value decreases with increasing ρ . But no matter the non-zero value of ρ , the effective sample size is bounded from above. This

follows as the limit of (5.5):

$$\lim_{n_i \rightarrow \infty} \frac{n_i}{1 + \rho(n_i - 1)} = \frac{1}{\rho}. \quad (5.6)$$

This property is called the information limit by ?. Note that this does not apply to negative ρ . In that case, the lower limit of the correlation depends on n , as it equals $-1/(n - 1)$. In other words, should $n \rightarrow \infty$, the correlation cannot be smaller than 0, contradicting a negative correlation. Thus, it is sensible to restrict attention to non-negative correlation. This lower bound for the intra-class correlation, ρ , comes from the fact that for a smaller ρ the covariance matrix will be non-positive definite, i.e., the variance would be negative.

For positive correlation, the amount of information from CS clusters is bounded from above. As a consequence, after some point, adding more replications to a cluster would no longer lead to useful efficiency improvements.

To make this point more clearly, consider the variance of estimating the parameter μ of a multivariate normal vector with CS covariance structure using N clusters all of size n . This variance is ?:

$$\text{Var}(\hat{\mu}_n) = \frac{\sigma^2 + n\tau}{Nn} = \frac{\sigma^2}{Nn} \frac{(\sigma^2 + n\tau)}{\sigma^2}. \quad (5.7)$$

The quantity σ^2/Nn is the variance of $\hat{\mu}_n$ if the data were uncorrelated ($\tau = 0$), the factor $\sigma^2 + n\tau/\sigma^2 > 1$ is the price we pay because of non-zero correlation in terms of variance. This quantity is often

called the variance inflation factor (VIF), i.e., the correlation inflates the variance comparing with the case of non-correlated data. As non-zero correlation would inflate the variance, it deflates the information. We may define the variance deflation factor, VDF, as $1/\text{VIF}$. As one may see, for the CS case, the VIF goes to infinity as $n \rightarrow \infty$. Hence, VDF goes to zero.

The asymptotic relative efficiency (ARE) of estimating μ with N clusters of size $n \rightarrow \infty$, compared to estimating it using N clusters of finite size $n_f < \infty$ is:

$$\lim_{n \rightarrow \infty} \frac{\text{Var}(\hat{\mu}_{n_f})}{\text{Var}(\hat{\mu}_n)} = \frac{\sigma^2 + \tau n_f}{\tau n_f} = 1 + \frac{(1 - \rho)}{n_f \rho} > 1. \quad (5.8)$$

(This definition, rather than its inverse, is used because ensuing expressions are easiest to manipulate.) Therefore, using a finite random subset of each cluster, the ARE is finite. Hence, one may choose n_f such that the efficiency loss compared to clusters of infinite size becomes arbitrarily small. Indeed, for given ρ , set the required ARE equal to $1 + \varepsilon$. Then,

$$n_f = \frac{1 - \rho}{\rho \varepsilon} = \frac{\sigma^2}{\tau \varepsilon}. \quad (5.9)$$

For example, set $\varepsilon = 0.01$, a very small increase of variability. For $\rho = 0.5$, $n_f = 100$ would reach this goal. Should $\rho = 0.9$, then $n_f \simeq 11$ would fulfill the requirement. These are important reductions in big-data settings, or whenever n_f/n becomes very

small. The corresponding expression for finite (typically large) n is:

$$n_f = \frac{n(1 - \rho)}{\varepsilon(1 + \rho + n\rho) + 1 + \rho}.$$

Considering (5.7), with a cluster of infinite size, the variance is:

$$\lim_{n \rightarrow \infty} \frac{\sigma^2 + n\tau}{Nn} = \frac{\tau}{N}. \quad (5.10)$$

In other words, with the given number of clusters, the minimum variance we can reach is (5.10). That is to say, in such a situation, for a given number of clusters, increasing the cluster sizes cannot make the variance smaller than a specific value. Obtaining a smaller variance is still possible by adding new clusters (increasing N). Such considerations can be found in the literature on determining sample and cluster sizes for cluster randomized trials. For example, ? have considered a 3-level model, with CS structure for variance at both levels (each with its own correlation); ? extended these calculations to generalized linear models. The information limit property of CS-based estimators is considered from a different point of view in ?, ?, and ?.

Finite Information Limit (FIL) Estimators

It is useful to generally define FIL. We also examine some other examples, and provide an important counterexample.

Consider the same clustered setting as before. Let θ be the parameter of interest and $\hat{\theta}$ a corresponding estimator. Then, Cov($\hat{\theta}$)
Page 132

has the FIL property if, for fixed N , $\lim_{n_i \rightarrow \infty} \mathcal{I}_{\boldsymbol{\theta}} < \infty$, where $\mathcal{I}_{\boldsymbol{\theta}} = \text{Cov}(\widehat{\boldsymbol{\theta}})^{-1}$ is the Fisher information.

The practical implication is the same as with CS. Let us say that there are K unique cluster sizes among the n_i ; index these as n_k and assume that for each k there are c_k clusters in the dataset. Then, $N = \sum_{k=1}^K c_k$. Because of FIL, one can consider a cluster size n_f , ideally $n_f \ll \min_k\{n_k\}$, such that the efficiency loss is less than a small, pre-specified amount. In some cases, it may be necessary to include all data from the subset of smaller clusters, while sub-sampling the larger clusters.

In effective sample size terminology, the estimator $\widehat{\boldsymbol{\theta}}$ has the FIL property if and only if the difference between effective sample sizes of original and sub-sampled data can be made arbitrarily small. Adopting a VDF viewpoint, $\widehat{\boldsymbol{\theta}}$ has the FIL property, if as the cluster size $n \rightarrow \infty$, the VDF goes to zero with at least linear rate.

While CS is a clear example, it is not the only one. Multivariate normal models with variance-covariance structure where the correlation between two replications within a cluster does not approach zero with increasing (spatial or temporal) distance between them are FIL candidates. These include correlation structures induced by linear mixed models with, for example, random intercept and random slope in time. For practical examples, consider craniofacial growth in rats ?, several products examined by an expert panel ?, genetic data where long sequences of genetic data are available for each subject ?, etc.

Consider a linear mixed model $\mathbf{Y}_i \sim N(X_i\beta, \Sigma_i)$, where $X_i\beta = \mu\mathbf{1}$ and $\Sigma_i = Z_i D Z_i + V_i$. Obviously, the $\bar{\mathbf{Y}} = 1/n(\mathbf{1}'\mathbf{Y}_i) \sim N(\mu, 1/n^2\mathbf{1}'\Sigma_i\mathbf{1})$. The FIL exists if the variance $1/n^2\mathbf{1}'\Sigma_i\mathbf{1}$ does not go to zero when cluster size tends to infinity. In other words, for a \sqrt{n} -consistent estimator, FIL does not apply.

Evidently, a linear mixed model with only a random intercept and independent errors with variance σ^2 is identical to the CS model. We have already seen that FIL exists in such a case. For a linear mixed model with random intercept b_{0i} and random slope b_{1i} in time one can derive,

$$\text{Var}(\widehat{\mu_{IS}}) = d_{22} \frac{(n+1)^2}{4} + d_{12}(n+1) + d_{11} + \frac{\sigma^2}{n},$$

where $d_{11} = \text{Var}(b_{0i})$, $d_{22} = \text{Var}(b_{1i})$ and $d_{12} = \text{Cov}(b_{0i}, b_{1i})$. Also, IS in μ_{IS} refers to a random intercept-slope model. Obviously, as $n \rightarrow \infty$, $\text{Var}(\widehat{\mu_{IS}}) \rightarrow \infty$, so $\mathcal{I}_{\widehat{\mu_{IS}}}$ is finite and FIL applies in this case as well. If one also looks at the VDF here, it goes to zero at a cubic rate.

Considering the CS case with $\rho = 0$, this would be an independent identically distributed sample from $N(\mu, \sigma^2)$. It is well-known and also straightforward to see that the estimator of μ is \sqrt{n} -consistent in this case, hence its variance goes to 0 as n tends to infinity. Therefore, the information is unbounded. However, such \sqrt{n} -consistent estimators are not specific to independent (non-clustered) data. It can happen for correlated (clustered) data as well.

A clear counterexample of non-FIL for clustered data is the multivariate normal vector with first-order autoregressive (AR(1)) variance-covariance. Then, $\Sigma = \sigma^2 C_n$ where the (j, k) element of C_n is $\rho^{|j-k|}$. Clearly, the correlation between Y_{ij} and Y_{ik} decreases rapidly with $|j - k|$.

The variance of $\hat{\mu}$ in this case, based on N clusters of size n , is ?:

$$\text{Var}(\hat{\mu}_{AR1}) = \frac{\sigma^2(1 + \rho)}{N[n - (n - 2)\rho]}. \quad (5.11)$$

Obviously,

$$\lim_{n \rightarrow \infty} \frac{1}{\text{Var}(\hat{\mu}_{AR1})} = \infty. \quad (5.12)$$

The ARE of $\text{Var}(\hat{\mu}_{AR1})$ when using $n_f < n$ measurements from each cluster, when $n \rightarrow \infty$, is:

$$\lim_{n \rightarrow \infty} \frac{\text{Var}(\hat{\mu}_{n_f})}{\text{Var}(\hat{\mu}_n)} = \lim_{n \rightarrow \infty} \frac{(1 - \rho)n + 2\rho}{(1 - \rho)n_f + 2\rho} = \infty. \quad (5.13)$$

The above follows from the ESS as well. For AR(1), derived ?:

$$\tilde{n} = \frac{n - (n - 2)\rho}{1 + \rho}. \quad (5.14)$$

Figure 5.1 (second row) shows the effective sample sizes in the AR(1) case for $\rho = \{0.2, 0.5, 0.8\}$ and $n = \{1, 2, \dots, 100\}$. As one may see, from (5.14) and the figure, and in contrast to the CS case, no matter what is ρ or n , the effective sample size does not converge to any asymptote. Of course, because of the correlation, the ESS is less than n ; for larger ρ 's, the ESS is smaller, but it is not bounded from above. For example, when $\rho = 0.5$, $\tilde{n} = (n+2)/3$;

for $\rho = 0.9$, we find $\tilde{n} = (n + 18)/19$. Hence, there is no information limit and adding cluster replications will add information without approaching a limit. Related, one may verify that the VDF for AR(1) case does not go to zero neither. This property ensures that time-series analysis works for AR(1) structures or one of the many generalizations thereof.

It is worth remarking that, in cases such as AR(1), increasing the cluster size means increasing the distance, $|j - k|$, (spatially, temporally, etc.) between measurements. As is implied by the structure, that would make the correlation smaller, so unlike in cases with FIL property, the correlation here is not persistent. In other words, no matter how large the correlation, if the lag between two measures increases, the correlation between them will become arbitrarily small, so the information added by such a new measurement will become virtually the same as for an independent additional measurement. Therefore, the information is not bounded in such structures.

Although FIL cannot be used to deal with high-dimensional AR(1) data, there are other possible solutions. Recently, ? have derived closed-form solutions for AR(1) model parameters in case the cluster sizes are all equal. They presented fast and stable performance of such solutions compared to existing iterative methods. Furthermore, they have proposed a sample-splitting solution to deal with clusters of unequal sizes for the case of AR(1) data. That would extend the use of the derived closed-form solutions in the case of unequally
Page 136

sized clusters.

Simulations I

A first simulation study is conducted to illustrate the FIL property for CS data. To this end, CS clusters are generated. Consider c_k clusters of unique sizes n_k for $k = 1, \dots, 6$. The simulation settings are:

- $\mu = 0$, $\tau = 5$ and $\sigma^2 \in \{20, 5, 1.25\}$ to produce $\rho \in \{0.2, 0.5, 0.8\}$
- $n_k \in \{500, 600, 700, 800, 900, 1000\}$, and $c_k \in \{100, 100, 100, 100, 100\}$.
- $n_f = \{5, 10, 50\}$.

A set of data is generated 50 times for each setting. Parameters are estimated for the full data using the MIXED procedure in SAS 9.4 for Windows. A sub-sample of size $n_f \in \{5, 10, 50\}$ is taken and the parameter μ and its variance estimated using balanced-data closed forms ?. The ARE's are computed by comparing the variance using a sub-sample of size n_f with the entire dataset.

Figure 5.2 shows the results for $n_f = \{5, 10, 50\}$ and $\rho = \{0.2, 0.5, 0.8\}$. Clearly, for $\rho = 0.2$, a very high efficiency is reached using the sub-samples of size $n_f = 50$, which gets even better for the higher correlations: indeed, when $\rho = 0.5$, one can suffice with $n_f = 10$, while for $\rho = 0.8$, a sub-sample of size $n_f = 5$ leads to almost full efficiency. In other words, even for a correlation as small as $\rho = 0.2$, analyzing only 6% of the data

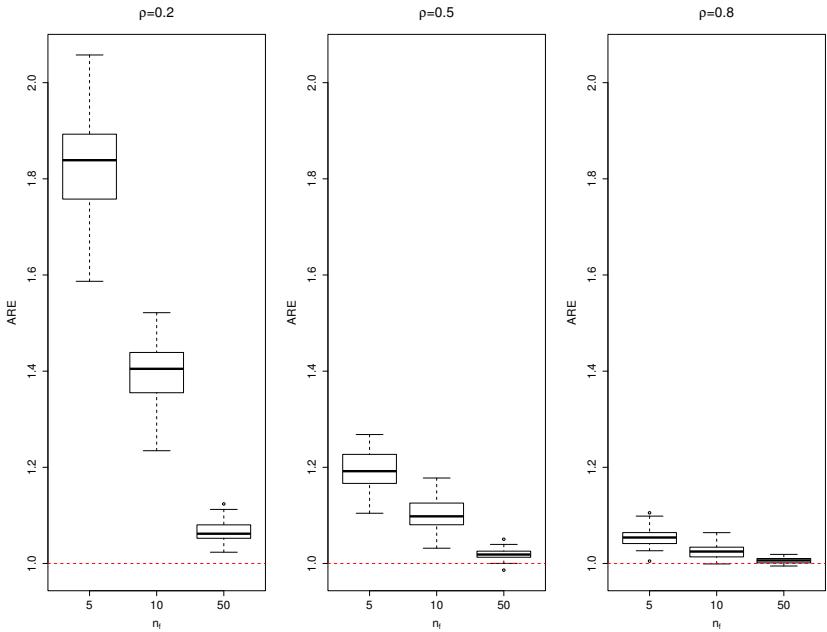


Figure 5.2: Comparing the ARE of full data with sub-samples of sizes $n_f = \{5, 10, 50\}$ for $\rho = \{0.2, 0.5, 0.8\}$. The dashed horizontal line shows $ARE = 1$.

produces almost the same efficiency as using the entire dataset. When the correlation becomes as high as $\rho = 0.8$, this proportion decreases to as little as 0.6%.

Using such a (tiny) sub-sample has two main advantages. First, the computation time drops substantially. This can be seen from Table 5.2. On the other hand, this sub-sample can be chosen with all cluster sizes constant, which further enables the use of closed-forms ?, further reducing computation time and avoiding any convergence issues. Thus, sacrificing a tiny bit of efficiency can be considered a

good bargain.

Extensions

From Sections 5 and 5, the advantages of FIL are clear. When CS is known to be true or assumed, sub-sampling can be used, unlike in the AR(1) case. But the question that remains is how to determine the sub-sampling size, n_f .

For the CS case, by pre-specifying the ARE rate, $1 + \varepsilon$, one may use (5.9) to determine the needed cluster size to achieve the desired level of efficiency comparing with a cluster of infinite size. Of course, the formula in (5.9) involves unknown parameters. In practice, one would take a small manageable sub-sample from the data (e.g., $n_f = 5$) to estimate the unknown parameters. To consider the extra uncertainty introduced into this estimator via replacing parameters with their estimates, a confidence interval can be constructed for n_f and its upper-bound can be used, to be on the safe side. As (5.9) involves the ratio of two random variables (as the parameters are replaced by their estimators), Fieller's method ? or the Delta method can be used to find such a confidence interval. Details of calculating such intervals will be provided in Appendix 10.

While for a FIL estimator taking only one sub-sample can provide the desired efficiency compared with using the full sample, when computational power available, it worth to consider taking more sub-samples of smaller sizes. Assume a single cluster of size N with multivariate normal distribution and CS structure for its covariance

matrix. Suppose we take M_0 sub-samples of size n_0 from this cluster, we have derived:

$$\text{ARE} = \frac{1}{2} + \frac{1}{2} \frac{N}{n_0} \frac{1 + (n_0 - 1)\rho}{1 + (N - 1)\rho}. \quad (5.15)$$

For example, for sufficiently large N , $M_0 = 2$, $\rho = 0.5$, and efficiency equal to $5/4$ one may derive $n_f = 2$, i.e., taking only two sub-samples of size 2 would produce such efficient estimators. The ARE's in (5.8) and (5.15) are derived in Appendix 10.

As stated earlier, a CS covariance structure is an important example of FIL estimators, while AR(1) is an important counterexample. The next question that needs to be addressed is how to examine, in general terms, whether the model at hand actually allows for a FIL estimator.

A priori, variance-covariance structures purely induced by serial or spatial processes would not be good FIL candidates, while structures induced at least in part by random effects might offer more hope. Still, the above is intuitive only and thus there is the need for a tool to detect this property in cases where the analyst is not sure. Of course, the ‘ideal’ but impractical solution is to analyze the full dataset as well as a small subset of it and to compare the results. While useless as such, it points to an alternative strategy. Because of the very information limit, it is not necessary to compare the use of subsets of size n_f to the full dataset, but rather to a subset of clusters of a larger size, but still smaller than the actual cluster sizes.

Using the same data setting as before, the following algorithm can be executed:

Step 1. Determine n_{f_ℓ} for $\ell = 1, \dots, L$ such that $n_{f_1} < n_{f_2} < \dots < n_{f_L} \ll \min_k \{n_k\}$.

Step 2. For given ℓ , take M random sub-samples of size n_{f_ℓ} and compute $\widehat{\mu}_{\ell m}$ and its variance $\text{Var}(\widehat{\mu}_{\ell m})$ for $m = 1, \dots, M$. Combine $\text{Var}(\widehat{\mu}_{\ell m})$ ($m = 1, \dots, M$) to obtain $\text{Var}(\widetilde{\mu}_{n_{f_\ell}})$ using the combination rules (5.16), (5.17), and (5.18). Repeat this step for $\ell = 1, \dots, L$.

Step 3. If there exists an ℓ such that $\frac{\text{Var}(\widetilde{\mu}_{n_{f_\ell}})}{\text{Var}(\widetilde{\mu}_{n_{f_L}})} < 1 + \varepsilon$, for a pre-specified ε , then the estimator under investigation can be taken to have the FIL property.

Step 4. In case FIL is concluded in Step 3, the final estimate can be computed using $\bar{\mu} = \sum_{\ell=1}^L w_\ell \widetilde{\mu}_\ell$ and $\text{Var}(\bar{\mu}) = \sum_{\ell=1}^L w_\ell \text{Var}(\widetilde{\mu}_\ell)$, where $w_\ell = \frac{n_{f_\ell}}{\sum_{\ell=1}^L n_{f_\ell}}$.

Several remarks are worth making. First, $L = 2$ might be enough, but depending on the size of the dataset and the computational power available, one can use more than two n_f to reach better estimators. While usually a very small L is sufficient, this parameter can also be gauged empirically, through similar studies. In other words, this implies that experience will grow through sequences of similar data analyses. Second, the sub-sampling can be done only once for each n_{f_ℓ} , i.e., $M = 1$, whence no combination rules are needed. Third, generating and analyzing the samples for each

(ℓ, m) pair is an array of independent tasks; doing them in parallel is straightforward. Fourth, the sub-sampling can be done more efficiently by splitting the data based on the cluster sizes; this approach is explained in the Appendix ?? and an R implementation of it is presented. Fifth, note that not being FIL does not merely follow from not being able to find an ℓ that satisfies the condition in Step 3.

The combination rules in Step 2 are based on the so-called multiple outputation idea ???. Let us briefly sketch the method. Consider M sub-samples of size n_s , sampled with replacement. Suppose $\hat{\boldsymbol{\theta}}_m$ as the estimated parameter in the m -th sub-sample with $\hat{\Sigma}_m$ as its covariance. The combined estimate $\tilde{\boldsymbol{\theta}}$ and its variance can be computed using the following combination rule:

$$\tilde{\boldsymbol{\theta}} = \frac{1}{M} \sum_{m=1}^M \hat{\boldsymbol{\theta}}_m, \quad (5.16)$$

and its covariance is computed as,

$$\tilde{\Sigma} = \frac{1}{M} \sum_{m=1}^M \hat{\Sigma}_m - \frac{M-1}{M} S_{\boldsymbol{\theta}}^2, \quad (5.17)$$

where,

$$S_{\boldsymbol{\theta}}^2 = \frac{1}{M-1} \sum_{m=1}^M (\hat{\boldsymbol{\theta}}_m - \tilde{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}}_m - \tilde{\boldsymbol{\theta}})'. \quad (5.18)$$

Simulations II

To study the performance of the proposed procedure in Section 5 the simulation study in Subsection 5 is extended by repeating the sub-sampling $M = 1, 3, 5$ times. Again, a set of data is generated 50 times for each $\rho \in \{0.2, 0.5, 0.8\}$ and then the parameter μ and its variance are estimated for the full data using the MIXED procedure in SAS 9.4 for Windows. A sub-sample of size $n_f \in \{5, 10, 50\}$ is taken $M \in \{1, 3, 5\}$ times. Estimation of μ and its variance is done in the same ways as in the first simulation. Two ARE's are computed. First, the variance of each sub-sample (for different combinations of n_f and M) is compared with the one from the full data. Second, to study the performance of the detection method in Section 5, the variance of each sub-sample, except the largest one ($n_f = 50, M = 5$) is compared with the variance of the largest sub-sample. Note that the negative ARE in some cases where $M > 1$ occurs because of negative variance estimates using the correction in (5.17). This matter is well-known ? and can be avoided by increasing M or simply use $M = 1$ with sufficiently large n_f .

Figure 5.3 shows the results for $\rho = 0.2$. As one may see in this case using $n_f = 5$ is not as efficient as expected. While $n_f = 10$ could be accepted, $n_f = 50$ (6.68% of data) provides almost fully efficient results. Looking at Figure 5.4 for $\rho = 0.5$ establishes that using only $n_f = 10$ (1.33% of data) could provide almost fully efficient results. When it comes to $\rho = 0.8$ in Figure 5.5, even using $n_f = 5$ (0.67% of data) is almost fully efficient.

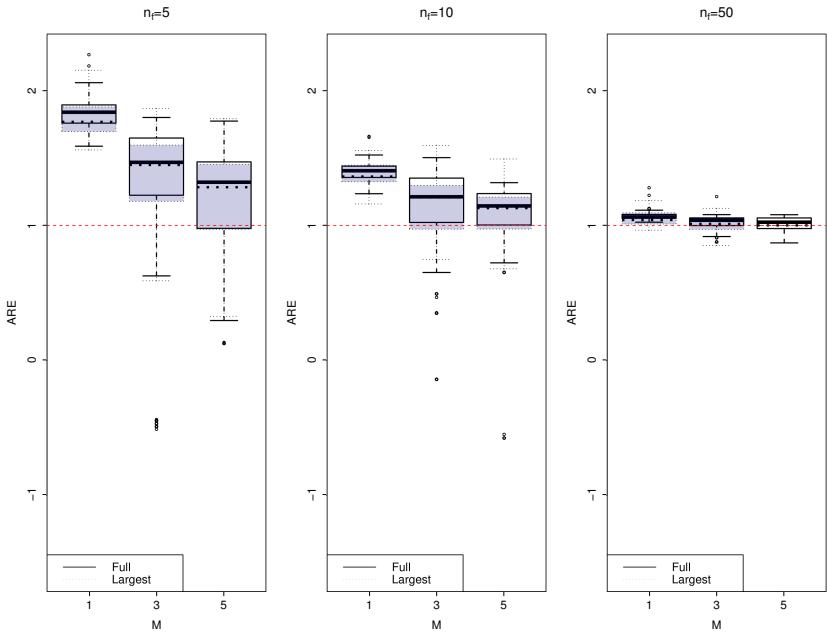


Figure 5.3: Comparing the ARE of full data (the boxplot with background color and solid lines) and sub-sampling of size $n_f = 50$, $M = 5$ times (the box plots with transparent color and dotted lines) with different combinations of n_f and M for $\rho = 0.2$. The dashed horizontal line shows $ARE = 1$.

Another interesting observation from Figures 5.3, 5.4, and 5.5 is that the proposed procedure in Section 5, i.e., comparing the efficiency with a sufficiently large n_f , provides results that are in line with the results when the comparison is done with the full data. This provides evidence that this procedure can be trusted to detect the FIL. Table 5.2 shows the computation time using only a tiny sub-sample for different combinations of n_f and M , next to the time for analyzing the full data. As one may see, depending on the selected n_f and M the computation could become between 500

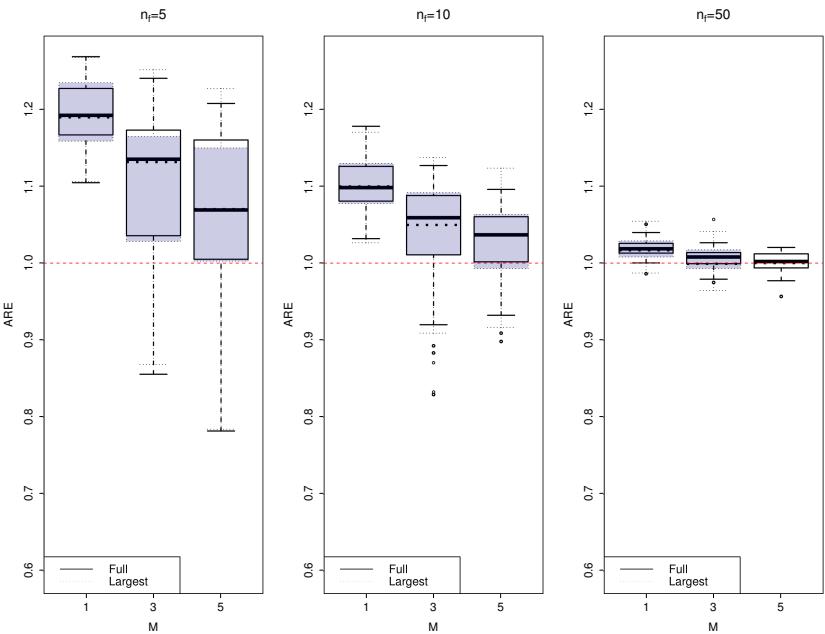


Figure 5.4: Comparing the ARE of full data (the boxplot with background color and solid lines) and sub-sampling of size $n_f = 50$, $M = 5$ times (the box plots with transparent color and dotted lines) with different combinations of n_f and M for $\rho = 0.5$. The dashed horizontal line shows $ARE=1$.

to 1700 times faster. We may remark that all the computations are done on a single core in a laptop. Of course, distributing the embarrassingly parallel nature of this methodology among different machines could possibly boost the computation more than this. Finally, as the FIL is detected using $\rho = 0.2, 0.5, 0.8$, following Step 4 in the FIL detection procedure, one can improve the estimation by combining the results from $M = 1, 3, 5$ using weights proportional to the corresponding n_f . Figure 5.6 shows the results of ARE's after taking the weighted average. A further ARE improvement is

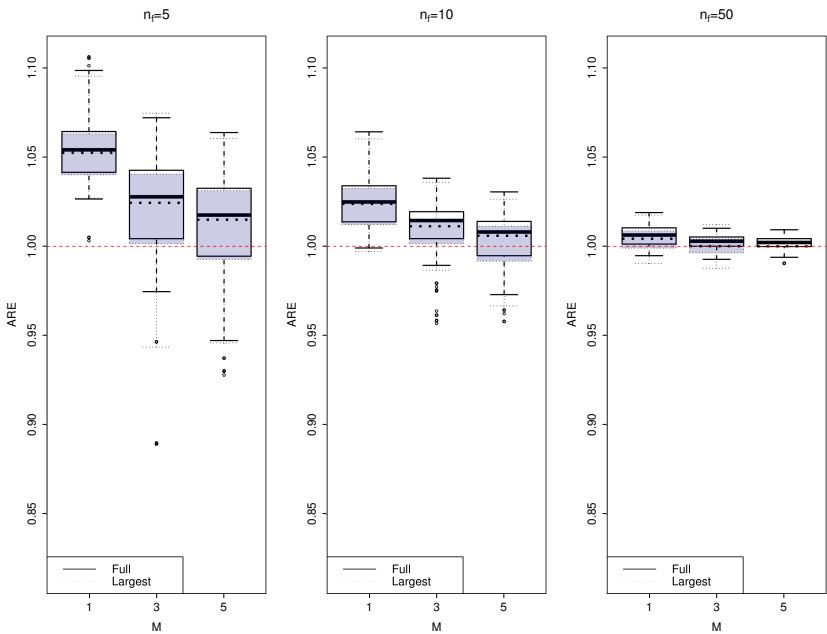


Figure 5.5: Comparing the ARE of full data (the boxplot with background color and solid lines) and sub-sampling of size $n_f = 50$, $M = 5$ times (the box plots with transparent color and dotted lines) with different combinations of n_f and M for $\rho = 0.8$. The dashed horizontal line shows $ARE = 1$.

observed, compared with Figure 5.2, where for $\rho = 0.2$ the upper bound of the ARE is around 2, while after combining the multiple sub-samples, it decreases to around 1.2; the corresponding decrease for $\rho = 0.5$ is from 1.25 to 1.06, and still for $\rho = 0.9$, from 1.10 to 1.02.

Table 5.2: Computation time (in seconds) for different combinations of n_f and M and the full data. The results from different ρ 's are aggregated for each combination of n_f and M .

M	n_f	Computation time	
		Mean	SD
1	5	0.704	0.059
	10	0.671	0.034
	50	0.710	0.043
	5	1.171	0.058
3	10	1.158	0.056
	50	1.311	0.066
	5	1.613	0.075
5	10	1.638	0.089
	50	1.981	0.094
Full sample		1133.973	153.387

Case study: Amazon.com book ratings

The effectiveness of detecting FIL is examined via simulations in Sections 5 and 5. In this section, we consider the ratings of the books on the Amazon website. The data are obtained from ?. The dataset contains 2,330,066 books, which are rated at least 1 and at most 21,398 times. Obviously, this is an example of clustered data, as the same book is rated many times. Figure 5.7 shows the histogram of the number of all and unique cluster sizes.

Fitting the CS model to the full data is not feasible; attempting it in Proc MIXED in SAS would lead to the same error as in the Table 5.1. Therefore, detecting FIL and applying it would make it possible to fit a model to this data. As a CS model is used, following the discussions in Section 5, one may use Fieller's upper-bound of
Page 147

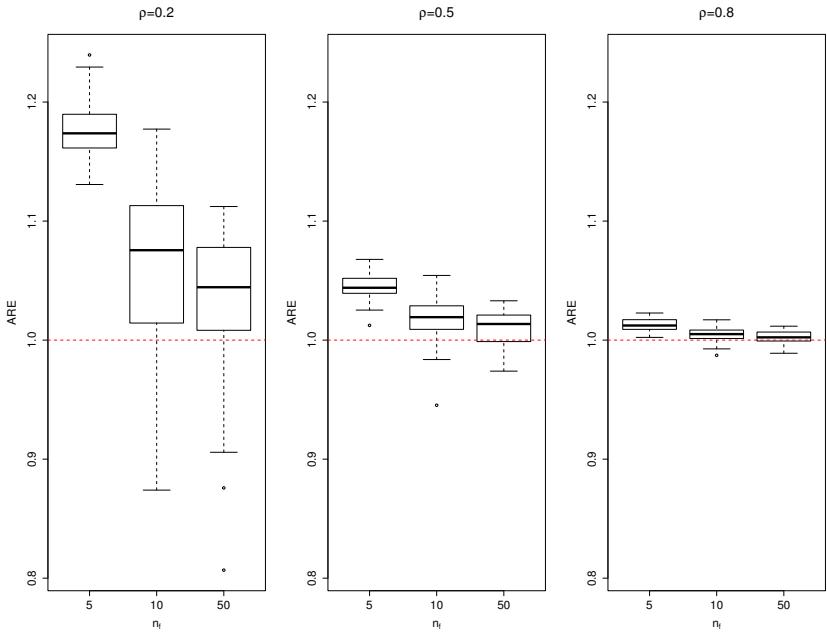


Figure 5.6: Comparing the ARE of full data with sub-samples of sizes $n_f = 5, 10, 50$ when the results from $M = 1, 3, 5$ are combined using the weights $w_i = (5/65, 10/65, 50/65)$.

(5.9) to determine the sub-sampling size for a pre-specified ε . To estimate the parameters, a sub-sample of size $n_1 = 5$ is used. Figure 5.8 shows the estimated Fieller's upper-bound for n_f for several ε 's using $\alpha = 0.05$. As one may see, $\varepsilon \approx 0.09$ can be obtained by a sub-sample of size almost 50. In order to make sure whether $n_f = 50$ is good enough, a sub-sample of size $n_f = 100$ is also taken for the sake of comparison.

The results of model fitting using $n_f = 5, 50$, and 100 are given in Table 5.3. As one may see for estimating μ , the efficiency using

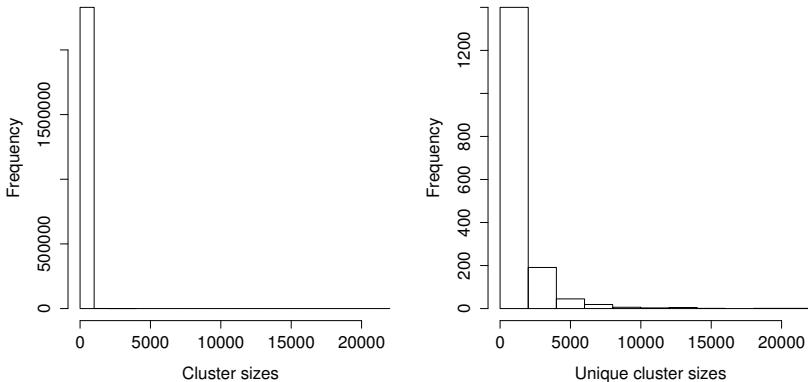


Figure 5.7: Histogram of number of all (left) and unique (right) cluster sizes in Amazon books ratings data.

$n_f = 50$ comparing with $n_f = 100$ does not change substantially (efficiency is equal to 0.99699). Therefore, one may conclude the FIL property applies and using a sub-set of size $n_f = 50$ or 100 one may be almost as efficient as using the full data. So, while fitting the model to the full data is not feasible, we could obtain almost fully efficient estimates using only a sub-set of the data.

As one may see in Table 5.3, $\hat{\rho} = 0.17$ and is significant. Another interesting point about these results is that the average rating is estimated as 4.3. Taking this into account that the maximum possible rating is 5, this shows those people who like a book tend to rate it. That would lead to an important conclusion: no matter how large the data set is, an inappropriate model may lead to biased

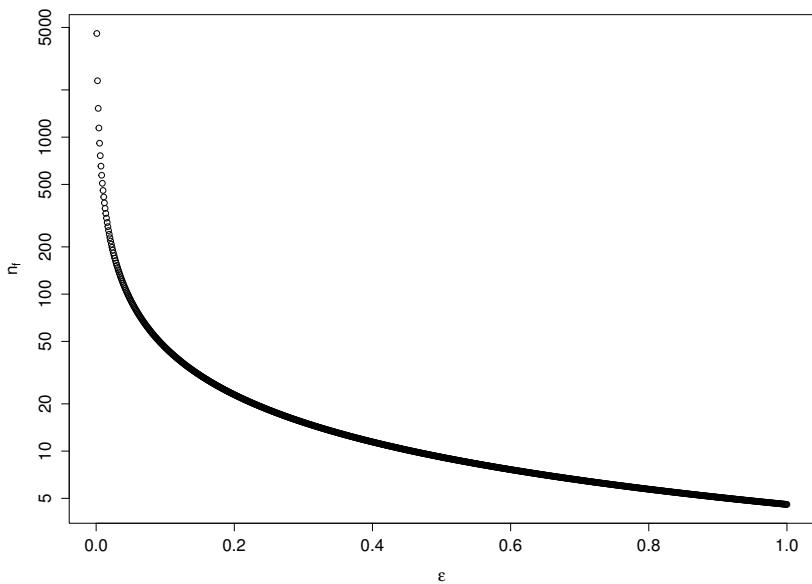


Figure 5.8: Amazon books ratings data. Fieller's upper-bound of n_f (log-scale) for different values of ε using $\alpha = 0.05$. The parameters are estimated using a sub-sample of size $n_1 = 5$.

estimates, i.e., the majority of the respondents are among the people who have liked the book, so it would not be a representative sample. This matter needs to be foreseen in the model.

It is worth noting that, as in any model, some assumptions are made here. The two main assumptions considered in this section are normality of the ratings and the CS structure for their covariance matrix. We admit that extensions would be needed for other classes of models, but normality if not crucial for the current setting, as inferences are robust with respect to deviations from normality,

Table 5.3: Estimating parameter vector (μ, τ, σ^2) and its standard error using $n_f = 5, 50, 100$.

n_f	μ		τ		σ^2	
	Estimate	StdErr	Estimate	StdErr	Estimate	StdErr
5	4.30390	0.00053	0.22737	0.00061	1.03355	0.00069
50	4.30505	0.00048	0.21496	0.00046	1.03127	0.00040
100	4.30500	0.00048	0.21435	0.00046	1.03285	0.00037

especially in the linear model. The CS structure assumption states that two observations can be exchanged, i.e., that observers have similar characteristics. If not, extensions with covariates might be of interest (to correct for such differences), or one may consider separate analyses for different groups of observers.

Conclusion

Inspired by compound-symmetry models for clustered data, the finite information limit (FIL) property for estimators based on hierarchical data is proposed and investigated both theoretically and via simulations. Loosely speaking, FIL means that for increasing cluster sizes, the amount of information is bounded from above. This implies that, after some point, adding new measurements would not substantially add to the information provided by the data, hence the efficiency would not importantly increase further. A consequence of this property is that analyzing a small, or sometimes even tiny, fraction of a dataset with FIL property (e.g., 1%–2%) would produce results almost as efficient as using the entire dataset, see Figures 5.2–5.6. This property is investigated from a theoretical

point of view and a simple procedure is proposed to detect the property in a dataset. This is not needed if simple models known to have the FIL property, such as CS, are used. However, with complex variance-covariance structures, such an empirical check can be handy. The proposed procedure together with the FIL itself were investigated in simulation studies. Furthermore, an efficient sub-sampling procedure is implemented in R and available in the Appendix.

Investigating the FIL property on the Amazon.com book ratings enabled us to fit a model which was not feasible for the full data. Based on the obtained results, the efficiency loss was not substantial neither. Such examples would show the importance of properties like FIL to deal with large clustered data, specially when an analysis is needed on a daily basis.

The computation time gain can be enormous, see Table 5.2. To add to the computational efficiency, unlike in many raw dataset, the sub-sample may be chosen balanced, which adds to the computational efficiency ?). As proposed in Section 5, the sub-sampling can be repeated several times to increase efficiency. In addition to the advantages mentioned, the proposed procedure based on the sub-sampling of various sizes several times consists of so-called “embarrassingly parallel steps.” Therefore, it can be implemented as fast as the computational resources allow. Even on a single core laptop scale, the computation gain could become 1500 times faster than using the full data. One may note that fitting the model to

the full (unbalanced) data is usually and out of necessity done using iterative algorithms, which are very complex to make parallel.

