Non-linear Models for Longitudinal Data

Jan Serroyen[1,2], **Geert Molenberghs**[1,3], **Geert Verbeke**[3,1], **Marie Davidian**[4]

[1] Methodology and Statistics, University Maastricht,
Peter Debyeplein 1, 6229 HA Maastricht, the Netherlands

[2] I-BioStat, Hasselt University,
Agoralaan 1, 3590 Diepenbeek, Belgium

[3] I-BioStat, Katholieke Universiteit Leuven,
Kapucijnenvoer 35, 3000 Leuven, Belgium

[4] Department of Statistics, North Carolina State University,
Raleigh, North Carolina

A Covariance and Correlation Matrices

$$
\begin{aligned}
\widehat{V}_{\mathsf{r.e.}} &= 0.1 \cdot \boldsymbol{N}\boldsymbol{N}' + 0.006 \cdot I_7 \\
&= \begin{pmatrix}
0.008 & 0.005 & 0.007 & 0.010 & 0.012 & 0.013 & 0.014 \\
0.005 & 0.017 & 0.015 & 0.023 & 0.027 & 0.029 & 0.031 \\
0.007 & 0.015 & 0.027 & 0.031 & 0.037 & 0.039 & 0.042 \\
0.010 & 0.023 & 0.031 & 0.053 & 0.056 & 0.059 & 0.063 \\
0.012 & 0.027 & 0.037 & 0.056 & 0.071 & 0.070 & 0.075 \\
0.013 & 0.029 & 0.039 & 0.059 & 0.070 & 0.081 & 0.080 \\
0.014 & 0.031 & 0.042 & 0.063 & 0.075 & 0.080 & 0.091
\end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\widehat{V}_{\mathsf{marg}} &= 0.042 \cdot \exp\left(-24.72\,|\boldsymbol{t} - \boldsymbol{t}'|\right) + 0.005 \cdot I_7 \\
&= \begin{pmatrix}
0.047 & 0.036 & 0.034 & 0.029 & 0.027 & 0.025 & 0.023 \\
0.036 & 0.047 & 0.039 & 0.034 & 0.031 & 0.029 & 0.027 \\
0.034 & 0.039 & 0.047 & 0.037 & 0.033 & 0.032 & 0.029 \\
0.029 & 0.034 & 0.037 & 0.047 & 0.038 & 0.036 & 0.033 \\
0.027 & 0.031 & 0.033 & 0.038 & 0.047 & 0.040 & 0.036 \\
0.025 & 0.029 & 0.032 & 0.036 & 0.040 & 0.047 & 0.039 \\
0.023 & 0.027 & 0.029 & 0.033 & 0.036 & 0.039 & 0.047
\end{pmatrix}
\end{aligned}
$$

$$V_{\mathsf{obs}} = \begin{pmatrix} 0.000 & 0.001 & 0.002 & 0.003 & 0.004 & 0.004 & 0.004 \\ 0.001 & 0.007 & 0.013 & 0.016 & 0.017 & 0.018 & 0.017 \\ 0.002 & 0.013 & 0.030 & 0.043 & 0.046 & 0.050 & 0.051 \\ 0.003 & 0.016 & 0.043 & 0.067 & 0.075 & 0.083 & 0.084 \\ 0.004 & 0.017 & 0.046 & 0.075 & 0.085 & 0.096 & 0.097 \\ 0.004 & 0.018 & 0.050 & 0.083 & 0.096 & 0.108 & 0.109 \\ 0.004 & 0.017 & 0.051 & 0.084 & 0.097 & 0.109 & 0.111 \end{pmatrix}$$

$$R_{\mathsf{r.e.}} = \begin{pmatrix} 1.000 & 0.862 & 0.802 & 0.699 & 0.637 & 0.602 & 0.553 \\ 0.862 & 1.000 & 0.930 & 0.810 & 0.739 & 0.698 & 0.641 \\ 0.802 & 0.930 & 1.000 & 0.871 & 0.795 & 0.751 & 0.690 \\ 0.699 & 0.810 & 0.871 & 1.000 & 0.912 & 0.862 & 0.792 \\ 0.637 & 0.739 & 0.795 & 0.912 & 1.000 & 0.945 & 0.868 \\ 0.602 & 0.698 & 0.751 & 0.862 & 0.945 & 1.000 & 0.919 \\ 0.553 & 0.641 & 0.690 & 0.792 & 0.868 & 0.919 & 1.000 \end{pmatrix}$$

