STAT/BIOST 571: Homework 6

Philip Pham

February 26, 2019

Problem 1: Fitting and interpreting the results of a linear mixed effects model; robust standard error estimation (20 points)

Download the `creatinine.csv` dataset from the course website. This file contains repeated observational data for 619 subjects, some of whom have hypertension and some of whom have a hereditary kidney disease, as indicated by the `group` variable, according to the coding in the Table 1. The outcome variable is `scr`, the reciprocal of serum creatinine. Serum creatinine is a measure of kidney function, with lower values indicating better kidney function. Higher values of the reciprocal reported in `scr` indicate better kidney function. The observations were taken at arbitrary times from each subject, with the number of observations ranging from 1 to 22. Ignoring hypertension status, we are interested in estimating the rate of change of `scr` for subjects with and without hereditary kidney disease. Thus, the only fixed effect covariates in your model should be age, kidney disease status, and possibly an interaction between these. In order to account for correlation within subjects, you will be fitting a linear mixed effects model with uncorrelated random slopes and intercepts and serial correlation of residuals that follows a spherical correlation model, including a nugget (correlation should be based on the timing of observations). Please use the `lme()` function in the `nlme` package in R to fit your models (i.e., do not code your own nonlinear optimization).

Group	Kidney disease	Hypertension	Sample size
1	Yes	Yes	294
2	Yes	No	103
3	No	Yes	73
4	No	No	149

Table 1: Measurements of serum creatinne reciprocals from 619 subjects in four groups

(a) *Fit the model by ML and report the estimated values of all variance parameters.*

The parameter estimates of fitting the model without and with an interaction term can be seen in Tables 3 and 4. The variance parameters can be found in Table 2. Their meaning is detailed in the subsequent paragraphs. The R model summaries can be found in the Appendix.

	Model	
Parameter	Without interaction (Equation 1)	With interaction (Equation 2)
$\hat{\sigma}$	0.2633414	0.225834
$\hat{\sigma}_{\gamma_0}$	0.04643211	0.1317468
$\hat{\sigma}_{\gamma_1}$	0.00522239	0.004823202
$\hat{\alpha}_r$	7.8894707	4.6700641
$\hat{\alpha}_n$	0.1759323	0.2299764

Table 2: Variance parameters for ML-fitted models.

Let t_{ij} be the age of the subject i at observation j. Let x_i indicate whether the subject has has kidney disease. Without an interaction term, the mean model is

$$Y_{ij} = (\beta_0 + \gamma_0) + \beta_2 x_i + (\beta_1 + \gamma_1)\, t_{ij} + \epsilon_{ij}. \tag{1}$$

With the interaction term, the mean model is

$$Y_{ij} = (\beta_0 + \gamma_0) + \beta_2 x_i + (\beta_1 + \beta_3 x_i + \gamma_1)\, t_{ij} + \epsilon_{ij}. \tag{2}$$

γ_j are the random effects, where γ_0 is subject-specific adjustment to the intercept, and γ_1 is the subject-specific adjustment to the slope.

The covariance structure of a cluster i can be described by the matrix

$$\Sigma_i = \sigma^2 \left(Z_i G Z_i^{\intercal} + R_i \right) \tag{3}$$

Z_i is a $m_i \times 2$ matrix, where the first column entries are all 1s, and the second column entries are ages for each subject t_{ij}.

G is a 2×2 diagonal matrix that describes the variance of the random effects γ_0 and γ_1:

$$G = \frac{1}{\sigma^2} \begin{pmatrix} \sigma_{\gamma_0}^2 & 0 \\ 0 & \sigma_{\gamma_1}^2 \end{pmatrix} \tag{4}$$

R_i is an $m_i \times m_i$ matrix that describes the correlations between the ϵ_{ij}s for different js with a nugget parameter $0 \leq \alpha_n < 1$ and range parameter $\alpha_r > 0$. $R_{ijj} = 1$ and $R_{ijj'} = (1 - \alpha_n) \exp\left(-\frac{|t_{ij} - t_{ij'}|}{\alpha_r} \right)$. Estimates for these parameters can be found in Table 2.

(b) *Report point estimates and standard errors for all fixed effect coefficients in your model. Include three versions of standard error estimates: (i) robust/empirical sandwich SEs that correctly account for clustering of the data, (ii) bootstrap SEs that correctly account for clustering of the data, and (iii) model based SE estimates based on the assumed random effect model being correct.*

Solution: See Table 3 for the standard errors.