Chapter 6

Structured Vertical Splitting

In this chapter we review the structured vertical splitting with its application in pairwise modeling of several responses simultaneously.

Fast precision estimation in high-dimensional multivariate joint models

Introduction

There are different approaches to deal with modelling multivariate clustered data. Authors like ? proposed to analyze each response conditioned on all the others. As an alternative, one may specify a marginal model for each response separately and then join them using an appropriate way. Among such ways are using a latent

variable (?), copulas (?), or random effects (?). The latter is of interest in this short note.

Let y_{rij} be the j th measurement taken on the i th cluster for the r th outcome, $i = 1, \dots, N$, $r = 1, \dots, m$ and $j = 1, \dots, n_{ri}$. Obviously, the number of measurements available for subject i is $\sum_{r=1}^m n_{ri}$. Note that the various outcomes do not need to be of the same kind (continuous, binary, count data, etc.). Also there is no need to have sequences of the same length for each cluster or each outcome. Let $\mathbf{Y}_{\mathbf{ri}}$ be the n_{ri} measurements taken on cluster i for outcome r . The model assumes that each \mathbf{Y}_{ij} satisfies a mixed model:

$$\mathbf{Y}_{\mathbf{ri}} = \boldsymbol{\mu}_{\mathbf{ri}} + \boldsymbol{\epsilon}_{\mathbf{ri}}, \quad (6.1)$$

where

$$\boldsymbol{\mu}_{\mathbf{ri}} = \mathbf{h}(X_{ri}\boldsymbol{\beta}_r + Z_{ri}\mathbf{b}_{\mathbf{ri}}). \quad (6.2)$$

As one may see in (6.2), each outcome can have its own set of fixed effects. In order to capture the associations between different responses, the mixed model approach for joint modelling of the m outcomes allows for correlation between the random effects of each outcome. Let $\mathbf{b}_i = (\mathbf{b}_{1i}, \dots, \mathbf{b}_{mi})'$; the joint model assumes:

$$\mathbf{b}_i = \begin{bmatrix} b_{1i} \\ b_{2i} \\ \vdots \\ b_{mi} \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, D \right), \quad D = \begin{bmatrix} D_{11} & D_{12} & \dots & D_{1m} \\ D_{21} & D_{22} & \dots & D_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ D_{m1} & D_{r2} & \dots & D_{mm} \end{bmatrix}. \quad (6.3)$$

The matrix D_{rs} ($r, s = 1, \dots, m$) represents the covariance between

\mathbf{b}_{ri} and \mathbf{b}_{si} .

While using the joint modelling approach for the multivariate clustered data is a flexible methodology, it can easily become intractable when the number of outcomes becomes large. As the matrix D contains the association between all outcome sequences, its dimension can quickly grow large. Therefore, it is easily possible that such a flexible joint model cannot be used in practice.

In order to overcome this drawback, ? and ? proposed a pseudo-likelihood based approach which considers all possible pairs (or higher order subsets) of the r outcomes and fit them separately. Obviously, using this methodology some parameters are estimated more than once. Averaging multiply estimated parameters gives a final estimate for each parameter. To find a correct estimate of the variance, ? and ? proposed a sandwich-type correction of the form $J^{-1}KJ^{-1}$, where J and K are formed based on cluster-wise Hessians and gradients of the log-pseudo-likelihood function, respectively.

The need for cluster-wise derivatives of the first and second order makes computing this sandwich estimator difficult and possibly expensive. ? and ? provided **SAS** implementations for the case of linear mixed models. Also, ? considered the case of generalized linear models using **Proc NL MIXED** in **SAS**. However, the cluster-wise calculation can become very expensive for large samples. Also, the focus of these implementations is on the fixed effects only. Implementing the variance estimation for, e.g., matrix D (6.3) is far

more complicated. Also, obtaining the right precision quantities in commonly available software like R demands new implementations. Furthermore, a different model needs its own implementation which includes evaluating gradients and Hessians of the log-(pseudo-)likelihood function.

? proposed within-cluster re-sampling to analyze clustered data when the intra-class correlation is considered a nuisance. Their idea was explored in ? who gave it the name multiple outputation (MO). MO consists of repeatedly taking sub-samples of size 1 from each cluster to form a sub-sample of independent observations, to then apply the standard methodologies for such data. The estimated parameters and their precisions from several sub-samples will be combined using appropriate rules. ? have shown that their variance estimator is consistent.

Note that using pairs of variables instead of analyzing them jointly is a trick which is not restricted to likelihood-based methods. It has been used in different contexts. For example, ? have used pair-copulas to model complex multivariate dependencies. Especially with Gaussian outcomes, pairwise approaches (e.g., vine copulas) allow retrieval of the full multivariate association structure, without the need for full joint modelling.

In this short note we extend the idea of MO to compute the variance of a pairwise-fitted multivariate joint model as a faster and easier way. The MO-based correction for the variance will be presented in Section 6. A simulation study will compare its performance with a

sandwich correction in Section 6. The two corrections are compared in analyzing a real dataset in Section 6, and finally the paper is concluded in Section 6.

A fast variance estimator based on multiple output- ation

Consider $y_{rij}^{(s)}$ as the j th measurement on i th cluster for r th outcome in the s th sub-sample, where $s = 1, \dots, M$, $j = 1, \dots, n_{ri}$, $i \in \{1, \dots, N\}$, and $r \in \{1, \dots, m\}$. Also consider $\boldsymbol{\theta}^{(s)}$ a vector of size p_s ($s = 1, \dots, M$), containing parameters estimated using the variables involved in the s th sub-sample, obviously, $\boldsymbol{\theta}^{(s)} \subseteq \boldsymbol{\theta}$. Let $\boldsymbol{\Theta}' = (\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(M)})$ the stacked vector combining all parameters estimated in different sub-samples. $\hat{\boldsymbol{\Theta}}$ replaces each $\boldsymbol{\theta}^{(s)}$ by its estimate. Also, $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$ is a vector of length $\sum_{s=1}^M \times p_s$, with its elements the variance of each element of $\hat{\boldsymbol{\Theta}}$.

The MO in the current paper extends the one in ? and ? in two main directions. First, the sub-samples in our approach are of size larger than one. The second aspect of our proposed extension of MO concerns the sub-sampling scheme. The MO in the sense of ? and ? takes random sub-samples of size 1 from each cluster many times to form sub-samples of independent observations. In our approach, the sub-samples are not taken at random but based on a pre-defined structure which implies a fixed number of needed sub-samples. One may note that the sub-sampling in our proposal is performed at the response level. Therefore, as it is not necessary to measure every response for all subjects, each subject will be

presented in the sub-samples which the responses presented there are measured for it.

For example, using a pairwise approach, each sub-sample consists of all the observations available for each subject corresponding to the two outcomes in the pair under analysis. Therefore, having m outcomes, we need to take $M = m(m - 1)/2$ sub-samples, each with a size larger than one. Note that, in a pairwise approach, $M = m(m - 1)/2$ is the number of all possible sub-samples. Taking less sub-samples would fail to estimate some parameters. On other hand, taking more sub-samples is not possible without repeating some already taken sub-samples. As it was remarked, it is not necessary for every subject to be present in every sub-sample. In a pairwise approach, suppose we only have three responses y_1, y_2, y_3 and for a particular subject only y_1 is measured. Then this subject will only be present in pairs (y_1, y_2) and (y_1, y_3) .

As one may have already noticed, some of the parameters in $\boldsymbol{\theta}$ would have more than one counterpart in $\boldsymbol{\Theta}$. In order to bring the $\boldsymbol{\Theta}$ back to the desired parameter space, one may take the average of possible several counterparts of each element of $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$. This can be done using an appropriate weight matrix pre-multiplied by $\boldsymbol{\Theta}$. Consider A , a $(p \times \sum_{s=1}^M p_s)$ matrix where its (ℓ, k) th element, $\ell = 1, \dots, p$, $k = 1, \dots, (p \times \sum_{s=1}^M p_s)$, is defined as follows:

$$A_{\ell,k} = \begin{cases} 1 & \text{if } \boldsymbol{\Theta}_k \text{ is a component of } \boldsymbol{\theta}_{\ell}, \\ 0 & \text{otherwise,} \end{cases} \quad (6.4)$$

where Θ_k and θ_ℓ are the k th and ℓ th elements in Θ and θ , respectively. If W is defined as A but with the rows divided by their sum then the combination rules for combining the estimates from M sub-samples are as follows,

$$\begin{cases} \tilde{\theta} = W\hat{\Theta}, \\ \text{diag}\{\text{Var}(\tilde{\theta})\} = W\Sigma_{\Theta} - W\text{diag}\left\{\left(\hat{\Theta} - \bar{\Theta}\right)\left(\hat{\Theta} - \bar{\Theta}\right)'\right\}, \end{cases} \quad (6.5)$$

where A' is the transpose of A , and $\bar{\Theta} = \hat{\Theta} - A'\tilde{\theta}$. For the proof, see ?, Appendix 2. These authors have shown that $I^{1/2}(\tilde{\theta} - \theta) \xrightarrow{d} N(0, \Sigma)$, where \xrightarrow{d} means convergence in distribution and $\text{Var}(\tilde{\theta})$ is a consistent estimator of Σ .

We may note that the combination rules in (6.5) are only estimating the diagonal elements of the covariance matrix of $\tilde{\theta}$. If one is interested in estimating the off-diagonal elements as well, the combination rules are the same. One would only need to construct Σ_{Θ} , which takes into account the off-diagonal elements also, then an appropriate W can be used to combine the estimated variances and covariances obtained from each sub-sample. The corresponding vector of unique parameters in this covariance matrix would be $p(p + 1)/2$, where each sub-sample would estimate part of them.

As one may see in (6.5), the combination rule for the variance has a subtraction, so as pointed out by both ? and ?, theoretically, it is possible to estimate a non-positive definite covariance matrix.

Although the chance for such an occurrence is small, and we have not observed it even once in our simulations, it is still necessary to discuss this matter. We can consider three situations in which such a phenomenon would occur: (1) the sample size, N , is not sufficiently large, (2) the number of sub-samples, M , is not sufficiently large, and (3) the size of each sub-sample is not sufficiently large.

? proposed to increase the number of sub-samples, M , to solve this problem. As it was pointed out, unlike the usual MO, in our approach the number of sub-samples is fixed. Therefore, in case of a non-positive definite covariance estimate, we suggest to take larger sub-samples each time. Note that taking such a strategy is not possible in the usual MO, since it takes one observation per cluster, so the sub-sampling size is fixed. Taking larger sub-samples in our case means to consider triples, quadruples, etc., instead of pairs. We expect the problem of non-positive definite covariance estimate to vanish when taking larger sub-samples.

When maximum likelihood estimator (MLE) is used, ? proposed a variance estimator based on MLE approximation, using first order Taylor expansion of the score function. This approach requires the evaluation of the subject-wise gradient and Hessian of the log-likelihood function, which is similar to the requirements of the sandwich estimator. Therefore, in case where even with larger sub-samples a non-positive definite covariance estimate persists, we suggest to use the sandwich estimator instead of MO.

Simulations

In this section the proposed idea will be explored and examined via two simulations studies. Sub-section 6 considers joint modelling of linear mixed models, and Sub-section 6 takes a joint model of ordinal data in a generalized linear mixed models context.

Gaussian responses

Consider the linear case with identity link in (6.2), as follows:

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \epsilon_i, \quad (6.6)$$

where \mathbf{X}_i is the design matrix regarding fixed effects, β represents the fixed effects, \mathbf{Z}_i is the design matrix related to the random effects, and $\mathbf{b} \sim N(0, D)$ shows the random effects. The errors are represented by $\epsilon_i \sim N(0, R)$. In order to evaluate the proposed method in a simulation study, consider the model in (6.6) with parameters:

$$\beta = (5, 5, 5, 5), D = \begin{pmatrix} 60 & 40 & 25 & 40 \\ & 50 & 35 & 30 \\ & & 45 & 40 \\ & & & 55 \end{pmatrix}, R = 196I_{n_i}, \quad (6.7)$$

where I_{n_i} is the identity matrix of order n_i , with n_i the length of the i -th cluster. This means that both of the \mathbf{X}_i and \mathbf{Z}_i are formed by columns of 1's. The data are generated from such model

100 times, each consists of 100 clusters of size $n_i = 5$. The mixed model is fitted to the full sample as well as each of the 6 pairs using Proc MIXED in SAS. Then, the estimates and variances are combined using (6.5). Also, the sandwich estimator is calculated using %jointpair macro in SAS in ?.

Figure 6.1 shows the simulation results. As one may see, the relative efficiency of estimating fixed effects using sandwich and MO corrections are very similar. Furthermore, the relative efficiency for estimating parameters in the D matrix are also presented. These parameters are only estimated using the MO correction, as the implementation was not available for sandwich-correction. The interesting result here is the computation time. As one may see, using MO-correction the computation can be done more than 2500 times faster. On the other hand, in case that computing the full likelihood model is possible (as it is the case in our simulation study), MO-correction can also possibly be computed faster than the full-likelihood. Therefore, it is not only an alternative for infeasible likelihoods, but also, it can be used to accelerate computations that are strictly speaking doable.

In order to see how our proposal performs comparing with sandwich estimator when the correlation between the random effects is small, the simulation above is repeated, but this time the off-diagonal elements of D matrix in (6.7) are divided by 10. Figure 6.2 shows the relative efficiencies of the variances obtained using MO and sandwich corrections. As one may see, these relative efficiencies are

again very close to 1.

Non-Gaussian responses

In order to investigate the performance of our proposal for non-Gaussian responses and more complex models a simulation is done based on the Leuven diabetes project. This dataset is analyzed in ?, details about this study can be found there. Here we only review parts necessary for our simulations. In order to study how well the diabetes is controlled, three outcomes were measured for each patient at baseline (T_0) and one year later (T_1). These three outcomes of interest were LDL-cholesterol (low-density lipoprotein cholesterol, mg/dl), HbA1c (glycosylated hemoglobin, %), and SBP (systolic blood pressure, mmHg). Although these three variables are continuous, the interest was in expert-specified cut-off values of these outcomes. The cut-off values are defined as $\{< 100, [100, 115), [115, 130), \geq 130\}$ (mg/dl) for LDL-cholesterol, $\{< 7, [7, 8), \geq 8\}\%$ for HbA1c, and $\{\leq 130, (30, 140], (140, 160], > 160\}$ mmHg for SBP.

In order to analyze these 3 ordinal outcomes together, a joint modelling approach with a proportional odds mixed model (POMM) for each response was considered in ?. The POMM for each response is as follows:

$$\text{logit}[P(Y_{rij} < k)] = \xi_{r0k} + \xi_{r1}t_{ij} + \xi_{r2}X_i + b_{ri} + \epsilon_{rij}, \quad (6.8)$$

where $r = 1, 2, 3$ indicates the outcome of interest, and $k =$
Page 165

$1, \dots, K - 1$ with K as the number of levels of the corresponding outcome of interest. Furthermore, X_i shows the gender of the subject, and b_{ri} 's are the random intercepts corresponding to each outcome. Following the mixed model approach for joint modelling of multiple outcomes, we assume the three random intercepts to be normally distributed with the following covariance matrix.

$$\begin{bmatrix} b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ & D_{22} & D_{23} \\ & & D_{33} \end{bmatrix} \right). \quad (6.9)$$

The parameter values for generating the data were obtained by fitting the joint model to the Leuven diabetes dataset. The data were generated 200 times using **SAS Proc IML**. Each time the parameters in the 3 pairs were estimated using adaptive Gaussian quadrature (?) with 3 quadrature points in **SAS Proc NL MIXED** and the combined variance was estimated using sandwich and MO corrections. Considering the complexity of the model, in order to do the simulation quicker, the codes were run on VSC cluster in Leuven, Belgium.

The results of the simulations are summarized in Table 6.1. Column 3 of Table 6.1 shows the parameter values which were used to generate the data. The last three columns show the means and standard deviations over 200 replications for estimated variance using MO, sandwich, and their relative efficiency, respectively. As one may see, for all of the variables the estimated variances are very close, therefore, one can conclude that our proposal performs

well also in a complex model with non-Gaussian responses.

Table 6.1: Simulations (non-Gaussian outcomes). The estimated model for three ordinal outcomes. The first columns indicates different outcomes, and the second column shows the estimated effect. The data obtained parameter-values which were used to generate the data are presented in column 3. Columns four and five show the average (standard deviation) of standard error using MO and sandwich, obtained over 200 replications, respectively. The last column shows the relative efficiency of MO compared to sandwich: $\text{Var}(\hat{\theta}_{\text{MO}})/\text{Var}(\tilde{\theta}_{\text{Sandwich}})$.

Outcome	Effect	Parameter	MO	Sandiwh	ARE (MO/Sandwich)
LDL-chol.	Int. 1	$\xi_{101} = -0.740$	0.039 (0.006)	0.039 (0.006)	0.993 (0.009)
	Int. 2	$\xi_{102} = 0.480$	0.038 (0.006)	0.038 (0.006)	0.998 (0.009)
	Int. 3	$\xi_{103} = 1.580$	0.042 (0.006)	0.041 (0.007)	1.012 (0.010)
	Time	$\xi_{11} = 1.050$	0.032 (0.004)	0.032 (0.004)	1.024 (0.008)
	Gender	$\xi_{12} = 0.450$	0.048 (0.008)	0.048 (0.008)	0.989 (0.001)
	RI sd	$\sqrt{D_{11}} = 1.880$	0.036 (0.005)	0.034 (0.005)	1.105 (0.010)
HbA1c	Int. 1	$\xi_{201} = 270$	0.042 (0.007)	0.042 (0.007)	0.997 (0.008)
	Int. 2	$\xi_{202} = 2.610$	0.052 (0.009)	0.051 (0.009)	1.043 (0.010)
	Time	$\xi_{21} = 1.040$	0.036 (0.004)	0.035 (0.005)	1.034 (0.010)
	Gender	$\xi_{22} = -0.090$	0.054 (0.009)	0.054 (0.009)	0.979 (0.001)
	RI sd	$\sqrt{D_{22}} = 2.150$	0.042 (0.007)	0.040 (0.007)	1.131 (0.012)
	Int. 1	$\xi_{301} = -0.190$	0.035 (0.006)	0.035 (0.006)	0.995 (0.009)
SBP	Int. 2	$\xi_{302} = 1.380$	0.038 (0.007)	0.038 (0.007)	1.012 (0.009)
	Int. 3	$\xi_{303} = 3.700$	0.056 (0.009)	0.055 (0.010)	1.028 (0.015)
	Time	$\xi_{31} = 0.510$	0.030 (0.003)	0.030 (0.003)	1.013 (0.005)
	Gender	$\xi_{32} = 0.190$	0.044 (0.008)	0.044 (0.008)	0.991 (0.001)
	RI sd	$\sqrt{D_{32}} = 1.620$	0.034 (0.004)	0.033 (0.005)	1.073 (0.015)
LDL-col. vs. HbA1c	Cov RI's	$D_{12} = 0.808$	0.067 (0.013)	0.068 (0.014)	0.961 (0.014)
LDL-col. vs. SPB	Cov RI's	$D_{13} = 1.218$	0.057 (0.012)	0.057 (0.012)	0.994 (0.015)
HbA1c vs. SBP	Cov RI's	$D_{23} = 1.393$	0.065 (0.014)	0.065 (0.014)	0.993 (0.016)

Application

To evaluate the hearing performance of a subject, hearing threshold sound pressure levels (measured in dB) are considered. By definition, a hearing threshold is the lowest signal intensity possible to perceive by a subject at a specific frequency. In a study considered in ? and ?, hearing threshold measured in 11 different frequencies

between 125 Hz and 8000 Hz for left and right ears obtained on 603 male participants from the Baltimore Longitudinal Study of Ageing, BLSA (?). The number of visits for each subject which varies between 1 to 15 are unequally spaced. In this illustration all subjects with missing values are ignored to make a fair comparison between MO and sandwich corrections.

In order to model different frequencies simultaneously, ? used a joint model approach. Because of several outcomes of interest, they have proposed a pairwise approach to jointly fit the model to the data. In this section we use this dataset to illustrate our proposal and compare it with the sandwich correction proposed in ?. Consider $y_i(t)$ as the hearing threshold for subject i at time t for some frequency. The mixed model fitted to this single response is as follows:

$$y_i(t) = (\beta_1 + \beta_2 Age_i + \beta_3 Age_i^2 + a_i) + (\beta_4 + \beta_5 Age_i + b_i)t + \beta_6 V(t) + \epsilon_i(t), \quad (6.10)$$

where t represents time and $V(t)$ is a dummy variable indicating whether the measurement is taken at the first time visit to account for systematically higher response values at the first visit. Also, Age indicates the age of the participant at first visit. Furthermore, a_i and b_i are the random intercepts and slopes, respectively; and ϵ is the error term. Using a joint modelling approach to model the hearing threshold at several frequencies one needs to let the random intercepts and slopes from various models to be jointly distributed.

Here a multivariate normal distribution is considered.

For the sake of our illustration, we consider two models: once three outcomes are modelled simultaneously (frequencies 125 Hz, 250Hz, and 500 Hz for the right ear) and once 5 outcomes are considered (frequencies 125 Hz, 250 Hz, 500 Hz, 750 Hz, and 1000 Hz for the right ear). A joint model of m outcomes using (6.10) for each response will have $6m$ fixed effects to estimate, as well as $2k(2k + 1)/2$ parameters in D . For $k = 3$ this is 39 parameters (on top of the parameters in the error matrix). Taking $k = 5$ the number of parameters in fixed effects and D matrix is 85. A pairwise approach is used to estimate the parameters of such models. The precision of the parameter estimates are computed using MO and sandwich corrections. Whenever possible ($k = 3$) the model is fitted directly using the full likelihood.

Figure 6.3 shows the ARE for estimating fixed effects using MO and the sandwich corrections. The left panel of this figure shows the results when 3 responses are modelled jointly. In this case, estimating the parameters using the full likelihood was also possible, so the ARE's are computed relative to the variance obtained from fitting the model using full likelihood. As one may see in Figure 6.3 (left), the MO correction gives slightly better results compared to the sandwich. Fitting the full likelihood to the 5-response model was not feasible. Therefore, the precision was only estimated using MO and sandwich corrections. Figure 6.3 (right) shows the ARE (MO divided by sandwich) for the 30 fixed effects of a 5-response

model. As one may see, the ARE's are around 1; nevertheless, MO still gives slightly more precise estimates for most of the parameters.

While both MO and sandwich corrections give similar precision estimates, their computation time differs dramatically. Fitting a 3-response model using the full likelihood takes only 38.12 seconds. Fitting such a model with a pairwise approach (it needs fitting 3 pairs) with sandwich correction takes 2.29 hours (216.66 times more than the full likelihood) using ?. The computation time using the MO correction is only 25.34 seconds (0.66 times the full likelihood computation time).

When considering 5 responses, fitting the full likelihood in Proc MIXED in SAS give **ERROR: The SAS System stopped processing this step because of insufficient memory**. Using a pairwise approach (with 10 pairs) and sandwich correction the computation time is 7.98 hours, while the MO correction takes only 1.45 minutes which is 331.03 faster comparing with sandwich correction implemented in ?.

We may note that the implementation in ? is not fully efficient. Also, as fitting different pairs is a so-called embarrassingly parallel task, using parallel computation is straightforward. So a more efficient implementation of sandwich correction will become faster (? , as an example). But the same discussion goes for the MO correction. Using parallel computation would also make this approach considerably faster. The interesting point is that on a single-core laptop scale, the MO correction is fast enough, also straightforward to

implement, which would avoid the need of using cluster computing, difficult implementations, etc.

Note that there could be other approaches to deal with this problem. In some cases, smoothing splines have been considered in the context of mixed models. For example, ? and ? consider random smoothing splines for the random-effects structure, and couple this with efficient computation. See also ?, pp. 381-384. Evidently, smoothing splines can also be used to specify the mean structure. Other useful tools are fractional polynomials, which can be used in the mean function, the residual variance function, etc. Examples of both of these uses can be found in ?, pp. 478-479 and pp. 137-139, respectively.

Conclusions

In this short note the idea of multiple ouputation was extended to provide a fast precision estimation in high-dimensional multivariate joint models. Such precision estimation can be used in the pairwise approach of ?? to replace the timely expensive and complicate to compute and implement sandwich estimator. While both MO and sandwich corrections give asymptotically the same preicisons, the main advantages of our MO-based proposal compared with sandwich correction are as follows:

1. MO correction is considerably faster, even on a laptop scale using a single core. In our simulations (Section 6) the time gain was more than 2500 times and in real data analysis (Section 6) it was about 350 times. As our simulations and

real data analysis results show, MO correction can possibly be computed even faster than the full likelihood (when feasible).

2. The Mo correction can be computed using the usual outcomes of any standard software. i.e., estimates and their precision. Unlike the sandwich correction which needs the first and second order derivatives. Therefore, MO correction is straightforward to implement with no need of complicated computations in any one of the usual software packages. This property is true for fixed effects, variance components, or other types of parameters of interest. Computing the derivatives of log-(pseudo-)likelihood function can possibly more difficult e.g. for variance components comparing with the fixed effects.
3. Using the MO correction for estimating the precision in a pairwise approach (??) each pair can be treated completely independent from the others. Therefore, if one cluster contains no missing values for one pair it will be used to estimate the parameters of that specific pairs. This is not true for the sandwich correction. The reason is the need to compute cluster-wise gradients and Hessians.

Considering the above advantages of MO correction over sandwich correction, it can be considered as an effective and efficient alternative for precision estimation when dealing with high-dimensional multivariate joint models.

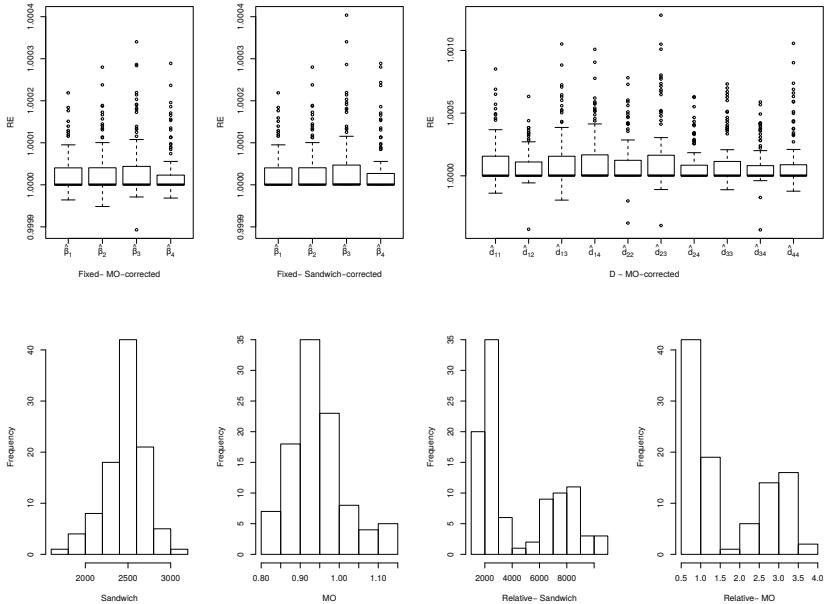


Figure 6.1: Simulations (Gaussian outcomes). Top: Relative efficiency of estimated fixed effects using MO-correction (left) and sandwich-correction (middle), estimated D -matrix using MO-correction (right) compared to the full model. The relative efficiency is defined as the ratio of the variance of the alternative method and the full likelihood, e.g., for MO $\text{Var}(\hat{\theta}_{\text{MO}})/\text{Var}(\hat{\theta}_{\text{full}})$ is computed. Bottom: Histograms of computation time (in seconds) using sandwich and MO corrections (first and second). Histograms of computation time (in seconds) using sandwich and MO corrections (third and fourth) relative to the full model computation time.

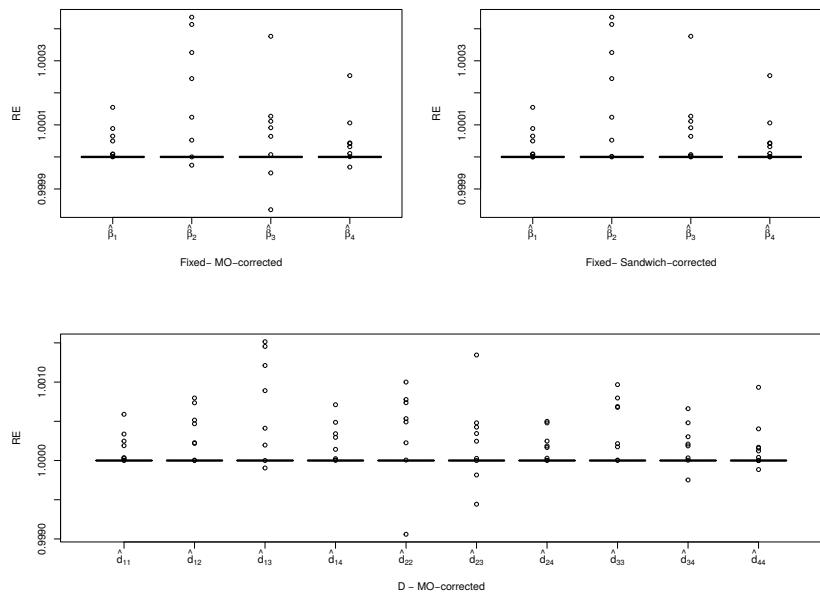


Figure 6.2: Simulations (Gaussian outcomes). Top: Relative efficiency of estimated fixed effects using MO-correction (left) and sandwich-correction (right). Bottom: estimated D -matrix using MO-correction compared to the full model. The relative efficiency is defined as the ratio of the variance of the alternative method and the full likelihood, e.g., for MO $\text{Var}(\hat{\theta}_{\text{MO}})/\text{Var}(\hat{\theta}_{\text{full}})$ is computed.

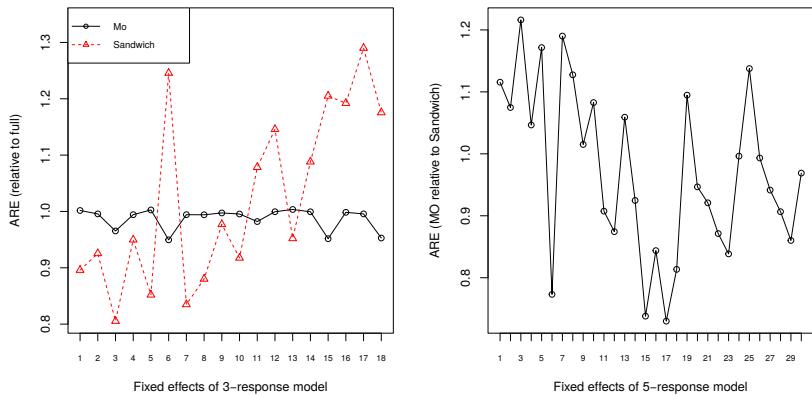


Figure 6.3: Application. Left: comparing the ARE (relative to the variance using full likelihood) of estimated fixed effects of MO and sandwich corrections in a model with three responses. Right: the $ARE = \text{Var}(\text{MO})/\text{Var}(\text{sandwich})$ of estimated fixed effects in a model with 5 responses. The x-axes show the indexes of the fixed effects (18 parameters for 3-response model and 30 parameters for 5-response model.)

Chapter 7

Supplementary topics

We have discussed the several similarities of multiple imputation (??) and multiple outputation (??). In this chapter we present two of our contributions that consider determining number of imputed datasets (equivalently number of outputed datasets), and performing multiple imputation (equivalently multiple outputation) in a case where the combination rule for estimates cannot be used in a direct way.

Iterative Multiple Imputation: A Framework to Determine the Number of Imputed Datasets

Introduction

Since Rubin's seminal work on multiple imputation (MI) (???), the method has been broadly applied, methodologically extended, and expanded towards ever more areas of application (????). Multiple imputation is a commonly used approach to analyze incomplete data. A growing literature and increasing number of software implementations have contributed to the spread of the method. One of the attractions of MI is its very good to excellent performance even with a relatively small number of imputed datasets. This was important when Rubin created the method, about forty years ago, in view of, among others, the US Census. It still is today because of ever increasing data streams.

Broadly, MI replaces a missing value with several plausible values, sampled from an appropriate predictive distribution for the missing values, given observed information. The method produces several completed datasets to replace the initial partially observed dataset.

An evident practical question is how many imputed datasets, M say, are sufficient for reliable results, knowing that full efficiency would be reached for $M = +\infty$. Precision should be balanced against computational expense. It has been stated repeatedly, and practically confirmed, that a small number of imputations

oftentimes gives very acceptable results. Sources like ? and the classic ? quote values as low as 2–5. While attractive, especially when faced with large and complex databases, such low numbers may not always apply. Features that would require larger numbers of imputation include: increasing amounts of missing information, and the need for hypothesis testing rather than merely parameter and precision estimation. It is reassuring that ? indicated that, even under undesirable circumstances, it is rare for M needing to be in excess of twenty. This observation was based on extensive simulation, rather than theory.

Some of the approaches for dealing with selecting the number of imputations are reviewed in Section 7. Our proposal to handle the choice is presented in Section 7. Section 7 reports some simulations, while real-life applications are offered in Section 7.

Number of imputed datasets: A review

Upon producing multiply imputed datasets, a standard analysis is routinely applied to each of the completed datasets. Let $\boldsymbol{\theta}$ be the parameter vector of interest and $\hat{\boldsymbol{\theta}}_m$ its estimate from the m th imputed dataset. If M is the number of imputed datasets, then $\tilde{\boldsymbol{\theta}}$, the multiple imputation estimator is defined as:

$$\tilde{\boldsymbol{\theta}} = \frac{1}{M} \sum_{m=1}^M \hat{\boldsymbol{\theta}}_m. \quad (7.1)$$

If $\widehat{\Sigma}_m$ is the estimated variance-covariance of $\widehat{\boldsymbol{\theta}}_m$, then the within-imputation variability is:

$$\widehat{W} = \frac{1}{M} \sum_{m=1}^M \widehat{\Sigma}_m, \quad (7.2)$$

and the between-imputation variability is:

$$\widehat{B} = \frac{1}{M-1} \sum_{m=1}^M (\widehat{\boldsymbol{\theta}}_m - \widetilde{\boldsymbol{\theta}}) (\widehat{\boldsymbol{\theta}}_m - \widetilde{\boldsymbol{\theta}})'. \quad (7.3)$$

Then, asymptotically, for the sample size as well as M going to infinity, we have

$$\widetilde{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \widehat{B} + \widehat{W}). \quad (7.4)$$

For finite M , the variance of the normal distribution in (7.4) becomes (?):

$$\widehat{V} = (1 + M^{-1})\widehat{B} + \widehat{W}. \quad (7.5)$$

The term \widehat{B}/M takes into account the increased variability stemming from finite M . Obviously, for $M \rightarrow \infty$, this extra term vanishes. Consider $r = \frac{1}{q} (\text{tr}(\widehat{B}\widehat{W}^{-1})) (1 + M^{-1})$ and $\nu = (M-1)(1+r^{-1})^2$ with $\text{tr}(A)$ indicating the trace of the matrix A , and q the length of parameter vector $\boldsymbol{\theta}$. For the k th element of $\boldsymbol{\theta}$ we have:

$$\widetilde{\theta}_k \approx \theta_k + t_\nu \sqrt{\widehat{V}_{kk}}, \quad (7.6)$$

where t_ν is Student's t distribution with ν degrees-of-freedom and \widehat{V} can be computed using (7.5). These expressions are derived and discussed in detail by ? or ?. The degrees-of-freedom ν are

computed by ?. Small sample degrees-of-freedom are reviewed by ?. Note that (7.6) applies to the scalar case.

Now, if I_M is the information based on M imputations and I_∞ is the information for $M \rightarrow +\infty$, then,

$$\frac{I_M}{I_\infty} = \left(1 + \frac{\gamma}{M}\right)^{-1}, \quad \gamma = \frac{r + 2/(\nu + 3)}{r + 1}, \quad (7.7)$$

where γ in (7.7) can be regarded as the fraction of missing information. ? suggested that for many applications with a moderate amount of missingness, 3–5 imputations might well be sufficient. For example, using these expressions, with 10% missingness, $M = 3$ would provide about 97% efficiency when compared with an infinite number of imputed datasets. It is not surprising that this rule of thumb is relatively broadly accepted in practice. For example, according to SAS/STAT(R) 9.4 User’s Guide, the default number of imputations in the MI procedure is set equal to 5.

However, in the latest version of SAS (SAS/STAT(R) 14.1) the default number of imputed datasets (**NIMPUTE**) is increased to 25. Furthermore, following ?, an option is provided in this version of SAS’ **PROC MI** to set the number of imputed datasets equal to the fraction of incomplete cases (**NIMPUTE=PCTMISSING**). This indicates a level of acceptance among commercial software developers for the need for larger M ’s, also alternative procedures to determine it.

A practical difficulty with this heuristic is the need for γ . The quantity should be estimated to begin with, but obviously, the

quality of this estimate is hugely affected by the number of imputed datasets itself. Furthermore, as ? also suggests, the rule for the number of imputed datasets would take into account three parameters: $\tilde{\theta}$, its variance, and the degrees-of-freedom of the Student's t distribution. When one is interested in controlling the p -values when conducting hypothesis tests, or other quantities, then perhaps more imputations are needed (??).

? distinguished between two cases. Their proposal is to use a small number of imputed datasets when the inference is clear-cut, but if it is less clear-cut and an accurate estimate of the p -value or γ is needed, they proposed to set $M = 100$.

Given that for small M , the approximation in (7.6) may not be accurate enough, ? proposed an iterative procedure to select M such that the confidence level remains at a selected level. He proposed to select M such that the coefficient of variation of $t_{\nu} \sqrt{V_{kk}}$ for the worst-case parameter becomes less than the type I error rate α ; this author used the conventional $\alpha = 0.05$.

? have performed a simulation study to investigate the effect of M on various characteristics, such as power, mean squared error, fraction of missingness, etc. ? explored various factors that are affected by increasing M . His findings were used in ? who considered the case of longitudinal data and then focused on determining M to stabilize MI-based inferences. Stability in these articles is defined in terms of four conditions regarding the conditional standard errors of the MI estimator, the test statistics,

the missingness fraction, as well as the coefficient of variation of the half confidence interval. Under some circumstances, these authors proposed to use as many as 200 imputed datasets.

? proposed a jackknife procedure (?) to estimate $\sqrt{B/M}$, the so-called Monte Carlo error. These authors then propose to select the sufficient number of imputed datasets based on achieving a pre-defined level of precision. Their approach is implemented in the command `mim` in STATA.

Number of imputed datasets: an alternative proposal

As we have seen, in most of the proposed rules for choosing M , the comparison is always between characteristics under a given, finite M on the one hand, and $M \rightarrow +\infty$ on the other. In contrast, our proposal is to compare the quantity of interest with its successor (i.e., under M versus $M + 1$). As the estimated parameter and its variance using MI will converge to some asymptotic values as $M \rightarrow \infty$, monitoring this convergence is insightful. This implies an iterative perspective, which is convenient for practice. We call this procedure the iterative multiple imputation (imi). The imi procedure is formulated as follows:

1. **Start.** Select an initial number of imputed datasets, M_0 ,
$$\tilde{\boldsymbol{\theta}}_{M_0} = \sum_{i=1}^{M_0} \hat{\boldsymbol{\theta}}_i / M_0.$$
2. **Update.** For $m > M_0$,

$$\tilde{\boldsymbol{\theta}}_{m+1} = \frac{m \tilde{\boldsymbol{\theta}}_m + \hat{\boldsymbol{\theta}}_{m+1}}{m + 1}. \quad (7.8)$$

3. **Distance.** Compute: $d_{m+1} = d(\tilde{\boldsymbol{\theta}}_{m+1}, \tilde{\boldsymbol{\theta}}_m)$ using an appropriate distance.
4. **Stopping rule.** $d_j < \varepsilon$ for $j = m + 1, \dots, m + k_0$.