$$R_{\mathsf{marg}} = \begin{pmatrix} 1.000 & 0.771 & 0.717 & 0.624 & 0.570 & 0.538 & 0.494 \\ 0.771 & 1.000 & 0.831 & 0.724 & 0.661 & 0.624 & 0.573 \\ 0.717 & 0.831 & 1.000 & 0.779 & 0.710 & 0.671 & 0.616 \\ 0.624 & 0.724 & 0.779 & 1.000 & 0.815 & 0.770 & 0.707 \\ 0.570 & 0.661 & 0.710 & 0.815 & 1.000 & 0.844 & 0.775 \\ 0.538 & 0.624 & 0.671 & 0.770 & 0.844 & 1.000 & 0.821 \\ 0.494 & 0.573 & 0.616 & 0.707 & 0.775 & 0.821 & 1.000 \end{pmatrix}$$

$$R_{\mathsf{obs}} = \begin{pmatrix} 1.000 & 0.909 & 0.933 & 0.893 & 0.883 & 0.861 & 0.839 \\ 0.909 & 1.000 & 0.902 & 0.752 & 0.702 & 0.656 & 0.636 \\ 0.933 & 0.902 & 1.000 & 0.958 & 0.922 & 0.887 & 0.885 \\ 0.893 & 0.752 & 0.958 & 1.000 & 0.988 & 0.972 & 0.974 \\ 0.883 & 0.702 & 0.922 & 0.988 & 1.000 & 0.996 & 0.995 \\ 0.861 & 0.656 & 0.887 & 0.972 & 0.996 & 1.000 & 0.998 \\ 0.839 & 0.636 & 0.885 & 0.974 & 0.995 & 0.998 & 1.000 \end{pmatrix}$$

$$R_{\mathsf{marg,2}} = \begin{pmatrix} 1.000 & 0.558 & 0.419 & 0.244 & 0.170 & 0.136 & 0.097 \\ 0.558 & 1.000 & 0.751 & 0.437 & 0.304 & 0.243 & 0.174 \\ 0.419 & 0.751 & 1.000 & 0.582 & 0.405 & 0.324 & 0.232 \\ 0.244 & 0.437 & 0.582 & 1.000 & 0.697 & 0.557 & 0.398 \\ 0.170 & 0.304 & 0.405 & 0.697 & 1.000 & 0.799 & 0.572 \\ 0.136 & 0.243 & 0.324 & 0.557 & 0.799 & 1.000 & 0.716 \\ 0.097 & 0.174 & 0.232 & 0.398 & 0.572 & 0.716 & 1.000 \end{pmatrix}$$

$$\mathsf{diag}(A_3) = (0.000, 0.011, 0.039, 0.156, 0.179, 0.142, 0.065)'$$

$$
R_{\mathsf{marg},3} = \begin{pmatrix} 1.000 & 0.737 & 0.635 & 0.478 & 0.396 & 0.352 & 0.296 \\ 0.737 & 1.000 & 0.861 & 0.649 & 0.537 & 0.477 & 0.401 \\ 0.635 & 0.861 & 1.000 & 0.753 & 0.624 & 0.555 & 0.466 \\ 0.478 & 0.649 & 0.753 & 1.000 & 0.828 & 0.736 & 0.618 \\ 0.396 & 0.537 & 0.624 & 0.828 & 1.000 & 0.889 & 0.747 \\ 0.352 & 0.477 & 0.555 & 0.736 & 0.889 & 1.000 & 0.840 \\ 0.296 & 0.401 & 0.466 & 0.618 & 0.747 & 0.840 & 1.000 \end{pmatrix}
$$

B SAS Programs for the Random-effects Models

- Model (3):

```
proc nlmixed data=mydata qpoints=20;
   num = beta1+b;
   ex = exp(-(time-beta2)/beta3);
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   random b ~ normal(0,d11) subject=tree;
   predict num/den;
   run;
```