Let $\hat{V}_i$ be the result of substituting the variance parameter estimates in Table 2 into Equation 3. Let $W_i = \hat{V}_i^{-1}$ be the weight matrix for each cluster. Let X_i be the matrix of cluster covariates,

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.532222	0.039557	0.039348	0.041531
age	-0.012915	0.000964	0.000941	0.001030
kidney.disease	-0.281916	0.026260	0.025027	0.034516

Table 3: Standard error estimates for fixed effect parameters.

1s in the first column, age in the second column, and an indicator for kidney disease in the third column. Let Y_i be the cluster response.

If we assume the random effect model is correct, then the covariance matrix for the paramter estimates $\hat{\beta}$ is

$$\hat{\text{var}}\left(\hat{\beta}\right) = \left(\sum_{i=1}^{n} X_i^\intercal W_i X_i\right)^{-1}, \tag{5}$$

and we take the square root of the diagonals to obtain the standard errors for the ML Standard Error column.

For the sandwich standard errors, we use the covariance matrix

$$\hat{\text{var}}\left(\hat{\beta}\right) = \left(\sum_{i=1}^{n} X_i^\intercal W_i X_i\right)^{-1} \left(\sum_{i=1}^{n} X_i^\intercal W_i \left(Y_i - X_i\hat{\beta}\right)\left(Y_i - X_i\hat{\beta}\right)^\intercal W_i X_i\right) \left(\sum_{i=1}^{n} X_i^\intercal W_i X_i\right)^{-1}, \tag{6}$$

which we use to get the Sandwich Standard Error column.

For the bootstrap standard errors, we resample clusters, and then fit a model to the resampled clusters to get samples $\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \ldots, \hat{\beta}^{(L)}$. In this case, $L = 2^{10}$. Then, $\hat{\text{var}}\left(\hat{\beta}\right)$ is estimated by taking the unbiased covariance estimate of the samples. Taking the square root of the diagonals gives us the Bootstrap Standard Error column.

In this case, the standard errors assuming the model is correct and the sandwich standard errors are very similar with the sandwich standard errors being slightly smaller. The bootstrap standard errors are largest. The difference is more than just Monte Carlo error, particularly for β_2 the coefficient for kidney disease, which hints at its interaction with other covariates.

(c) *Now, give point estimates and three versions of standard error estimates for the marginal rates of change in `scr` in subjects with and without kidney disease. As in part (b), your three versions of SE estimates should be: (i) robust/empirical sandwich SEs that correctly account for clustering of the data, (ii) bootstrap SEs that correctly account for clustering of the data, and (iii) model based SE estimates based on the assumed random effect model being correct. Depending on how you parameterized your model, these answers may or may not coincide with estimates you reported in part (b). You may parameterize the model any way you wish, but you must answer this question based on the output of a single call to `lme()`.*

Solution: Table 4

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.190676	0.054232	0.050269	0.050770
age	-0.003108	0.001436	0.001194	0.001217
kidney.disease	0.313575	0.071373	0.068325	0.070047
age:kidney.disease	-0.016649	0.001857	0.001705	0.001706

Table 4: Standard error estimates for fixed effect parameters with interaction term.

Appendix

Code for fitting models and generating tables is attached on the subsequent pages.

LME and Creatinine

Joining Data

In [1]:

```
library(data.table)
library(nlme)
library(parallel)
library(xtable)

creatinine.data <- data.table(read.csv('creat.csv'), key='group')
head(creatinine.data)
```

id	group	age	scr
1	1	35.765	0.182
1	1	37.990	0.088
3	1	51.083	0.156
3	1	52.386	0.116
3	1	52.805	0.087
3	1	52.997	0.067

In [2]:

```
group.data <- data.table(
    group=c(1,2,3,4),
    kidney.disease=c(1,1,0,0),
    hypertension=c(1,0,1,0),
    key='group')
group.data
```

group	kidney.disease	hypertension
1	1	1
2	1	0
3	0	1
4	0	0

In [3]:

```
creatinine.group.data <- creatinine.data[group.data]
setkey(creatinine.group.data, id)
head(creatinine.group.data)
```

id	group	age	scr	kidney.disease	hypertension
1	1	35.765	0.182	1	1
1	1	37.990	0.088	1	1
2	2	24.997	1.429	1	0
2	2	27.441	1.111	1	0
2	2	30.524	1.429	1	0
3	1	51.083	0.156	1	1