Here, M_0 is an integer indicating the initial number of imputations. For $M_0 = 2$, the stopping rule will be examined from the beginning, but in situations where the user knows a minimum number of imputations are needed (based on proportion of missing data, etc.) a larger M_0 can be used. $\hat{\boldsymbol{\theta}}_m$ is the estimated parameter in the m -th iteration and $\tilde{\boldsymbol{\theta}}_m = \sum_{i=1}^M \hat{\boldsymbol{\theta}}_m / M$. Note that $\tilde{\boldsymbol{\theta}}$ can be replaced with other quantities of interest, e.g., p -values. Clearly, for various quantities of interest, the corresponding combination rule should be applied in (7.8). Since the convergence of this procedure is not monotone, one needs to determine an integer k_0 as the number of successive steps that the stopping rule should be validated. This would prevent an early declaration of convergence. Again, $k_0 = 1$ would stop the first time this criterion is met, but a larger k_0 is recommended. Our proposal is to use $k_0 = 3$ (see Sections 7 and 7).

As can be seen, the number of imputed datasets in the proposed iterative procedure has two components: a deterministic part: $M_0 + k_0$ and a stochastic part which is determined by the iterations. Our implementation of this procedure in R package `imi` makes it possible to determine these two parameters interactively after observing the computation time of imputing the dataset and fit the model for $M_0 = 2$ times.

According to the convergence rate of the multivariate central limit theorem (??), roughly speaking, the convergence of $\tilde{\boldsymbol{\theta}}$ to $\boldsymbol{\theta}$ as $M \rightarrow \infty$ in (7.4) depends on 3 different aspects: sample size, dimension of the random vector (parameter space), and the third moment of the components of the random vector. In our iterative procedure, the sample size consists of two aspects: the number of imputed datasets, M , also the proportion of missing data. A larger M makes the convergence faster, while a larger proportion of missing data makes it slower. On the other hand, another important issue is the dimension of the parameter space, the more parameters to estimate the more slowly the convergence will be achieved. The third moment highlights the role of the model, for example, the convergence for a linear model would be different from a logistic regression (see Section 7). Therefore, an appropriate method for determining M should consider all these aspects. In other words, it should be sensitive to changes in any of these factors. As our proposed method uses the goal of the analyses (parameter estimate, hypothesis testing, etc.) in the process of determining M , it includes such aspects. Simulation results in Section 7 support this claim as well.

One important aspect of the proposed iterative procedure is the choice of an appropriate distance in Step 3. One immediate choice is to use an Euclidean distance, as follows:

$$d_{m+1}^{\text{Euc}} = \sqrt{(\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m)^T (\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m)}. \quad (7.9)$$

Alternatively, one may try to make the largest difference smaller, then an ℓ_∞ -norm would be the appropriate distance. Obviously, for $q = 1$ these two are equivalent:

$$d_{m+1}^{\ell_\infty} = \max \left\{ |\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m| \right\}. \quad (7.10)$$

The common problem with ℓ_p -norm type distance measures in (7.9)–(7.10) is the fact that they would emphasize the elements in the parameter vector which have larger magnitude. Also, they are not robust against changes in units. This can also be seen from the general definition of the ℓ_p -norm of a vector $\mathbf{x} = (x_1, \dots, x_k)$, $\|\mathbf{x}\|_p$:

$$\|\mathbf{x}\|_p = (|x_1|^p + \dots + |x_k|^p)^{1/p}, p \leq 1.$$

When $p \rightarrow \infty$, one can show:

$$\|\mathbf{x}\|_\infty = \max\{|x_1|, \dots, |x_k|\}.$$

As an alternative, we propose using the Mahalanobis distance (?):

$$d_{m+1}^{\text{Mah}} = \sqrt{\left(\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m\right)^T S^{-1} \left(\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m\right)}. \quad (7.11)$$

For using (7.11), one needs to define an appropriate S . One can show:

$$\tilde{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m = \frac{\hat{\boldsymbol{\theta}}_{m+1} - \tilde{\boldsymbol{\theta}}_m}{m+1}. \quad (7.12)$$

Considering the fact that $\hat{\boldsymbol{\theta}}_{m+1}$ and $\tilde{\boldsymbol{\theta}}_m$ are independent given the

observed part of the sample, one may use:

$$\text{Cov} \left(\frac{\widehat{\boldsymbol{\theta}}_{m+1} - \widetilde{\boldsymbol{\theta}}_m}{m+1} \right) = \frac{1}{(m+1)^2} [\text{Var}(\widehat{\boldsymbol{\theta}}_{m+1}) + \text{Var}(\widetilde{\boldsymbol{\theta}}_m)] = \frac{1}{(m+1)^2} [\text{Var}(\widehat{\theta}_m)] \quad (7.13)$$

The division by $m+1$ in (7.12) makes this an inappropriate candidate for S , as it changes every time, while we need to monitor convergence of the procedure, relative to a sufficiently stable yardstick.

Other choices for S include the within, between, or combined covariance matrix of the parameters at step $m+1$, i.e., \widehat{W}_{m+1} , \widehat{B}_{m+1} , or \widehat{V}_{m+1} , respectively; here \widehat{V} is computed as in 7.5. Our simulation results show that \widehat{B}_{m+1} cannot be an appropriate choice since being normalized by the variability of the differences, the resulting Mahalanobis distance is not sensitive to the fraction of missing data. This will be discussed in more detail in Section 7.

An appropriate distance should be sensitive to the fraction of missing data, but robust against rescaling model parameters. Both \widehat{W} and \widehat{V} serve these purposes, but our simulation results show that (see Section 7) for some large values of variance components, \widehat{W} would show chaotic behavior from one iteration to another in some cases, while \widehat{V} balances between and within variability, hence behave more stably. Therefore, our proposal for an appropriate distance in Step 3 is the Mahalanobis distance with $S = \widehat{V}_{m+1}$. In cases where the quantity of interest is unitless, e.g., a correlation coefficient, a coefficient of variation, a coefficient of determination, a p -value, etc., using ℓ_p -norm type distances such as in (7.9)–(7.10)

is recommended, especially when obtaining the variance of the quantity of interest is not straightforward. Of course, one may use any tailored distance according to the problem at hand, this would not change the proposed iterative procedure.

As mentioned earlier, the convergence speed of the imi procedure depends on various factors such as the model, the parameter space, and the fraction of missing data. Also, the choice of the distance in Step 3 of the procedure plays a determining role when it comes to deciding when to stop. Therefore, formulating a fixed universal threshold for ϵ in Step 3 of the proposed iterative procedure seems not to be possible. However, based on extensive simulation results, when using a Mahalanobis-types distance as in (7.11) with $S = \hat{V}_{m+1}$, considering $\epsilon = 0.05$ leads to a liberal choice of M . If one wants to select a conservative M , we propose to set $\epsilon = 0.01$. In the same setting, such choices of ϵ would select M 's that are approximately in agreement with Rubin's classic suggestion, e.g., to use $M = 5$ for 10% missing data (See Table 7.1).

In general, for selecting ϵ , one needs to consider the nature of the quantity of interest when deciding on ϵ . For example, when dealing with p -values, one needs to ensure that increasing the number of imputed datasets would not change the result of the test, so the distance between two p -values from imputed datasets number m and $m + 1$, near or smaller than the significance, α , should be smaller than, for example, $\alpha/10$; see Section 7. Therefore, in case of p -values, we propose to use a Euclidean distance with

$\epsilon = \alpha/10$. Note that, given the null hypothesis, the p -value is uniformly distributed. Therefore, for an appropriate ϵ and up to a constant coefficient, using a Mahalanobis distance is equivalent to the Euclidean distance.

As ? pointed out, the two central procedures in theory of statistics are parameter estimation and hypothesis testing. To summarize our discussion above, we proposed to use a different ϵ for each of these procedures. In case of parameter estimation one may use 0.05 or 0.01, and in case of hypothesis testing (using p -values) one may use $\alpha/10$.

Simulation study

In order to study our proposed method, two simulation plans are considered. One with a multivariate normal vector, and the other with a logistic regression model. Using these two simulation settings, the performance of our proposed procedure for both parameter estimation and hypothesis testing will be studied. The results will be displayed, compared, and discussed, after presenting the plans.

Simulation plan

A simulation study is undertaken for a multivariate normal vector as well as data generated from a logistic regression model. Two types of covariance structures are considered for the multivariate normal model: compound-symmetry (CS) and first-order autoregressive (AR(1)). Consider \mathbf{Y}_i a vector of length k , then $\mathbf{Y}_i \sim N(\mu\mathbf{1}_k, \Sigma)$. Under CS, $\Sigma = \sigma^2 I_k + \tau J_k$, for $\sigma^2, \tau > 0$ and I_k and J_k the

identity and all-ones matrices, respectively. For AR(1), we have $\Sigma = \sigma^2 C$ where the (i, j) element of C is defined as $\rho^{|i-j|}$, for $\sigma^2 > 0, -1 \leq \rho \leq +1$.

All data are generated for $\mu = 0$. The length of each random vector is set to 5, and the sample size is considered to be 100. For each covariance structure in the multivariate normal case 18 different scenarios are considered as combinations of the following settings:

- Two proportions of missing data are used: 10% and 70%;
- Three σ^2 values are used: 1, 16, 64;
- Three ρ values are used: 0.2, 0.5, 0.8. In case of CS, the corresponding τ is computed given the specified ρ using $\tau = \rho\sigma^2/(1 - \rho)$.

Logistic regression data are generated with parameters $(\beta_0, \beta_1, \beta_2) = (0.2, -2, 0.5)$, where the design matrix is generated using each of the 9 considered CS settings. The parameters of the CS and AR(1) models are estimated using the results in ? and ?, respectively. The logistic regression is fitted using R function `glm`. The proportion of missing values are the same as in the multivariate normal scenarios (10% and 70%).

The missing data for all of these scenarios are generated under a missing at random (MAR) mechanism using the function `ampute` in the R package `mice`; see ? and ?. The function `ampute` implements a multivariate approach to generate missing data by assigning a

weight to each observation. In case of a missing at random (MAR) mechanism, these weights only depend on the observed part of the sample. Based on the assigned weights, a logistic distribution will be used to compute the probability of missingness. An observation with a larger weight has more chance to be missing.

Each incomplete dataset is imputed 500 times using a multivariate normal predictive model in the R package `Amelia2`, and each scenario is repeated 100 times as well.

To compute the distance between two steps in our iterative procedure, we use five different measures: Euclidean distance (7.9), ℓ_∞ -norm (7.10), Mahalanobis distance (7.11) with S selected as within (\widehat{W}), between (\widehat{B}), and combined covariance matrix (\widehat{V}) of the estimated parameter vector.

There are two main outcomes of interest from different considered scenarios. First, the convergence plot for different distances. Second, the M for which the iterative procedure terminates. Such M is computed using seven values $\epsilon = 0.005, 0.01, 0.02, 0.03, 0.04$ and 0.05 , and for each ϵ , three successive validation steps are considered: $k_0 = 1, 3$ and 5 .

In order to evaluate our proposed method in case of p -values, one-sample t -tests are applied with $H_0 : \mu = d_0$ on the first column of the data generated from each setting of CS and AR(1) covariance structure. In addition, paired t -tests are also performed for testing $H_0 : \mu_1 - \mu_2 = d_0$ for the first and the second columns of data

generated using each setting of CS and AR(1) covariance matrices. To see the effect of the test value on the convergence, we have considered $d_0 = 0$ and $d_0 = 10$. As the p -value is a scalar, we use Euclidean distance (7.9) to measure distance of two successive steps. In case of p -values, again the convergence plots are provided; also, the sufficient number of imputed datasets is computed for two values of $\epsilon = 0.05/10$ and $0.05/100$. For each ϵ , three successive validation steps are considered: $k_0 = 1, 3$ and 5 .

Simulation results

As displaying the results of 18 different scenarios takes a lot of space, here we discuss all of the outcomes but only display the results for one of them. The rest of the results will be available in the R package `imi` via 9 lists of datasets, the function `imi.make.plots` can be used to produce similar plots as in Figure 7.1 and Figure 7.2 for other scenarios. Further information can be found in the documentation of the package.

Figure 7.1 shows the convergence plots for the scenario with $\sigma^2 = 16$ and $\rho = 0.5$. As one may see, while using all of the distance functions a decreasing trend can be observed, but as expected Euclidean and ℓ_∞ norms, as well as Mahalanobis distance with $S = \widehat{B}$ are not performing well. On the other hand, the latter with both $S = \widehat{W}$ and $S = \widehat{V}$ performs fine. Note that for 70% missing data for some of the imputed datasets, quasi-complete separation occurs when fitting logistic regression. In such cases, a new imputed dataset is generated.

An interesting observation from Figure 7.1 is the different convergence rate for logistic regression compared with the multivariate normal case. As one may see, for 10% missing data, the logistic regression converges almost with the same speed as 70% missing data in case of a multivariate normal vector parameter estimates. That highlights the effect of the model (thus not only the proportion of missing data) when determining a sufficient number of imputed datasets.

Figure 7.2 shows the convergence plots using $M = 500$ for the p -values of one sample and paired t -tests for two different test values (0, 10). The distance between two p -values is computed using the Euclidean distance. As one may see, one important factor that should be taken into account when determining M is the test value. When it is far from the actual parameter convergence is achieved for a much smaller M .

To explore the selected M using various ϵ 's and k_0 's to evaluate our proposals ($\epsilon = 0.01$ or 0.05 , and $k_0 = 3$), Tables 7.1 and 7.2 show the selected M for parameter estimation in multivariate normal and logistic regression (Table 7.1) and p -values of one sample and paired t -tests (Table 7.2). As one may see, again the effect of proportion of missing data and the model is verified on the selected sufficient M . While a dataset with a larger proportion of missing data needs a larger M , the number of sufficient imputed datasets in case of logistic regression is different from multivariate normal, even for a smaller proportion of missing data. This would highlight the role

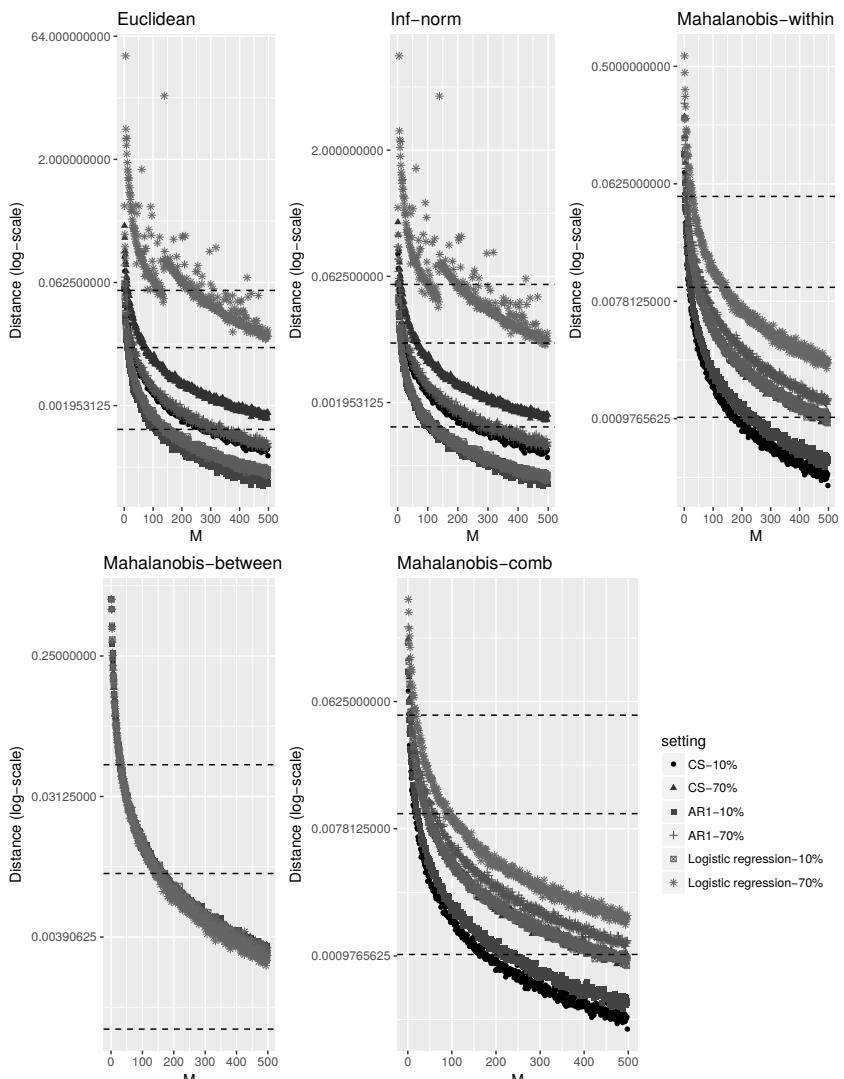


Figure 7.1: Convergence plots for multivariate normal random vectors parameter estimates with CS and AR(1) covariance matrix structures (with $\sigma^2 = 16$ and $\rho = 0.5$), as well as a logistic regression model, using five different distance functions: Euclidean norm, ℓ_∞ -norm, and Mahalanobis distance with within, between, and combined covariance matrices over imputed sets of data for two proportion of missing data: 10% and 70%. Each point in the plot is the average over 100 replications. The three horizontal dashed lines are $\epsilon = 0.05, 0.01, 0.001$, respectively.

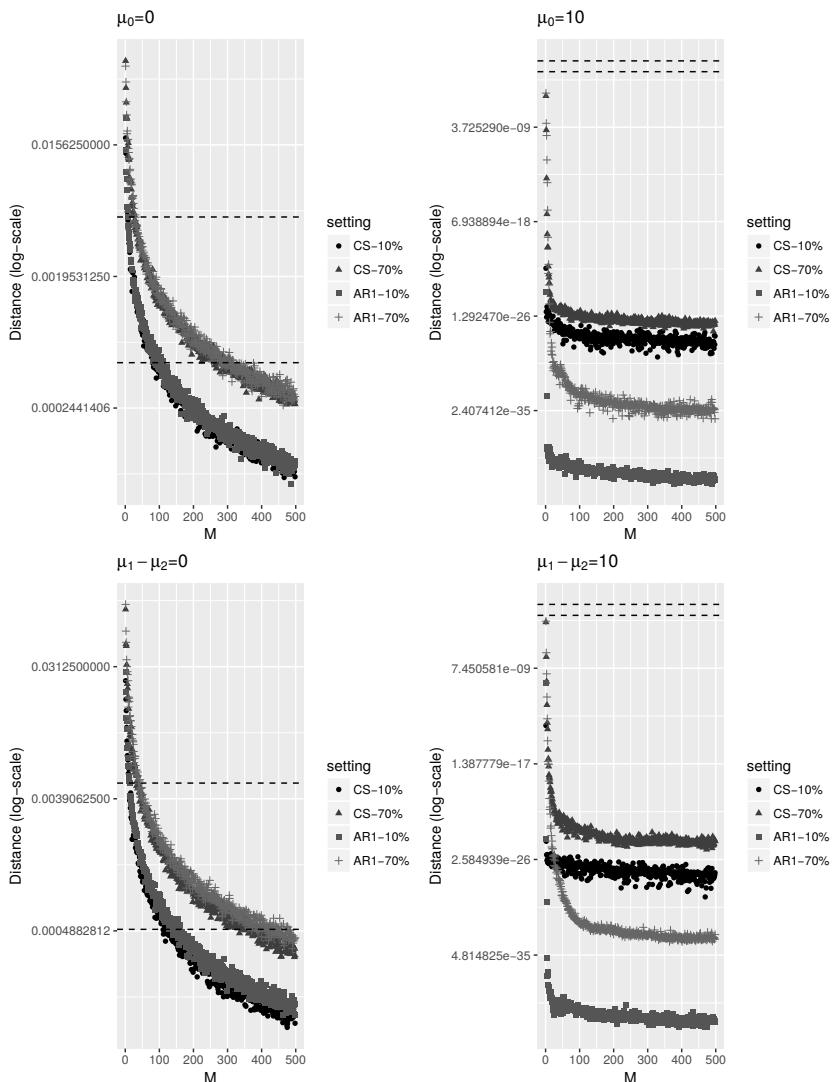


Figure 7.2: Convergence plots for the p -values of one sample (first row) and paired (second row) t -tests for different values of test statistics. The data are generated from a (multivariate) normal distribution with mean 0 and CS and AR(1) covariance matrices (with $\sigma^2 = 16$ and $\rho = 0.5$). The missing values are generated with proportions 10% and 70%. The two horizontal dashed lines are $\epsilon = 0.005, 0.0005$, respectively.

of a procedure that accounts for all of these aspects.

Also, it seems that our proposals for ϵ and k_0 are sensible in terms of comparing with Rubin's classical rule suggesting 3–5 imputed datasets for 10% missing data. In case of p -values it seems $\epsilon = 0.005$ together with $k_0 = 3$ works acceptably fine.

An interesting observation when comparing the selected M for different values of ρ is the effect of *missing information*. For the same fraction of missing data, when the correlation is larger, the selected M is smaller. That would suggest that what really matters is not only the fraction of missing data, but the fraction of missing information. As an extreme case, consider two variables X and Y in a given dataset, suppose X is complete and Y is partially missing. Assume further that $Y_i = X_i$ ($\rho = 1$), for every subject. In this case one imputation would be sufficient, no matter the fraction of missing items. Figure 7.3 shows the selected M for different scenarios with $\sigma^2 = 16$ and $\rho = 0.5$ and 0.8 , and $k_0 = 3$. As one may see, in most of the cases, for the same proportion of missing data and model, the selected M is smaller when ρ is larger.

Applications

In this section, two applications of multiple imputations are considered: fitting a logistic regression to incomplete data, as well as combining p -values from a one-sample t -test for a set of incomplete data. The number of imputed datasets will be determined using the proposed iterative procedure via the R package `imi`.

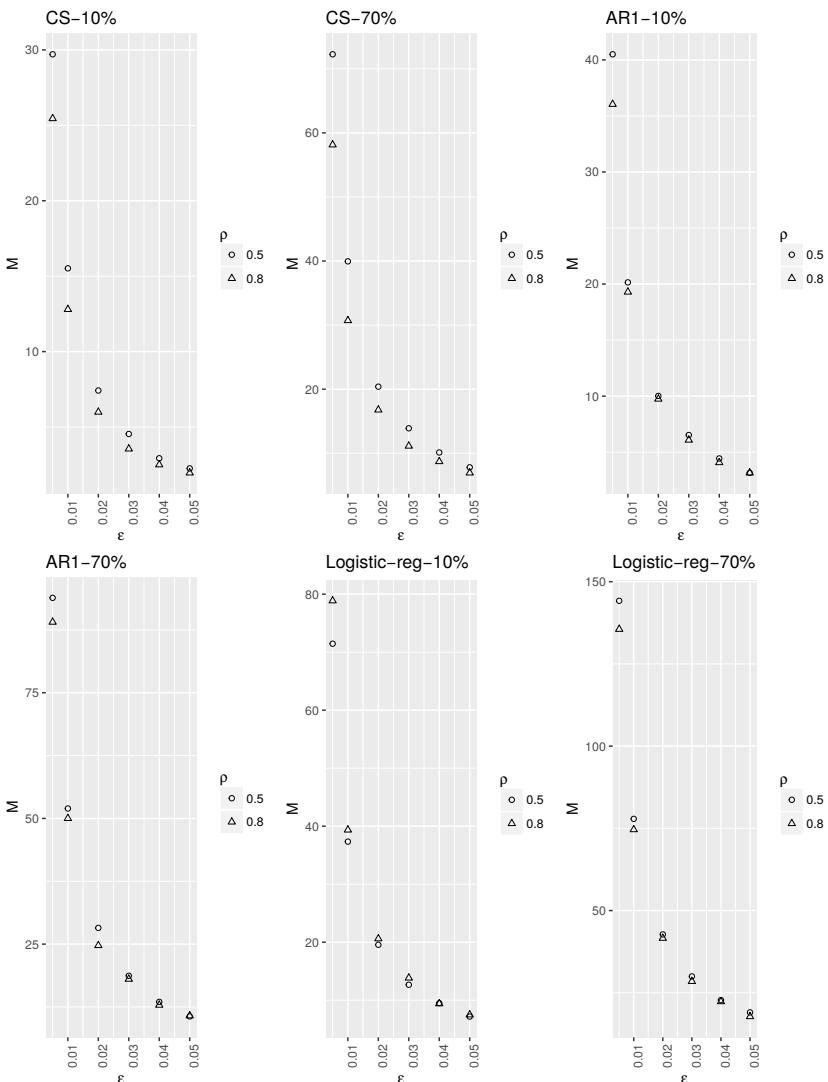


Figure 7.3: Averaged selected M with $k_0 = 3$ for data generated from a multivariate normal with mean 0 and covariance matrix with CS and AR(1) structures with $\sigma^2 = 16$ and $\rho = 0.5, 0.8$, as well as a logistic regression model. The proportions of missing data are 10% and 70%. Each point in the plot is the average over 100 replications.

Leuven Eye Study. Logistic regression

The Leuven Eye Study (LES; ?), is an extensive observational study of glaucoma performed at the ophthalmology department of UZ Leuven. The dataset so far consists of 141 variables measured for 585 subjects. As this study was performed in a clinical setting, it was not feasible to measure all these variables for all of the subjects. Because of that, missingness is a considerable problem. Among these 141 variables, 130 of them are selected to be used in multiple imputation. An analysis was performed to study which risk factors are relevant for the binary outcome defined as normal vs. glaucoma. The risk factors selected for the model include both structural and functional measurements of the eye (cup to disc ratio, corneal thickness, visual acuity), previous medical history (gender, sleep apnea, rhythm disorder, having had cataract surgery, being medicated with statins or calcium channel blockers) and more variable biometric measurements, like intra-ocular pressure and diastolic blood pressure.

Figure 7.4 (bottom-left) shows a histogram of the number of missing values (out of 585 patients) in the LES dataset. As one may see, the number of missing values is diverse. Also the patterns of missing values could be different from variable to variable. Therefore, determining M using classical or simulation-based approaches could be a challenge here. Using our proposal, it could be more convenient to determine the sufficient number of imputed datasets.

in order to impute the missing values the fully conditional specification (FCS) approach (??, and ?) is employed. For continuous variables, predictive mean matching, ?, is used for generating the imputed datasets. Also, a binary logistic regression predictive model is used to impute binary variables. In the specific case of one variable with three levels, a polytomous logistic regression predictive model is used. In order to create a sufficient number of imputed datasets we use the function `imi.glm` in the R package `imi`.

We have selected 2 for the initial number of imputations (M_0). Also, the convergence criterion should be successively validated 3 times to terminate the procedure ($k_0 = 3$). Figure 7.4 (bottom-right) shows the convergence plot for this model on the LES data. As one may see, with $\epsilon = 0.05$ (the liberal choice) and $k_0 = 3$ we need $M = 58$ imputed datasets, while with the conservative choice, $\epsilon = 0.01$, one may need to generate $M = 250$ imputed datasets.

Cholesterol data. One sample t -test p -values

The cholesterol dataset includes cholesterol levels for 28 patients treated at a Pennsylvania medical center. The cholesterol levels have been recorded on day 2, day 4 and day 14 after an attack for each patient. However, there are 9 missing values for day 14. This dataset was analyzed by ?, Chapter 5 and is publicly available in R package `norm2`.

Here we use our R implemented function `imi.t.test` to sufficiently impute this dataset and perform the t -test on it for two different

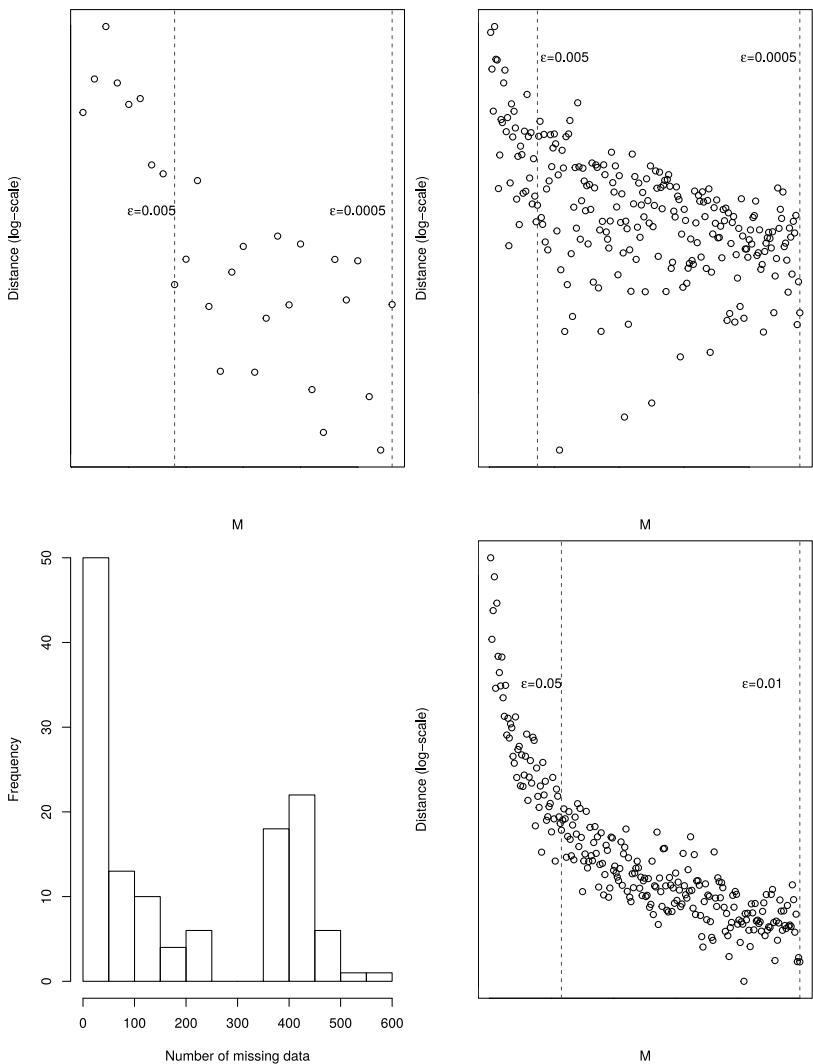


Figure 7.4: Convergence plots for p -values of t -test on cholesterol data (top) with $\mu_0 = 200$ (left) and $\mu_0 = 220$ (right) test values, and LES logistic regression (bottom-right), with $\epsilon = 0.05, 0.01$, $M_0 = 2$, and $k_0 = 3$. The dashed vertical line shows the selected M for different values of ϵ . The histogram for the number of missing values in LES dataset (bottom-left) is also presented.

test-values ($\mu_0 = 200, 220$).

Again, we ask for two initial imputations and then 3 successive validation steps ($M_0 = 2, k_0 = 3$). Figure 7.4 (top) shows the convergence plot for $\mu_0 = 200$ (left) and $\mu_0 = 220$ (right). In each case the vertical dashed line shows the selected M for different values of ϵ . As one may see, when testing $H_0 : \mu = 200$, for $\epsilon = 0.05/10$ and $k_0 = 3$, only $M = 9$ imputed datasets are sufficient, changing ϵ to $0.05/100$ will increase M to 28. When testing $H_0 : \mu = 220$, however, more imputations are needed. For $\epsilon = 0.05/10$ we get $M = 37$ and for $\epsilon = 0.05/100$ we obtain $M = 239$.

Discussion and Concluding Remarks

Multiple imputation is an appealing and extensively used method to analyze incomplete data, or, even more broadly, data that can be cast in a missing-data framework (?). There is widely held and largely justified wisdom that a small number of imputation suffices for many practical purposes, even though ? states that the number depends on the amount of missing information, the data type under study, and the inferential purpose. We have presented a very simple procedure, that is based on conventional large-sample convergence results, by merely using the fact that multiple imputation is in itself a sampling mechanism. It is based on comparing the estimates under M and $M + 1$ imputations, the procedure stops when the distance between two steps become smaller than a pre-defined ϵ . One needs to select an initial number of imputations as well as number of steps the stopping rule should be validated. This would

prevent early stopping when two consecutive quantities of interest are unusually close. Thus, when convergence is suggested, to play safe, one could still go on for a while, and wait until convergence has been confirmed. Stopping too early would lead to less precise estimates, or unwanted decision making, while the main problem with too high an M is the computational cost, also when the distance between two steps becomes very small, due to a small ϵ (or equivalently selecting a large M), some numerical underflow issues could occur.

There are two main parameters that are needed to be specified before applying our procedure on an incomplete dataset. The stopping rule for the distance between two steps, ϵ , and the number of steps this criterion should be successively validated, k_0 . Based on our numerical experiences, we suggest $k_0 = 3$ validation steps is enough in most of the applications. For ϵ , the choice depends on the chosen distance which itself depends on the purpose of the analysis: estimation or hypothesis testing. For estimation purposes, i.e., estimating the parameters and their precision, we suggest to use a Mahalanobis distance with $S = \widehat{V}_{M+1}$, then $\epsilon = 0.05$ as a liberal choice and $\epsilon = 0.01$ as a conservative choice. For p -values at $100(1 - \alpha)\%$ level of confidence, we suggest to use the Euclidean distance with $\alpha/10$ as a liberal choice, and as a conservative choice our proposal is to use $\alpha/100$. Of course, as discussed in Section 7, the convergence rate depends on various aspects (proportion of missing data, the model, dimension of the parameter space). Therefore, it is very well possible that for less conventional applications one

needs different tailor-made values for ϵ and k_0 .

Simulations and case studies, the latter with admittedly a large fraction of missingness, indicate that it might be needed, in many practical settings, to generate a number of imputations well above what common wisdom prescribes.

Our method has several advantages. First, it is applicable in any context where multiple imputation is useful, irrespective of the amount of missing information, the model used, or the target of inference. Second, imputed data sets can be generated one by one, and each time one can easily decide about the need to continue adding new imputed data sets, depending on the research question(s) to be answered. Third, the procedure can easily be automated since ‘expert judgement’ is not needed. Finally, no post-hoc sensitivity assessment to the number of imputations is required, allowing the stopping criterion to be specified prior to the data analysis or even prior to the data collection.

Furthermore, our implementation of this procedure in R package `imi` could make it more available for different applications. The current version of this package includes the *t*-test (one sample, two samples and paired), as well as linear and generalized linear regression models. For each application one has the possibility to generate a sufficient number of imputed datasets. In case of existence of some already generated imputed sets of data, one can examine their sufficiency for the desired analyses. When they are not sufficient it is also possible to generate new imputed datasets

till M becomes sufficiently large. The considered interactive options allow the user to select M_0 and/or k_0 based on the time it takes to generate 2 imputed datasets and perform the analyses on them.

On using multiple imputation for exploratory factor analysis of incomplete data

Introduction

Factor analysis and principal component analysis (PCA) are techniques mainly based on singular value decomposition of covariance matrices of multivariate data, e.g., a questionnaire, several measurements on a subject, etc. In multivariate data, it is very well possible that for some of the subjects, one or more of the variables are missing. One approach to deal with this phenomenon is listwise deletion, i.e., removing all subjects with missing values and only using the observed part of the data. This would lead to loss of information, and worse than that, biased estimates and conclusions.

As an alternative to listwise deletion, ? proposed the nonlinear iterative partial least squares estimation (NIPALS) procedure, which uses an alternating weighted least squares algorithm and estimates principal components one by one in a sequential manner. ? extended NIPALS to directly estimate the desired subspace, rather than sequentially. Iterative principal component analysis, ?, estimates the missing values and the principal components simultaneously. The main difference between iterative PCA and NIPALS is that NIPALS tries to estimate principal components regardless of the

missing values, while iterative PCA produces a single imputation for the missing values as well. However, a main problem with the iterative PCA of ? is overfitting. This problem would become more serious when the amount of missing data becomes larger. Authors like ? and ? proposed a regularized version of iterative PCA to overcome this problem.

Recently, authors like ?, ? and ? have considered multiple imputation (MI) in the sense of Rubin (???) to deal with the missing data problem in PCA and factor analysis. Rubin's multiple imputation first imputes the data using, for example, a joint (?) or conditional (?) model, then in the second step performs the usual analysis on each completed (imputed) data. The third and last step uses appropriate combination rules to combine the results from each imputed data. An appropriate combination rule needs to respect the fact that the imputed data are, after all, unobserved. Thus, it needs to take into account that each missing value is replaced with several plausible values. The focus of the current paper is on this last approach, where the multiple imputation is done prior to the desired analysis, e.g., PCA or factor analysis.

In this paper, the above problem will be described and difficulties of using MI in case of factor analysis and PCA will be discussed. Possible solutions will be reviewed and an alternative simple solution will be presented. A simulation study will evaluate the proposed method. Also, considering the "Divorce in Flanders" dataset as a case study, the application of the proposed method will be illustrated.

The paper ends with concluding notes.

Using multiple imputation with factor analysis: a review

Consider p correlated random variables $\mathbf{X}'_i = \{X_{i1}, \dots, X_{ip}\}$ with observed covariance matrix $\text{Cov}(\mathbf{X})$, of which the population value is Σ . In general, the idea of principal component analysis is to find as few as possible (uncorrelated) linear combinations of elements of \mathbf{X} such that their variance becomes as large as possible (needed) subject to proper standardization. One can show (?) if $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$ are the ordered eigenvalues of Σ , and (λ_i, e_i) is the i th eigenvalue-eigenvector pair, then the i th principal component is given by:

$$Y_i = e_i' \mathbf{X}, \quad \text{Var}(Y_i) = \lambda_i, \quad \text{Cov}(Y_i, Y_k) = 0, \quad i \neq k. \quad (7.14)$$

As one may see in (7.14), the PCA (or factor analysis from a wider perspective) are obtained using singular value decomposition of the covariance (correlation) matrix. The eigenvalues of Σ are the roots of the characteristic polynomial $|\Sigma - \lambda I|$ where $|A|$ denotes the determinant of matrix A and I is the identity matrix of the same order as Σ . Obviously, there is no natural ordering among roots of a polynomial. Further, as one may see in (7.14), λ_i represents a variance. It thus makes sense to order the eigenvalues in a descending manner. The problem arises when using multiple imputation prior to the factor analysis or PCA.

Consider $X_{ij}^{(m)}$ as the observed value for i th subject ($i = 1, \dots, N$) of j th variable ($j = 1, \dots, p$) in the m th imputed dataset. The eigenvector corresponding to the largest eigenvalue of $\Sigma^{(m)} = \text{Cov}(X^{(m)})$ gives the structure related to the first latent factor, and there is no guarantee that the eigenvector corresponding to the largest eigenvalue of $\Sigma^{(k)} = \text{Cov}(X^{(k)})$ ($k \neq m$) is comparable with the one from the m th imputed data. In other words, averaging the eigenvectors (principal axes, factor loadings) using the order or the obtained eigenvalues of the covariance matrix estimated from each imputed set is likely to lead to misleading or meaningless results.

