

Longitudinal Models

Sharif Mahmood

Lecturer
Institute of Statistical Research And Training
University of Dhaka

July 31, 2011

Outline

1 Mixed Models

- Definition
- Syntax
- Formula
- Inference
- Example

2 GEE

- Definition
- Syntax
- Correlation Structure
- Inference
- Example

Mixed Models

- 1 Subject-specific or cluster-specific model of correlated/clustered data
- 2 Basic premise is that there is natural heterogeneity across individuals in the study population that is the result of unobserved covariates; random effects account for the unobserved covariates.
- 3 The `lme4` package contains functions for fitting linear mixed models, generalized linear mixed models and nonlinear mixed models.
- 4 The `lme4` package uses S4 classes and methods.
- 5 For more information about fitting mixed models in R using `lme4` see the available vignettes, `vignette(package="lme4")`

lmer()

- The `lmer()` function in the `lme4` package is used to fit linear and generalized linear models.
- Basic syntax,

```
lmer(formula, data, family=NULL, REML=TRUE)
```

<code>formula</code>	Symbolic description of the model to be fitted
<code>data</code>	Optional dataframe
<code>family</code>	Description of the error distribution and link function, if <code>NULL</code> a linear mixed model is fitted
<code>REML</code>	Logical, if <code>TRUE</code> estimate using REML (provides a consistent estimate of the variance components); if <code>FALSE</code> estimate using ML

Formula `lmer()`

- A random-effects term in `lmer()` is specified by a linear model term and a grouping factor separated by '`|`'; i.e. a random effect is a linear model term conditional on the level of the grouping factor.
- The entire random-effects expression should be enclosed in parentheses since the precedence of '`|`' as an operator is lower than most other operators used in linear model formulas
- For example,
 - ▶ Random intercept,
`lmer(Reaction ~ Days + (1|Subject), data= sleepstudy)`
 - ▶ Random intercept and slope
`lmer(Reaction ~ Days + (Days | Subject), data=sleepstudy)`

Inference

- Functions used for inference and prediction,

<code>summary()</code>	Summarize model results
<code>anova()</code>	Sequential tests of fixed effects and model comparison
<code>VarCorr()</code>	Extract variance components
<code>ranef()</code>	Predict random effects
<code>residuals()</code>	Extract residuals
- See `?"mer-class"` for a complete list of available methods

Example - Mixed Models

```
# Install and load lme4 package
install.packages("lme4")
library(lme4)

# Sleep data
# Average reaction time per day for subjects in a sleep deprivation study
data(sleepstudy)
head(sleepstudy)
str(sleepstudy)

# Trellis plot of data
# Clear that there are different intercepts and slopes for each subject
# type = c("g","p","r"), plots grid lines, points, and regression line
xyplot(Reaction ~ Days | Subject, data=sleepstudy,
       type = c("g","p","r"),
       xlab = "Days of sleep deprivation",
       ylab = "Average reaction time (ms)", aspect = "xy")
```

Example - Mixed Models

```
# Random intercept model
fit1 <- lmer(Reaction ~ Days + (1 | Subject), data=sleepstudy)
summary(fit1)

# Extract information
names(fit1) # S4 class need to look at the slots
getSlots("mer") # Slot names for a mer object returned by lmer()

fit1@deviance # Get deviance

sum.fit1 <- summary(fit1) # Additional information returned by summary
class(sum.fit1)
getSlots("summary.mer")
sum.fit1@coefs # Coefficients

VarCorr(fit1) # Extract variance estimates
```


Example - Mixed Models

```
# Model Diagnostics
y.hat <- fitted(fit1) # Fitted values
int.hat <- ranef(fit1)[[1]][[1]] # Predicted intercepts
res.hat <- residuals(fit1) # Estimated residuals
qqnorm(int.hat, main="Random Intercepts"); qqline(int.hat)
qqnorm(res.hat, main="Residuals"); qqline(res.hat)
plot(y.hat, res.hat, xlab="Fitted Values", ylab="Residuals")
abline(h=0, lty=2)

# Random intercept and slope model
fit2 <- lmer(Reaction ~ Days + (Days | Subject), data=sleepstudy)
summary(fit2)

# Make outcome a binary variable
sleepstudy$react.YesNo <- with(sleepstudy, cut(Reaction,
      breaks=c(min(Reaction), mean(Reaction), max(Reaction)),
      labels=c("Yes", "No")))

# Generalized linear mixed model - binomial outcome
lmer(react.YesNo ~ Days + (Days | Subject),
      data=sleepstudy, family=binomial)
```

Generalized Estimating Equations

- 1 Population-average or marginal model, provides a regression approach for generalized linear models when the responses are not independent (correlated/clustered data)
- 2 Goal is to make inferences about the population, accounting for the within-subject correlation
- 3 The packages `gee` and `geepack` are used for GEE models in R
- 4 The major difference between `gee` and `geepack` is that `geepack` contain an ANOVA method that allows us to compare models and perform Wald tests.