- Model(5):

```
proc nlmixed data=mydata qpoints=20;
   num = beta1+b1;
   ex = exp(-(time-beta2-b2)/beta3);
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   random b1 b2 ~ normal([0,0],[d11, d12, d22]) subject=tree;
   predict num/den;
run;
```

- Model (5) with treatment effects:

```
proc nlmixed data=mydata qpoints=20;
   num = beta10+beta11*treat+b1;
   ex = exp(-(time-beta20+beta21*treat-b2)/(beta30+beta31*treat));
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   random b1 b2 ~ normal([0,0],[d11, d12, d22]) subject=tree;
   predict num/den;
run;
```

- Model (6):

```
proc nlmixed data=mydata qpoints=20;
   num = beta1+b1;
   ex = exp(-(time**(1+b2)-beta2)/beta3);
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   random b1 b2 ~ normal([0,0],[d11, d12, d22]) subject=tree;
   predict num/den;
run;
```

- Model (17):

```
proc nlmixed data=mydata qpoints=20;
   parms phim=0.64 phimdiff = 0
         eta=1.88 etadiff=0
         tau=3.68 taudiff=0
         gamma=-0.01 gdiff=0
         d11=0.01 sigma2=0.01 d22=0.01 d33=0.01;
   num = (phim + phimdiff * treat + vm)
         * (time ** (eta + etadiff * treat + n));
   den = ((tau + k + taudiff * treat) ** (eta + etadiff * treat + n))
         + (time ** (eta + etadiff * treat + n));
   mean = num/den + gamma + gdiff * treat;
   model y ~ normal(mean,sigma2);
   random vm k n ~ normal([0, 0, 0],[d11,d12,d22,d13,d23,d33])
          subject=bird;
run;
```