In order to overcome this problem, ? have averaged the imputed values to have one single complete dataset. Other authors like ? and ? proposed to first impute the data, then perform the PCA or factor analysis on each imputed dataset separately. After obtaining the eigenvectors (factor loadings), because of the discussed problem of ordering, one needs an intermediate step before the usual averaging. This step consists of the use of the class of generalized procrustes rotations (?) to make these matrices as similar as possible. After rotating the obtained factor loadings simultaneously, the next step would be the usual averaging.

One intermediate solution in place of averaging the imputed values (?) or averaging the factor loadings (?) could be to estimate the covariance matrix from imputed sets of data using Rubin's rules first, and then apply the PCA or factor analysis on this combined covariance matrix. This proposal will be discussed in the next
Page 207

section.

Using multiple imputation with factor analysis: a proposal

Consider $X^{(obs)}$ a dataset with missing values and $X^{(1)}, \dots, X^{(M)}$ as the M imputed datasets with estimated covariance matrices $\widehat{\Sigma^{(1)}}, \dots, \widehat{\Sigma^{(M)}}$. Using Rubin's rules (?) the multiple imputation estimate of Σ can be obtained as follows:

$$\tilde{\Sigma} = \frac{1}{M} \sum_{i=1}^M \widehat{\Sigma_M}. \quad (7.15)$$

Having $\tilde{\Sigma}$, one may perform PCA or FA directly on it, and then the problem of factor ordering would vanish. Of course, that would not come for free. Estimating the covariance matrix first, and then performing FA would make impossible the direct use of Rubin's combination rules for precision purposes. Therefore, one may consider indirect or alternative solutions. We will consider this point in more detail and propose a straightforward solution. It is worth noting that this is not required if no precision estimates are needed, e.g., when principal components are merely calculated for descriptive purposes or, in general, when there is no need to either make inferences about them or select a sufficiently small subset of principal components.

An important aspect of PCA or FA is determining the number of factors or principal axes. One popular criterion to determine the number of factors/PCs is the proportion of explained variance. The
Page 208

proportion of explained variance based on the first k factors, γ_k , is

$$\gamma_k = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^p \lambda_j}, \quad (7.16)$$

with λ_j as in (7.14). When using MI, one needs to ensure the correct amount of information present in the data is used. This is also important when estimating γ_k . This can be done by constructing a confidence interval for γ_k using the estimate and variance obtained by properly taking imputation into account. Consider $\Lambda = (\lambda_1, \dots, \lambda_p)$ and Δ a diagonal matrix with $\lambda_1 \geq \dots \geq \lambda_p$ as its diagonal elements. For large samples we have (???):

$$\widehat{\Lambda} \sim N_p \left(\Lambda, \frac{2}{N} \Delta^2 \right). \quad (7.17)$$

Therefore, for large N the estimated eigenvalues become uncorrelated. Consider the pair $(\sum_{j=1}^k \widehat{\lambda}_j, \sum_{j=1}^p \widehat{\lambda}_j)$. Using (7.17) leads to:

$$\text{Cov} = \begin{pmatrix} \sum_{j=1}^k \widehat{\lambda}_j \\ \sum_{j=1}^p \widehat{\lambda}_j \end{pmatrix} = \frac{2}{N} \begin{pmatrix} \sigma_{11} & \sigma_{11} \\ \sigma_{11} & \sigma_{22} \end{pmatrix}, \quad (7.18)$$

with,

$$\sigma_{11} = \sum_{j=1}^k \widehat{\lambda}_j^{-2}, \quad \sigma_{22} = \sum_{j=1}^p \widehat{\lambda}_j^{-2}. \quad (7.19)$$

Then, using the Delta method, the variance of $\widehat{\gamma}_k$ can be written as
Page 209

follows,

$$\text{Var}(\widehat{\gamma}_k) \approx \frac{2}{N} \left(\sum_{j=1}^p \widehat{\lambda}_j \right)^{-2} \left[(1 - 2\widehat{\gamma}_k) \sigma_{11} + \widehat{\gamma}_k^2 \sigma_{22} \right] \quad (7.20)$$

Now, equipped with $\widehat{\gamma}_k$ and $\text{Var}(\widehat{\gamma}_k)$, one finds the MI estimates of them using Rubin's rule. Consider $\widetilde{\gamma}_{km}$ and $\text{Var}(\widetilde{\gamma}_{km})$ the estimated γ_k and its variance in the m th imputed data, then

$$\widetilde{\gamma}_k = \frac{1}{M} \sum_{m=1}^M \widetilde{\gamma}_{km}, \quad \text{Var}(\widetilde{\gamma}_k) = \frac{1}{M} \sum_{m=1}^M \text{Var}(\widetilde{\gamma}_{km}) + \frac{M+1}{M} S_{\widetilde{\gamma}_k}^2, \quad (7.21)$$

where $S_{\widetilde{\gamma}_k}^2 = \sum_{m=1}^M (\widetilde{\gamma}_{km} - \widetilde{\gamma}_k)^2 / (M-1)$ is the sample variance of $\widetilde{\gamma}_{km}$ ($m = 1, \dots, M$). Having $\widetilde{\gamma}_k$ and its variance in (7.21), one may easily construct a confidence interval for the proportion of explained variance.

It is well-known that for ratios such as $\widetilde{\gamma}_k$ using Fieller's method (?) is a sometimes preferable alternative to the Delta method. Using (7.18) and (7.19), Fieller's confidence interval for (7.16) can be calculated as follows:

$$C_1^2 = \frac{\sigma_{11}}{\left(\sum_{j=1}^k \widehat{\lambda}_j \right)^2}, \quad C_2^2 = \frac{\sigma_{11}}{\left(\sum_{j=1}^k \widehat{\lambda}_j \right)^2}, \quad r = \frac{\sigma_{11}}{\sqrt{\sigma_{11}\sigma_{22}}},$$

$$A = C_1^2 + C_2^2 - 2rC_1C_2, \quad B = z_{\alpha/2}^2 C_1^2 C_2^2 (1 - r^2),$$

$$L = \widehat{\gamma_k} \frac{1 - z_{\alpha/2}^2 r C_1 C_2 - z_{\alpha/2} \sqrt{A - B}}{1 - z_{\alpha/2}^2 C_2^2}, \quad (7.22)$$

$$.U = \widehat{\gamma_k} \frac{1 - z_{\alpha/2}^2 r C_1 C_2 + z_{\alpha/2} \sqrt{A - B}}{1 - z_{\alpha/2}^2 C_2^2} \quad (7.23)$$

It is well-known (?) that $\text{tr}(\sum_{m=1}^M \Sigma_m) = \sum_{m=1}^M \text{tr}(\Sigma_m)$ where tr denotes the trace of the matrix. As $\text{tr}(A_n) = \sum_{j=1}^p \lambda_j$, where λ_j ($j = 1, \dots, p$) are the eigenvalues, by estimating $\tilde{\Sigma}$ using (7.15), the sum of all of the eigenvalues would not change. But unfortunately, as ? has shown, for matrices A , B , and C , with eigenvalues α_i , β_i, δ_i in descending order, respectively, for any $1 \leq k \leq p$ we have,

$$\sum_{i=1}^k \delta_i \leq \sum_{i=1}^k \alpha_i + \sum_{i=1}^k \beta_i. \quad (7.24)$$

That would mean the proportion of explained variance obtained using eigenvalues of $\tilde{\Sigma}$ in (7.15) is always smaller than $\widehat{\gamma_k}$ in (7.21), i.e., $\widetilde{\gamma_k}$ overestimates the proportion of explained variance using $\tilde{\Sigma}$. In order to check the validity of the proposed method, one may make sure that the estimated explained proportion of variance obtained using $\tilde{\Sigma}$ falls in the estimated confidence interval for γ_k directly obtained from the imputed sets of data. Otherwise, using the proposed approach is not recommended.

Simulations

In order to evaluate the proposed methodology a simple simulation is performed in this section. The sample size is set to $N = 1000$,
Page 211

number of variables is $p = 10$, and the data are generated from a multivariate normal distribution using Cholesky decomposition of the following covariance matrix:

$$\begin{bmatrix} 9.0 & 5.0 & 5.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 5.0 & 9.0 & 5.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 5.0 & 5.0 & 9.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 17.0 & 10.0 & 10.0 & 10.0 & 10.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 10.0 & 17.0 & 10.0 & 10.0 & 10.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 10.0 & 10.0 & 17.0 & 10.0 & 10.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 10.0 & 10.0 & 10.0 & 17.0 & 10.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 10.0 & 10.0 & 10.0 & 10.0 & 17.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 8.0 & 3.0 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 3.0 & 8.0 \end{bmatrix}$$

For creating missingness in the data, 5%, 10%, 50% of each column were randomly selected and set to NA. The factor analysis is done using function `fa` in R package `psych` with PCA as the method, and no rotation. First, the factor analysis is performed on complete data and the factor loadings and eigenvalues are obtained. Then, for the data including missing values, two different methodologies are applied:

- Impute the data using FCS (?) $M = 20$ times using package `MICE` in R, and then, estimate the covariance matrix using (7.15) and perform factor analysis on it.
- Using regularised iterative principal component analysis

(IPCA) in the R package `missMDA` to impute the data a single time, then apply the factor analysis on this imputed dataset.

Furthermore, the proportion of explained variance using complete data is obtained. Also, the γ_k obtained from $\tilde{\Sigma}$ together with $\tilde{\gamma}_k$ and its confidence interval are provided, together with the γ_k obtained for the data imputed using IPCA. In order to compare the factor loadings with the ones from complete data, the ℓ_1 -norm of the difference between two loading matrices is given, i.e., the maximum absolute column sum of this difference. The simulations are repeated 100 times. Table 7.3 summarizes the means and standard deviations of the obtained results.

As one may see in Table 7.3, using $\tilde{\Sigma}$, the obtained loadings are most similar to the ones from the complete data. On the other hand, for γ_k , this method would find the proportion of explained variance very close to the one from the full data. Also, it is always in the confidence interval found directly using Rubin's rules. As one may see, using IPCA, γ_k is estimated larger than its value for the complete data, and that would become even larger when the amount of missing values increases. One possible reason is that IPCA tries to impute the data in a way that the explained variance is maximized (imputing using PC axes), while MI only uses the available information.

Divorce in Flanders

In order to illustrate the proposed methodology we use the Divorce in Flanders (DiF) dataset (?). DiF contains a sample of marriages registered in the Flemish Region of Belgium, between 1971 and 2008 with an oversampling for dissolved marriages (2/3 dissolved and 1/3 intact marriages). As a part of this study, the participants were asked to complete the validated Dutch version (?) of Big Five Inventory (BFI) (?).

The sample at hand consists of 9385 persons in 4472 families. In each family, mother, father, step mother, step father, and one child over 14 were asked to fill in BFI. Note that, depending on the presented family roles and the number of people agreed to participate, the size of the families could vary between 1 and 5. Among these 9385 persons, there are 1218 persons with at least one non-response (out of 44 items). As our main purpose here was to illustrate the use of the proposed method, in order to get rid of the problem of intra-correlation within the families, one person from each family was selected at random to form a sample of uncorrelated subjects. As a result, a random sample of size 4472 was taken where 515 of them had at least one non-response.

This incomplete dataset was imputed using the fully conditional specification (FCS) of ? using the MICE package in R (?) with the Predictive Mean Matching (PMM) method (??). The imputation was done $M = 25$ times. The covariance matrix was estimated for each of the imputed sets of data and the factor analysis was done

on the averaged estimated covariance matrix, as well as on each imputed set. The latter was used to construct a confidence interval for the proportion of explained variance. The estimated factor loadings are presented in Table 7.4. The Delta method confidence interval for the proportion of explained variance is estimated as (0.418, 0.427) and the Fieller's confidence interval using (7.22) and (7.23) is estimated as (0.417, 0.428). The estimated proportion of explained variance of the first five factors, using the proposed methodology is 0.422, which falls within both of the estimated intervals close to their upper bounds. This is coherent with the validity of the proposed methodology for this incomplete dataset.

Conclusions

Nonresponse and missing values are among at the same time major and common problems in data analysis, especially when it comes to survey data. Multiple imputation, which was first introduced to deal with nonresponse in surveys (?) has become a key and effective tool to deal with this problem. However, when it comes to combining this methodology with techniques like factor analysis and principal component analysis, due to the problem of ordering factors/principal components, combining the results from different sets of imputed data becomes as issue.

This problem is addressed in this article and a pragmatic solution is proposed, which is justified by theoretical discussion and reasoning. Our proposal states to first estimate the covariance matrix of the correlated variables and then perform the FA/PCA on this single

matrix. The theoretical aspects of this methodology are studied and investigated. A confidence interval is proposed for the proportion of explained variance, which can be used as a criterion to decide on the validity of the the proposed method.

The simulation results show comparable performance of the proposed method, when compared to alternative methodologies. To evaluate our proposal in real situations, it is applied to an incomplete BFI dataset; the result was definitely acceptable. The main advantages of using the proposed methodology are: it is compatible with any imputation methodology; implementing it is very straightforward and no extra programming effort is needed. Therefore, it can be used within any desired statistical package or software. It is fast and practical. Also, the proposed confidence interval for proportion of explained variance can be used as an effective criterion to decide about validity of the proposed method.

Table 7.1: Mean, standard deviation (STD) for selected M given different ϵ and k_0 values, and number of times $M > 500$ (out of 100 replications) for different models and different ϵ 's using the Mahalanobis-type distance with $S = \hat{V}$. The results are presented for 3 different successive validation steps $k_0 = 1, 3, 5$.

Model	ϵ	$k_0 = 1$			$k_0 = 3$			$k_0 = 5$		
		Mean	STD	$M > 500$	Mean	STD	$M > 500$	Mean	STD	$M > 500$
CS-10%	0.005	13.82	5.02	0.00	29.16	8.54	0.00	37.63	9.24	0.00
	0.01	7.15	3.08	0.00	15.38	4.88	0.00	19.31	6.01	0.00
	0.02	3.88	1.96	0.00	6.98	3.05	0.00	8.62	3.64	0.00
	0.03	2.67	1.44	0.00	4.36	2.22	0.00	5.31	2.51	0.00
	0.04	2.09	1.06	0.00	3.27	1.71	0.00	3.63	1.91	0.00
	0.05	1.68	0.84	0.00	2.42	1.51	0.00	2.78	1.80	0.00
CS-70%	0.005	30.71	10.28	0.00	71.12	13.25	0.00	94.98	14.39	0.00
	0.01	18.12	5.24	0.00	37.93	8.01	0.00	50.62	9.72	0.00
	0.02	10.06	3.74	0.00	20.73	5.33	0.00	25.04	5.41	0.00
	0.03	6.45	2.59	0.00	13.65	3.95	0.00	16.85	4.39	0.00
	0.04	4.81	2.10	0.00	9.61	3.03	0.00	12.24	3.78	0.00
	0.05	3.94	1.94	0.00	7.67	2.70	0.00	9.40	3.18	0.00
AR(1)-10%	0.005	18.86	7.24	0.00	39.05	11.13	0.00	46.60	12.93	0.00
	0.01	10.37	4.28	0.00	18.72	6.58	0.00	23.81	8.44	0.00
	0.02	4.99	2.33	0.00	9.85	3.57	0.00	11.74	4.29	0.00
	0.03	3.45	1.59	0.00	5.96	2.77	0.00	7.44	3.33	0.00
	0.04	2.40	1.22	0.00	4.15	2.29	0.00	4.67	2.54	0.00
	0.05	2.03	1.07	0.00	3.02	1.65	0.00	3.64	2.15	0.00
AR(1)-70%	0.005	44.30	13.06	0.00	97.12	17.89	0.00	123.94	20.28	0.00
	0.01	23.71	8.08	0.00	51.85	9.49	0.00	64.14	11.04	0.00
	0.02	13.55	4.93	0.00	26.79	6.21	0.00	33.91	6.82	0.00
	0.03	9.28	3.36	0.00	17.81	3.97	0.00	21.94	5.11	0.00
	0.04	6.76	2.65	0.00	13.59	3.34	0.00	16.67	3.93	0.00
	0.05	5.44	2.26	0.00	10.84	2.98	0.00	13.12	3.17	0.00
Logreg-10%	0.005	39.06	16.12	0.00	71.73	24.19	0.00	90.08	29.32	0.00
	0.01	19.95	8.71	0.00	38.31	14.02	0.00	46.67	15.70	0.00
	0.02	11.14	5.26	0.00	20.38	7.42	0.00	25.47	9.62	0.00
	0.03	7.31	3.60	0.00	13.06	5.50	0.00	16.28	6.07	0.00
	0.04	5.36	2.82	0.00	9.98	4.10	0.00	11.72	4.71	0.00
	0.05	4.60	2.56	0.00	7.85	3.38	0.00	9.53	3.78	0.00
Logreg-70%	0.005	61.33	26.34	0.00	140.31	53.29	0.00	174.82	67.77	0.00
	0.01	36.68	15.25	0.00	76.05	25.85	0.00	94.65	32.48	0.00
	0.02	21.35	7.88	0.00	40.75	13.77	0.00	51.16	16.31	0.00
	0.03	14.68	6.37	0.00	28.97	8.96	0.00	34.92	10.65	0.00
	0.04	11.76	4.29	0.00	22.35	6.44	0.00	26.23	7.47	0.00
	0.05	10.01	4.00	0.00	18.99	5.44	0.00	21.77	6.26	0.00

Table 7.2: Mean, standard deviation (STD) for selected M given different ϵ and k_0 values, and number of times $M > 500$ for different models and different ϵ 's for t -test and paired t -test with test value $\mu = 0$. The results are presented for 3 different successive validation steps $k_0 = 1, 3, 5$.

Test	Model	ϵ	$k_0 = 1$			$k_0 = 3$			$k_0 = 5$		
			Mean	STD	$M > 500$	Mean	STD	$M > 500$	Mean	STD	$M > 500$
$\mu_1 = 0$	CS-10%	0.005	2.90	2.55	0.00	8.17	6.74	0.00	11.32	8.93	0.00
		5e-04	12.78	9.70	0.00	50.41	32.28	0.00	83.97	53.85	0.00
	CS-70%	0.005	5.41	3.85	0.00	20.22	10.08	0.00	29.20	14.79	0.00
		5e-04	21.75	12.38	0.00	119.70	57.55	0.00	211.70	93.22	0.00
	AR(1)-10%	0.005	2.76	2.03	0.00	7.55	6.39	0.00	10.55	9.08	0.00
		5e-04	12.73	10.51	0.00	53.61	34.37	0.00	88.68	63.04	0.00
$\mu_1 - \mu_2 = 0$	AR(1)-70%	0.005	5.91	4.34	0.00	20.98	10.62	0.00	30.73	15.85	0.00
		5e-04	22.88	15.28	0.00	120.90	57.80	1.00	214.83	97.30	1.00
	CS-10%	0.005	3.32	2.24	0.00	10.34	7.32	0.00	15.16	11.30	0.00
		5e-04	15.75	11.31	0.00	64.97	37.27	0.00	110.12	58.83	0.00
	CS-70%	0.005	6.62	4.14	0.00	25.41	13.59	0.00	36.49	19.45	0.00
		5e-04	25.94	17.69	0.00	157.88	74.58	6.00	243.37	124.17	6.00
	AR(1)-10%	0.005	4.02	2.78	0.00	11.46	7.58	0.00	16.18	10.01	0.00
		5e-04	14.37	10.65	0.00	74.38	40.09	0.00	134.99	70.34	0.00
	AR(1)-70%	0.005	8.31	5.11	0.00	29.14	14.83	0.00	43.30	19.48	0.00
		5e-04	31.39	17.23	0.00	160.09	72.00	2.00	288.81	112.58	2.00

Table 7.3: Comparing different methods of doing factor analysis with missing data. The confidence interval is calculated using the Delta method.

	Difference with complete data			Proportion of explained variance				
	$\bar{\Sigma}$	IPCA	Complete	IPCA		MI		
				$\bar{\Sigma}$	$\tilde{\gamma}_k$	Lower-CI	Upper-CI	
5%-mean	0.08494	0.11056	0.68423	0.69627	0.68402	0.68410	0.65651	0.71169
5%-sd	0.01930	0.01897	0.00822	0.00840	0.00837	0.00837	0.00774	0.00899
10%-mean	0.14074	0.19739	0.68512	0.71071	0.68516	0.68531	0.65746	0.71316
10%-sd	0.12210	0.04949	0.00722	0.00759	0.00769	0.00768	0.00713	0.00824
50%-mean	0.91345	1.23120	0.68472	0.86345	0.68358	0.68503	0.65478	0.71529
05%-sd	0.96514	0.14911	0.00805	0.00779	0.01220	0.01212	0.01159	0.01272

Table 7.4: Factor loadings using oblimin rotation of DiF data using the estimated covariance matrix from multiply imputed data using $M = 25$ imputations.

English items*	Factor load		
	C	O	N
19. worries a lot	0.097	0.076	0.653
14. can be tense	0.181	0.101	0.640
9r**. is relaxed, handles stress well	-0.134	-0.213	0.624
39. gets nervous easily	0.017	0.038	0.701
24r. is emotionally stable, not easily upset	-0.155	-0.186	0.461
34r. remains calm in tense situations	-0.214	-0.210	0.551
4. is depressed, blue	-0.020	0.042	0.369
29. can be moody	0.139	0.113	0.323
1. is talkative	0.093	0.128	0.115
21r. tends to be quiet	-0.065	-0.034	0.021
16. generates a lot of enthusiasm	0.290	0.335	-0.002
36. is outgoing, sociable	0.084	0.255	0.092
6r. is reserved	-0.047	-0.089	-0.113
31r. is sometimes shy, inhibited	0.015	-0.185	-0.223
11. is full of energy	0.358	0.209	-0.246
26. has an assertive personality	0.335	0.145	-0.202
40. likes to reflect, play with ideas	0.394	0.395	-0.025
25. is inventive	0.343	0.435	-0.169
30. values artistic, aesthetic experiences	0.083	0.447	-0.001
5. is original, comes up with new ideas	0.291	0.391	-0.073
15. is ingenious, a deep thinker	0.453	0.293	0.098
20. has an active imagination	0.019	0.520	0.006
10. is curious about many different things	0.297	0.420	-0.117
44. is sophisticated in art, music, or literature	-0.053	0.397	-0.032
41r. has few artistic interests	-0.047	0.256	-0.093
35r. prefers work that is routine	-0.007	0.089	-0.161
3. does a thorough job	0.614	-0.067	0.011
28. perseveres until the task is finished	0.685	-0.086	0.001
18r. tends to be disorganized	0.401	-0.519	0.035
23r. tends to be lazy	0.429	-0.402	0.008
13. is a reliable worker	0.554	-0.010	0.039
33. does things efficiently	0.630	Page 219 0.015	-0.017
38. makes plans and follows through with them	0.592	0.050	-0.062
43r. is easily distracted	0.355	-0.362	-0.280

Chapter 8

Software packages

Most of the techniques and methods that are introduced throughout this thesis should be implemented as a computer program to be effective. Therefore, to make our contributions complete, we have developed R packages to implement various methodologies introduced in the previous chapters. R is a freely available programming language for statistical computation and graphics. We have selected R, as it is well received not only among statisticians, but also many users of statistical methods from different areas.

Figure 8.1 shows results of the 16th annual KDnuggets Software Poll based on opinions of about 2800 voters, who chose from a record number of 93 different tools. As we may see, R was by far the most favorite tool. In the recent years, Python is attracting more users, but R is still among the top data science tools.

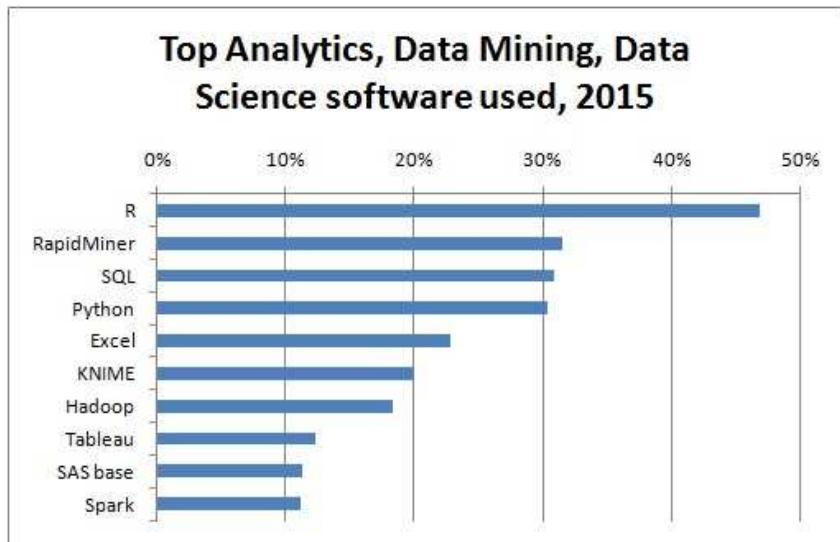


Figure 8.1: Top analytical tools for data science (source: <https://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html>).

We have implemented our proposed methods in 5 different R packages as follows:

- **fastCS**: implements various methods proposed for estimating clustered data with a covariance matrix with compound-symmetry structure.
- **fastAR1**: implements various methods proposed for estimating clustered data with a covariance matrix with AR(1) structure.
- **miscVSS**: implements various methods to perform vertical sample splitting. That includes both random and structured version of this approach.

- **imi**: implements the iterative multiple imputation technique.
- **mifa**: implements our proposal for explanatory factor analysis for incomplete data using multiple imputations.

The source codes for all of these R packages are publicly and freely available via author's github page (<https://github.com/vahidnassiri>), and can easily be installed from there using the command `install_github` which is available via R package `devtools`. Therefore, one needs to first install this R package using the following command:

```
install.packages("devtools")
```

Once `devtools` is installed, one can use the following command to install any of these packages directly from R environment. Some of the packages are dependent on other R packages. Such dependencies can also be installed automatically in case they user does not already have them installed.

R package **fastCS**

The R package `fastCS` provides functions to:

- Estimate parameters of balanced clustered data with CS structure covariance matrix
- Combine estimated and their variance from different splits using:

- ▷ parameter-free weights
 - ▷ scalar weights
 - ▷ optimal weights
- Perform cluster by cluster analysis

This covers our contributions in Section 4.

R package fastAR1

The R package **fastAR1** provides function to:

- Estimate parameters of balanced clustered data with AR1 covariance matrix
- Estimate variance of the estimates above
- Combine estimated parameters and their variances from different splits.

R package miscVSS

The R package **miscVSS** provides function to:

- Perform random vertical splitting using iterative multiple outputation with acceleration
- Prepare data for structured vertical splitting

- Combine the estimated parameters and variances from vertical splitting using appropriate combination rules.

This covers our contributions in Chapters 5, and 6.

R package `imi`

The R package `imi` provides function to:

- Impute incomplete data and determine the number of sufficient imputed datasets using iterative multiple imputation idea for:
 - ▷ Linear models
 - ▷ Generalized linear models
 - ▷ Student's *t*-test
- Study the convergence of the procedure for a given number of imputed datasets
- Visualize the convergence procedure and obtained results
- Combine estimates and covariance matrices from multiple imputation using appropriate rules.

This covers our contributions in Section 7.

R package `mifa`

The R package `mifa` provides function to:

- Impute the incomplete data, determine the number of sufficient imputed datasets, and perform exploratory factor analysis on them.
- Compute confidence interval for the proportion of explained variance for given numbers of factors using Fieller's method (?).
- Compute confidence interval for the proportion of explained variance for given numbers of factors using bootstrap.
- Combine estimates and covariance matrices from multiple imputation using appropriate rules.

This covers our contributions in Section 7.

Chapter 9

Conclusions

Chapter 10

Appendices

Incompleteness in the Compound-symmetry Model

Based on (4.1) and the multiplicity of the cluster sizes, the sufficient statistics are:

$$W_{1k} = \sum_{i=1}^{c_k} \sum_{j=1}^{n_k} Y_{ij}^{(k)}, \quad (\text{A.1})$$

$$W_2 = \sum_{k=1}^L \sum_{i=1}^{c_k} \sum_{j=1}^{n_k} \left(Y_{ij}^{(k)} \right)^2, \quad (\text{A.2})$$

$$W_{3k} = \sum_{i=1}^{c_k} \left(\sum_{j=1}^{n_k} Y_{ij}^{(k)} \right)^2, \quad (\text{A.3})$$

$$W_{4k} = c_k. \quad (\text{A.4})$$

The conditional and marginal expectations of (A.1)–(A.4) are:

$$E(W_{1k}|c_k) = c_k n_k \mu, \quad (\text{A.5})$$

$$E(W_{1k}) = N \mu \pi_k n_k, \quad (\text{A.6})$$

$$E(W_2|c_k) = \sum_{k=1}^L c_k n_k (\sigma^2 + d + \mu^2), \quad (\text{A.7})$$

$$E(W_2) = N(\sigma^2 + d + \mu^2) \sum_{k=1}^L \pi_k n_k, \quad (\text{A.8})$$

$$E(W_{3k}|c_k) = c_k \left\{ n_k (\sigma^2 + d + \mu^2) + n_k (n_k - 1) (d + \mu^2) \right\}, \quad (\text{A.9})$$

$$E(W_{3k}) = N \pi_k n_k \left\{ (\sigma^2 + d + \mu^2) + (n_k - 1) (d + \mu^2) \right\}, \quad (\text{A.10})$$

$$E(W_{4k}) = N \pi_k. \quad (\text{A.11})$$

Group all sufficient statistics in \mathbf{W} and define a function

$$g(\mathbf{w}) = \sum_{k=1}^L \lambda_k \frac{w_{1k}}{w_{4k}}. \quad (\text{A.12})$$

Then,

$$E\{g(\mathbf{W}|\mathbf{W}_4)\} = \sum_{k=1}^L \lambda_k \frac{E(W_{1k}|W_{4k})}{W_{4k}} = \mu \sum_{k=1}^L \lambda_k n_k,$$

and hence

$$E\{g(\mathbf{W})\} = \mu \sum_{k=1}^K \lambda_k n_k.$$

Thus, every solution $\boldsymbol{\lambda} \perp \mathbf{n}$, where $\mathbf{n} = (n_1, \dots, n_K)'$, provides a counterexample, establishing incompleteness.

Such a vector $\boldsymbol{\lambda}$ exists if and only if $K \geq 2$, for which it is assumed that at least two $c_k > 0$ (i.e., at least two different cluster sizes occur).

Likelihood-based Estimation of the CS Model

Score Functions

The score function has components:

$$\frac{\partial \ell}{\partial \mu_k} = \frac{1}{\sigma_k^2 + n_k d_k} \left(\sum_{i=1}^{c_k} \sum_{j=1}^{n_i} y_{ij}^{(k)} - c_k n_k \mu_k \right), \quad (\text{A.13})$$

$$\frac{\partial \ell}{\partial \sigma_k^2} = \frac{-c_k n_k}{2\sigma_k^2} \cdot \frac{\sigma_k^2 + (n_k - 1)d_k}{\sigma_k^2 + n_k d_k} + \frac{c_k n_k S_k}{2\sigma_k^4} \quad (\text{A.14})$$

$$- \frac{d_k (2\sigma_k^2 + n_k d_k) c_k n_k T_k}{2\sigma_k^4 (\sigma_k^2 + n_k d_k)^2}, \quad (\text{A.15})$$

$$\frac{\partial \ell}{\partial d_k} = \frac{-c_k n_k}{2(\sigma_k^2 + n_k d_k)} + \frac{c_k n_k T_k}{2(\sigma_k^2 + n_k d_k)^2}, \quad (\text{A.16})$$

with

$$S_k = \frac{1}{c_k n_k} Q_k = \frac{1}{c_k n_k} \sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} \mathbf{Z}_i^{(k)}, \quad (\text{A.17})$$

$$T_k = \frac{1}{c_k n_k} R_k = \frac{1}{c_k n_k} \sum_{i=1}^{c_k} \mathbf{Z}_i^{(k)'} \mathbf{J}_{\mathbf{n}_k} \mathbf{Z}_i^{(k)}. \quad (\text{A.18})$$

Lack of Closed-form Solution when $K \geq 2$

The lack of a closed form when $K \geq 2$ is well known, but we highlight a few relevant features here. More detail is given in Supplementary Materials 10. Function (4.4) can be turned into the log-likelihood kernel for the conventional situation where there is a common mean parameter and common variance components across all cluster sizes, i.e., $\ell(\mu, \sigma^2, d)$. The score functions follow from

summing the terms in (A.13)–(A.16) across cluster sizes:

$$\frac{\partial \ell}{\partial \mu} = \sum_{k=1}^K \frac{\partial \ell}{\partial \mu_k} \Big|_{\mu_k=\mu}, \quad \frac{\partial \ell}{\partial \sigma^2} = \sum_{k=1}^K \frac{\partial \ell}{\partial \sigma_k^2} \Big|_{\sigma_k^2=\sigma^2}, \quad \frac{\partial \ell}{\partial d} = \sum_{k=1}^K \frac{\partial \ell}{\partial d_k} \Big|_{d_k=d}. \quad (\text{A.19})$$

Solving the score equation in (A.19) for the mean, using that

$$\Sigma_{n_k}^{-1} = \frac{1}{\sigma^2} I_{n_k} - \frac{d}{\sigma^2(\sigma^2 + n_k d)} J_{n_k},$$

leads to the identity:

$$\hat{\mu} = \frac{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d} \bar{Y}^{(k)}}{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d}} = \frac{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d} \hat{\mu}_k}{\sum_{k=1}^K \frac{n_k c_k}{\sigma^2 + n_k d}}, \quad (\text{A.20})$$

where $\hat{\mu}_k$ as in (4.5). For the variance components, only implicit identities follow; they are functions of (A.17)–(A.18). These take the form of high-degree polynomials, for which no general explicit solution exists. While (A.20) is explicit, it is a weighted average of the cluster-size specific averages $\bar{Y}^{(k)}$, with weights depending on the variance components. This, combined with the result for the variance components, implies that there is no explicit solution, unless the variance components are known or the cluster size is constant.

Full Likelihood

Referring to the conventional situations, i.e. $\ell(\mu, \sigma^2, d)$ in (4.2) and the score equation in (A.19), also second derivatives can be calculated:

$$\frac{\partial^2 \ell}{\partial \mu^2} = \sum_{k=1}^K \frac{-c_k n_k}{\sigma^2 + n_k d}$$

$$\frac{\partial^2 \ell}{\partial \sigma^2 \partial \mu} = \sum_{k=1}^K \frac{-1}{(\sigma^2 + n_k d)^2} \left(\sum_{i=1}^{c_k} \sum_{j=1}^{n_k} \mathbf{y}_{ij}^{(k)} - c_k n_k \mu_k \right)$$

$$\frac{\partial^2 \ell}{\partial d \partial \mu_k} = \sum_{k=1}^K \frac{-n_k}{(\sigma^2 + n_k d)^2} \left(\sum_{i=1}^{c_k} \sum_{j=1}^{n_k} \mathbf{y}_{ij}^{(k)} - c_k n_k \mu_k \right)$$

$$\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} = \sum_{k=1}^K \left(\frac{-1}{\sigma^4} + \frac{d(2\sigma^2 + n_k)}{\sigma^4(\sigma^2 + n_k)} \right)$$

$$\frac{\partial^2 \ell}{(\partial \sigma^2)^2} = \sum_{k=1}^K \left(\frac{c_{kn_k}}{2\sigma^4} \cdot \frac{\sigma^2 + (n_k - 1)d}{\sigma^2 + n_k d} - \frac{c_k n_k}{2\sigma^2} \cdot \frac{d}{(\sigma^2 + n_k d)^2} - \frac{c_{kn_k} S_k}{\sigma^6} \right.$$

$$\left. - d c_k n_k T_k \frac{\sigma^2(\sigma^2 + n_k d) - (2\sigma^2 + n_k d)^2}{\sigma^6(\sigma^2 + n_k d)^3} \right)$$

$$\frac{\partial^2 \ell}{\partial d \partial \sigma^2} = \sum_{i=1}^K \left(\frac{c_k n_k}{2(\sigma^2 + n_k d)} - \frac{c_k n_k T_k}{\sigma^4} \cdot \frac{(\sigma^2 + n_k d)^2 - n_k d(2\sigma^2 + n_k d)}{(\sigma^2 + n_k d)^3} \right)$$

$$\frac{\partial^2 \ell}{\partial \mu \partial d} = \sum_{k=1}^K \frac{-n_k}{(\sigma^2 + n_k d)^2} \sum_{i=1}^{c_k} \sum_{j=1}^{n_k} Z_{ij}^{(k)}$$

$$\frac{\partial^2 \ell}{\partial \sigma^2 \partial d} = \sum_{k=1}^K \left(\frac{c_k n_k}{2(\sigma^2 + n_k d)^2} - \frac{c_k n_k T_k}{(\sigma^2 + n_k d)^3} \right)$$

$$\frac{\partial^2 \ell}{\partial d^2} = \sum_{k=1}^K \left(\frac{c_k n_k^2}{2(\sigma^2 + n_k d)^2} - \frac{c_k n_k^2 T_k}{(\sigma^2 + n_k d)^3} \right).$$

equals:

$$L \propto \prod_{i=1}^k \frac{1}{(2\pi)^{n_1/2} |\Sigma_{n_1}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{y}_{i1} - \boldsymbol{\mu}_{n_1})' \Sigma_{n_1}^{-1} (\mathbf{y}_{i1} - \boldsymbol{\mu}_{n_1}) \right\} \\ \times \left[\frac{\Phi(\alpha + \mathbf{y}'_{i1} \boldsymbol{\beta})}{\Phi \left(\frac{\alpha + \boldsymbol{\mu}'_{n_1} \boldsymbol{\beta}}{\sqrt{1 + \boldsymbol{\beta}' \Sigma_{n_1} \boldsymbol{\beta} / n_1}} \right)} \right] \quad (\text{A.22})$$

$$\times \prod_{i=k+1}^N \frac{1}{(2\pi)^{n_2/2} |\Sigma_{n_2}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_{n_2})' \Sigma_{n_2}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_{n_2}) \right\} \\ \times \left[\frac{1 - \Phi(\alpha + \mathbf{y}'_{i1} \boldsymbol{\beta})}{1 - \Phi \left(\frac{\alpha + \boldsymbol{\mu}'_{n_1} \boldsymbol{\beta}}{\sqrt{1 + \boldsymbol{\beta}' \Sigma_{n_1} \boldsymbol{\beta} / n_1}} \right)} \right] \quad (\text{A.24})$$

$$\ell \propto -\frac{1}{2} \sum_{i=1}^k \left\{ \ln |\Sigma_{n_1}| + (\mathbf{y}_i - \boldsymbol{\mu}_{n_1})' \Sigma_{n_1}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_{n_1}) \right\} \quad (\text{A.25})$$

$$-k \ln \Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \quad (\text{A.26})$$

$$-\frac{1}{2} \sum_{i=k+1}^N \left\{ \ln |\Sigma_{n_2}| + (\mathbf{y}_i - \boldsymbol{\mu}_{n_2})' \Sigma_{n_2}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_{n_2}) \right\} \quad (\text{A.27})$$

$$-(N-k) \ln \left\{ 1 - \Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \right\}, \quad (\text{A.28})$$

with $\tilde{\alpha} = \frac{\alpha}{\sqrt{1 + \boldsymbol{\beta}' \Sigma_{n_1} \boldsymbol{\beta} / n_1}}$ and $\tilde{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}}{\sqrt{1 + \boldsymbol{\beta}' \Sigma_{n_1} \boldsymbol{\beta} / n_1}}$

The corresponding score equations are:

$$\begin{aligned}\frac{\partial \ell}{\partial \mu} &= \frac{1}{\sigma^2 + n_1 d} \left(\sum_{i=1}^k \sum_{j=1}^{n_1} \mathbf{y}_{ij} - k n_1 \mu \right) - k n_1 \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}} \frac{\phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})}{\Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})} \\ &\quad + \frac{1}{\sigma^2 + n_2 d} \left(\sum_{i=k+1}^N \sum_{j=1}^{n_2} \mathbf{y}_{ij} - (N-k) n_2 \mu \right) \end{aligned} \quad (\text{A.30})$$

$$-(N-k) n_2 \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}} \frac{\phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})}{\Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})}, \quad (\text{A.31})$$

with $\frac{\partial \ell}{\partial \sigma^2}$ and $\frac{\partial \ell}{\partial d}$ identical to (A.15) and (A.16). The components

of the Hessian are:

$$\frac{\partial^2 \ell}{\partial \mu^2} = \frac{-kn_1}{\sigma^2 + n_1 d} - \frac{(N-k)n_2}{\sigma^2 + n_2 d} \quad (\text{A.32})$$

$$-kn_1 \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}} \left[\frac{-\Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot \phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot (\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}}}{\Phi^2(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})} \right. \\ \left. - \frac{\phi^2(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}}}{\Phi^2(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})} \right] \quad (\text{A.34})$$

$$+ (N-k)n_2 \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}} \quad (\text{A.35})$$

$$\times \left[\frac{-(1 - \Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}})) \cdot \phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot (\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}}}{(1 - \Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}))^2} \right. \\ \left. + \frac{\phi^2(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}) \cdot \mathbf{j}'_{\mathbf{n}_1} \tilde{\boldsymbol{\beta}}}{(1 - \Phi(\tilde{\alpha} + \boldsymbol{\mu}'_{n_1} \tilde{\boldsymbol{\beta}}))^2} \right] \quad (\text{A.36})$$

$$\frac{\partial^2 \ell}{\partial \sigma^2 \partial \mu} = \frac{-1}{(\sigma^2 + n_1 d)^2} \left(\sum_{i=1}^k \sum_{j=1}^{n_1} \mathbf{y}_{ij} - kn_1 \mu \right) \quad (\text{A.38})$$

$$- \frac{1}{(\sigma^2 + n_2 d)^2} \left(\sum_{i=k+1}^N \sum_{j=1}^{n_2} \mathbf{y}_{ij} - (N-k)n_2 \mu \right) \quad (\text{A.39})$$

$$\frac{\partial^2 \ell}{\partial d \partial \mu} = \frac{-n_1}{(\sigma^2 + n_1 d)^2} \left(\sum_{i=1}^k \sum_{j=1}^{n_1} \mathbf{y}_{ij} - kn_1 \mu \right) \quad (\text{A.40})$$

$$- \frac{n_2}{(\sigma^2 + n_2 d)^2} \left(\sum_{i=k+1}^N \sum_{j=1}^{n_2} \mathbf{y}_{ij} - (N-k)n_2 \mu \right) \quad (\text{A.41})$$

with $\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2}$, $\frac{\partial^2 \ell}{(\partial \sigma^2)^2}$, $\frac{\partial^2 \ell}{\partial d \partial \sigma^2}$, $\frac{\partial^2 \ell}{\partial \mu \partial d}$, $\frac{\partial^2 \ell}{\partial \sigma^2 \partial d}$, and $\frac{\partial^2 \ell}{\partial (d)^2}$ indentical to (10), (10), (10), (10), and (10).