Fitting LME Model

In [4]:

```
fit.scr.model <- function(data, interaction.term=FALSE) {
    formula = scr ~ age + kidney.disease
    if (interaction.term) { formula <- update(formula, . ~ . + age:kidney.disease)
}
    lme(formula,
        random=reStruct(~age|id, pdClass='pdDiag'),
        correlation=corExp(form=~age|id, nugget=TRUE),
        method='ML', data=data,
        control=lmeControl(maxIter=100, msMaxIter=100, niterEM=50))
}
```

Covariance for Subject i

In [5]:

```
make.covariance <- function(model, i) {
    # Error term, usually denoted epsilon
    error.correlation <- corMatrix(model$modelStruct$corStruct)[[as.character(i)]]
    if (is.null(error.correlation)) { error.correlation <- 1 }
    # Random effects correlation, Z * G * tranpose(Z).
    Z <- cbind(1, model$data[J(i), age])
    random.correlation <- Z %*% as.matrix(model$modelStruct$reStruct$id) %*% t(Z)
    # Convert correlation matrix into covariance matrix.
    (random.correlation + error.correlation)*(model$sigma*model$sigma)
}
```

β_j Covariance

Maximum Likelihood Estimate

This assumes that the random effects model is correct. I can also be used as the *bread* part of the sandwich estimator.

In [6]:

```
make.covariates <- function(model, data) model.matrix(model$terms, data)

make.ml.parameter.covariance <- function(model) {
    groups <- unique(model$groups)$id
    chol2inv(chol(Reduce(`+`, lapply(groups, function(i) {
        X <- make.covariates(model, model$data[J(i)])
        t(X) %*% chol2inv(chol(make.covariance(model, i))) %*% X
    }))))
}
```

Sandwich Estimate

In [7]:

```
make.response <- function(data) data$scr

make.sandwich.parameter.covariance <- function(model) {
    bread <- make.ml.parameter.covariance(model)
    meat <- Reduce(`+`, lapply(unique(model$groups)$id, function(i) {
        X <- make.covariates(model, model$data[J(i)])
        y <- make.response(model$data[J(i)])
        weights <- chol2inv(chol(make.covariance(model, i)))
        residuals <- as.numeric(make.response(model$data[J(i)]) - X %*% model$coefficients$fixed)
        empirical.covariance <- outer(residuals, residuals)
        t(X) %*% weights %*% empirical.covariance %*% weights %*% X
    }))
    bread %*% meat %*% bread
}
```

Bootstrap Estimate

To account for clustering of the data, we resample clusters.

In [8]:

```
resample.clusters <- function(data) {
    resampled.data <- data[
        data.table(id=sample(unique(data$id), replace=TRUE),
                   new.id=c(1:length(unique(data$id))),
                   key='id')]
    resampled.data[,id:=NULL]
    setnames(resampled.data, 'new.id', 'id')
    setkey(resampled.data, id)
    resampled.data
}
```

Models

Without Interaction Term

In [9]:

```
scr.model <- fit.scr.model(creatinine.group.data)
summary(scr.model, adjustSigma=FALSE)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
        AIC       BIC   logLik
  -53.94986 -11.00314 34.97493

Random effects:
 Formula: ~age | id
 Structure: Diagonal
        (Intercept)        age  Residual
StdDev:  0.04643211 0.00522239 0.2633414

Correlation Structure: Exponential spatial correlation
 Formula: ~age | id
 Parameter estimate(s):
    range    nugget
7.8894707 0.1759323
Fixed effects: list(formula)
                    Value  Std.Error  DF   t-value p-value
(Intercept)     1.5322224 0.03955663 965  38.73490       0
age            -0.0129154 0.00096383 965 -13.40006       0
kidney.disease -0.2819162 0.02626024 617 -10.73548       0
 Correlation:
               (Intr) age
age            -0.849
kidney.disease -0.352 -0.082

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.49081741 -0.56731338 -0.07887231  0.43113427  4.95897388

Number of Observations: 1585
Number of Groups: 619
```

In [10]:

```
ml.parameter.covariance <- make.ml.parameter.covariance(scr.model)
sandwich.parameter.covariance <- make.sandwich.parameter.covariance(scr.model)

bootstrap.parameter.samples <- do.call(rbind, mclapply(
    replicate(1024, creatinine.group.data, simplify=FALSE), function(data) {
        resampled.data <- resample.clusters(data)
        fit.scr.model(resampled.data)$coefficients$fixed
    }, mc.cores=4))
bootstrap.parameter.covariance <- cov(bootstrap.parameter.samples)
```