C R Code for the Marginal Models

The marginal models ca be fitted in R. We present annotated code.

```
#########################
#   DATA MANIPULATION   #
#########################
yi <- as.data.frame(matrix(NA,nrow=length(Orange[,1]),ncol=3))
names(yi) <- c("Tree","age","circumference")
rescale <- 100
yi[,3] <- Orange$circ /rescale
yi[,2] <- Orange$age / rescale
yi[,1] <- Orange$Tree
```

```
N <- length(unique(yi[,1]))
ni <- length(unique(yi[,2]))
zi <- matrix(1,ncol=ni,nrow=ni)
ti <- matrix(yi[,2][yi[,1]==1],ncol=ni,nrow=ni)
tij <- ti[,1]

##################################################################
#  Gradient function                                             #
#this function computes central difference approximations for 'f'#
##################################################################
cd <- function (x, f, ..., eps = 1e-04) {
    n <- length(x)
    res <- numeric(n)
    ex <- pmax(abs(x), 1)
    for (i in 1:n) {
        x1 <- x2 <- x
        x1[i] <- x[i] + eps * ex[i]
        x2[i] <- x[i] - eps * ex[i]
        diff.f <- c(f(x1, ...) - f(x2, ...))
        diff.x <- 2 * max(abs(c(x1[i] - x[i], x2[i] - x[i])))
        res[i] <- diff.f / diff.x
    }
    res
 }

#############################################
# Model (9): EXPONENTIAL serial correlation #
#############################################
loglik1 <- function(theta) {
  beta <- theta[1:3]
  sigma <- theta[4]
  tau <- theta[5]
  phi <- theta[6]
  ll <- 0
  covi <- tau^2*exp(-abs(ti-t(ti))/phi)
  Vi <- covi + sigma^2*diag(ni)
  Wi <- solve(Vi)
  Mu <- beta[1]/(1+exp(-(tij-beta[2])/beta[3]))

  # Marginal Negative LogLikelihood function
   lli <- function(x) 0.5*ni*log(2*pi)
          + 0.5*log(det(Vi)) + 0.5*(diag(t(x-Mu)%*%Wi%*%(x-Mu)))
   ll <- sum(tapply(yi[,3],yi[,1],FUN=lli))
   ll
```

```
}

# Calculate -Loglikelihood at starting values:
Start1 <- c(1.75,7,3, 0.1,0.2,7)
attr(Start1,"names") <- c("Asym","xmid","scal","sigma","tau","phi")
loglik1(Start1)

grad.loglik1 <- function(thetas) cd(thetas, loglik1)

# OPTIM: minimize neg. loglikelihood:
fit1 <- optim(Start1,loglik1,grad.loglik1,hessian=T,
              method="BFGS",control=list(trace=1,maxit=500))

se.fit1 <- sqrt(diag(solve(fit1$hess)))
cbind(fit1$par,se.fit1)

# Sigma^2:
  fit1$par[4]^2
# Delta method to obtain s.e. of sigma:
  sqrt(4*fit1$par[4]^2*(se.fit1[4]^2))

# Tau^2:
  fit1$par[5]^2
# Delta method to obtain s.e. of tau:
  sqrt(4*fit1$par[5]^2*(se.fit1[5]^2))

###############################################################
# Model (11): Exponential serial corr. + Linear Variance fct. #
###############################################################
loglik2 <- function(theta) {
  beta <- theta[1:3]
  sigma <- theta[4:5]
  phi <- theta[6]
  ll <- 0
  Amat <- diag(exp(sigma[1]+sigma[2]*tij))
  covi <- exp(-abs(ti-t(ti))/phi)
  Vi <- sqrt(Amat)%*% covi %*% sqrt(Amat)
  Wi <- solve(Vi)
  Mu <- beta[1]/(1+exp(-(tij-beta[2])/beta[3]))

  # Marginal Negative LogLikelihood function
   lli <- function(x) 0.5*ni*log(2*pi)
          + 0.5*log(det(Vi)) + 0.5*(diag(t(x-Mu)%*%Wi%*%(x-Mu)))
   ll <- sum(tapply(yi[,3],yi[,1],FUN=lli))
```

```
   ll
}

# Calculate -Loglikelihood at starting values:
Start2 <- c(1.75,7,3, -4,0.1,6)
attr(Start2,"names") <- c("Asym","xmid","scal","sigma0","sigma1","phi")
loglik2(Start2)

grad.loglik2 <- function(thetas) cd(thetas, loglik2)

# OPTIM: minimize neg. loglikelihood:
fit2 <- optim(Start2,loglik2,grad.loglik2,hessian=T,method="Nelder-Mead",
              control=list(trace=1,maxit=2000))

se.fit2 <- sqrt(diag(solve(fit2$hess)))
cbind(fit2$par,se.fit2)

################################################################
# Model (12): Exponential serial corr. + Quadr. Variance fct. #
################################################################
loglik3 <- function(theta) {
  beta <- theta[1:3]
  sigma <- theta[4:6]
  phi <- theta[7]
  ll <- 0
  Amat <- diag(exp(sigma[1]+sigma[2]*tij+sigma[3]*(tij^2)))
  covi <- exp(-abs(ti-t(ti))/phi)

  Vi <- sqrt(Amat)%*% covi %*% sqrt(Amat)
  Wi <- solve(Vi)
  Mu <- beta[1]/(1+exp(-(tij-beta[2])/beta[3]))

  # Marginal LogLikelihood function
   lli <- function(x) -(0.5*ni*log(2*pi)
         + 0.5*log(det(Vi)) + 0.5*(diag(t(x-Mu)%*%Wi%*%(x-Mu))))
   ll <- sum(tapply(yi[,3],yi[,1],FUN=lli))
   ll
}

# Calculate -Loglikelihood at starting values:
Start3 <- c(1.924, 7.250, 3.499, -5.090, 0.168, 0.000, 6.278)
attr(Start3,"names") <- c("Asym","xmid","scal",
                          "sigma0","sigma1","sigma2","phi")
loglik3(Start3)
```

```
# NO CONVERGENCE with optim() and nlminb()  !!!

# Newton-Raphson optimization
#-----------------------------
param <- Start3
tt <- NULL

for(iter in 0:250){
cat("iter:", iter, "\t", "loglik:", tt, "\n")
h=0.00001;
gradnum=rep(0,length(param))
hessnum=matrix(0,nrow=length(param),ncol=length(param))
for(i in 1:length(param)){
   parami=param;
   parami[i]=parami[i]+h;
   tt=loglik3(param);
   gradnum[i]=(loglik3(parami)-loglik3(param))/h;
   for(j in 1:length(param)){
      paramj=param;
      paramj[j]=paramj[j]+h;
      paramij=param;
      paramij[i]=paramij[i]+h;
      paramij[j]=paramij[j]+h;
      hessnum[i,j]=(loglik3(paramij)-loglik3(parami)
                    -loglik3(paramj)+loglik3(param))/h/h;
      hessnum[j,i]=hessnum[i,j];
      }
   }
param=param-solve(hessnum)%*%gradnum/20;
}

# Estimates and s.e.
rownames(param) <- c("Asym","xmid","scal","sigma0","sigma1","sigma2","phi")
cbind(param,sqrt(diag(solve(-hessnum))))
```