Pseudo-likelihood for Split Samples

General Considerations

Informally, a pseudo-likelihood function is one that replaces a computationally intractable or slow-to-maximize likelihood function, with another function that still produces a consistent and asymptotically normal estimator when maximised, i.e., by setting the first derivatives of the log pseudo-likelihood equal to zero and solving the resultant pseudo score equations. An early reference is ?, and details on various forms of pseudo-likelihood can be found in (? , Ch. 9, 12, 21, 22, 24, and 25). In the current clustered setting, the likelihood contribution of a cluster is often replaced by a product of contributions for various sub-vectors. In some cases, such a sub-vector can be conditioned upon another sub-vector.

? and ? used pseudo-likelihood to fit mixed models to high-dimensional multivariate longitudinal data. They supplemented the standard method with an additional device. They first replaced a set of M longitudinal sequences by the $M(M - 1)/2$ longitudinal pairs. This in itself is a standard application of pseudo-likelihood. They then assumed that each pair has its own parameter vector. Symbolically, this can be written as:

$$p\ell(\theta) \equiv p\ell(\mathbf{y}_{1i}, \mathbf{y}_{2i}, \dots, \mathbf{y}_{Mi} | \theta) = \sum_{r < s} \ell(\mathbf{y}_{ri}, \mathbf{y}_{si} | \theta_{rs}), \quad (\text{A.42})$$

where \mathbf{Y}_{ri} is the r th sequence for subject i . In (A.42), θ results from stacking all $M(M - 1)/2$ pair-specific parameter vectors θ_{rs} . The actual parameter vector of interest is θ^* , the set of non-redundant

parameters is θ .

To obtain θ^* , ? take averages of all available estimates for that specific parameter, implying that $\widehat{\theta}^* = A\widehat{\theta}$ for an appropriate linear combination matrix A . Further, combining this step with general pseudo-likelihood inference, a sandwich estimator is used:

$$\sqrt{N}(\widehat{\theta}^* - \theta^*) = \sqrt{N}(A\widehat{\theta} - A\theta) \stackrel{\text{approx.}}{\sim} N(\mathbf{0}, AI_0^{-1}I_1I_0^{-1}A'), \quad (\text{A.43})$$

where

$$I_0(\boldsymbol{\theta}) = E \left[\frac{\partial^2 p\ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}' \partial \boldsymbol{\theta}} \right], \quad I_1(\boldsymbol{\theta}) = E \left[\left(\frac{\partial p\ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)' \cdot \frac{\partial p\ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]. \quad (\text{A.44})$$

? took a very similar route to partition a potentially large sample into sub-samples.

To fix ideas, consider log-likelihood (4.4). When used as an instrument to estimate a single vector (μ, σ^2, d) , this function can be viewed a pseudo-likelihood. This setting can be generalized by assuming that a dataset, consisting of repeated measures per subject, is divided into K subgroups, each containing c_k independent replicates. Consider the pseudo-likelihood:

$$p\ell(\boldsymbol{\theta}) = \sum_{k=1}^K \ell(\boldsymbol{\theta}_k | \mathbf{y}_1^{(k)}, \dots, \mathbf{y}_{c_k}^{(k)}). \quad (\text{A.45})$$

While the underlying principle is similar to (A.42), it is not identical. The similarities are: (1) all $\boldsymbol{\theta}_k$ are assumed to be different, allowing
Page 240

for separate, even parallel, estimation; (2) $\boldsymbol{\theta}$ stacks all vectors $\boldsymbol{\theta}_k$; (3) the parameter of interest $\boldsymbol{\theta}^*$, is found from an appropriate combination of the $\boldsymbol{\theta}_k$. Parallel estimation was also followed by ? and ?.

There are important differences, however. Here, and in the remainder of the article, we assume that $\ell(\boldsymbol{\theta}_k | \mathbf{y}_1^{(k)}, \dots, \mathbf{y}_{c_k}^{(k)})$ is the likelihood that we would have, should group k be the only one in the data. That is, the individual likelihood contributions are not altered, rather the data are partitioned. This is similar to the independent partitioning done by ?. In line with their derivations, (A.43) can also be used here. Given that $\ell_k(\boldsymbol{\theta}_k)$ is a genuine likelihood, its contributions to $I_0(\boldsymbol{\theta})$ and $I_1(\boldsymbol{\theta})$ are identical, up to the sign. As a result, $I_0(\boldsymbol{\theta})^{-1}I_1(\boldsymbol{\theta})I_0(\boldsymbol{\theta})^{-1} = -I_0(\boldsymbol{\theta})^{-1}$, a block-diagonal matrix with blocks of the form $I_0(\boldsymbol{\theta}_k)$. We now turn to the split-sample case.

Pseudo-likelihood for Split Sample

? chose

$$A = \frac{1}{K}(I, \dots, I) \quad (\text{A.46})$$

to pass from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^*$. This is a sensible choice in the i.i.d. setting (e.g., when all clusters in a CS model have the same size) and with the same number of subjects per sub-sample. The estimator and

precision estimator then become:

$$\hat{\boldsymbol{\theta}}^* = \frac{1}{K} \sum_{k=1}^K \hat{\boldsymbol{\theta}}_k, \quad (\text{A.47})$$

$$\text{var}(\hat{\boldsymbol{\theta}}^*) = \frac{1}{K} H_{\hat{\boldsymbol{\theta}}}^{-1}, \quad (\text{A.48})$$

with $H_{\hat{\boldsymbol{\theta}}}^{-1} = -I_0(\boldsymbol{\theta}_k)$. In this special case, the expected information matrices are identical. Alternatively, one can use the observed information matrices, and then use instead:

$$\frac{1}{K^2} \sum_{k=1}^K \mathcal{H}_{\hat{\boldsymbol{\theta}}, k}^{-1}. \quad (\text{A.49})$$

where $\mathcal{H}_{\hat{\boldsymbol{\theta}}, k}$ is the observed information for sub-sample k .

In this particular case, pseudo-likelihood produces the same estimator as full likelihood. This stems from the fact that all subjects follow the same distribution, in contrast to, for example, the setting set out at the start of Section 4. Should the subjects have identical distributions, but with c_k variables, the above results can be modified accordingly. For example, (A.47) would be replaced by

$$\hat{\boldsymbol{\theta}}^* = \sum_{k=1}^K \frac{c_k}{N} \hat{\boldsymbol{\theta}}_k,$$

with similar modification to precision estimation. In this case, full likelihood would be obtained.

In the next section, we will consider the more general case where different subjects may have a different distribution such as, for
Page 242

example, the CS case with a different number of measurements per cluster.

Derivation of Optimal Scalar Weights for Compound-symmetry Case

To find the optimal scalar weight with minimum variance for μ we use the method of Lagrange with the constraint that the weights a_k need to sum to 1:

$$Q = \sum_{k=1}^K a_k^2 \frac{\sigma^2 + n_k d}{c_k n_k} - \lambda \left(\sum_{k=1}^K a_k - 1 \right). \quad (\text{A.50})$$

Solving the first partial derivative we become an expression for a_k involving λ . Summing this one produces an expression for λ , leading to the complete formula for a_k . Precisely, the system of equations is:

$$\frac{\partial Q}{\partial a_k} = 2a_k \frac{\sigma^2 + n_k d}{c_k n_k} - \lambda = 0, \quad (\text{A.51})$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{k=1}^K a_k - 1 = 0, \quad (\text{A.52})$$

or, alternatively:

$$a_k = \frac{\lambda}{2} \frac{c_k n_k}{\sigma^2 + n_k d}, \quad (\text{A.53})$$

$$\lambda = \left(\frac{1}{2} \sum_{k=1}^K \frac{c_k n_k}{\sigma^2 + n_k d} \right)^{-1}, \quad (\text{A.54})$$

and hence

$$a_k = \frac{\frac{c_k n_k}{\sigma^2 + n_k d}}{\sum_{m=1}^K \frac{c_m n_m}{\sigma^2 + n_m d}}.$$

In the same manner, expressions for b_k and g_k can be found, as we will show next. For σ^2 :

$$Q = 2\sigma^4 \sum_{k=1}^K b_k^2 \frac{1}{c_k(n_k - 1)} - \lambda \left(\sum_{k=1}^K b_k - 1 \right),$$

producing the system:

$$\frac{\partial Q}{\partial b_k} = \frac{4\sigma^4 b_k}{c_k(n_k - 1)} - \lambda = 0, \quad (\text{A.55})$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{k=1}^K b_k - 1 = 0, \quad (\text{A.56})$$

which can be rewritten as:

$$4\sigma^4 b_k = \lambda c_k(n_k - 1), \quad (\text{A.57})$$

$$\lambda = \frac{4\sigma^4}{\sum_{k=1}^K c_k(n_k - 1)}, \quad (\text{A.58})$$

and finally:

$$b_k = \frac{c_k(n_k - 1)}{\sum_{m=1}^K c_m(n_m - 1)}.$$

For d , the objective function is:

$$Q = \sum_{k=1}^K g_k^2 v_k - \lambda \left(\sum_{k=1}^K g_k - 1 \right),$$

with

$$v_k = \frac{2}{c_k n_k} \left(\frac{\sigma^4}{n_k - 1} + 2d\sigma^2 + n_k d^2 \right).$$

The system of equations now is:

$$\frac{\partial Q}{\partial g_k} = 2g_k v_k - \lambda = 0, \quad (\text{A.59})$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{k=1}^K g_k - 1 = 0, \quad (\text{A.60})$$

leading to:

$$g_k = \lambda \frac{1}{2v_k}, \quad (\text{A.61})$$

$$\lambda = \frac{2}{\sum_{k=1}^K \frac{1}{v_k}}, \quad (\text{A.62})$$

giving the solution:

$$g_k = \frac{\frac{c_k n_k}{\frac{\sigma^4}{n_k - 1} + 2d\sigma^2 + n_k d^2}}{\sum_{m=1}^K \frac{c_m n_k}{n_m - 1} + 2d\sigma^2 + n_m d^2}.$$

Cluster-by-cluster Analysis

We study the case of the most extreme partitioning, i.e., where each of the clusters is analyzed separately. This can be relevant in cases with perhaps a limited number of very to extremely large clusters. This means that $c_k \equiv 1$ throughout. Clearly, the n_k will then no longer be unique. We will examine this case in detail, and contrast a first weighted estimator with an *ad hoc* one.

The Weighted Estimator for the Cluster-by-cluster Case

The estimator follows from setting $c_k \equiv 1$ and hence $K \equiv N$ throughout. For example, this special case can easily be considered for all expressions in Sections 6.1.1-6.1.3. Because c_k enters the inverse of the variance-covariance matrix multiplicatively, as is seen from (4.8)-(4.9), the optimal estimator that is obtained when each cluster is considered to be its own stratum, is identical to the one obtained when strata are defined in terms of all clusters of a given size. The same is true for the scalar weights.

It is insightful to consider in more detail the special case where further the cluster sizes are all identical to n . One then easily obtains:

$$\hat{\mu} = \frac{1}{Nn} \sum_{i=1}^N \sum_{j=1}^n Y_{ij}, \quad (\text{A.63})$$

$$\hat{\sigma}^2 = \frac{1}{Nn(n-1)} \left(n \sum_{i=1}^N \mathbf{Z}'_i \mathbf{Z}_i - \sum_{i=1}^N \mathbf{Z}'_i J_n \mathbf{Z}_i \right) \quad (\text{A.64})$$

$$= \frac{1}{Nn(n-1)} (nQ - R), \quad (\text{A.65})$$

$$\hat{d} = \frac{1}{Nn(n-1)} \left(\sum_{i=1}^N \mathbf{Z}'_i J_n \mathbf{Z}_i - \sum_{i=1}^N \mathbf{Z}'_i \mathbf{Z}_i \right) \quad (\text{A.66})$$

$$= \frac{1}{Nn(n-1)} (Q - R), \quad (\text{A.67})$$

with obvious notation for Q and R , inspired by (A.17)-(A.18).
The corresponding variance-covariance elements, similar in spirit to
Page 246

(4.8)-(4.9), are:

$$\text{var}(\hat{\mu}) = \frac{\sigma^2 + nd}{Nn}, \quad (\text{A.68})$$

$$\text{var} \begin{pmatrix} \hat{\sigma^2} \\ \hat{d} \end{pmatrix} = \begin{pmatrix} \frac{2\sigma^4}{N(n-1)} & -\frac{2\sigma^4}{Nn(n-1)} \\ -\frac{2\sigma^4}{Nn(n-1)} & \frac{2}{Nn} \left[\frac{\sigma^4}{n-1} + 2\sigma^2 d + nd^2 \right] \end{pmatrix} \quad (\text{A.69})$$

This estimator coincides with the MLE, as is known from ?.

A Two-stage Estimator for Compound Symmetry

In linear mixed models, there is a method of estimation, sometimes called the two-stage approach (??), in which each cluster is analyzed separately to begin with, using linear regression, after which the cluster-specific parameters are summarized into fixed effects. Although the above cluster-by-cluster analysis is superficially similar to this, it is not equivalent. In particular, there is no bias (as can be seen in the two stage method), and the maximum likelihood estimator is recovered.

This approach is most useful when cluster sizes are not constant, and in models that are more complex than compound symmetry. However, to gain some insight, we develop the details of the method for the CS model with constant cluster size.

For the mean, (A.63) is retained, as the average of the cluster-

specific averages \bar{Y}_i . Further, define:

$$s^2 = \frac{1}{Nn} \sum_{i=1}^N \sum_{j=1}^n (Y_{ij} - \bar{Y}_i)^2, \quad (\text{A.70})$$

$$t^2 = \frac{1}{N} \sum_{i1}^N (\bar{Y}_i - \hat{\mu})^2. \quad (\text{A.71})$$

Straightforward algebra shows:

$$E(s^2) = \frac{n-1}{n} \sigma^2, \quad (\text{A.72})$$

$$E(t^2) = \frac{N-1}{N} \left(d + \frac{1}{n} \sigma^2 \right). \quad (\text{A.73})$$

Should n and N approach infinity, then it follows that s^2 and t^2 are asymptotically unbiased estimators for σ^2 and d , respectively. However, this is not always reasonable. In applications such as the NTP data (Section 2), it is fair to say that the cluster size has a biological upper limit. In other situations, however, such as meta-analyses, it is sensible to assume that both n and N approach infinity.

In the next section, we will study the consequences of removing the bias. For now, a small, obvious modification is:

$$s_*^2 = \frac{n}{n-1} s^2, \quad (\text{A.74})$$

$$t_*^2 = \frac{N}{N-1} t^2. \quad (\text{A.75})$$

Now, s_*^2 is unbiased, while $E(t_*^2) = d + \sigma^2/n$, the bias σ^2/n can be made to disappear asymptotically provided it is sensible to let n

grow large.

It is of interest to consider the variance-covariance structure of the estimators s^2 , t^2 , s_*^2 , and t_*^2 , as well as to make relative efficiency considerations. This will be done next.

Connections Between Estimators

Comparing algebraic expressions (A.65)–(A.67) with (A.70)–(A.71), leads to the linear relationships:

$$s^2 = \frac{n-1}{n} \widehat{\sigma^2} + 0 \cdot \widehat{d}, \quad (\text{A.76})$$

$$t^2 = \frac{N-1}{Nn} \widehat{\sigma^2} + \frac{N-1}{N} \widehat{d}. \quad (\text{A.77})$$

Relationships (A.76)–(A.77) can be combined with (A.69) to produce:

$$\text{var} \begin{pmatrix} s^2 \\ t^2 \end{pmatrix} = \begin{pmatrix} \frac{2(n-1)\sigma^4}{Nn^2} & 0 \\ 0 & \frac{2(N-1)^2}{N^2 n} \left[\frac{\sigma^4}{n} + 2\sigma^2 d + nd^2 \right] \end{pmatrix} \quad (\text{A.78})$$

and, similarly,

$$\text{var} \begin{pmatrix} s_*^2 \\ t_*^2 \end{pmatrix} = \begin{pmatrix} \frac{2\sigma^4}{N(n-1)} & 0 \\ 0 & \frac{2}{Nn} \left[\frac{\sigma^4}{n} + 2\sigma^2 d + nd^2 \right] \end{pmatrix}. \quad (\text{A.79})$$

From its definition it follows that $s_*^2 \equiv \widehat{\sigma^2}$. The same is not true for t_*^2 . One reason to consider it nevertheless is its independence from s_*^2 . Indeed, $(\widehat{\mu}, s_*^2 \equiv \widehat{\sigma^2}, t_*^2)'$ is an estimator with mutually independent components. While the same is true when s^2 and t^2 are used instead, the biases are larger.

For this case then, the choice between \hat{d} and t_*^2 is in terms of a trade-off between efficiency and independence.

To gauge the efficiency loss when using t_*^2 , the mean squared error is:

$$MSE(t_*^2) = \frac{2}{Nn} \left(\frac{\sigma^4}{n} + 2\sigma^2 d + nd^2 \right) + \frac{1}{n^2} \sigma^4,$$

and hence the relative MSE:

$$RMSE(t^2; \hat{d}) = \frac{2 \left(\frac{\sigma^4}{n} + 2\sigma^2 d + nd^2 \right) + \frac{N}{n} \sigma^4}{2 \left(\frac{\sigma^4}{n-1} + 2\sigma^2 d + nd^2 \right)}, \quad (\text{A.80})$$

which approaches infinity when N does, while n would remain constant. In other words, this estimator is inconsistent unless it is being applied in situations where n can also be considered to be large.

There are three distinct situations. First, when $N/n = \lambda n + o(n)$, for some λ , i.e., when N is of the order of n^2 , then, based on (A.80), the ARE is $2(d^2 + \lambda\sigma^4)/[2d^2]$. The magnitude of the efficiency loss depends on sizes of the parameters involved. Second, when $\mathcal{O}(N) < \mathcal{O}(n^2)$, the ARE equals 1. This includes the cases where N is constant, $N = n^{1/2}$, $N = n$, and $N = n^{3/2}$, for example. A constant or slowly increasing N is plausible in a meta-analytic context. Third, if N/n increases too quickly, i.e., $\mathcal{O}(N/n) > \mathcal{O}(n)$, then the estimator t_*^2 is inconsistent. This is the case, in particular, for bounded n .

The estimators s^2 and t^2 can be combined linearly to produce
Page 250

unbiased estimators. In other words, based on (A.72)–(A.73), the following corrections can be applied to (A.70)–(A.71):

$$s_{\text{corr}}^2 = \frac{n}{n-1} s^2, \quad (\text{A.81})$$

$$t_{\text{corr}}^2 = \frac{N}{N-1} t^2 - \frac{N}{(n-1)(N-1)} s^2. \quad (\text{A.82})$$

Interestingly, this requirement reproduces (A.76)–(A.77): the requirement of an unbiased estimator reproduces $\widehat{\sigma^2}$ and \widehat{d} , presented in (A.65)–(A.67) and hence also with their variance.

Details About the First Simulation Study

The simulation study, summarized in Section 7, is described in detail here.

Simulation Method

The design of the simulation study is as follows.

- Each generated set of data consists of c_k clusters of size n_k , for $k = 1, \dots, K$. We choose $K = 4$ throughout.
- For a generated set of data, the splitting is done by placing all clusters of a given size in one sub-sample.
- The CS model parameters are $\mu = 0$, $d = 1$, and $\sigma^2 = 2$.
- After estimating the three model parameters within each sub-sample, they are combined using the following weighting

methods: (a) equal, (b) proportional, where the weights are

$$w_k = \frac{c_k}{\sum_{\ell=1}^4 c_\ell},$$

and (c) size-proportional, where the weights for μ and d are:

$$w_k \frac{c_k n_k}{\sum_{\ell=k}^4 c_\ell n_\ell},$$

while for σ^2 we take:

$$w_k = \frac{c_k(n_k - 1)}{\sum_{\ell=1}^4 c_\ell(n_\ell - 1)}.$$

- Per setting, 100 replications are considered.

These settings are applied to various combinations of the n_k and c_k , now described in turn.

Setting 1: Equal $c_k \cdot n_k$, Different c_k and n_k .

Consider 150 samples in each split, as follows: $(c_1, n_1) = (3, 50)$, $(c_2, n_2) = (5, 30)$, $(c_3, n_3) = (10, 15)$, and $(c_4, n_4) = (15, 10)$. The results are presented in Table 1. Graphical depictions can be found in Figures 1 and 2. Figure 1 shows that there is a different amount of information in the various sub-samples. This is not a problem, rather a consequence of the way the splits are created and the different amounts of information carried in each. It reminds us that we need to be judicious how the information from the splits will be weighted. It is not a surprise that equal weights are a poor choice. The other methods perform similarly, and all do very well.

To varying degrees, the same will be seen in Settings 2 and 3.

Table 1: First simulation study. Setting 1. Average of split-specific and combined (weighted) parameters and their precision estimates.

	μ	$\text{var}(\mu)$	d	$\text{var}(d)$	σ^2	$\text{var}(\sigma^2)$
split1	-0.00396	0.05779	0.68143	0.41395	1.98676	0.00296
split2	0.05697	0.03071	0.80997	0.19712	1.98578	0.00304
split3	-0.02111	0.01174	0.95161	0.07869	1.97690	0.00319
split4	0.01123	0.00626	0.98870	0.04677	1.98056	0.00347
Equal	0.01078	0.03769	0.85793	0.09988	1.98250	0.01406
Prop	0.00698	0.03230	0.92245	0.08568	1.98081	0.01907
Size prop	0.01078	0.03769	0.85793	0.09988	1.98260	0.01405
Full	0.00780	0.03513	0.98016	0.08614	1.98257	0.01392

Setting 2: Different $c_k \cdot n_k$, Equal c_k , Different n_k

To see the effect of split size, the following choices are made:

$(c_1, n_1) = (4, 25)$, $(c_2, n_2) = (4, 50)$, $(c_3, n_3) = (4, 125)$, and $(c_4, n_4) = (4, 250)$. As a consequence, the size of the splits will be 100, 200, 500, and 1000, respectively. Table 2 summarized the results, with graphical displays presented in Figures 3 and 4.

Setting 3: Different $c_k \cdot n_k$, Different c_k , Equal n_k

We now choose: $(c_1, n_1) = (10, 20)$, $(c_2, n_2) = (20, 20)$, $(c_3, n_3) = (50, 20)$, and $(c_4, n_4) = (100, 20)$. Table 3 summarizes the results.

Graphs can be found in Figures 5 and 6.

Optimal, Approximate Optimal, and Iterated Optimal Weights

Optimal weights were discussed in Section 6.1.1. When we plug the MLE's into the optimal weights, the result of using these weights is the MLE's itself. Of course, this is a circular reasoning, which

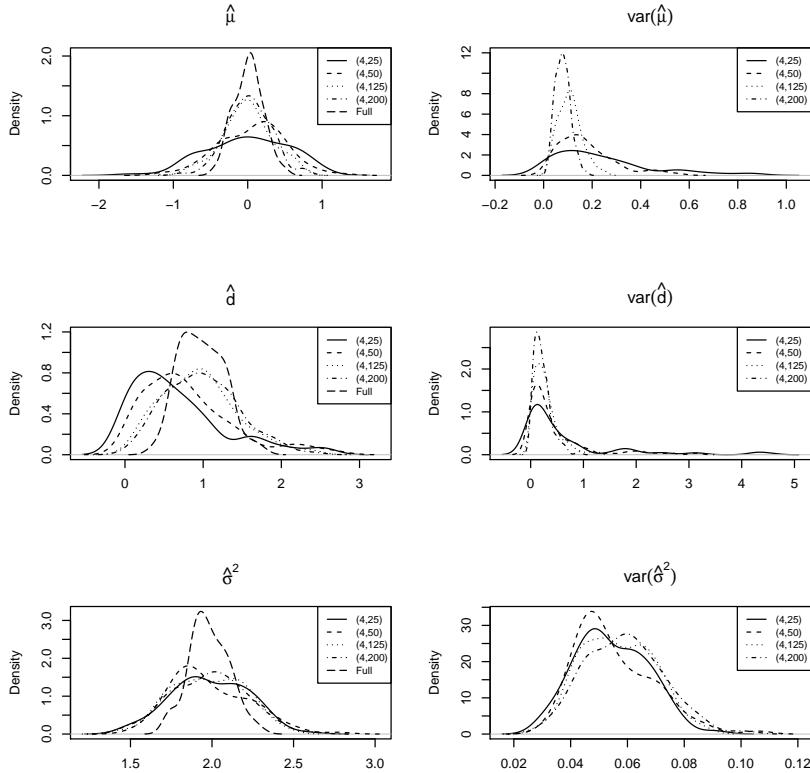


Figure 1: First simulation study. Setting 1. Split-specific results.

is why one needs to resort to, for example, the approximate or iterated optimal weights derived in Section 6.1.2. For both of these, using Settings 1–3, we conducted simulations. They are reported in Figures 7–9.

It is noteworthy that the behavior of the iterated optimal weights depends on c_k and n_k . First, they often but not always converge in a single iteration; the maximum number of iterations observed

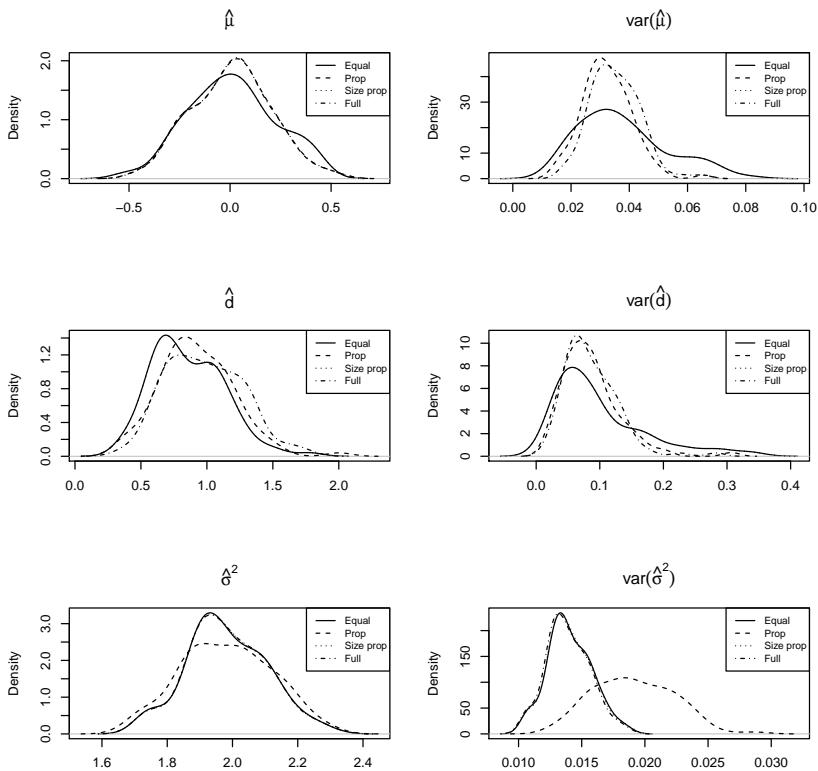


Figure 2: First simulation study. Setting 1. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.

in our simulations being 6. Second, the iterated optimal weights converge to size optimal weights for σ^2 and to proportional weights for d .

Taken together, it follows that both approximately optimal and iterated optimal weights provide excellent results. The specific attraction of the approximate optimal weights is that they obviate

Table 2: First simulation study. Setting 2. Average of split-specific and combined (weighted) parameters and their precision estimates.

	μ	$\text{var}(\mu)$	d	$\text{var}(d)$	σ^2	$\text{var}(\sigma^2)$
split1	-0.02515	0.05227	0.83440	0.52423	2.00347	0.00730
split2	0.01287	0.05157	0.86891	0.57904	1.97285	0.00160
split3	0.06812	0.03586	0.74147	0.23681	2.00165	0.00026
split4	-0.03676	0.02979	0.68241	0.14117	1.99216	0.00006
Equal	0.00477	0.05111	0.78180	0.14770	1.99253	0.00935
Prop	0.00477	0.05111	0.78180	0.14770	1.99253	0.00935
Size prop	-0.00147	0.07139	0.72798	0.16585	1.99328	0.00447
Full	0.00530	0.06339	0.89599	0.14604	1.99333	0.00446

Table 3: First simulation study. Setting 3. Average of split-specific and combined (weighted) parameters and their precision estimates.

	μ	$\text{var}(\mu)$	d	$\text{var}(d)$	σ^2	$\text{var}(\sigma^2)$
split1	0.00343	0.00900	0.84739	0.05169	2.02445	0.00190
split2	0.02553	0.00304	1.00224	0.01754	2.01962	0.00047
split3	-0.00010	0.00045	0.95794	0.00212	1.99765	0.00007
split4	0.01151	0.00012	1.01226	0.00064	1.98944	0.00002
Equal	0.01009	0.01139	0.95496	0.02694	2.00779	0.00486
Prop	0.00939	0.00604	0.98690	0.01369	1.99702	0.00234
Size prop	0.00939	0.00604	0.98690	0.01369	1.99702	0.00234
Full	0.00939	0.00614	1.00487	0.01372	1.99702	0.00233

the need for iteration, which is a factor of stability and speed.

Details About the Second Simulation Study

The aim of this study is to compare the proposed method to two alternatives:

1. full maximum likelihood;
2. the proposed sample-splitting method, allowing for closed forms;

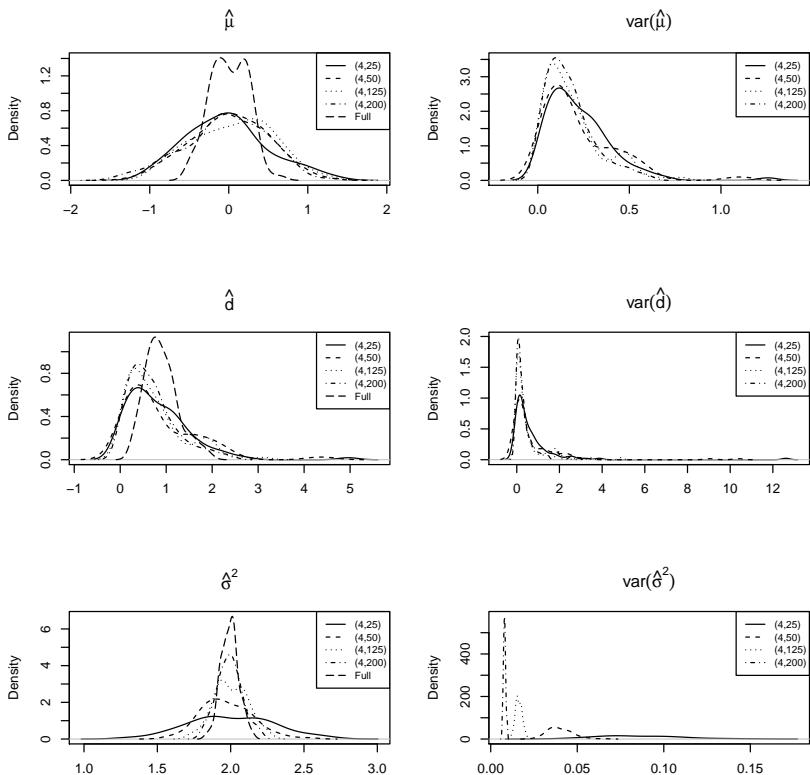


Figure 3: First simulation study. Setting 2. Split-specific results.

3. using multiple imputation (MI) first, to render the clusters of equal sizes, and then apply closed-form solutions to the augmented balanced data, together with the combination rules.

Simulation Plan

In order to study the effect of cluster sizes (n_k) and number of clusters of each size (c_k), 5 different configurations are considered:
Page 257

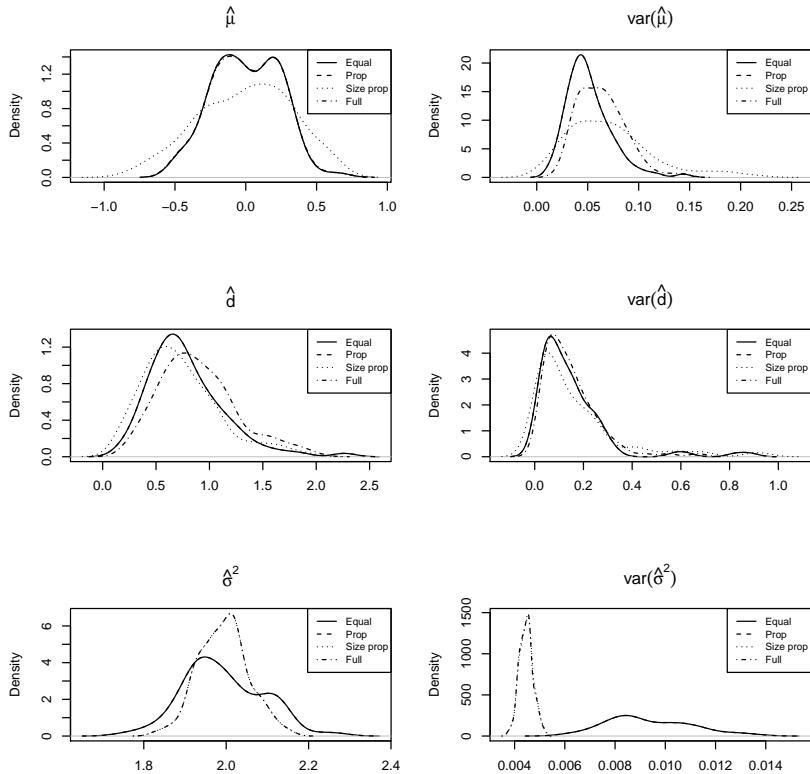


Figure 4: First simulation study. Setting 2. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.

Config. 1. $c_k = (15, 25, 30, 20, 10)$, $n_k = (8, 5, 3, 9, 15)$;

Config. 2. $c_k = (150, 250, 300, 200, 100)$, $n_k = (8, 5, 3, 9, 15)$;

Config. 3. $c_k = (1500, 2500, 3000, 2000, 1000)$, $n_k = (8, 5, 3, 9, 15)$;

Config. 4. $c_k = (15, 25, 30, 20, 10)$, $n_k = (80, 50, 30, 90, 150)$;

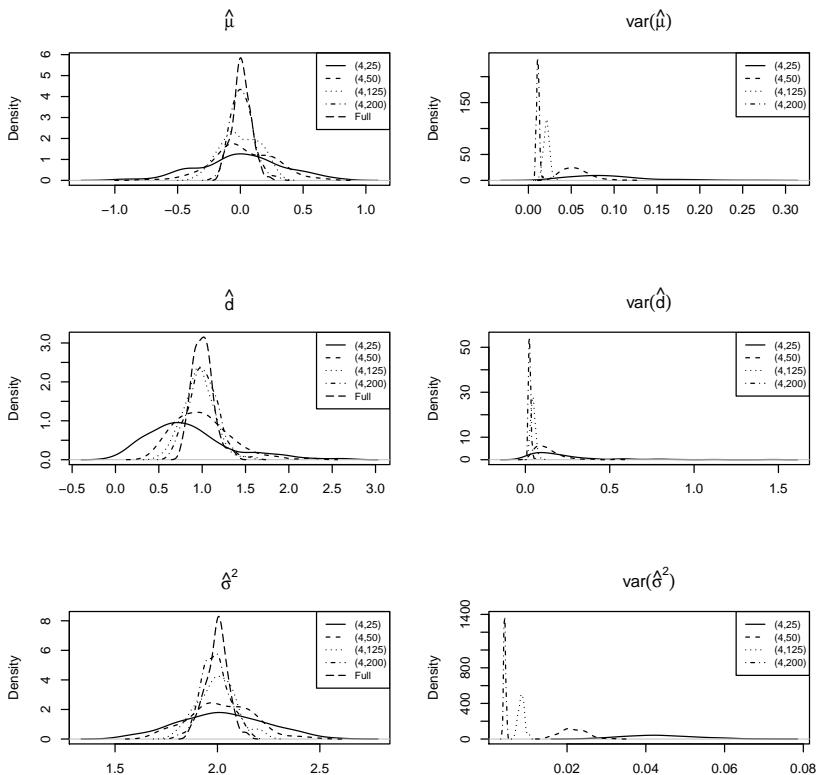


Figure 5: First simulation study. Setting 3. Split-specific results.

Config. 5. $c_k = (15, 25, 30, 20, 10)$, $n_k = (800, 500, 300, 900, 1500)$.

