Package 'nmw'

March 14, 2017

Version 0.1.1

Title Understanding Nonlinear Mixed Effects Modeling for Population Pharmacokinetics

Description This shows how NONMEM(R) <http://www.iconplc.com/innovation/nonmem/> software works. NONMEM's classical estimation methods like 'First Order(FO) approximation', 'First Order Conditional Estimation(FOCE)', and 'Laplacian approximation' are explained.

Depends R (>= 3.0.0)

ByteCompile yes

License GPL-3

Copyright 2017, Kyun-Seop Bae

Author Kyun-Seop Bae

Maintainer Kyun-Seop Bae <k@acr.kr>

URL http://optimizer.r-forge.r-project.org/

NeedsCompilation no

Repository

Date 2017-03-14

R topics documented:

nmw-package *Understanding Nonlinear Mixed Effects Modeling for Population Pharmacokinetics*

Description

This shows how NONMEM(R) <http://www.iconplc.com/innovation/nonmem/> software works.

Details

This package explains 'First Order(FO) approximation' method, 'First Order Conditional Estimation(FOCE)' method, and 'Laplacian(LAPL)' method of NONMEM software.

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

NONMEM Users guide

Examples

```
DataAll = Theoph
colnames(DataAll) = c("ID", "BWT", "DOSE", "TIME", "DV")
DataAll[,"ID"] = as.numeric(as.character(DataAll[,"ID"]))

nTheta = 3
nEta = 3
nEps = 2

THETAinit = c(2, 50, 0.1) # Initial estimate
OMinit = matrix(c(0.2, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.1, 0.2), nrow=nEta, ncol=nEta)
OMinit
SGinit = matrix(c(0.1, 0, 0, 0.1), nrow=nEps, ncol=nEps)
SGinit

LB = rep(0, nTheta) # Lower bound
UB = rep(1000000, nTheta) # Upper bound

PRED = function(THETA, ETA, DATAi) # Prediction function
{
  DOSE = 320
  TIME = DATAi[,"TIME"]

  KA = THETA[1]*exp(ETA[1])
  V  = THETA[2]*exp(ETA[2])
  K  = THETA[3]*exp(ETA[3])

  TERM1 = DOSE/V * KA/(KA - K)
  TERM2 = exp(-K*TIME)
  TERM3 = exp(-KA*TIME)

  F  = TERM1 * (TERM2 - TERM3)
  G1 = -F*K/(KA - K) + KA*TIME*TERM1*TERM3
  G2 = -F
  G3 = (F/(KA - K) - TIME*TERM1*TERM2) * K
  H1 = F
  H2 = 1

  if (METHOD=="LAPL") {
    D11 = DOSE*(KA*V**-1.0*(-1.0*KA*(-2.0*KA*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-2.0)+
          -1.0*KA*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)+
          KA*TIME*(-1.0*KA*TIME*TERM3*(-1.0*K+KA)**-1.0+
          -1.0*KA*TERM3*(-1.0*K+KA)**-2.0)+
          KA*TIME*TERM3*(-1.0*K+KA)**-1.0)+
          KA*V**-1.0*(-1.0*K+KA)**-1.0*(-1.0*TERM3+TERM2)+
          2.0*KA*V**-1.0*(-1.0*KA*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-1.0))
    D21 = -G1
    D22 = F
    D31 = DOSE*(KA*V**-1.0*(KA*K*TIME*TERM2*(-1.0*K+KA)**-2.0+
          K*(-2.0*KA*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-2.0))+
          KA*V**-1.0*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)))
    D32 = -G3
    D33 = DOSE*KA*V**-1.0*(-1.0*K*TIME*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*TERM2*(-1.0*K+KA)**-2.0)+-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-2.0+
          2.0*K*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2))+
          K*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2))
  } else {
    D11 = 0
    D21 = 0
    D22 = 0
    D31 = 0
    D32 = 0
    D33 = 0
  }

  return(cbind(F, G1, G2, G3, H1, H2, D11, D21, D22, D31, D32, D33))
}
#########
METHOD = "ZERO" # PRED function refers this.
InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)
(EstRes = EstStep())            # It will take about 3 secs.
(CovRes = CovStep())            # It will take about 1 sec.
PostHocEta() # FinalPara from EstStep()

#########
#METHOD = "COND" # PRED function refers this.
#InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
#         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)
#(EstRes = EstStep())            # It will take about 4 mins.
#(CovRes = CovStep())            # It will take about 40 secs.
#get("EBE", envir=e)

######### "LAPL" usually fails due to numerical difficulties.
#METHOD = "LAPL" # PRED function refers this.
#THETAinit = c(4, 50, 0.2) # It is changed for better convergence.
#InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
#         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)
#(EstRes = EstStep())            # It will take about 3 mins. Succeeded with R-3.3.3 x64
#(CovRes = CovStep())            # It will take about 1 min.
#get("EBE", envir=e)
```