D GAUSS Programs for the Marginal Models

The marginal models can also be fitted in GAUSS. We present the corresponding likelihood functions, that are then optimized using a generic optimizer.

- Model (9):

```
proc loglik(param);
```

```
local beta1,beta2,beta3,sigma2,phi,tau2,ll,lli,i,ni,yi,nni,mui,vi,u;
beta1=param[1];
beta2=param[2];
beta3=param[3];
tau2=param[4];
phi=param[5];
sigma2=param[6];

ll=0;
u=abs(timevec-timevec');
vi=sigma2*eye(7)+tau2*exp(-u/phi);
nni=1+exp(-(timevec-beta2)/beta3);
mui=beta1./nni;

i=1;
do while i<=5;
   yi=tree[.,i];
   lli=-0.5*7*ln(2*pi)-0.5*ln(det(vi))-0.5*(yi-mui)'inv(vi)*(yi-mui);
   ll=ll+lli;
i=i+1;
endo;

retp(ll);
endp;
```

- Model (11):

```
proc loglik(param);
local beta1,beta2,beta3,phi,ll,lli,i,ni,yi,
      nni,mui,vi,u,varp1,varp2,amat,varfunc;
beta1=param[1];
beta2=param[2];
beta3=param[3];
varp1=param[4];
varp2=param[5];
phi=param[6];

ll=0;

u=abs(timevec-timevec');
varfunc=exp(varp1+timevec*varp2);

amat=diagrv(eye(7),varfunc);
rmat=exp(-u/phi);
```

```
vi=(amat^(0.5))*rmat*(amat^(0.5));

vmat=vi;
nni=1+exp(-(timevec-beta2)/beta3);
mui=beta1./nni;

i=1;
do while i<=5;
   yi=tree[.,i];
   lli=-0.5*7*ln(2*pi)-0.5*ln(det(vi))-0.5*(yi-mui)'inv(vi)*(yi-mui);
   ll=ll+lli;
i=i+1;
endo;

retp(ll);
endp;
```

- Model (12):

```
proc loglik(param);
local beta1,beta2,beta3,phi,ll,lli,i,ni,yi,nni,
      mui,vi,u,varp0,varp1,varp2,amat,varfunc;
beta1=param[1];
beta2=param[2];
beta3=param[3];
varp0=param[4];
varp1=param[5];
varp2=param[6];
phi=param[7];

ll=0;

u=abs(timevec-timevec');
varfunc=exp(varp0+varp1*timevec+varp2*(timevec^2));

amat=diagrv(eye(7),varfunc);
rmat=exp(-u/phi);
vi=(amat^(0.5))*rmat*(amat^(0.5));
vmat=vi;
nni=1+exp(-(timevec-beta2)/beta3);
mui=beta1./nni;

i=1;
```

```
do while i<=5;
   yi=tree[.,i];
   lli=-0.5*7*ln(2*pi)-0.5*ln(det(vi))-0.5*(yi-mui)'inv(vi)*(yi-mui);
   ll=ll+lli;
i=i+1;
endo;

retp(ll);
endp;
```

E SAS Programs for the Transition Models

- Model (14):

```
proc nlmixed data=mydata;
   num = beta1+gamma*yprev;
   ex = exp(-(time-beta2)/beta3);
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   predict num/den;
   run;
```

- Model(16):

```
proc nlmixed data=mydata;
   num = beta1;
   ex = exp(-(time-beta2)/beta3+gamma*yprev);
   den = 1 + ex;
   model y ~ normal(num/den,sigma2);
   predict num/den;
   run;
```