Each configuration is repeated 100 times.

Each cluster is generated from a CS model with $\mu_0 = 0$, $d_0 = 1$, and $\sigma_0^2 = 4$.

For estimating the parameters using the full unbalanced data, PROC MIXED in SAS (Version 9.4) is used with the covariance structure Page 259

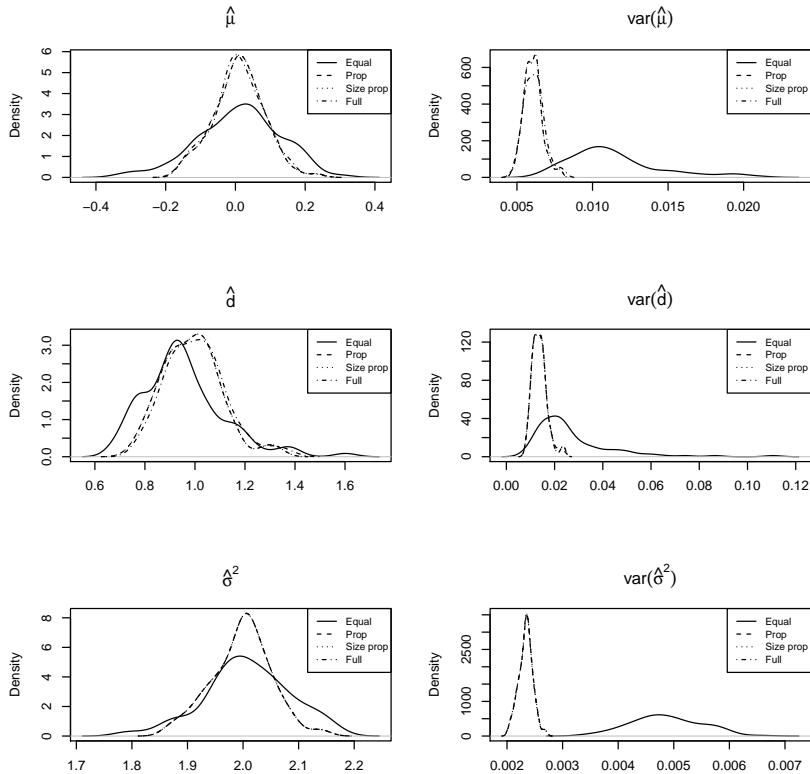


Figure 6: First simulation study. Setting 3. Combining the results from the four splits, using equal, proportional, and size proportional weights. This is compared with full maximum likelihood.

in the REPEATED statement set to `type=cs`.

The closed form solutions and their variances are implemented in R in three different ways. First, the formulas are implemented directly using ‘for’ loops. Following the ideas in ?, it might be faster to replace ‘for’ loops with vectorized computation. For $\hat{\mu}$ it is straightforward, since one just needs to compute an arith-

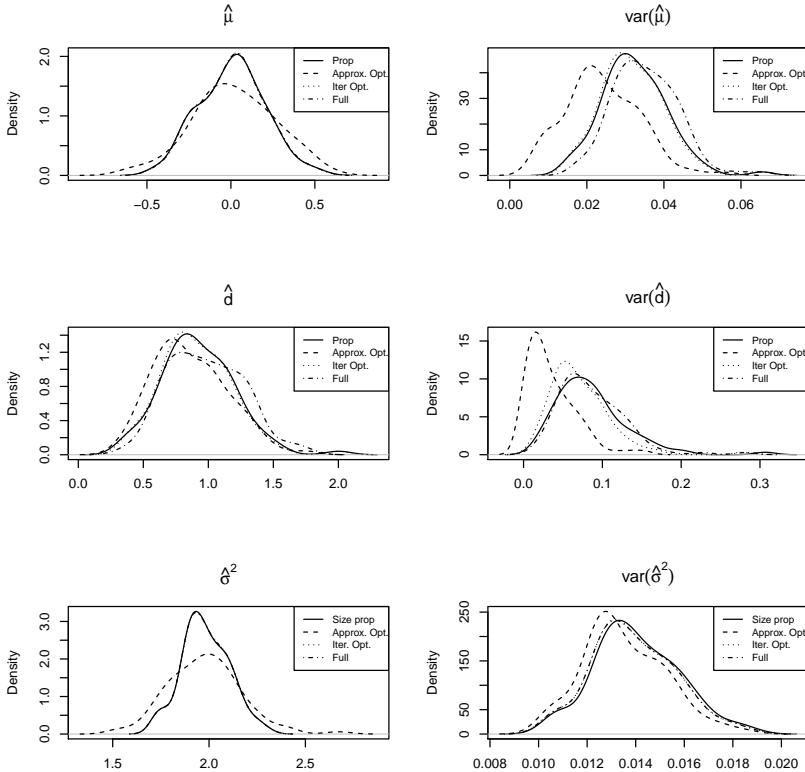


Figure 7: First simulation study. Setting 1. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.

metic average. If Z is a n_k times c_k matrix with its i th column defined as $Z_i^{(k)} = (Y_i^{(k)} - \mu_k \mathbf{1}_{n_k})$, then computing $\sum_{i=1}^{c_k} Z_i^{(k)'} Z_i^{(k)}$ is equivalent to replacing each element in matrix Z by its square, and then sum over the sum of its columns. Furthermore, $J_{n_k} Z_i^{(k)}$ would simply compute the sum of columns in matrix Z . Therefore, $\sum_{i=1}^{c_k} Z_i^{(k)'} J_{n_k} Z_I^{(k)}$ is equivalent to post-multiplying Z by the sum

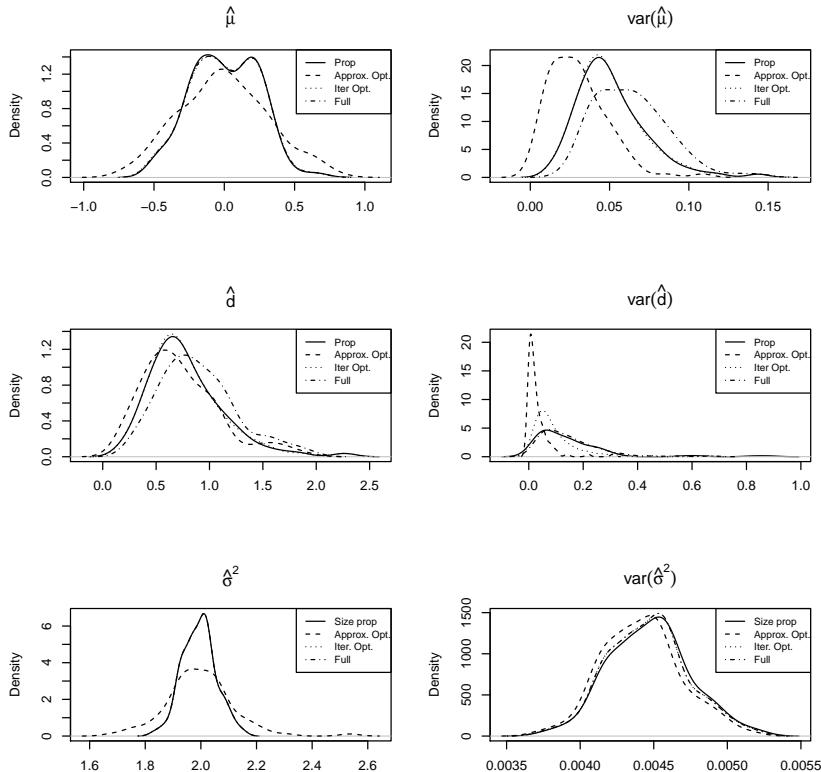


Figure 8: First simulation study. Setting 2. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.

of its columns and then sum over this vector. In this way, within each split, the parameters can be estimated avoiding ‘for’ loops.

Second, it is also possible to find the estimates for all of the splits at once instead of computing them separately.

A third way consists of calculating all the estimates together and

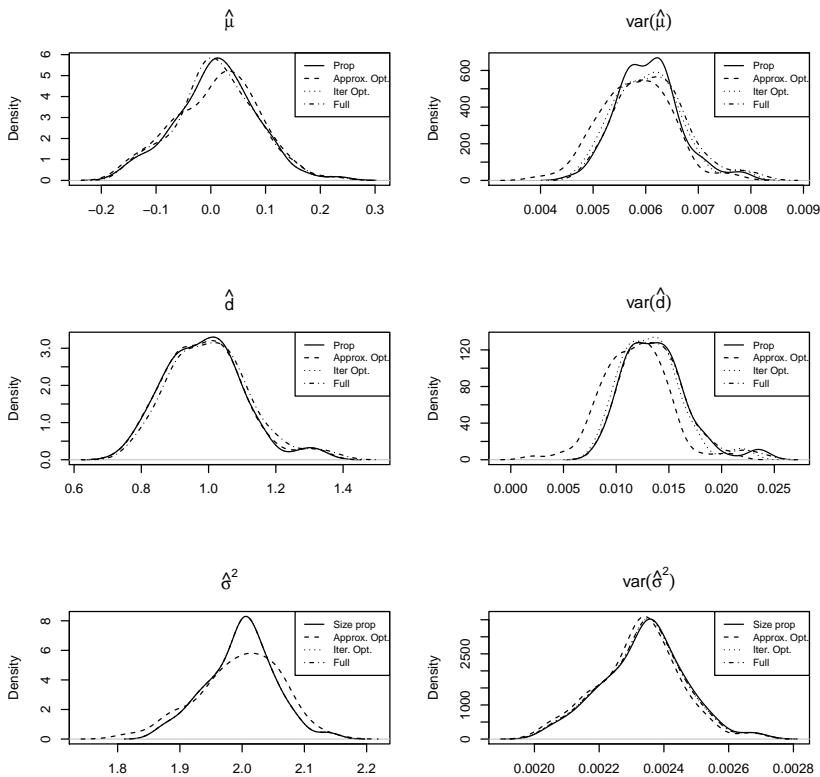


Figure 9: First simulation study. Setting 3. (Size) proportional, approximate, and iterated optimal weights, as well as full maximum likelihood.

not split by split in a ‘for’ loop. This approach is possible via imposing balance through adding missing values in the matrix but, when multiplying and summing, ignoring the missing values. This is very easy in R.

We will compare computation time between these three approaches, for the five configurations.

Additionally, to combine the results from sample splitting, the same weights as used in the case study are considered here as well: equal, proportional, approximate scalar, scalar, and approximate optimal. In the case of the approximate optimal weights both simple and proper variances are calculated.

For the multiple imputation based approach, $M = 20$ imputations are considered and the conventional combination rules applied.

Note that the MI approach cannot be used with configurations 1, 4, and 5, because the number of available subjects in the observed dataset is less than the number of repeated measurements, leading to a singular covariance matrix. From the remaining configurations 2 and 3, we have chosen #2, which implies smaller numbers and hence is more challenging.

For each configuration we report three results: the estimated parameters, their standard errors, and the mean square error (MSE). Furthermore, we report computation time.

Simulation results

Based on the simulation results, it appears that using equal weights is not recommended, while using proportional weights produces results comparable with ML. Of course, in case of σ^2 the approximate scalar weights work better comparing with ML. An interesting outcome of the simulation is that by keeping the number of clusters of different sizes constant, but allowing the cluster sizes to increase, improves estimation of σ^2 , while increasing the number of clusters

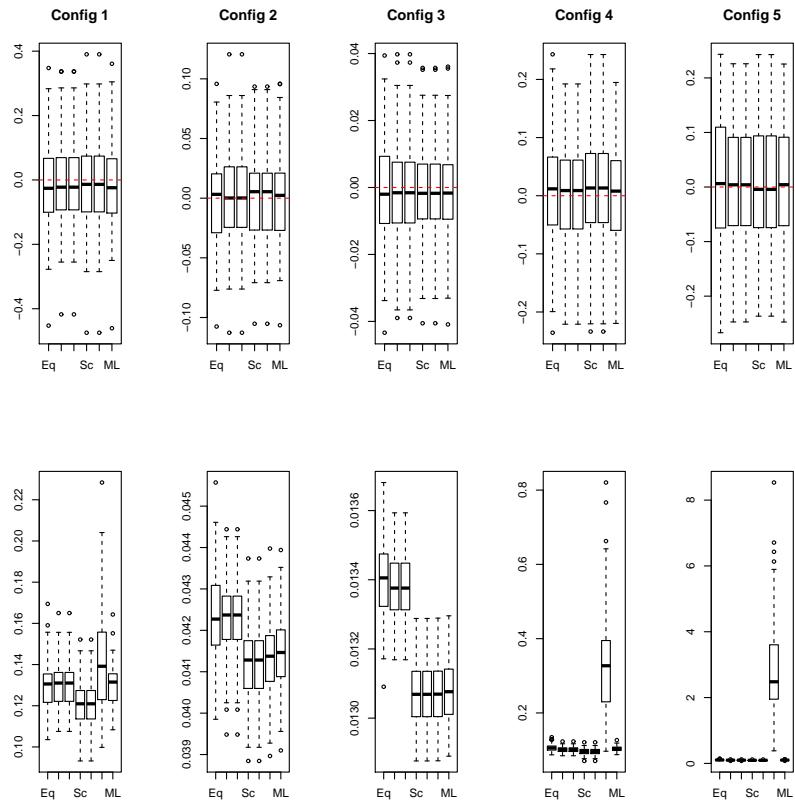


Figure 10: Second simulation study. Estimates for μ (first row) and its standard error (second row).

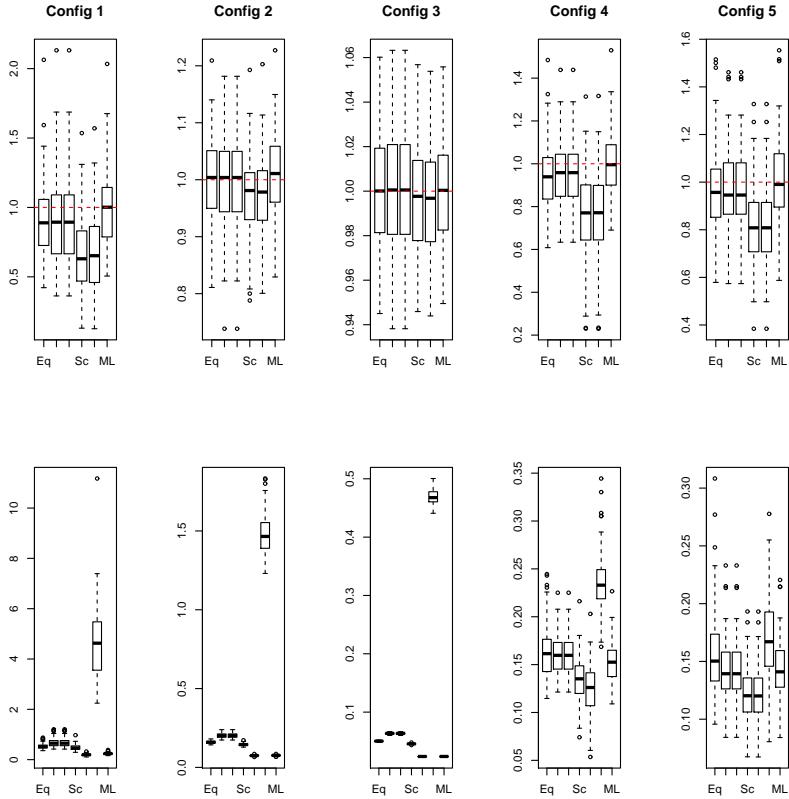


Figure 11: Second simulation study. Estimates for d (first row) and standard errors (second row).

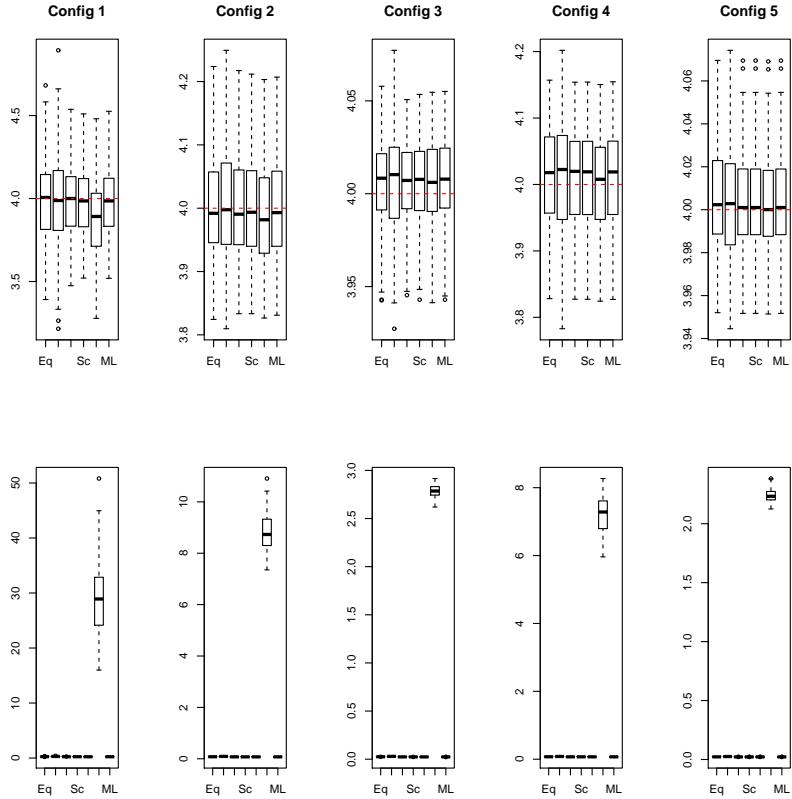


Figure 12: Second simulation study. Estimates for σ^2 (first row) and standard errors (second row).

Table 4: Second simulation study. Mean, standard deviation (S.D.) and MSE for μ among 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop	Approx. sc.	Scalar	Approx. opt.	ML
1	Mean	-1.72277E-02	1.51605E-02	-1.51605E-02	1.21296E-02	-1.21296E-02	1.56028E-02
	S.D.	(1.32751E-01)	(1.28989E-01)	(1.28989E-01)	(1.32435E-01)	(1.32435E-01)	(1.29150E-01)
	MSE	1.77434E-02	1.67015E-02	1.67015E-02	1.75108E-02	1.75108E-02	1.67564E-02
2	Mean	-9.39926E-04	6.93419E-04	6.93419E-04	1.27613E-03	1.27613E-03	7.59533E-04
	S.D.	(3.93526E-02)	(3.95534E-02)	(3.95534E-02)	(3.84321E-02)	(3.84321E-02)	(3.83996E-02)
	MSE	1.53403E-03	1.54930E-03	1.54930E-03	1.46389E-03	1.46389E-03	1.46036E-03
3	Mean	-8.30934E-04	-1.25609E-03	-1.25609E-03	-1.26810E-03	-1.26810E-03	-1.31356E-03
	S.D.	(1.44839E-02)	(1.47545E-02)	(1.47545E-02)	(1.41310E-02)	(1.41310E-02)	(1.41704E-02)
	MSE	2.08376E-04	2.17095E-04	2.17095E-04	1.99298E-04	1.99298E-04	2.00517E-04
4	Mean	9.30928E-03	2.53713E-03	2.53713E-03	8.29367E-03	8.29367E-03	2.82086E-03
	S.D.	(9.26881E-02)	(8.32009E-02)	(8.32009E-02)	(9.76672E-02)	(9.76672E-02)	(8.28999E-02)
	MSE	8.59183E-03	6.85960E-03	6.85960E-03	9.51227E-03	9.51227E-03	6.81163E-03
5	Mean	9.77532E-03	1.02173E-02	1.02173E-02	8.72381E-03	8.72381E-03	1.02422E-02
	S.D.	(1.09769E-01)	(1.04876E-01)	(1.04876E-01)	(1.04982E-01)	(1.04982E-01)	(1.04847E-01)
	MSE	1.20243E-02	1.09934E-02	1.09934E-02	1.09872E-02	1.09872E-02	1.09880E-02

Table 5: Second simulation study. Mean and standard deviation (S.D.) for standard errors of μ estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop	Approx. sc.	Scalar	Simple opt.	Proper opt.	ML
1	Mean	1.29844E-01	1.29733E-01	1.29733E-01	1.20593E-01	1.20593E-01	1.41085E-01	1.29648E-01
	S.D.	(1.18175E-02)	(1.07496E-02)	(1.07496E-02)	(1.09122E-02)	(1.09122E-02)	(2.25155E-02)	(9.95614E-03)
2	Mean	4.23655E-02	4.23056E-02	4.23056E-02	4.11657E-02	4.11657E-02	4.12635E-02	4.14298E-02
	S.D.	(1.08386E-03)	(8.88747E-04)	(8.88747E-04)	(8.71465E-04)	(8.71465E-04)	(8.90259E-04)	(8.66432E-04)
3	Mean	1.34008E-02	1.33822E-02	1.33822E-02	1.30725E-02	1.30725E-02	1.30728E-02	1.30799E-02
	S.D.	(1.21188E-04)	(9.92324E-05)	(9.92324E-05)	(9.68871E-05)	(9.68871E-05)	(9.68974E-05)	(9.62652E-05)
4	Mean	1.06373E-01	1.01232E-01	1.01232E-01	9.60807E-02	9.60807E-02	3.30358E-01	1.03382E-01
	S.D.	(9.80278E-03)	(7.26031E-03)	(7.26031E-03)	(8.36242E-03)	(8.36242E-03)	(1.39042E-01)	(7.17708E-03)
4	Mean	1.05176E-01	9.84615E-02	9.84615E-02	9.45066E-02	9.45066E-02	2.81427E-00	1.00553E-01
	S.D.	(1.10711E-02)	(8.18259E-03)	(8.18259E-03)	(8.14229E-03)	(8.14229E-03)	(1.41211E-00)	(8.28537E-03)

and keeping their sizes constant improves the estimation of d . This is not surprising, because d is the between-cluster variability, which is easier to estimate from a larger number of clusters. This should be seen against the background of relatively small differences anyway.

The results based on MI are not comparable with sample-splitting results. In particular, the variance component d is underestimated using MI, while σ^2 is overestimated. The larger standard errors in this case suggest that the sample-splitting methods use information more efficiently.

Table 6: Second simulation study. Mean, standard deviation (S.D.) and MSE for d estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop.	Approx. sc.	Scalar	Approx. opt.	ML
1	Mean	9.09580E-01	9.00788E-01	9.00788E-01	6.63293E-01	6.71111E-01	9.86549E-01
	S.D.	(2.66885E-01)	(2.99736E-01)	(2.99736E-01)	(2.70556E-01)	(2.79060E-01)	(2.55874E-01)
	MSE	7.86910E-02	9.87862E-02	9.87862E-02	1.85840E-01	1.85264E-01	6.49975E-02
2	Mean	9.98984E-01	9.97534E-01	9.97534E-01	9.72269E-01	9.73771E-01	1.00712E+00
	S.D.	(7.44958E-02)	(8.28143E-02)	(8.28143E-02)	(7.28186E-02)	(7.22762E-02)	(7.08463E-02)
	MSE	5.49515E-03	6.79570E-03	6.79570E-03	6.01854E-03	5.85958E-03	5.01696E-03
3	Mean	1.00004E+00	9.99697E-01	9.99697E-01	9.97218E-01	9.97153E-01	1.00019E+00
	S.D.	(2.61046E-02)	(2.82480E-02)	(2.82480E-02)	(2.48898E-02)	(2.48068E-02)	(2.44715E-02)
	MSE	6.74637E-04	7.90063E-04	7.90063E-04	6.21496E-04	6.17332E-04	5.92900E-04
4	Mean	9.40257E-01	9.50756E-01	9.50756E-01	7.65219E-01	7.65362E-01	9.96005E-01
	S.D.	(1.53414E-01)	(1.48216E-01)	(1.48216E-01)	(1.91865E-01)	(1.91614E-01)	(1.49451E-01)
	MSE	2.68697E-02	2.41734E-02	2.41734E-02	9.15661E-02	9.14038E-02	2.21282E-02
5	Mean	9.63459E-01	9.68182E-01	9.68182E-01	8.21266E-01	8.21270E-01	1.00958E+00
	S.D.	(1.73767E-01)	(1.63471E-01)	(1.63471E-01)	(1.72162E-01)	(1.72163E-01)	(1.69235E-01)
	MSE	3.12283E-02	2.74677E-02	2.74677E-02	6.12894E-02	6.12881E-02	2.84459E-02

Table 7: Second simulation study. Mean and standard deviation (S.D.) for standard errors of d estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop.	Approx. sc.	Scalar	Simple opt.	Proper opt.	ML
1	Mean	5.30888E-01	6.76159E-01	6.76159E-01	4.83519E-01	1.95328E-01	4.66561E+00	2.36408E-01
	S.D.	(1.00495E-01)	(1.66365E-01)	(1.66365E-01)	(1.10589E-01)	(4.01066E-02)	(1.40173E+00)	(3.91752E-02)
2	Mean	1.60365E-01	2.03126E-01	2.03126E-01	1.45284E-01	7.48056E-02	1.47955E+00	7.60798E-02
	S.D.	(8.29444E-03)	(1.42616E-02)	(1.42616E-02)	(8.70034E-03)	(3.35411E-03)	(1.31587E-01)	(3.26483E-03)
3	Mean	5.02596E-02	6.34861E-02	6.34861E-02	4.56061E-02	2.39463E-02	4.67989E-01	2.39748E-02
	S.D.	(8.43976E-04)	(1.44201E-03)	(1.44201E-03)	(8.34510E-04)	(3.68251E-04)	(1.24414E-02)	(3.66865E-04)
4	Mean	1.62410E-01	1.59656E-01	1.59656E-01	1.34256E-01	1.24552E-01	2.35351E-01	1.51740E-01
	S.D.	(2.83009E-02)	(2.05482E-02)	(2.05482E-02)	(2.31559E-02)	(2.51091E-02)	(3.00109E-02)	(2.11541E-02)
5	Mean	1.54122E-01	1.43313E-01	1.43313E-01	1.22117E-01	1.22024E-01	1.70868E-01	1.43764E-01
	S.D.	(3.48725E-02)	(2.50371E-02)	(2.50371E-02)	(2.30972E-02)	(2.31064E-02)	(3.70951E-02)	(2.38992E-02)

the clear winners, further enhanced by smaller standard errors. Furthermore, it follows that computing the estimates in a semi-parallel fashion, thus avoiding ‘for’ loops within the splits but using then between splits, is most efficient. Unless the n_k ’s are very large, computing all estimates at once is the most efficient. Of course, if the estimates for different splits can be done in parallel (without ‘for’ loops), this is more efficient than estimating them all at once.

The Balanced Conditionally Independent Model

In this case, one imposes the following structure on (4.37):
Page 269

Table 8: Second simulation study. Mean, standard deviation (S.D.) and MSE for σ^2 estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop.	Approx. sc.	Scalar	Approx. opt.	ML
1	Mean	3.98608E+00	3.99739E+00	3.98571E+00	3.98364E+00	3.87487E+00	3.98075E+00
	S.D.	(2.50882E-01)	(2.95821E-01)	(2.35138E-01)	(2.33650E-01)	(2.38167E-01)	(2.30477E-01)
	MSE	6.25088E-02	8.66420E-02	5.49414E-02	5.43140E-02	7.18141E-02	5.29588E-02
2	Mean	4.00184E+00	4.00681E+00	4.00177E+00	4.00087E+00	3.99027E+00	4.00064E+00
	S.D.	(7.85190E-02)	(9.05849E-02)	(7.57443E-02)	(7.56486E-02)	(7.50477E-02)	(7.51739E-02)
	MSE	6.10698E-03	8.16998E-03	5.68296E-03	5.66624E-03	5.67055E-03	5.59502E-03
3	Mean	4.00509E+00	4.00491E+00	4.00575E+00	4.00590E+00	4.00472E+00	4.00587E+00
	S.D.	(2.45760E-02)	(2.82395E-02)	(2.36027E-02)	(2.36983E-02)	(2.37694E-02)	(2.36697E-02)
	MSE	6.23282E-04	8.13646E-04	5.84605E-04	5.90806E-04	5.81652E-04	5.89081E-04
4	Mean	4.01402E+00	4.01190E+00	4.01302E+00	4.01304E+00	4.00363E+00	4.01306E+00
	S.D.	(7.69655E-02)	(8.78962E-02)	(7.32088E-02)	(7.31202E-02)	(7.20589E-02)	(7.31207E-02)
	MSE	6.06101E-03	7.79021E-03	5.47558E-03	5.46319E-03	5.15372E-03	5.46370E-03
5	Mean	4.00346E+00	4.00338E+00	4.00292E+00	4.00292E+00	4.00192E+00	4.00292E+00
	S.D.	(2.56561E-02)	(2.79741E-02)	(2.46670E-02)	(2.46664E-02)	(2.46901E-02)	(2.46669E-02)
	MSE	6.63599E-04	7.86128E-04	6.10884E-04	6.10854E-04	6.07187E-04	6.10909E-04

Table 9: Second simulation study. Mean and standard deviation (S.D.) for standard errors of σ^2 estimates in 100 replications for each configuration using different combination weights comparing with full sample MLE.

Config.		Equal	Prop.	Approx. sc.	Scalar	Simple opt.	Proper opt.	ML
1	Mean	2.53758E-01	2.95524E-01	2.40482E-01	2.38699E-01	2.30855E-01	2.88730E-01	2.35804E-01
	S.D.	(1.98892E-02)	(3.33124E-02)	(1.45355E-02)	(1.39849E-02)	(1.38813E-02)	(6.93951E-03)	(1.34905E-02)
2	Mean	7.98210E-02	9.26570E-02	7.57927E-02	7.53242E-02	7.48395E-02	8.82494E-02	7.49872E-02
	S.D.	(1.84469E-03)	(3.08841E-03)	(1.45443E-03)	(1.42354E-03)	(1.40575E-03)	(6.92140E-03)	(1.39989E-03)
3	Mean	2.52202E-02	2.92034E-02	2.39687E-02	2.38353E-02	2.37400E-02	2.75559E-02	2.37444E-02
	S.D.	(1.85701E-04)	(3.12856E-04)	(1.42587E-04)	(1.40888E-04)	(1.40071E-04)	(6.70930E-02)	(1.39784E-04)
4	Mean	7.22370E-02	8.07793E-02	7.01684E-02	7.01663E-02	6.99970E-02	7.23107E-02	7.01238E-02
	S.D.	(1.54384E-03)	(2.35043E-03)	(1.28712E-03)	(1.28385E-03)	(1.26263E-03)	(5.28710E-01)	(1.27765E-03)
5	Mean	2.25782E-02	2.52054E-02	2.19702E-02	2.19702E-02	2.19647E-02	2.23651E-02	2.19689E-02
	S.D.	(1.55999E-04)	(2.29649E-04)	(1.35364E-04)	(1.35358E-04)	(1.35462E-04)	(5.50362E-02)	(1.35379E-04)

- $X_i^{(k)}$ can be rewritten in terms of a first matrix that imposes structure between clusters (e.g., treatment effect), termed $A_i^{(k)}$, and a second one that imposes structure within clusters (e.g., time evolution), $T_i^{(k)'} = (Z_i^{(k)'}, Q_i^{(k)'})'$.
- The matrices $A_i^{(k)}$, $Z_i^{(k)}$, and $Q_i^{(k)}$ are constant among all clusters of size n_k .
- The matrix $\Sigma_i^{(k)} = \sigma^2 I_{n_k}$.

This is the general, balanced growth-curve model as studied by ? and ?. Building on their development, we will now derive sufficient statistics and associated maximum likelihood estimators for the Page 270

Table 10: Second simulation study. Computation time (in seconds) using closed-form solutions with different implementation forms, compared to PROC MIXED.

Config.		Split by split		Together	PROC MIXED
		without loops	using loops		
1	Mean	0.00520	0.00980	0.00640	0.34155
	S.D.	(0.00882)	(0.00943)	(0.00823)	(0.17711)
2	Mean	0.02150	0.07340	0.05660	0.35575
	S.D.	(0.01617)	(0.02006)	(0.09662)	(0.03073)
3	Mean	0.17980	0.73480	0.43400	0.81783
	S.D.	(0.02292)	(0.05835)	(0.11543)	(0.02582)
4	Mean	0.00220	0.00610	0.00360	2.58808
	S.D.	(0.00579)	(0.01497)	(0.00689)	(0.40720)
5	Mean	0.04030	0.27490	0.02130	629.58333
	S.D.	(0.01521)	(0.01941)	(0.00872)	(116.09435)

Table 11: Second simulation study. Mean, standard deviation (S.D.) and MSE for CS parameter estimates in 100 replications for configuration 2 using different combination weights comparing with full sample MLE and MI-MLE.

	Equal	Prop.	Approx. sc.	Scalar	Approx. opt.	MI	ML
μ	Mean	-5.09643E-03	-2.62061E-03	-2.62061E-03	-2.88933E-03	-8.04195E-03	-3.28224E-03
	S.D.	(4.85424E-02)	(4.91772E-02)	(4.91772E-02)	(4.68032E-02)	(4.68032E-02)	(6.00662E-02)
	MSE	2.35877E-02	2.40108E-03	2.40108E-03	2.17698E-03	2.17698E-03	3.63654E-03
d	Mean	9.98392E-03	9.95216E-01	9.95216E-01	9.70589E-01	9.71627E-01	9.92960E-01
	S.D.	(7.33193E-02)	(7.46946E-02)	(7.46946E-02)	(7.59516E-02)	(7.53362E-02)	(1.83983E-02)
	MSE	5.32455E-03	5.54638E-03	5.54638E-03	6.42380E-03	9.31343E-01	5.62737E-03
σ^2	Mean	4.00782E+00	4.00791E+00	4.00581E+00	4.00544E+00	3.99400E+00	5.32627E+00
	S.D.	(7.66500E-02)	(8.98881E-02)	(7.19708E-02)	(7.13441E-02)	(7.19080E-02)	(1.55770E-01)
	MSE	5.87763E-03	8.06169E-03	5.16181E-03	5.06871E-03	5.15505E-03	1.78301E+00

parameters in this model. This can be expressed

$$\mathbf{Y} = A(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) \begin{pmatrix} Z \\ Q \end{pmatrix} + BZ + \boldsymbol{\epsilon}.$$

Here, \mathbf{Y} is an $N \times n$ matrix stacking the outcomes of all clusters of size c , A , Z , and Q group the designs mentioned in Section 4, the vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ contain the fixed effects, B contains N rows of length q , representing the q -dimensional random-effects vector, and $\boldsymbol{\epsilon}$ shares its dimensions with \mathbf{Y} .

Table 12: Second simulation study. Mean and standard deviation (S.D.) for the standard error of CS parameter estimates in 100 replications for configuration 2 using different combination weights comparing with full sample MLE and MI-MLE.

	Equal	Prop.	Approx. sc.	Scalar	Simple opt.	Proper opt.	MI	ML
μ	4.24162E-02	4.22766E-02	4.22766E-02	4.11356E-02	4.11356E-02	4.12439E-02	4.95431E-02	4.14004E-02
	S.D. (1.21548E-03)	(8.59430E-04)	(8.59430E-04)	(9.31222E-04)	(9.31222E-04)	(9.42598E-04)	(7.65032E-03)	(9.11785E-04)
d	1.60545E-01	2.02922E-01	2.02922E-01	1.45623E-01	7.47531E-02	1.48865E+00	9.10211E-02	7.54391E-02
	S.D. (8.88524E-03)	(1.47416E-02)	(1.47416E-02)	(9.84170E-03)	(3.76197E-03)	(1.29914E-01)	(5.18621E-03)	(3.38044E-03)
σ^2	Mean 7.99442E-02	9.26315E-02	7.58626E-02	7.54139E-02	7.49171E-02	8.86450E+00	1.64310E-01	7.50377E-02
	S.D. (1.87358E-03)	(3.14842E-03)	(3.19531E-03)	(1.34244E-03)	(1.33807E-03)	(6.56714E-01)	(2.27104E-02)	(1.42688E-03)

Now, define K the projection matrix such that $K'K = I_{r-q}$, for an appropriate dimension r , and $ZK = 0$. Then, set $P = QK$ and consider the projection model:

$$\mathbf{Y}_1 \equiv YK = A\beta_2 P + \epsilon K.$$

The variance of a cluster is $\sigma^2 I_{r-q}$. Next, define H such that $H'H = I_q$ and $QH = 0$. A second projection model emerges:

$$\mathbf{Y}_2 \equiv YH = A\beta_1 + B + \epsilon H.$$

The variance of a cluster is $\sigma^2 I_q + H'DH$, with D the variance-covariance matrix of the vector of random effects. Importantly, projections $\mathbf{Y}_1 \perp \mathbf{Y}_2$.

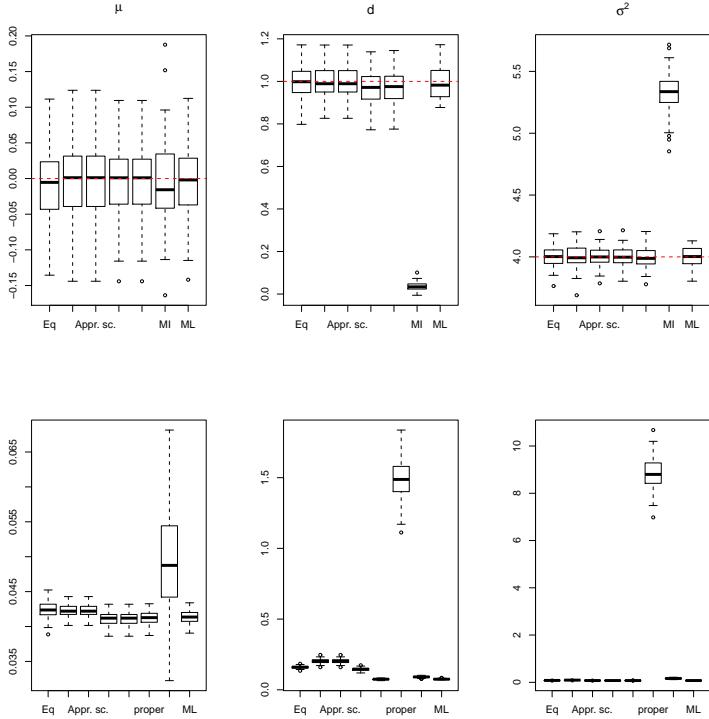


Figure 13: Second simulation study. Estimated CS parameters (first row) and their standard error (second row) using sample splitting, MI-MLE, and MLE.

cient statistics:

$$T_1 = (A'A)^{-1} A' \mathbf{Y}_1 P'(PP')^{-1}, \quad (\text{A.83})$$

$$T_2 = \text{tr} \left\{ \mathbf{Y}'_1 \left[I - A(A'A)^{-1} A' \right] \mathbf{Y}_1 \right\}, \quad (\text{A.84})$$

$$T_3 = (A'A)^{-1} A' \mathbf{Y}_2, \quad (\text{A.85})$$

$$T_4 = \mathbf{Y}'_2 \left[I - A(A'A)^{-1} A' \right] \mathbf{Y}_2. \quad (\text{A.86})$$

Sufficient statistics (A.83)–(A.86) lead to the maximum likelihood
Page 273

estimators:

$$\hat{\beta}_1 = T_1, \quad (\text{A.87})$$

$$\hat{\beta}_2 = T_3, \quad (\text{A.88})$$

$$\hat{\sigma}^2 = \frac{1}{N(n-q)} T_2, \quad (\text{A.89})$$

$$\hat{D} = \frac{1}{N} T_4 - \hat{\sigma}^2 I_q. \quad (\text{A.90})$$

Note that the estimators for the fixed effects do not involve the variance components.