CovStep *Covariance Step*

Description

It calculates standard erros and various variance matrices with the `e$FinalPara` after estimation step.

Usage

```
CovStep()
```

Details

Because `EstStep` uses nonlinear optimization, covariance step is separated from estimation step. It caculcates variance-covariance matrix of estimates on the original scale.

Value

`Time` consumed time

`Standard Error` standard error of the estimates in the order of theta, omega, and sigma

`Covariance Matrix of Estimates`
covariance matrix of estimates in the order of theta, omega, and sigma. This is inverse(R) x S x inverse(R) by default.

`Correlation Matrix of Estimates`
correlation matrix of estimates in the order of theta, omega, and sigma

`Inverse Covariance Matrix of Estimates`
inverse covariance matrix of estimates in the order of theta, omega, and sigma

`Eigen Values` eigen values of covariance matrix

`R Matrix` R matrix of NONMEM, second derivative of log likelihood function with respect to esimation parameters

`S Matrix` S matrix of NONMEM, sum of individual cross-product of first derivative of log likelihood function with respect to esimation parameters

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

NONMEM Users Guide

See Also

`EstStep`, `InitStep`

Examples

```
# Only after InitStep and EstStep
#CovStep()
```

EstStep *Estimation Step*

Description

This estimates upon the conditions with `InitStep`.

Usage

```
EstStep()
```

Details

It does not have arguments. All necessary arguments are stored in the `e` environment. It assumes "INTERACTION" between eta and epsilon for `"COND"` and `"LAPL"` options. The output is basically same with NONMEM output.

Value

`Initial OFV`	initial value of objective function
`Time`	time consumed for this step
`Optim`	the raw output from `optim` function
`Final Estimates`	final estimates in the original scale

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

NONMEM Users Guide

See Also

`InitStep`

Examples

```
# Only After InitStep
#EstStep()
```

InitStep *Initialization Step*

Description

It recevies parameters for the estimation and stores them into e environment.

Usage

```
InitStep(DataAll, THETAinit, OMinit, SGinit, nTheta, LB = rep(0, nTheta),
         UB = rep(0, nTheta), Pred, METHOD = METHOD)
```

Arguments

DataAll	Data for all subjects. It should contain columns which Pred function uses.
THETAinit	Theta initial values
OMinit	Omega matrix initial values
SGinit	Sigma matrix initial values
nTheta	Number of thetas
LB	Lower bounds for theta vector
UB	Upper bounds for theta vector
Pred	Prediction function name
METHOD	one of the estimation methods "ZERO", "COND", "LAPL"

Details

Prediction function should return not only prediction values(F or IPRED) but also G (first derivative with respect to etas) and H (first derivative of Y with respect to epsilon). For the "LAPL", prediction function should return second derivative with respect to eta also. All objective functions assume NONMEM "INTERACTION" option for "COND" and "LAPL" option. Omega matrix should be full block one. Sigma matrix should be diagonal one.