In [11]:

```
standard.errors <- data.frame(
    `Estimate`=scr.model$coefficients$fixed,
    `ML Standard Error`=sqrt(diag(ml.parameter.covariance)),
    `Sandwich Standard Error`=sqrt(diag(sandwich.parameter.covariance)),
    `Bootstrap Standard Error`=sqrt(diag(bootstrap.parameter.covariance)),
    check.names=FALSE)

print(xtable(standard.errors,
             caption='Standard error estimates for fixed effect parameters.',
             label='tab:standard_errors_no_interaction',
             digits=c(0, 6, 6, 6, 6)),
      booktabs=TRUE, file='standard_errors_no_interaction.tex',
      sanitize.colnames.function=identity,
      sanitize.rownames.function=identity,
      size='small')

standard.errors
```

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.53222240	0.0395566329	0.0393479582	0.041530790
age	-0.01291536	0.0009638288	0.0009409584	0.001030028
kidney.disease	-0.28191623	0.0262602392	0.0250273809	0.034516098

With Interaction Term

In [12]:

```
scr.model.interaction <- fit.scr.model(creatinine.group.data, interaction.term=TRU
E)
summary(scr.model.interaction, adjustSigma=FALSE)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: data 
        AIC       BIC   logLik
  -127.9923 -79.67728 72.99617

Random effects:
 Formula: ~age | id
 Structure: Diagonal
        (Intercept)         age Residual
StdDev:   0.1317468 0.004823202 0.225834

Correlation Structure: Exponential spatial correlation
 Formula: ~age | id 
 Parameter estimate(s):
    range    nugget 
4.6700641 0.2299764 
Fixed effects: list(formula) 
                        Value  Std.Error  DF   t-value p-value
(Intercept)         1.1906763 0.05423200 964 21.955237  0.0000
age                -0.0031083 0.00143568 964 -2.165016  0.0306
kidney.disease      0.3135748 0.07137345 617  4.393438  0.0000
age:kidney.disease -0.0166492 0.00185709 964 -8.965190  0.0000
 Correlation: 
                   (Intr) age    kdny.d
age                -0.928              
kidney.disease     -0.760  0.705       
age:kidney.disease  0.718 -0.773 -0.935

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max 
-2.30553348 -0.53490304 -0.07976542  0.36022352  5.46804808 

Number of Observations: 1585
Number of Groups: 619 
```

In [13]:

```
ml.parameter.covariance.interaction <- make.ml.parameter.covariance(scr.model.inte
raction)
sandwich.parameter.covariance.interaction <- make.sandwich.parameter.covariance(sc
r.model.interaction)

bootstrap.parameter.samples.interaction <- do.call(rbind, mclapply(
    replicate(1024, creatinine.group.data, simplify=FALSE), function(data) {
        resampled.data <- resample.clusters(data)
        fit.scr.model(resampled.data, interaction.term=TRUE)$coefficients$fixed
    }, mc.cores=4))
bootstrap.parameter.covariance.interaction <- cov(bootstrap.parameter.samples.inte
raction)
```

In [14]:

```
standard.errors.interaction <- data.frame(
    `Estimate`=scr.model.interaction$coefficients$fixed,
    `ML Standard Error`=sqrt(diag(ml.parameter.covariance.interaction)),
    `Sandwich Standard Error`=sqrt(diag(sandwich.parameter.covariance.interaction
)),
    `Bootstrap Standard Error`=sqrt(diag(bootstrap.parameter.covariance.interactio
n)),
    check.names=FALSE)

print(xtable(standard.errors.interaction,
             caption='Standard error estimates for fixed effect parameters with in
teraction term.',
             label='tab:standard_errors_interaction',
             digits=c(0, 6, 6, 6, 6)),
      booktabs=TRUE, file='standard_errors_interaction.tex',
      sanitize.colnames.function=identity,
      sanitize.rownames.function=identity,
      size='small')

standard.errors.interaction
```

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.190676306	0.054231995	0.050269334	0.050769991
age	-0.003108265	0.001435677	0.001193725	0.001216667
kidney.disease	0.313574816	0.071373449	0.068324860	0.070047199
age:kidney.disease	-0.016649153	0.001857089	0.001705009	0.001706189