Algebraic Derivations in the AR(1) Case

Here, we present more detailed derivations of the key algebraic expressions presented in Sections 4 and 4.

Some Useful Expressions

Consider,

$$C = \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \dots \\ & 1 & \rho & \rho^2 & \dots \\ & & \ddots & \ddots & \ddots \\ & & & 1 & \rho \\ & & & & 1 \end{pmatrix}, \quad (\text{A.91})$$

then,

$$\Sigma = \sigma^2 C. \quad (\text{A.92})$$

It can be shown that:

$$\det(C) = (1 - \rho^2)^{n-1} \quad (\text{A.93})$$

The inverse of C can be calculated as follows:

$$C^{-1} = \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho & \dots & 0 \\ -\rho & 1+\rho^2 & \ddots & \dots \\ \ddots & \ddots & \ddots & \ddots \\ & & 1+\rho^2 & -\rho \\ & & -\rho & 1 \end{pmatrix}, \quad (\text{A.94})$$

as one may see C^{-1} is a symmetric-tridiagonal matrix with constant diagonal except for the outer entries, and constant first off-diagonal.

Consider:

$$C^{-1} = \frac{1}{1-\rho^2} G. \quad (\text{A.95})$$

Then, by taking the derivative with respect to ρ :

$$\frac{\partial C^{-1}}{\partial \rho} = \frac{2\rho}{(1-\rho^2)^2} G + \frac{1}{1-\rho^2} H, \quad (\text{A.96})$$

where, $H = \frac{\partial G}{\partial \rho}$ and has the form:

$$H = \begin{pmatrix} 0 & -1 & \dots & 0 \\ -1 & 2\rho & \ddots & \dots \\ \ddots & \ddots & \ddots & \ddots \\ & \ddots & 2\rho & -1 \\ 0 & & -1 & 0 \end{pmatrix}. \quad (\text{A.97})$$

Also, considering the fact $CC^{-1} = I$, one can derive:

$$\frac{\partial C^{-1}}{\partial \rho} = -C^{-1} \frac{\partial C}{\partial \rho} C^{-1}. \quad (\text{A.98})$$

The Likelihood Estimators in a Given Cluster

The likelihood function for an n_k -dimensional multivariate normal sample of size c_k has the following form:

$$L = \prod_{i=1}^{C_k} \frac{1}{|\Sigma|^{1/2} (2\pi)^{n_k/2}} \exp \left\{ -\frac{1}{2} (y_i - \mu_i)' \Sigma^{-1} (y_i - \mu_i) \right\}. \quad (\text{A.99})$$

Therefore, the non-constant terms of the log-likelihood are as follows:

$$\ell \propto -\frac{C_k}{2} \ln |\Sigma| + \frac{1}{2} \sum_{i=1}^{C_k} (y_i - \mu_i)' \Sigma^{-1} (y_i - \mu_i), \quad (\text{A.100})$$

which considering (A.93) for AR(1):

$$|\Sigma| = \left(\sigma^2 \right)^{n_k} \left(1 - \rho^2 \right)^{n_k - 1}. \quad (\text{A.101})$$

As a general case, if we consider the mean as linear model with the form $\mu_i = X_i \beta$, one can derive:

$$\frac{\partial \ell}{\partial \mu_i} = \Sigma^{-1} \sum_{i=1}^{C_k} (y_i - \mu_i) = 0 \Rightarrow \hat{\beta} = (X' X)^{-1} X' y. \quad (\text{A.102})$$

Now expanding the log-likelihood for σ^2 and ρ , we have:

$$\ell \propto -\frac{C_k}{2} n_k \ln \sigma^2 - \frac{C_k}{2} (n_k - 1) \ln (1 - \rho^2) - \frac{1}{2} \sum_{i=1}^{C_k} (y_i - \mu_i)' \Sigma^{-1} (y_i - \mu_i). \quad (\text{A.103})$$

Considering $\Sigma = \sigma^2 C$ and (A.94), the derivative with respect to σ^2

is as follows:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{C_k n_k}{2} \frac{1}{\sigma^2} + \frac{1}{2} \frac{1}{(\sigma^2)^2} \sum_{i=1}^{C_k} (y_i - \mu_i)' C^{-1} (y_i - \mu_i). \quad (\text{A.104})$$

Solving $\frac{\partial \ell}{\partial \sigma^2} = 0$ gives:

$$\hat{\sigma}^2 = \frac{1}{C_k n_k} \sum_{i=1}^{C_k} (y_i - \mu_i)' C^{-1} (y_i - \mu_i). \quad (\text{A.105})$$

One may notice that C^{-1} contains the parameter ρ .

Taking the derivative of (A.103) with respect to ρ gives:

$$\frac{\partial \ell}{\partial \rho} = \frac{C_k(n_k - 1)}{2} \frac{2\rho}{1 - \rho^2} - \frac{1}{\sigma^2} \sum_{i=1}^{C_k} (y_i - \mu_i)' \frac{\partial C^{-1}}{\partial \rho} (y_i - \mu_i). \quad (\text{A.106})$$

Setting $\frac{\partial \ell}{\partial \rho} = 0$ gives:

$$\hat{\sigma}^2 \frac{2\hat{\rho}}{1 - \hat{\rho}^2} = \frac{1}{C_k(n_k - 1)} \sum_{i=1}^{C_k} (y_i - \mu_i)' \frac{\partial C^{-1}}{\partial \rho} (y_i - \mu_i). \quad (\text{A.107})$$

Solving (A.105) and (A.107) gives $\hat{\sigma}^2$ and $\hat{\rho}$. For any $(n_k \times n_k)$ matrix Q , $\sum_i (y_i - \mu_i)' Q (y_i - \mu_i)$ equals $\text{tr}\{SQ\}$, where tr denotes the trace of a matrix, and $S = \sum_i (y_i - \mu_i)(y_i - \mu_i)'$. Hence, from (A.105), (A.107), (A.95), (A.96), and (A.98), one can write:

$$\begin{cases} (1 - \hat{\rho}^2)\hat{\sigma}^2 = \frac{1}{C_k n_k} \text{tr}\{SG\}, \\ (1 - \hat{\rho}^2)\hat{\sigma}^2 = \frac{1}{C_k n_k} \text{tr}\{SG\} + \frac{1 - \hat{\rho}^2}{2\hat{\rho}} \frac{1}{C_k(n_k - 1)} \frac{1}{C_k(n_k - 1)} \text{tr}\{SH\}. \end{cases} \quad (\text{A.108})$$

Set $g = \text{tr}\{SG\}$ and $h = \text{tr}\{SH\}$, it follows that

$$\frac{g}{n_k} + \frac{1 - \hat{\rho}^2}{2\hat{\rho}} h = 0. \quad (\text{A.109})$$

Given that both g and h are functions of ρ only, ρ can be estimated using (A.109). Given ρ , one can use one of equations in (A.108) to estimate σ^2 .

Let us consider some special cases. For $n_k = 2$:

$$G = \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Therefore, g and h can be computed as:

$$g = \text{tr} \left[\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \right] = S_{11} - 2\rho S_{12} + S_{22}.$$

$$h = \text{tr} \left[\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \right] = -2S_{12}.$$

Now using (A.109):

$$\hat{\rho}(S_{11} - 2\hat{\rho}S_{12} + S_{22} + (1 - \hat{\rho}^2)(-2S_{12}),$$

which gives:

$$\hat{\rho} = \frac{2S_{12}}{S_{11} + S_{22}}. \quad (\text{A.110})$$

Then, using first equation in (A.108):

$$(1 - \hat{\rho}^2)\hat{\sigma}^2 = \frac{1}{2C_k}(S_{11} - 2\hat{\rho}S_{12} + S_{22}),$$

which gives:

$$\hat{\sigma}^2 = \frac{S_{11} + S_{22}}{2C_k}. \quad (\text{A.111})$$

For $n_k = 3$:

$$g = \text{tr} \left[\begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{pmatrix} \begin{pmatrix} 1 & -\rho & 0 \\ -\rho & 1 + \rho^2 - \rho & \\ 0 & -\rho & 1 \end{pmatrix} \right] = S_{11} + S_{22} + S_{33} - 2\rho R$$

$$h = \text{tr} \left[\begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 2\rho & -1 \\ 0 & -1 & 0 \end{pmatrix} \right] = -2(S_{12} + S_{23}) + 2\rho S_{22}.$$

Let,

$$\begin{cases} S = S_{11} + S_{22} + S_{33} \\ R = S_{12} + S_{23} \end{cases} \Rightarrow \begin{cases} g = S + \rho^2 S_{22} - 2\rho R \\ h = -2R + 2\rho S_{22} \end{cases}$$

Using (A.109):

$$2S_{22}\rho^3 - R\rho^2 - (S + 3S_{22})\rho + 3R = 0. \quad (\text{A.114})$$

Considering the results for $n_k = 2$ and $n_k = 3$, one can calculate
Page 279

(A.109) for the general case $n_k = n$ as follows.

$$(n-1)\tilde{S}\rho^3 - (n-2)R\rho^2 - (n\tilde{S} + S)\rho + nR = 0 \quad (\text{A.115})$$

with:

$$\begin{cases} S = S_{11} + \dots + S_{nn}, \\ \tilde{S} = S_{22} + \dots + S_{n-1,n-1}, \\ R = S_{12} + S_{23} + \dots + S_{n-1,n}. \end{cases}$$

Then using (A.108):

$$\hat{\sigma}^2 = \frac{1}{C_n} \frac{1}{(1 - \hat{\rho}^2)} (S + \hat{\rho}^2 \tilde{S} - 2\hat{\rho}R). \quad (\text{A.116})$$

For $n_k > 2$, (A.115) is a third-degree polynomial. One can show that this equation has only one root in $[-1, 1]$.

Proof. Consider:

$$f(\rho) = (n-1)\tilde{S}\rho^3 - (n-2)R\rho^2 - (n\tilde{S} + S)\rho + nR$$

$$f'(\rho) = 3(n-1)\tilde{S}\rho^2 - 2(n-2)R\rho - (n\tilde{S} + S)$$

$$f''(\rho) = 6(n-1)\tilde{S}\rho - 2(n-2)R$$

The discriminant of $f'(\rho)$ is as follows:

$$\Delta_{f'(\rho)} = (n-2)^2 R^2 + 3(n\tilde{S} + S)(n-1)\tilde{S} \geq 0.$$

Therefore $f'(\rho)$ has no root and hence $f(\rho)$ is monotone. One may see $f'(0) \leq 0$, therefore, $f(\rho)$ is a monotonically decreasing function (I).
Page 280

One can show $f(1) \leq 0$ and $f(-1) \geq 0$ (II). Considering (I) and (II) together, one may conclude $f(\rho)$ must necessarily cross the horizontal line only once between $[-1, 1]$. \square

This shows the unique $\hat{\rho}$ can be easily estimated solving (A.115) using Cardano's formula (?).

Hessians, Covariance Matrices, and Optimal Weights

Given the MLEs for the AR(1) covariance structure, the Hessians and covariance matrices of the MLEs can be derived. Following the general results obtained about optimal weights, they can be used to compute the exact optimal weights in the case of the AR(1) structure. As mean and variance parameters are orthogonal in the normal case, we can consider the second derivative for fixed effects and variance components separately.

Second derivative with respect to fixed effects

As

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{C_k} X_i' \Sigma^{-1} (y_i - \mu_i),$$

we have:

$$\begin{aligned} E \left[\frac{\partial \ell}{\partial \beta} \left(\frac{\partial \ell}{\partial \beta} \right)' \right] &= \sum_{i=1}^{C_k} X_i' \Sigma^{-1} E(y_i - \mu_i)(y_i - \mu_i)' \Sigma^{-1} X_i \\ &= \sum_{i=1}^{C_k} X_i' \Sigma^{-1} X_i. \end{aligned}$$

For the special case of just an intercept $X_i = \mathbf{1}$:

$$\mathbb{E} \left[\frac{\partial \ell}{\partial \beta} \left(\frac{\partial \ell}{\partial \beta} \right)' \right] = \sum_{i=1}^{C_k} \mathbf{1}' \Sigma^{-1} \mathbf{1} = \frac{C_k}{\sigma^2(1-\rho^2)} \left[(n_k - 2)\rho^2 - 2(n_k - 1)\rho + n_k \right] \quad (\text{A.117})$$

Therefore, the variance for $\hat{\mu}$ can be computed as inverse of (A.117).

Second derivative with respect to variance components

To calculate the derivatives with respect to variance components rather than $\frac{\partial C^{-1}}{\partial \rho}$, we need $K = \frac{\partial C^{-1}}{\partial \rho^2}$. Using these derivatives:

$$\begin{cases} \frac{\partial}{\partial \rho} 2 \left(\frac{\rho}{1-\rho^2} \right) = 2 \frac{1+\rho^2}{(1-\rho^2)^2}, \\ \frac{\partial}{\partial \rho} \frac{\rho}{(1-\rho^2)^2} = \frac{1+3\rho^2}{(1-\rho^2)^3}, \\ \frac{\partial}{\partial \rho} \frac{1+\rho^2}{(1-\rho^2)^2} = \frac{2\rho(3+\rho^2)}{(1-\rho^2)^3}. \end{cases} \quad (\text{A.118})$$

it follows that

$$\frac{\partial C^{-1}}{\partial \rho^2} = K = \frac{1}{(1-\rho^2)^3} \begin{pmatrix} 2(1+3\rho^2) & -2\rho(3+\rho^2) & & \\ -2\rho(3+\rho^2) & 4(1+3\rho^2) & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & -2\rho(3+\rho^2) & 2(1+3\rho^2) \end{pmatrix} \quad (\text{A.119})$$

The second-derivatives are:

$$\begin{cases} \frac{\partial^2 \ell}{\partial (\sigma^2)^2} = \frac{C_k n_k}{2} \frac{1}{(\sigma^2)^2} - \frac{1}{(\sigma^2)^3} \sum_{i=1}^{C_k} (y_i - \mu_i)' C^{-1} (y_i - \mu_i), \\ \frac{\partial^2 \ell}{\partial \rho^2} = \frac{C_k (n_k - 1)(1+\rho^2)}{(1-\rho^2)^2} - \frac{1}{2\sigma^2} \sum_{i=1}^{C_k} (y_i - \mu_i)' K (y_i - \mu_i), \\ \frac{\partial^2 \ell}{\partial \rho \partial \sigma^2} = \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{C_k} (y_i - \mu_i)' \frac{\partial C^{-1}}{\partial \rho} (y_i - \mu_i). \end{cases} \quad (\text{A.120})$$

To construct the expected Hessian and covariance matrix, one needs to find the expectations of the expressions in (A.120).

$$E\left(\frac{\partial^2 \ell}{\partial(\sigma^2)^2}\right) = -\frac{C_k n_k}{2} \frac{1}{(\sigma^2)^2}. \quad (\text{A.121})$$

This follows from the fact that:

$$E\left(\sum_{i=1}^{C_k} (y_i - \mu_i)' C^{-1} (y_i - \mu_i)\right) = C_k \text{tr} \left\{ E[(y_i - \mu_i)' (y_i - \mu_i)] C^{-1} \right\},$$

and $E[(y_i - \mu_i)(y_i - \mu_i)'] = \sigma^2 C$.

For the second derivative with respect to ρ :

$$E\left[\frac{\partial^2 \ell}{\partial \rho^2}\right] = \frac{C_k(n_k - 1)(1 + \rho^2)}{(1 - \rho^2)^2} - \frac{C_k}{2} \text{tr}\{KS\}. \quad (\text{A.122})$$

Likewise:

$$E\left[\frac{\partial^2 \ell}{\partial \rho \partial \sigma^2}\right] = \frac{C_k}{2\sigma^2} \text{tr} \left\{ C \frac{\partial C^{-1}}{\partial \rho} \right\}. \quad (\text{A.123})$$

Substituting for $\text{tr}\{KS\}$ and $\text{tr}\{C \frac{\partial C^{-1}}{\partial \rho}\}$ we get:

$$E\left[\frac{\partial^2 \ell}{\partial \rho \partial \sigma^2}\right] = \frac{C_k(n_k - 1)}{\sigma^2} \frac{\rho}{1 - \rho^2}. \quad (\text{A.124})$$

$$E\left[\frac{\partial^2 \ell}{\partial \rho^2}\right] = -C_k(n_k - 1) \frac{1 + \rho^2}{(1 - \rho^2)^2}. \quad (\text{A.125})$$

Using (A.121), (A.124), and (A.125) one obtains the 2×2 Hessian
Page 283

matrix as follows:

$$H = -C_k \begin{pmatrix} \frac{n_k}{2(\sigma^2)^2} & -\frac{n_k-1}{\sigma^2} \frac{\rho}{1-\rho^2} \\ -\frac{n_k-1}{\sigma^2} \frac{\rho}{1-\rho^2} & (n_k-1) \frac{1+\rho^2}{(1-\rho^2)^2} \end{pmatrix}. \quad (\text{A.126})$$

The determinant of the Hessian in (A.126) is as follows:

$$\det(H) = \frac{C_k^2(n_k-1)(n_k-(n_k-2)\rho^2)}{2(\sigma^2)^2(1-\rho^2)^2}. \quad (\text{A.127})$$

So,

$$-H^{-1} = \frac{1}{C_k(n_k-(n_k-2)\rho^2)} \begin{pmatrix} 2(\sigma^2)^2(1+\rho^2) & 2\rho\sigma^2(1-\rho^2) \\ 2\rho\sigma^2(1-\rho^2) & \frac{n_k}{n_k-1}(1-\rho^2)^2 \end{pmatrix}. \quad (\text{A.128})$$

The Hessian for the unbiased estimator differs slightly from its MLE counterpart:

$$\tilde{H} = -C_k \begin{pmatrix} \frac{n_k-1}{2(\sigma^2)^2} & -\frac{n_k-1}{\sigma^2} \frac{\rho}{1-\rho^2} \\ -\frac{n_k-1}{\sigma^2} \frac{\rho}{1-\rho^2} & (n_k-1) \frac{1+\rho^2}{(1-\rho^2)^2} \end{pmatrix}, \quad (\text{A.129})$$

$$\det(\tilde{H}) = \frac{C_k^2(n_k-1)^2}{2(\sigma^2)^2(1-\rho^2)}. \quad (\text{A.130})$$

Therefore,

$$-\tilde{H}^{-1} = \frac{1}{C_k(n_k-(n_k-2)\rho^2)} \begin{pmatrix} 2(\sigma^2)^2(1+\rho^2) & 2\rho\sigma^2(1-\rho^2) \\ 2\rho\sigma^2(1-\rho^2) & (1-\rho^2)^2 \end{pmatrix}. \quad (\text{A.131})$$

Having the covariance matrix, one may easily find the optimal

weights using

$$W_{opt.} = \frac{V_k^{-1}}{\sum_{i=1}^K V_i^{-1}} \quad (\text{A.132})$$

The variance of an estimator obtained using the optimal weights in (A.132) can be calculated as $\left(\sum_{i=1}^K V_i^{-1}\right)^{-1}$.

Proof of Proposition 4.1

Proof. Consider an estimator of the form:

$$\tilde{\mu}_\alpha = \frac{1}{c} \sum_{i=1}^c \sum_{j=1}^n \alpha_j Y_{ij}, \quad (\text{A.133})$$

for a vector of weights $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)'$. Because the clusters are i.i.d. it is evident that the components of $\boldsymbol{\alpha}$ do not depend on the cluster index i . Clearly, the requirement that $E(\tilde{\mu}_\alpha) = \mu$ implies the condition

$$\sum_{j=1}^n \alpha_j = 1. \quad (\text{A.134})$$

An expression of the variance of $\tilde{\mu}_\alpha$ combined with this requirement produces the objective function:

$$Q = \sigma^2 \left(\sum_{j=1}^n \alpha_j^2 + 2 \sum_{j < k} \alpha_j \alpha_k \rho^{|j-k|} \right) - \lambda \left(\sum_{j=1}^n \alpha_j - 1 \right), \quad (\text{A.135})$$

with λ a Lagrange multiplier. Taking the derivative of (A.135) w.r.t. $\boldsymbol{\alpha}$ leads to, after rearrangement:

$$\boldsymbol{\alpha} = \frac{\lambda}{2\sigma^2} C^{-1} \mathbf{1}.$$

Given that we have an explicit form for C^{-1} , it follows that

$$\alpha = \frac{\lambda}{2\sigma^2(1+\rho)} \boldsymbol{\rho}^{(1)}, \quad (\text{A.136})$$

with $\boldsymbol{\rho}^{(1)} = (1, 1-\rho, \dots, 1-\rho, 1)'$. Combining (A.136) with constraint (A.134) leads to $\lambda = 2\sigma^2(1+\rho)/[2 + (n-2)(1-\rho)]$, hence

$$\alpha = \frac{1}{[2 + (n-2)(1-\rho)]} \boldsymbol{\rho}^{(1)},$$

establishing the MLE. This completes the proof.

Optimal weights in case of a general mean structure

$$X_i^{(k)} \boldsymbol{\beta}$$

Cluster size specific expressions are:

$$\widehat{\boldsymbol{\beta}}_k = \left(\sum_{i=1}^{c_k} X_i^{(k)'} \Sigma_k^{-1} X_i^{(k)} \right)^{-1} \left(\sum_{i=1}^{c_k} X_i^{(k)'} \Sigma_k^{-1} Y_i^{(k)} \right) \quad (\text{A.137})$$

and

$$\text{var}(\widehat{\boldsymbol{\beta}}_k) = V_k = \left(\sum_{i=1}^{c_k} X_i^{(k)'} \Sigma_k^{-1} X_i^{(k)} \right)^{-1}. \quad (\text{A.138})$$

The combination rule is

$$\tilde{\boldsymbol{\beta}}_k = \sum_{i=1}^K A_k \widehat{\boldsymbol{\beta}}_k, \quad (\text{A.139})$$

with

$$V_k^{-1} = \frac{1}{\sigma^2} \sum_{i=1}^{c_k} X_i^{(k)'} C_k^{-1} X_i^{(k)} \quad (\text{A.140})$$

and C_k is as described in Supplementary Materials 10.

The first factor in (A.137) can be split into three parts:

$$(1 - \rho^2) X_i^{(k)'} C_k^{-1} X_i^{(k)} = X_i^{(k)'} (1 + \rho^2) I_k X_i^{(k)}$$

$$- \rho^2 X_i^{(k)'} \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ \vdots & & & 0 & 0 \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix} X_i^{(k)}$$

$$- \rho X_i^{(k)'} \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} X_i^{(k)}.$$

(10) simplifies to $(1 + \rho^2) X_i^{(k)'} X_i^{(k)}$, while (10) equals

$$\rho^2 \begin{pmatrix} x_{ki11}^2 + x_{kin_k 1}^2 & x_{ki11} \cdot x_{ki22} + x_{kin_k 1} \cdot x_{kin_k 2} & \dots & x_{ki11} \cdot x_{ki1p} + x_{kin_k 1} \cdot x_{kin_k p} \\ & x_{ki12}^2 + x_{kin_k 2}^2 & & \\ & & \ddots & \\ & & & x_{ki1p}^2 + x_{kin_k p}^2 \end{pmatrix} \quad (\text{A.141})$$

Defining $\mathbf{X}_{i1}^{(k)} = (x_{ki11} \ \dots \ x_{ki1p})^t$ and $\mathbf{X}_{in_k}^{(k)} = (x_{kin_k 1} \ \dots \ x_{kin_k p})^t$,

(10) equals

$$\rho^2 [\mathbf{X}_{i1}^{(k)} \ 0 \ \dots \ 0 \ \mathbf{X}_{in_k}^{(k)}] \mathbf{X}_i^{(k)}. \quad (\text{A.142})$$

For the third term, define $\mathbf{X}_{ij}^{(k)-} = (x_{ki2j} \ \dots \ x_{kin_k j})^t$ and $\mathbf{X}_{ij}^{(k)+} =$

$(x_{ki1j} \dots x_{kin_k-1j})^t$. As a consequence, (10) will equal

$$-\rho[X_i^{(k)-'} \cdot X_i^{(k)+} + X_i^{(k)+'} \cdot X_i^{(k)-}]. \quad (\text{A.143})$$

In summary:

$$\begin{aligned} (1 - \rho^2)X_i^{(k)'}C_k^{-1}X_i^{(k)} &= (1 + \rho^2)X_i^{(k)'}X_i^{(k)} \\ &\quad - \rho^2[\mathbf{X}_{i1}^{(k)} \ 0 \dots 0 \ \mathbf{X}_{in_k}^{(k)}]X_i^{(k)} \\ &\quad - \rho[X_i^{(k)-'} \cdot X_i^{(k)+} + X_i^{(k)+'} \cdot X_i^{(k)-}] \\ &\stackrel{\text{notation}}{=} F_{1k}. \end{aligned} \quad (\text{A.144})$$

The second factor in (A.137), using the same notations for $Y_i^{(k)}$ as described above, can be rewritten as:

$$(1 - \rho^2)X_i^{(k)'}C_k^{-1}Y_i^{(k)} = (1 + \rho^2)X_i^{(k)'}Y_i^{(k)} \quad (\text{A.145})$$

$$-\rho^2 \begin{pmatrix} x_{ki11} \cdot y_{ki1} + x_{kin_k 1} \cdot y_{kin_k} \\ x_{ki12} \cdot y_{ki1} + x_{kin_k 2} \cdot y_{kin_k} \\ \vdots \\ x_{ki1p} \cdot y_{ki1} + x_{kin_k p} \cdot y_{kin_k} \end{pmatrix} \quad (\text{A.146})$$

$$-\rho[X_i^{(k)-'} \cdot Y_i^{(k)+} + X_i^{(k)+'} \cdot Y_i^{(k)-}] \quad (\text{A.147})$$

$$\stackrel{\text{not.}}{=} F_{2k}. \quad (\text{A.148})$$

Combining (A.144) and (A.148) the overall estimate equals:

$$\begin{aligned}\tilde{\beta}_k &= \sum_{i=1}^K A_k \widehat{\beta}_k \\ &= \sum_{i=1}^K \left(\sum_{m=1}^K F_{1m} \right)^{-1} F_{2k}\end{aligned}\tag{A.149}$$

Delta Method for the Mean Estimator

By plugging in ρ_k and defining $a'_k = c_k[n_k - (n_k - 2)\rho_k]$, equation (A.150) simplifies to

$$a_k = \frac{a'_k}{\sum_{m=1}^K a'_m},\tag{A.150}$$

and (4.59) becomes

$$\text{var}(\widehat{\mu}_k) = v_k = \frac{\sigma_k^2(1 + \rho_k)}{a'_k}.\tag{A.151}$$

The first derivatives equal

$$\begin{aligned}\frac{\partial \tilde{\mu}}{\partial \mu_k} &= a_k = \frac{a'_k}{\sum_{m=1}^K a'_m}, \\ \frac{\partial \tilde{\mu}}{\partial \sigma_k^2} &= 0, \\ \frac{\partial \tilde{\mu}}{\partial \rho_k} &= \frac{-c_k(n_k - 2) \sum_{m=1}^K a'_m (\mu_k - \mu_m)}{\left(\sum_{m=1}^K a'_m \right)^2},\end{aligned}\tag{A.152}$$

and these can be combined using the delta method, resulting in (4.65):

$$\begin{aligned}\text{var}(\tilde{\mu}) &= \sum_{i=1}^K \frac{a_k'^2}{\left(\sum_{k=1}^K a_k'\right)^2} \cdot \frac{\sigma_k^2(1 - \rho_k^2)}{a_k'} \\ &\quad + \frac{\sum_{k=1}^K \left[c_k(n_k - 2) \sum_{m=1}^K a_m' (\mu_k - \mu_m)\right]^2}{\left(\sum_{k=1}^K a_k'\right)^4} \cdot \frac{1 - \rho_k^2}{c_k(n_k - 1)}\end{aligned}\tag{A.153}$$

Calculating $\hat{\rho}$ and $\hat{\sigma}^2$ in R

In this section, we consider the implementation of the calculations for the variance components via the R package **fastAR1**. This can be done with a few simple lines of code. For fixed $C_k = C$ and $n_k = n$, and given the data y , the function **est.ar1** estimates the variance components and provides a plot for the third-degree polynomial in (A.115). This visually underscores that there is only one root in $[-1, 1]$. Figure 14 shows (A.115) for 10 simulated data sets; clearly, there is a single root only in $[-1, 1]$. For convenience, the R code is given in Supplementary Materials ???. Other functions to find variances and iterated optimal weights are also available.

Details on Additional Simulations

Simulations with Proportional and Size-proportional Weights

Here we consider $C_1 = 500, C_2 = 250, C_3 = 250, C_4 = 500$, and $n_1 = 5, n_2 = 10, n_3 = 10, n_4 = 5$. Parameters are set to $\mu = 0, \sigma = 2$ and $\rho = (0.1, 0.5, 0.8)$. The data are generated 100 times and

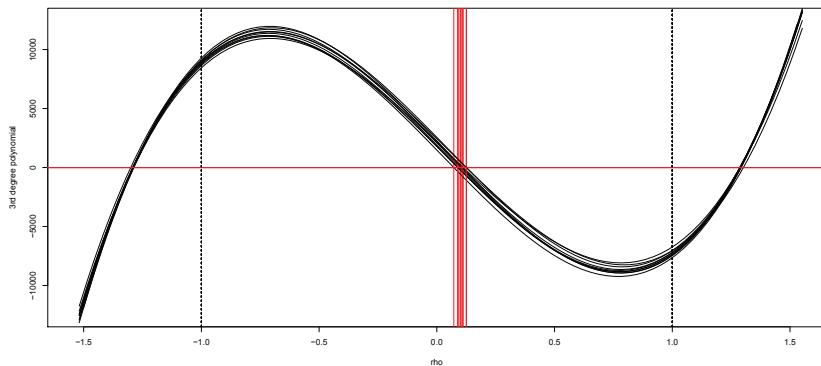


Figure 14: The third degree polynomial in (A.115) for 10 different generated data. The red vertical line shows $\hat{\rho}$.

the model is fitted using PROC MIXED in SAS (for a single overall intercept). For combining the results from different sub-samples we have used proportional weights and size-proportional weights:

$$\begin{cases} \text{Prop} = \frac{C_k}{\sum_l C_l}, \\ \text{Size.Prop} = \frac{C_k n_k}{\sum_l C_l n_l}. \end{cases} \quad (\text{A.155})$$

The results are compared with full likelihood (Table 13). In contrast to the compound-symmetry case, the size-proportional weights show much better results than the proportional weights. Furthermore, the size-proportional weights in the current simulation are identical with the equal weights. The n_k 's have a much larger influence in the AR(1) case compared to CS. Figures 15, 16, and 17 make the comparisons easier.

Table 13: Comparing proportional, size-proportional and iterated optimal weights with full likelihood for AR(1) covariance structure.

		$\hat{\mu}$	Sd	$\hat{\rho}$	Sd	$\hat{\sigma}^2$
$\rho = 0.1$	Prop.	-0.00190	0.01615	0.10027	0.01158	1.99642
	Size.Prop.	-0.00207	0.01538	0.10024	0.01080	1.99793
	It.Opt.	-0.00206	0.01538	0.10024	0.00993	1.99792
	ML	-0.00207	0.01538	0.10032	0.01078	1.99793
$\rho = 0.5$	Prop.	-0.00212	0.02221	0.49966	0.00954	2.00349
	Size.Prop	-0.00155	0.02156	0.49955	0.00898	2.00257
	It.Opt.	-0.00168	0.02149	0.49956	0.00826	2.00265
	ML	-0.00170	0.02150	0.49986	0.00896	2.00259
$\rho = 0.8$	Prop.	0.00195	0.02890	0.79923	0.00549	1.99529
	Size.Prop	0.00234	0.02904	0.79911	0.00530	1.99542
	It.Opt.	0.00213	0.02855	0.79915	0.00486	1.99538
	ML	0.00212	0.02855	0.79937	0.00527	1.99519

Simulations with Proportional and Size-proportional Weights: ρ near 0/1

We now present a comparison between proportional and size-proportional weights. We see that, for ρ 's near 1 (i.e., near CS), size-proportional weights are worse than proportional weights.

We consider $c_1 = 500$, $c_2 = 250$, $c_3 = 250$, $c_4 = 500$, and $n_1 = 5$, $n_2 = 10$, $n_3 = 10$, $n_4 = 5$. Parameters are set as $\mu = 0$, $\sigma = 2$ and $\rho \in \{0.01, 0.2, 0.5, 0.8, 0.9, 0.95, 0.99\}$. The data are generated 100 times and the model is fitted using PROC MIXED in SAS (for a single overall intercept). For combining results from different subsamples we have used proportional weights and size-proportional weights as in (A.155). The results are compared with full likelihood results.

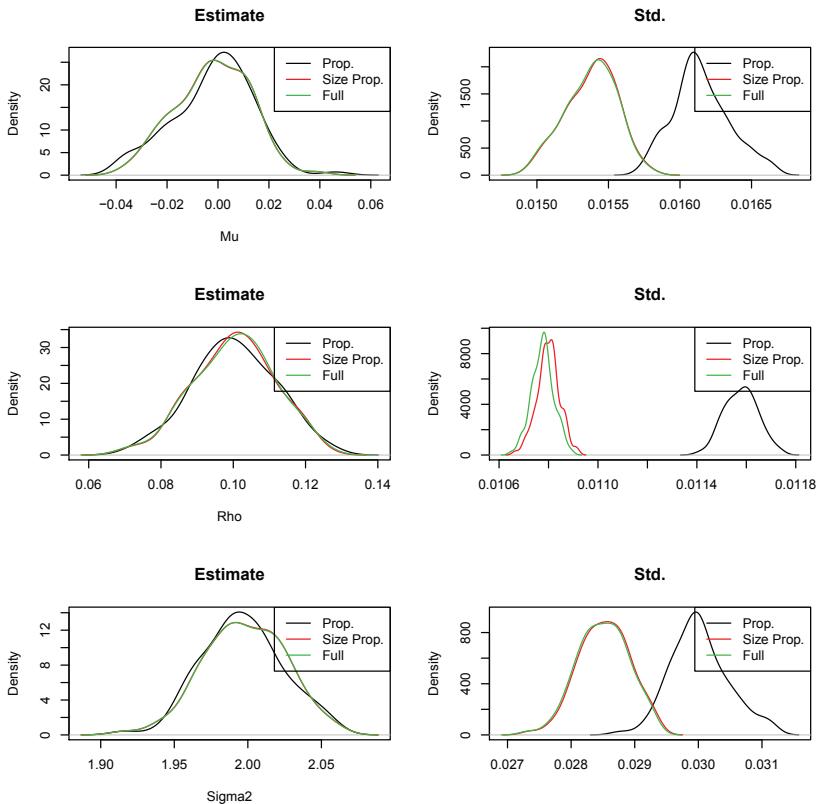


Figure 15: Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.1$.

In Figure 18, for $\rho = 0.99$ and 0.95 , the size-proportional weights perform worse than the proportional weights. This is expected, because in this case AR(1) approaches CS. This result is clearer in the left panel of Figure 18, where the standard deviations are shown. For ρ 's near 1, the proportional weights are as efficient as full likelihood, while as ρ moves further from 1 this would happen

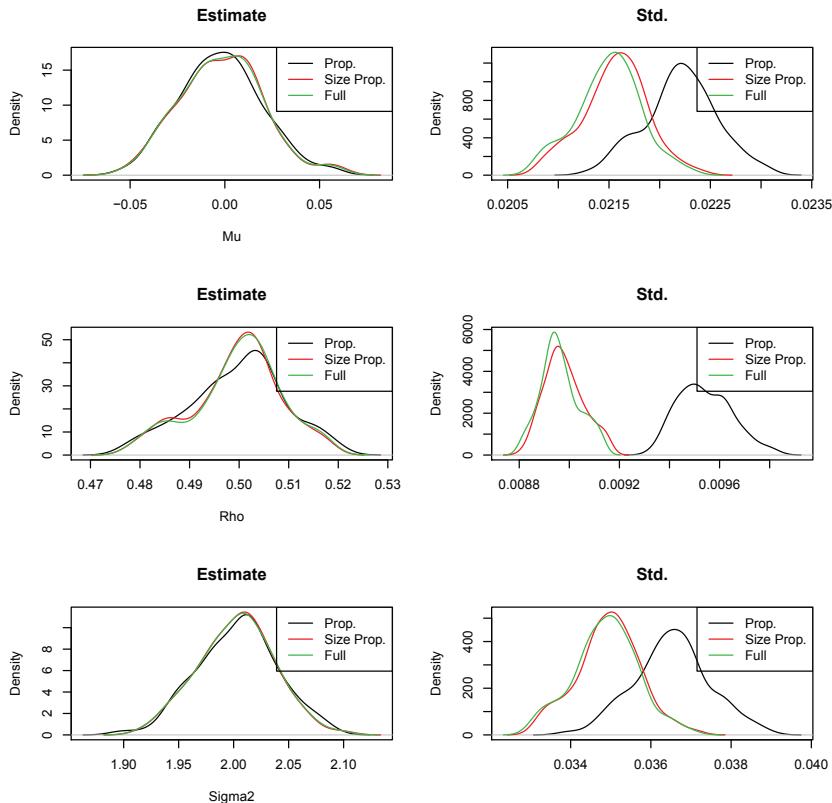


Figure 16: Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.5$.

for size-proportional weights.

Figure 19 shows this phenomenon more clearly, as for some selected ρ 's (0.01, 0.5, 0.95) the density plot for all 100 simulated datasets is plotted rather than a boxplot. The size-proportional weights are better than proportional weights if ρ is not very close to 1. As soon as ρ becomes 0.95 or 0.99, the size-proportional weights become

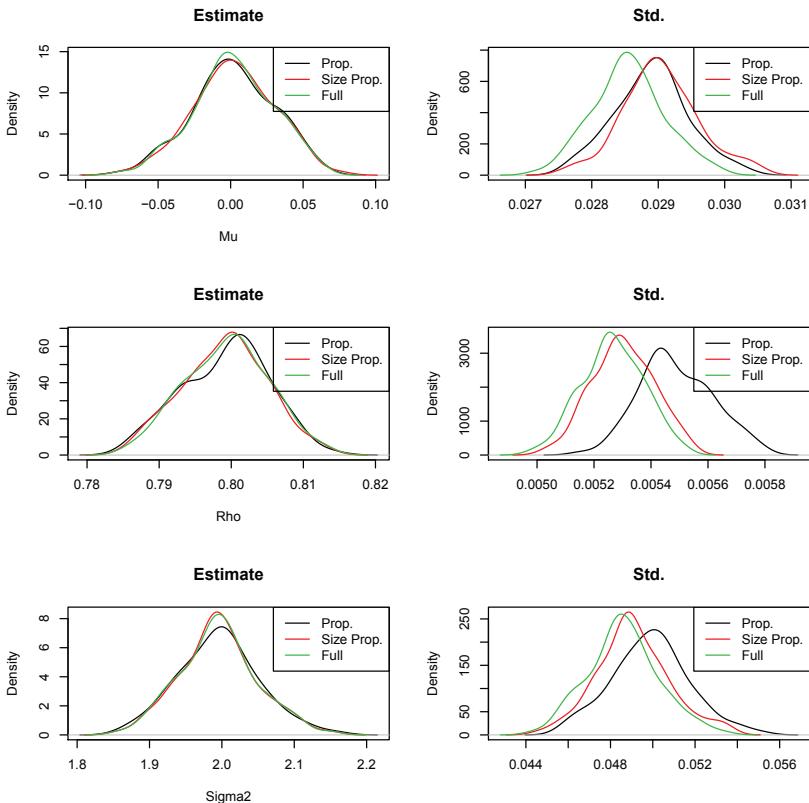


Figure 17: Comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.8$.

worse.