Value

This does not return values, but stores necessary values into the environment e.

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

NONMEM Users Guide

Examples

```
DataAll = Theoph
colnames(DataAll) = c(”ID”, ”BWT”, ”DOSE”, ”TIME”, ”DV”)
DataAll[,”ID”] = as.numeric(as.character(DataAll[,”ID”]))

nTheta = 3
nEta = 3
nEps = 2

THETAinit = c(2, 50, 0.1) # Initial estimate
OMinit = matrix(c(0.2, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.1, 0.2), nrow=nEta, ncol=nEta)
OMinit
SGinit = matrix(c(0.1, 0, 0, 0.1), nrow=nEps, ncol=nEps)
SGinit

LB = rep(0, nTheta) # Lower bound
UB = rep(1000000, nTheta) # Upper bound

PRED = function(THETA, ETA, DATAi) # Prediction function
{
  DOSE = 320
  TIME = DATAi[,”TIME”]

  KA = THETA[1]*exp(ETA[1])
  V  = THETA[2]*exp(ETA[2])
  K  = THETA[3]*exp(ETA[3])

  TERM1 = DOSE/V * KA/(KA - K)
  TERM2 = exp(-K*TIME)
  TERM3 = exp(-KA*TIME)

  F  = TERM1 * (TERM2 - TERM3)
  G1 = -F*K/(KA - K) + KA*TIME*TERM1*TERM3
  G2 = -F
  G3 = (F/(KA - K) - TIME*TERM1*TERM2) * K
  H1 = F
  H2 = 1

  if (METHOD==”LAPL”) {
    D11 = DOSE*(KA*V**-1.0*(-1.0*KA*(-2.0*KA*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-2.0)+
          -1.0*KA*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)+
          KA*TIME*(-1.0*KA*TIME*TERM3*(-1.0*K+KA)**-1.0+
          -1.0*KA*TERM3*(-1.0*K+KA)**-2.0)+
          KA*TIME*TERM3*(-1.0*K+KA)**-1.0)+
          KA*V**-1.0*(-1.0*K+KA)**-1.0*(-1.0*TERM3+TERM2)+
          2.0*KA*V**-1.0*(-1.0*KA*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-1.0))
    D21 = -G1
    D22 = F
    D31 = DOSE*(KA*V**-1.0*(KA*K*TIME*TERM2*(-1.0*K+KA)**-2.0+
          K*(-2.0*KA*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2)+
          KA*TIME*TERM3*(-1.0*K+KA)**-2.0))+
          KA*V**-1.0*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2)))
    D32 = -G3
    D33 = DOSE*KA*V**-1.0*(-1.0*K*TIME*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*TERM2*(-1.0*K+KA)**-2.0)+-1.0*K*TIME*TERM2*(-1.0*K+KA)**-1.0+
          K*(-1.0*K*TIME*TERM2*(-1.0*K+KA)**-2.0+
          2.0*K*(-1.0*K+KA)**-3.0*(-1.0*TERM3+TERM2))+
          K*(-1.0*K+KA)**-2.0*(-1.0*TERM3+TERM2))
  } else {
    D11 = 0
    D21 = 0
    D22 = 0
    D31 = 0
    D32 = 0
    D33 = 0
  }

  return(cbind(F, G1, G2, G3, H1, H2, D11, D21, D22, D31, D32, D33))
}
#########
METHOD = "ZERO" # PRED function refers this.
InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)

######### OR
METHOD = "COND" # PRED function refers this.
InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)

######### OR
METHOD = "LAPL" # PRED function refers this.
THETAinit = c(4, 50, 0.2) # It is changed for better convergence for Theoph example.
InitStep(DataAll, THETAinit=THETAinit, OMinit=OMinit, SGinit=SGinit, nTheta=nTheta,
         LB=LB, UB=UB, METHOD=METHOD, Pred=PRED)
```

Index