Generalized Estimating Equations

- Basic Syntax for `geeglm()` from the `geepack` package; has a syntax very similar to `glm()`

```
geeglm(formula, family=gaussian, data, id, zcor=NULL, constr,  
        std.err="san.se")
```

<code>formula</code>	Symbolic description of the model to be fitted
<code>family</code>	Description of the error distribution and link function
<code>data</code>	Optional dataframe
<code>id</code>	Vector that identifies the clusters
<code>zcor</code>	Enter a user defined correlation structure
<code>constr</code>	Working correlation structure: "independence", "exchangeable", "ar1", "unstructured", "userdefined"
<code>std.err</code>	Type of standard error to be calculated. Default "san.se" is the robust (sandwich) estimate; use "jack" for approximate jackknife variance estimate

Correlation Structure

- Independence,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Exchangeable,

$$\begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix}$$

- Autoregressive order 1,

$$\begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

- Unstructured,

$$\begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix}$$

- GEE model will give valid results with a misspecified correlation structure when the sandwich variance estimator is used

Inference

- For a `geeglm` object returned by `geeglm()`, the functions `drop1()`, `confint()` and `step()` do not apply; however `anova()` does apply.
- The function `esticon()` in the `doBy` package computes and test linear functions of the regression parameters for `lm`, `glm` and `geeglm` objects
- Basic syntax,

```
esticon(obj, cm, beta0, joint.test=FALSE)
```

<code>obj</code>	Model object
<code>cm</code>	Matrix specifying linear functions of the regression parameters (one linear function per row and one column for each parameter)
<code>beta0</code>	Vector of numbers
<code>jointtest</code>	If TRUE joint Wald test of the hypothesis $L\beta = \beta_0$ is made, default is one test for each row, $(L\beta)_i = \beta_{0,i}$

esticon()

- Let $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p)$ denote the estimated parameters. Also let $k = (k_1, k_2, \dots, k_p)$ denote a vector of constants; one row of the matrix for the `cm` argument. Then $c = k'\beta = k_1\beta_1 + k_2\beta_2 + \dots + k_p\beta_p$
- `esticon()` calculates the linear combinations of the parameter estimates c , the standard error and the confidence interval
- Specify a value for `beta0` to test $H_0 : c = \text{beta0}$
- If `joint.test=TRUE` then all of the linear combinations are tested jointly

Example - GEE

```
# Install and load package geepack
install.packages("geepack")
library(geepack)

# ohio dataset from geepack - Health effect of air pollution
# Children followed for four years, wheeze status recorded annually
data(ohio) # Load the dataset
head(ohio)
str(ohio)

# Response is binary - fit a logistic GEE model
# Treat time (age) as continuous
fit.exch <- geeglm(resp~age+smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
fit.unstr <- geeglm(resp~age+smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "unstructured", std.err="san.se")
summary(fit.exch)
summary(fit.unstr)
```

Example - GEE

```
# Treat time (age) as categorical
fit <- geeglm(resp~factor(age)+smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
summary(fit)

# Test the effect of smoke using anova()
fit1 <- geeglm(resp~factor(age)+smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
fit2 <- geeglm(resp~factor(age), family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
anova(fit1, fit2)

# Individual Wald test and confidence interval for each parameter
est <- esticon(fit, diag(5))

# Odds ratio and confidence intervals
OR.CI <- exp(cbind(est$Estimate, est$Lower, est$Upper))
rownames(OR.CI) <- names(coef(fit))
colnames(OR.CI) <- c("OR", "Lower OR", "Upper OR")
```




Example - GEE

```
# Odds ratio of wheezing for a 9-year old with a mother who smoked
# during the first year of the study compared to an 8-year old with a
# mother who did not smoke during the first year of the study
# That is estimate, [smoke+factor(age)0] - [factor(age)-1]
esticon(fit, c(0,-1,1,0,1))
exp(.Last.value$Estimate)
# 9-year old with mother who smoked is at greater risk of wheezing
# Jointly test effects using esticon()
fit <- geeglm(resp~factor(age)*smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
summary(fit)

L = cbind(matrix(0, nrow=3, ncol=5), diag(3))
esticon(fit, L, joint.test=TRUE)

# Could also use anova()
fit1 <- geeglm(resp~factor(age)*smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
fit2 <- geeglm(resp~factor(age)+smoke, family=binomial(link="logit"),
  data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
anova(fit1, fit2)
```

References

-  Pinheiro, J.C. and Bates, D.M.
Mixed-Effects Models in S and S Plus.
Springer, 2000.
-  Fitzmaurice, G.M., Laird, N.M. and Ware, J.H.
Applied Longitudinal Analysis.
Wiley Interscience, 2004.