Simulations With Optimal Weights

Given the covariance matrix of the parameter estimators, finding the optimal weights is straightforward, but in practice the unknown parameters therein need to be estimated. Here we compare the



Figure 18: Simulation study. Boxplots comparing proportional and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.99, 0.95, 0.9, 0.8, 0.5, 0.2, 0.01$. In every section of the boxplots (which are separated by dashed lines) the first out of three represents the proportional weights, the middle of is size-propotional weights and the one on the right shows the results for the full likelihood. The first row presents the estimates while the second row shows the standard deviations of these estimates.



Figure 19: Simulation study. Comparing proportional, size-proportional and full likelihood results via their empirical density for the 100 replications. In all of the figures $\mu = 0$ and $\sigma^2 = 2$. The first row is for $\rho = 0.01$, the middle one is for $\rho = 0.5$ and last one corresponds to $\rho = 0.99$. In each figure the ticker dotted line corresponds to full likelihood, the dashed line is for size-proportional weights and the solid line is for proportional weights.

iterative weights with size-proportional weights and ML. See Figures 20, 21, 22, and Table 13. As expected, the optimal weights lead to estimates very close to the MLE; the difference between them being numerical.

Size-proportional weights are used as the initial weights to begin the iterative procedure. One interesting outcome of this simulation is that the iterative procedure always converged after just one iteration. This means, iterated optimal weights are just like the approximated optimal weights, but there, instead of using $\hat{\theta}_k$ from each sub-sample, one may use $\tilde{\theta}$ obtained from all sub-samples using a non-optimal but good weight.

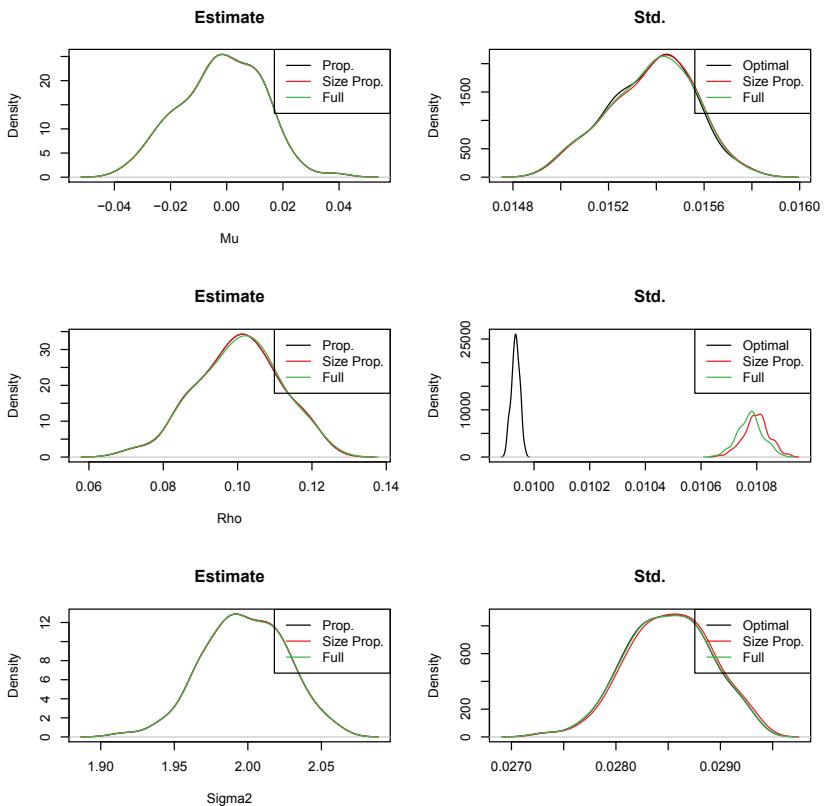


Figure 20: Comparing iterated optimal and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.1$.

Simulations on Computation Time

Here, some summary tables are presented to summarize the results which are already presented via figures earlier. Furthermore, a table and a figure are added to compare computation time for closed form solutions to numerical ones.

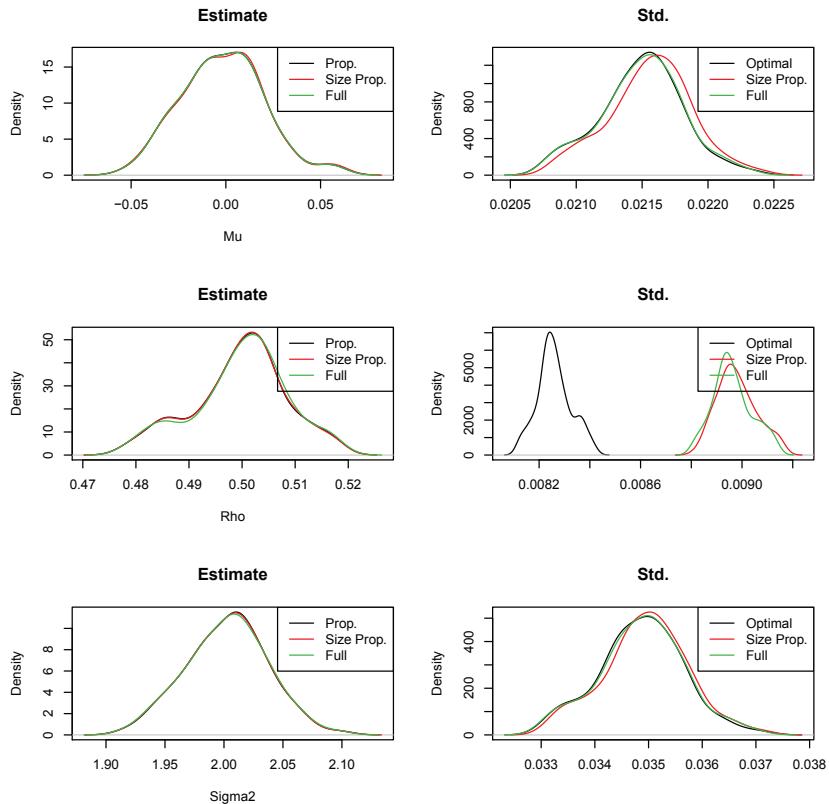


Figure 21: Comparing iterated optimal and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.5$.

In each table the mean of the estimated parameter and its standard deviation using the 100 replications are given, together with the standard deviation of those 100 numbers (in parentheses). If θ is the parameter of interest, $\hat{\theta}$ is its estimate and θ_0 is its real value,

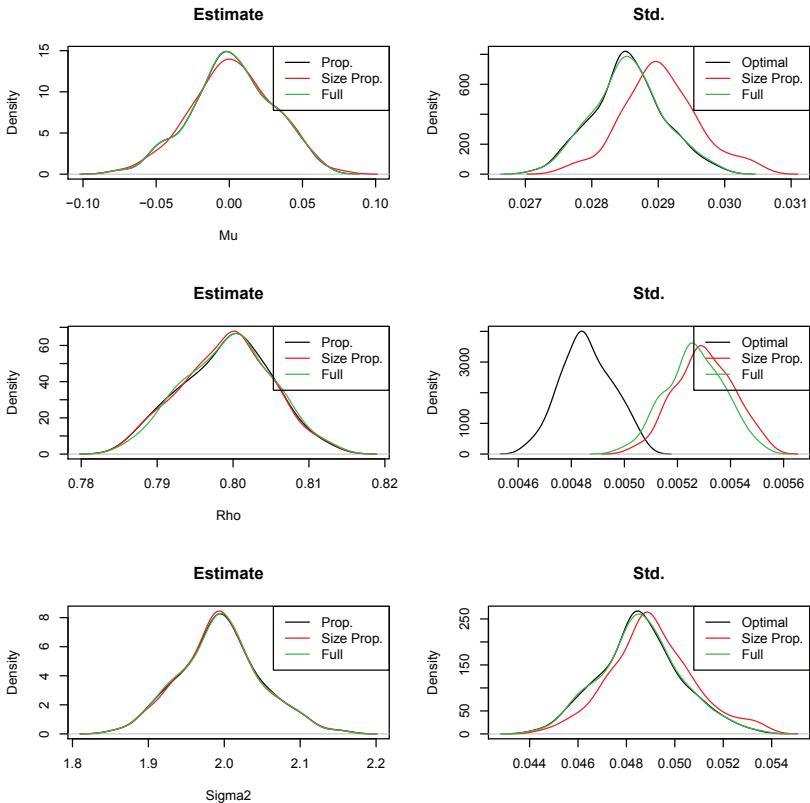


Figure 22: Comparing iterated optimal and size-proportional weights with full likelihood for 100 replications with $\mu = 0$, $\sigma^2 = 2$ and $\rho = 0.8$.

then the MSE is computed as follows:

$$\text{MSE}(\hat{\theta}) = \frac{1}{100} \sum_{i=1}^{100} (\hat{\theta}_i - \theta_0)^2. \quad (\text{A.156})$$

Table 14 summarizes the results for μ . The sample splitting estimates are computed using proportional and size-proportional
Page 299

(identical to equal weights in this case) weights. The results using the full sample are also given. The third column in Table 14 presents the averaged (over 100 replications) estimated μ and its standard deviation. The fourth column presents the averaged estimated standard deviation for $\hat{\mu}$ (over 100 replications) and its standard deviation. The last column shows the MSE computed using (A.156) for $\mu_0 = 1$. Tables 15 and 16 shows the same results for ρ and σ^2 ($\sigma_0^2 = 2$), respectively.

Table 17 compares the computation time between closed-form and interative methods. The closed-form solutions are implemented in R and for the numerical methods the MIXED procedure in SAS is used, with error covariance structure set to AR(1).

The data are generated using $n = 10$ for all clusters, with c is varying from 100 to 1000000. Therefore, the design is balanced and the point of this comparison is to see how the computation time is reduced in each split.

As one may see in Table 17 and Figure 23, using closed form solutions significantly reduces the computation time. This means, as well as the computation time reduction due to splitting the data, using closed form solutions within each split the computation time reduction is also huge: for example, for a million clusters, the reduction is from almost one hour to less than 5 seconds. Figure 23 shows that computation time using closed form solution changes linearly with the number of clusters, while this will be exponential using an iterative solution.

To assess the effect of the overall size of the dataset, the model is fitted to two concatenated copies of the same set. Computation time results are presented in Table 17 and Figure 23. The data are generated with $\mu = 0$, $\sigma^2 = 2$, and $\rho = 0.25$.

Table 14: Simulation study. Estimating μ and its standard deviation. The mean (standard deviation) of the 100 replications are given together with mean squared errors for $\rho = 0.01, 0.2, 0.5, 0.8, 0.9, 0.99$ using proportional and size-proportional weights comparing with the full likelihood results.

ρ_0	method	mean($\hat{\mu}$) (s.d.)	mean(s.e.($\hat{\mu}$)) (s.d.)	MSE $\times 10^4$
0.01	Prop.	-0.00271 (0.01462)	0.01503 (0.00020)	2.19000
	Size Prop.	-0.00277 (0.01320)	0.01429 (0.00018)	1.80172
	Full	-0.00275 (0.01319)	0.01428 (0.00018)	1.79828
0.2	Prop.	0.00158 (0.01677)	0.01752 (0.00025)	2.81056
	Size Prop.	0.00085 (0.01616)	0.01673 (0.00021)	2.59147
	Full	0.00090 (0.01615)	0.01672 (0.00021)	2.58880
0.5	Prop.	0.00391 (0.02244)	0.02217 (0.00037)	5.13770
	Size Prop.	0.00397 (0.02191)	0.02155 (0.00035)	4.91201
	Full	0.00396 (0.02182)	0.02148 (0.00034)	4.87038
0.8	Prop.	0.00130 (0.02790)	0.02894 (0.00050)	7.72450
	Size Prop.	-0.00049 (0.02710)	0.02912 (0.00045)	7.27464
	Full	0.00053 (0.02713)	0.02862 (0.00045)	7.29130
0.9	Prop.	-0.00828 (0.03006)	0.03221 (0.00056)	9.63393
	Size Prop.	-0.00727 (0.03145)	0.03306 (0.00070)	10.3224
	Full	-0.00803 (0.02998)	0.03207 (0.00057)	9.54258
0.99	Prop.	0.00162 (0.03663)	0.03597 (0.00064)	13.3123e
	Size Prop.	-0.00007 (0.03930)	0.03797 (0.00088)	15.2876
	Full	0.00156 (0.03666)	0.03597 (0.00064)	13.3305

Table 15: Simulation study. Estimating ρ and its standard deviation. The mean (standard deviation) of the 100 replications are given together with mean squared errors for $\rho = 0.01, 0.2, 0.5, 0.8, 0.9, 0.99$ using proportional and size-proportional weights comparing with the full likelihood results.

ρ_0	method	mean($\hat{\rho}$) (s.e.)	mean(s.e.($\hat{\rho}$)) (s.e.)	MSE
0.01	Prop.	0.01077 (0.01237)	0.01165 (0.00006)	1.52178e-04
	Size Prop.	0.01115 (0.01200)	0.01087 (0.00004)	1.43974e-04
	Full	0.01123 (0.01203)	0.01084 (0.00004)	1.44675e-04
0.2	Prop.	0.19960 (0.01213)	0.01133 (0.00007)	1.45806e-04
	Size Prop.	0.19973 (0.01145)	0.01058 (0.00005)	1.29974e-04
	Full	0.19986 (0.01142)	0.01056 (0.00005)	1.29174e-04
0.5	Prop.	0.49956 (0.00963)	0.00954 (0.00011)	9.19119e-05
	Size Prop.	0.49965 (0.00904)	0.00898 (0.00008)	8.11057e-05
	Full	0.49990 (0.00904)	0.00896 (0.00008)	8.08877e-05
0.8	Prop.	0.79973 (0.00541)	0.00548 (0.00012)	2.90660e-05
	Size Prop.	0.79990 (0.00483)	0.00529 (0.00009)	2.31132e-05
	Full	0.80018 (0.00489)	0.00525 (0.00009)	2.36726e-05
0.9	Prop.	0.90017 (0.00286)	0.00321 (0.00008)	8.14862e-06
	Size Prop.	0.90013 (0.00297)	0.00318 (0.00008)	8.76257e-06
	Full	0.90040 (0.00292)	0.00312 (0.00008)	8.60114e-06
0.99	Prop.	0.98994 (0.00038)	0.00039 (0.00001)	1.45292e-07
	Size Prop.	0.98992 (0.00042)	0.00041 (0.00002)	1.77848e-07
	Full	0.98997 (0.00037)	0.00039 (0.00001)	1.37289e-07

Details on PANSS Data Analysis

As one may see from Table 18, by far the majority of the study subjects have complete data and hence belong to the first pattern.

Figure 24 presents boxplots for the entire set of data, for the subjects from the first pattern only, and for various split samples.

To examine the choice of an AR(1) covariance structure, Table 19
Page 302

Table 16: Simulation study. Estimating σ^2 and its standard deviation. The mean (standard deviation) of the 100 replications are given together with mean squared errors for $\rho = 0.01, 0.2, 0.5, 0.8, 0.9, 0.99$ using proportional and size-proportional weights comparing with the full likelihood results.

ρ_0	method	mean($\hat{\sigma}^2$) (s.e.)	mean(s.e.($\hat{\sigma}^2$)) (s.e.)	MSE
0.01	Prop.	1.99964 (0.02960)	0.02981 (0.00049)	8.67280e-04
	Size Prop.	2.00165 (0.02836)	0.02834 (0.00040)	7.98842e-04
	Full	2.00167 (0.02832)	0.02832 (0.00040)	7.97002e-04
0.2	Prop.	2.00581 (0.02907)	0.03093 (0.00055)	8.70077e-04
	Size Prop.	2.00484 (0.02778)	0.02936 (0.00044)	7.87298e-04
	Full	2.00473 (0.02772)	0.02933 (0.00044)	7.83248e-04
0.5	Prop.	1.99783 (0.03860)	0.03638 (0.00097)	1.47960e-03
	Size Prop.	1.99890 (0.03747)	0.03488 (0.00085)	1.39116e-03
	Full	1.99897 (0.03748)	0.03481 (0.00085)	1.39152e-03
0.8	Prop.	2.00013 (0.04900)	0.05002 (0.00166)	2.37744e-03
	Size Prop.	2.00136 (0.04423)	0.04930 (0.00137)	1.93881e-03
	Full	2.00101 (0.04569)	0.04880 (0.00142)	2.06754e-03
0.9	Prop.	2.00122 (0.05767)	0.05872 (0.00196)	3.29407e-03
	Size Prop.	2.00089 (0.06037)	0.05915 (0.00230)	3.60829e-03
	Full	2.00115 (0.05876)	0.05793 (0.00198)	3.41986e-03
0.99	Prop.	1.99683 (0.06941)	0.07117 (0.00254)	4.77911e-03
	Size Prop.	1.99527 (0.07598)	0.07484 (0.00344)	5.73813e-03
	Full	1.99641 (0.06940)	0.07093 (0.00253)	4.78099e-03

shows three model selection criteria to compare different error covariance structures. Changing from independence structure ($R = \sigma^2 I$) to compound-symmetry ($R = \sigma^2 I$) the criteria decrease with a large amount, and the same when changing to AR(1). The step to an unstructured covariance does not make a big difference (considering that the unstructured covariance would have 21 parameters to estimate compared to 2 parameters in the AR(1) model). Therefore,
Page 303

Table 17: Simulation study. The computation time for a sample with $n = 10$ and $c = 1e+02, 1e+03, 1e+04, 5e+04, 1e+05, 3e+05, 5e+05, 7e+05, 9e+05, 1e+06$. The closed form solution is obtained by implementing the results of this paper in R, and the numerical solution is obtained using PROC MIXED in SAS to estimate a repeated measurement model with AR(1) covariance structure.

time (s)	1e+02	1e+03	1e+04	5e+04	1e+05	3e+05	5e+05	7e+05
Closed form	0.00	0.00	0.03	0.23	0.34	1.45	2.07	3.37
Numerical	0.08	0.13	1.04	10.45	34.74	268.96	770.74	1611.4

□□

Figure 23: Simulation study. Comparing computation time using closed form (left) and numerical (right) solutions. The horizontal axis shows number of clusters (c) and the vertical axis shows the computation time in seconds.

AR(1) seems to be a good choice.

The 95% confidence intervals, accompanying (4.67), are presented in Figure 25. In order to give more insight in these results, Figure 26 shows the 95% confidence interval in each split, comparing with the full sample splits (the horizontal dashed line in the figure).

The 95% confidence intervals, accompanying (4.68), are presented in Figure 27. Figure 28 shows the 95% confidence intervals for the parameter estimates in each split comparing with the full sample estimate (the horizontal dashed-like in the figure.)

Table 18: PANSS data. Number of clusters in each trial for each cluster pattern.

n	Pattern	Trial				
		FIN-1	FRA-3	INT-2	INT-3	IN
*	*
2	*	.	*	.	.	.
*	.	.	.	*	.	.
*	*	*
3	*	.	*	.	*	.
*	*	.	.	*	.	.
*	*	*	.	*	.	.
*	*	*	.	*	.	.
4	*	.	*	.	*	.
*	*	*	.	*	.	.
*	*	*	*	.	.	.
*	*	*	*	.	.	.
*	*	*	*	.	*	.
*	*	*	*	*	.	.
*	*	*	*	*	*	.
5	*	*	*	.	*	*
*	*	*	*	.	*	.
*	*	.	*	.	*	*
*	*	*	*	.	*	*
*	*	*	*	*	*	.
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
6	*	*	*	.	*	*
*	*	*	*	.	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*

Fieller's method and Delta method

Here, the Fieller's method to construct a confidence interval for (5.9) as well as the Delta method are discussed. The ratio which
Page 305



Figure 24: PANSS data. Boxplots for the entire set of data, for the subject from the first pattern only, and for various split samples.

Table 19: PANSS data. Comparing different error covariance structures using three model comparison criteria for model (4.67) (residual log-likelihood value; AIC; BIC). Three R structures: Ind. : independence structure ($R = \sigma^2 I$), CS: compound-symmetry structure ($R = \sigma^2 I + dJ$), AR(1): AR(1) structure ($R_{ij} = \sigma^2 \rho^{|i-j|}$), UN: unstructured ($R_{ij} = \sigma_{ij}^2$).

Model	-2 Res.log.lik.	AIC	BIC
Unstructured	80005.1	80047.1	80164.1
AR(1)	80522.6	80526.6	80537.8
Compound symm.	82683.1	82687.1	82698.3
Independence	89546.1	89548.1	89553.7

we require a confidence interval for is:

$$n_f = \frac{\hat{\sigma}^2}{\hat{\tau}\varepsilon}.$$

Suppose n_1 is a small-feasible sub-sampling size chosen to estimate the parameters in the model (e.g., $n_1 = 5$). From ?, one finds:
Page 306

$$\begin{aligned} \text{Var} \begin{pmatrix} \widehat{\sigma^2} \\ \widehat{\tau\varepsilon} \end{pmatrix} &= \\ \frac{2\sigma^4}{Nn_1(n_1-1)} \begin{pmatrix} n_1 & -\varepsilon \\ -\varepsilon & \varepsilon^2 \frac{\sigma^4 + 2(n_1-1)\tau\sigma^2 + n_1(n_1-1)\tau^2}{\sigma^4} \end{pmatrix} \\ &= \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{pmatrix}. \quad (\text{A.157}) \end{aligned}$$

The Fieller's confidence interval for (5.9) can be calculated following these steps.

$$\begin{aligned} C_1^2 &= \frac{s_{11}}{\widehat{\sigma^2}}, \\ C_2^2 &= \frac{s_{22}}{(\varepsilon\widehat{\tau})^2}, \\ r &= \frac{s_{12}}{\sqrt{s_{11}s_{22}}}, \\ A &= C_1^2 + C_2^2 - 2rC_1C_2, \\ B &= t^2 C_1^2 C_2^2 (1 - r^2), \\ L &= \frac{\widehat{\sigma^2} (1 - z_{\alpha/2}^2) r C_1 C_2 - z_{\alpha/2} \sqrt{A - B}}{\varepsilon\widehat{\tau} (1 - z_{\alpha/2}^2 C_2^2)}, \\ U &= \frac{\widehat{\sigma^2} (1 - z_{\alpha/2}^2) r C_1 C_2 + z_{\alpha/2} \sqrt{A - B}}{\varepsilon\widehat{\tau} (1 - z_{\alpha/2}^2 C_2^2)}. \end{aligned} \quad (\text{A.158})$$

One may take $n_f = \max\{L, U\}$. Of course, in case were $z_{\alpha/2}^2 C_2^2 < 1$, there would be no finite interval. As it was mentioned, one may also use the Delta method as follows:

$$\text{Var}\left(\widehat{\frac{\sigma^2}{\varepsilon\hat{\tau}}}\right) \approx \frac{1}{\varepsilon^2\hat{\tau}^2} \left(s_{11} - 2\frac{\sigma^2}{\tau}s_{12} + \frac{\sigma^4}{\tau^2}s_{22} \right). \quad (\text{A.159})$$

Random vertical data splitting for compound-symmetry structure

Assume a single cluster of size N with multivariate normal distribution and compound-symmetry structure for its covariance matrix, see Section 5. Suppose we take M_0 sub-samples of size n_0 from this cluster. Following ?, the parameter μ should be estimated within each sub-sample, $\hat{\mu}^{(m)}$ ($m = 1, \dots, M_0$) and then the overall estimate can be obtained by averaging these estimates. Furthermore, one needs the within and between sub-samples variabilities to compute the overall variability, see (5.17).

Within each sub-sample the variance of $\hat{\mu}^{(m)}$ can be computed using the variance formula for a mean estimator in CS-multivariate normal ? :

$$S_W = \frac{\sigma^2}{n_0} [1 + (n_0 - 1)\rho]. \quad (\text{A.160})$$

Estimating the between variability requires computing $E\left\{(\bar{\mu} - \hat{\mu})^2\right\}$, where $\bar{\mu}$ is the average of $\hat{\mu}^{(m)}$'s ($m = 1, \dots, M_0$). Set S_m as the index of members of sub-sample m and $S_{m'}$ the rest

of the sample, then,

$$\hat{\mu}^{(m)} - \bar{\mu} = \frac{1}{n_0} \left\{ \left(1 - \frac{1}{M_0}\right) \sum_{i \in S_m} y_i - \frac{1}{M_0} \sum_{m=1}^{M_0} \sum_{i \in S_{m'}} y_i \right\}.$$

Therefore,

$$E \left\{ (\hat{\mu}^{(m)} - \bar{\mu})^2 \right\} = \frac{1}{n_0} (T_1 + T_2 + T_3 + T_4), \quad (A.161)$$

where,

$$\begin{cases} T_1 = \left(1 - \frac{1}{M_0}\right) E \left[\sum_{i \in S_m} y_i \right]^2 \\ T_2 = \frac{1}{M_0^2} (M_0 - 1) E \left[\sum_{i \in S_m} y_i \right]^2 \\ T_3 = -2 \left(1 - \frac{1}{M_0}\right) \frac{1}{M_0} (M_0 - 1) E \left\{ \left(\sum_{i \in S_{m'}} y_i\right) \left(\sum_{i' \in S_{m'}} y_{i'}\right) \right\} \\ T_4 = 2 \frac{1}{M_0} \frac{(M_0 - 1)(M_0 - 2)}{2} E \left\{ \left(\sum_{i \in S_m} y_i\right) \left(\sum_{i' \in S_{m'}} y_{i'}\right) \right\}. \end{cases} \quad (A.162)$$

From (A.162) one may find,

$$\begin{cases} T_1 + T_2 = \frac{M_0 - 1}{M_0} n_0 \sigma^2 [1 + (n_0 - 1)\rho] \\ T_3 + T_4 = -\frac{M_0 - 1}{M_0} \frac{\sigma^2 n_0^2}{N} [1 + (N_1)\rho]. \end{cases} \quad (A.163)$$

Therefore, the between variability can be computed as follows,

$$S_B = E \left\{ (\hat{\mu}^{(m)} - \bar{\mu})^2 \right\} = \frac{M_0 - 1}{M_0} \left[\frac{1 + (n_0 - 1)\rho}{n_0} - \frac{1 + (N_1)\rho}{N} \right]. \quad (A.164)$$

Using (A.160) and (A.164), the total variability can be computed

as follows,

$$S_T = \frac{1}{M_0} \sigma^2 \frac{1 + (n_0 - 1)\rho}{n_0} + \frac{M_0 - 1}{M_0} \sigma^2 \frac{1 + (N - 1)\rho}{N} \quad (\text{A.165})$$

Comparing the variance in (A.165) with the variance when estimating μ using the full sample would give,

$$\text{ARE} = \frac{M_0 - 1}{M_0} + \frac{1}{M_0} \frac{N}{n_0} \frac{1 + (n_0 - 1)\rho}{1 + (N - 1)\rho} \quad (\text{A.166})$$

For example, if sub-sampling is done only once ($M_0 = 1$), then,

$$\text{ARE} = \frac{N}{n_0} \frac{1 + (n_0 - 1)\rho}{1 + (N - 1)\rho}, \quad (\text{A.167})$$

The same calculation is possible for $M_0 = 2$,

$$\text{ARE} = \frac{1}{2} + \frac{1}{2} \frac{N}{n_0} \frac{1 + (n_0 - 1)\rho}{1 + (N - 1)\rho}. \quad (\text{A.168})$$

By considering the desired ARE as $1 + \epsilon$, one may find n_0 .

Combination rule for p -values in LES dataset analysis (IMI)

The Kruskal-Wallis test statistic asymptotically follows a χ^2 distribution. Therefore, in order to find the combined p -value, we may follow the procedure proposed by ?. More details can be found in ? and ?. The combination is done using `micombine.chisquare` in package `miceadds` in R, ?.

- **Averaging.** Compute the test statistic for each imputed data and take their average: $\bar{\chi^2} = \frac{1}{M} \sum_{m=1}^M \chi_m^2$.
- **Relative variance increase.** $r = \left(1 + \frac{1}{M}\right) \frac{\sum_{m=1}^M (\sqrt{\chi_m^2} - \sqrt{\bar{\chi^2}})^2}{M-1}$
- **Test statistic.** $D = \frac{\frac{\chi^2}{\kappa} - \frac{M+1}{M-1}r}{1+r}$, where κ is the degrees-of-freedom of the χ_m^2 . For a χ^2 -test of independence, it is $(k_1 - 1)(k_2 - 1)$ where k_1 and k_2 are the number of levels of two categorical variables. Take five diagnosis groups and variable Eye with two levels, then $\kappa = (5 - 1)(2 - 1) = 4$. For the Kruskal-Wallis test, κ is the number of groups minus 1. So, when comparing values of a continuous variable among three types of diagnosis, $\kappa = 2$. When this comparison is done for all five types, $\kappa = 4$.
- **Computing p-value.** $p\text{-value} = P(F_{\kappa, \nu} > D)$, where F is the F -distribution and $\nu = \kappa^{-3/M}(M - 1) \left(1 + \frac{1}{r}\right)^2$

The surrogate model

The model takes the form:

$$\begin{cases} S_{ij} = \mu_S + m_{S_i} + (\alpha + a_i)Z_{ij} + \varepsilon_{S_{ij}}, \\ T_{ij} = \mu_T + m_{T_i} + (\beta + b_i)Z_{ij} + \varepsilon_{T_{ij}}, \end{cases} \quad (\text{A.169})$$

where S_{ij} and T_{ij} are the surrogate and true endpoints, and Z_{ij} is the treatment indicator (1 for active; 0 for placebo), respectively. Index i refers to the trial and j to the subject. Further, μ_S and μ_T are fixed intercepts and m_{S_i} and m_{T_i} are random intercepts,

for surrogate and true endpoint, respectively. The fixed treatment effects are α (for S) and β (for T), and the centre-specific treatment effects are a_i (for S) and b_i (for T). Random intercepts and slopes are assumed to follow $(m_{S_i}, m_{T_i}, a_i, b_i)' \sim N(0, D)$, where

$$D = \begin{pmatrix} d_{SS} & & & \\ d_{ST} & d_{TT} & & \\ d_{Sa} & d_{Ta} & d_{aa} & \\ d_{Sb} & d_{Tb} & d_{ab} & d_{bb} \end{pmatrix}. \quad (\text{A.170})$$

Furthermore, for the error terms,

$$\begin{pmatrix} \varepsilon_{S_{ij}} \\ \varepsilon_{T_{ij}} \end{pmatrix} \sim N(0, \Sigma), \quad \Sigma = \begin{pmatrix} \sigma_{SS} & & \\ \sigma_{ST} & \sigma_{TT} & \end{pmatrix}. \quad (\text{A.171})$$

The parameters of interest for the surrogate evaluation are trial- and individual-level coefficients of determination, R_{trial}^2 and R_{indiv}^2 , respectively:

$$R_{\text{trial}}^2 = \frac{1}{d_{ab}} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}' \begin{pmatrix} d_{SS} & d_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}, \quad R_{\text{indiv}}^2 = \frac{\sigma_{ST}^2}{\sigma_{SS}\sigma_{TT}}. \quad (\text{A.172})$$

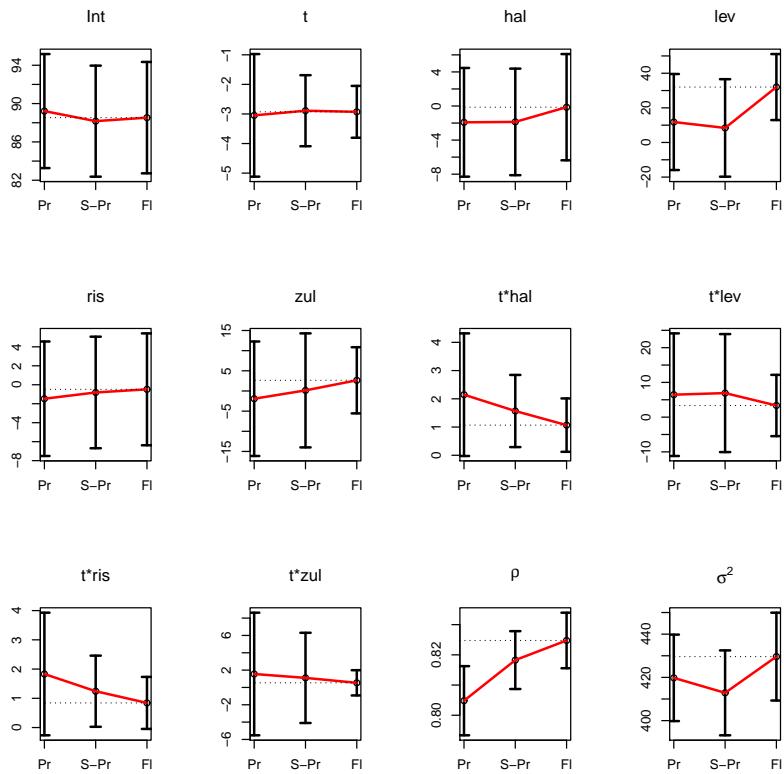


Figure 25: PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates using sample splitting, combined with proportional (Pr - first) and size-proportional (S-Pr - second) weights, and full likelihood (Fl - third). The dashed horizontal line shows the full likelihood estimate. The model used in here is without trial effect (4.67).

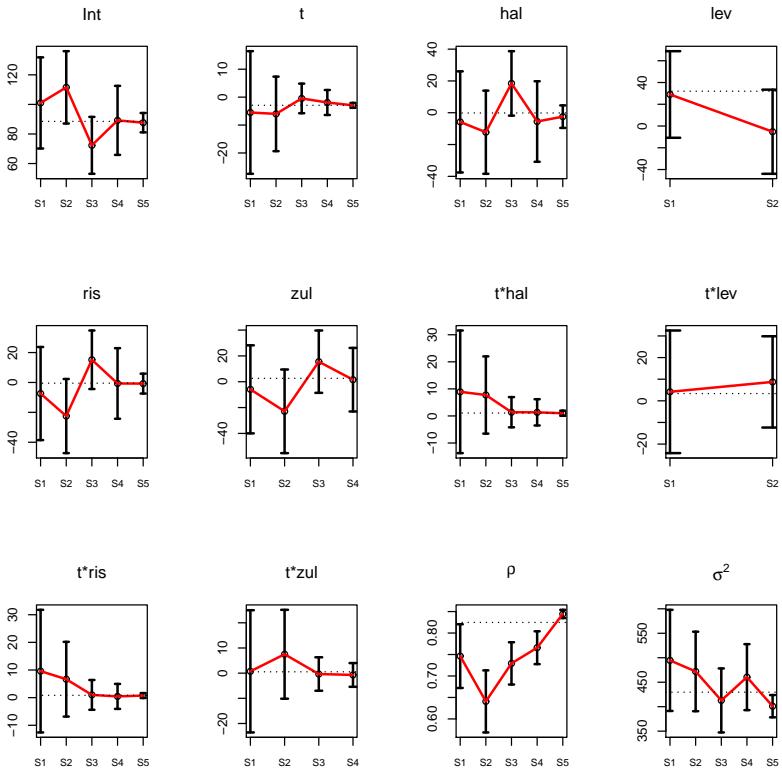


Figure 26: PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates within each split. The dashed horizontal line shows the full likelihood estimate. The model used here is without trial effect (4.68).

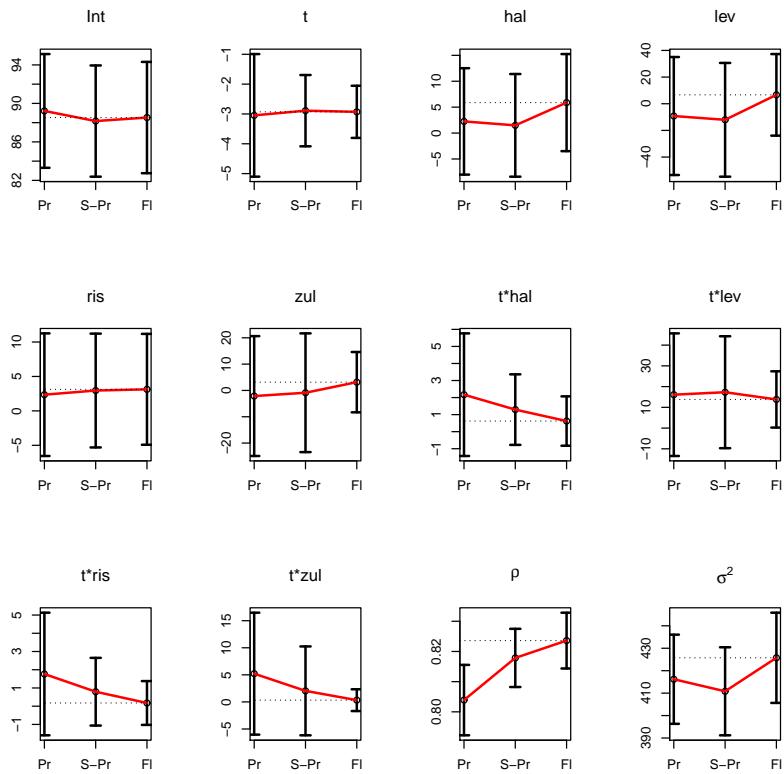


Figure 27: PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates using sample splitting, combined with proportional (Pr - first) and size-proportional (S-Pr - second) weights, and full likelihood (Fl - third). The dashed horizontal line shows the full likelihood estimate. The model used in here is with trial effect (4.68).

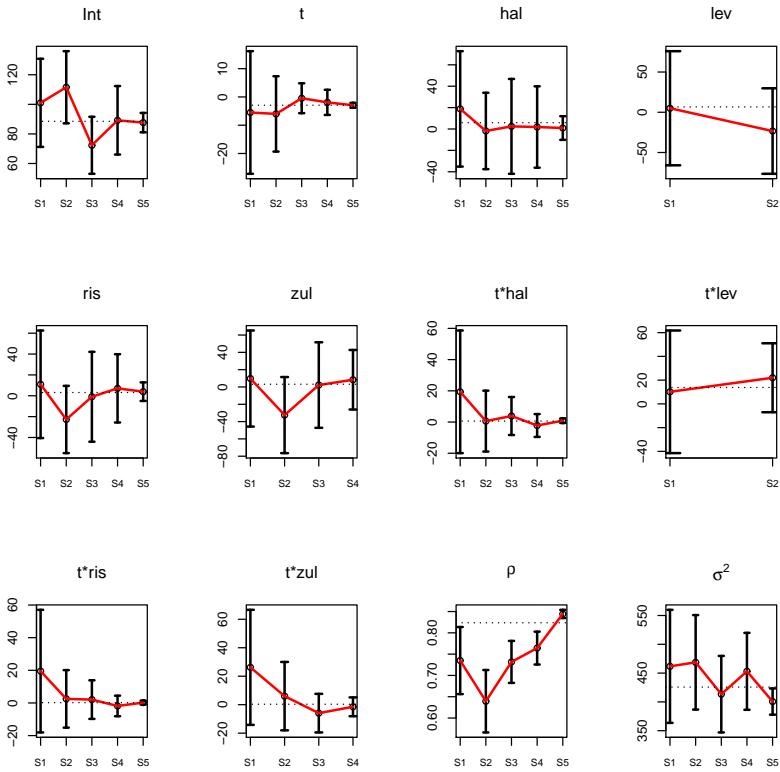


Figure 28: PANSS data. 95% confidence intervals for fixed effects and variance components estimates and the standard deviations of these estimates within each split. The dashed horizontal line shows the full likelihood estimate. The model used in here is with trial effect (4.68).