

Introduction to PSM

with focus on linear models

June 6, 2008

DTU Informatik
Institut for Informatik og Matematisk Modellering

PSM: Population Stochastic Modelling

- Linear and non-linear Mixed-effects models using stochastic differential equations
 - Multivariate response
 - Multivariate input
 - Missing observations
 - Dosage regimen for PK/PD modelling
- Parameter estimation in mixed-effects model using maximum likelihood
 - It is possible to use a simple pooled likelihood function by using a model without random effects

Overview of PSM

- PSM is written for R
 - Partly based on FORTRAN-code
- Main functions
 - **PSM.simulate**
Simulate data for multiple individuals.
 - **PSM.estimate**
Estimate population parameters for any linear or non-linear model.
 - **PSM.smooth**
Optimal estimates of model states based on estimated parameters.

Linear case:

A time-invariant model

DATA: y_{ij} , $i = 1 \dots N$, $j = 1 \dots n_i$

- 1st stage model

$$dX_t = [\mathbf{A}(\varphi_i)X_t + \mathbf{B}(\varphi_i)U_t]dt + \boldsymbol{\sigma}_\omega(\varphi_i)d\omega_t$$

$$Y_{ij} = \mathbf{C}(\varphi_i)X_{ij} + \mathbf{D}(\varphi_i)U_{ij} + e_k \quad , e_k \in N(0, S)$$

- 2nd stage model

$$\varphi_i = h(\theta, \eta_i, Z_i) \quad , \eta_i \in N(0, \Omega), \quad Z_i \text{ are covariates}$$

Input arguments

Main functions

- `PSM.simulate(Model, Data, THETA, deltaTime)`
- `PSM.estimate(Model, Data, Par, CI = FALSE)`
- `PSM.smooth(Model, Data, THETA, subsample = 0)`
- Model, Data and Par are
 - R-type: **list**

Input argument: **Model** (linear)

Model is a list containing:

- **Matrices**=function(phi): [matA, matB, matC, matD]
- **X0**=function(Time, phi, U): **col.matrix**, Initial state
- **SIG**=function(phi): **matrix**, σ_{ω}
- **S**=function(phi): **matrix**, S
- **h**=function(eta,theta,covar): **list**, phi φ_i
- **ModelPar**=function(THETA Θ): [theta θ , OMEGA Ω]

[...] denotes a list.

Input argument: **Data**

Data is an unnamed list of lists $[[1],[2],[3],\dots,[N]]$

- Each element is a list containing
 - **Time**, **vector** n_i elements
 - **Y**, **matrix**
Observations by column, may contain NA.
 - **U**, **matrix** Input to system at sample times
 - **covar**, **user defined**, input argument to h
 - **Dose**: [Time, State, Amount]

Input argument: **THETA**

- **THETA** Θ – (named) vector
 - is the parameters in θ, Ω to be *ML-estimated*

$$\max_{\Theta} L(\Theta)$$

Example:

```
THETA = c(k=.3, om2=0.04)
ModelPar = function(THETA) {
  list(
    theta = c(k=THETA['k'], S=5.0, A0=10.0,...),
    OMEGA = diag(c(.3,THETA['om2']) )
  )
}
```

Input argument: **Par**

Par is a list containing:

- **Init**, (named) vector THETA Θ - Initial estimates
- **LB**, **UB**, vectors - Bounds for search

Eks.

```
p$LB = c(k=0.004, om2 = 0.75 )  
p$Init = c(k=0.005, om2 = 100.00)  
p$UB = c(k=1.600, om2 = 300.0 )
```

Note: ModelPar: $\Theta \rightarrow (\theta, \Omega)$

Output from PSM.simulate

```
MyData <- PSM.simulate(Model, Data, THETA, deltaT)
```

- Returns a **Data-list** `[[1],[2],[3],...,[length(Data)]]`
 - **X** - Simulated states sampled at time points for measurements
 - **Y** - Simulated measurements
 - **Time** - Time points for measurements
 - **U** - Input vector used in simulation
 - **Dose** - Dose specification used in simulation
 - **eta** - The random effects used in the simulation
 - **longX** - Entire outcome of simulated states
 - **longTime** - Time points for longX.

Output from PSM.estimate

```
MyFit <- PSM.estimate(Model, Data, Par, CI = FALSE)
```

- Returns a list containing
 - **NegLogL** - Value of the negative log-likelihood function at optimum.
 - **THETA** - Population parameters at optimum
 - **CI** - 95% confidence interval for the estimated parameters
 - **SD** - standard error of estimated parameters
 - **COR** - correlation matrix for estimated parameters
 - **sec** - Time for the estimation in seconds
 - **opt** - Raw output from the optimizer

Output from PSM.smooth

```
out <- PSM.smooth(Model, Data, THETA, subsamp=0)
```

- Returns an unnamed list [[1],[2],[3],...,[N]]
 - **TIME** – (subsampled) time-vector
 - **Xs, Ps** - Smoothed state and state co-variance estimate
 - **Ys** - Response based on smoothed state
 - **Xf, Pf** - Filtered state and state co-variance estimate
 - **Xp, Pp** - Predicted state and state co-variance estimate
 - **Yp, R** - Predicted observations and observation variances
 - **Eta** - Estimated eta
 - **negLogL** - Value of the negative log-likelihood .

Example

- Simulate from the model

$$dA_t = [-k]A_t dt$$

$$Y_{ij} = [1/V]A_{ij} + e_k \quad , \quad e_k \in N(0, S)$$

- $N = 1$, $t = c(0, 1, 2, 3, 5, 7, 10, 13, 16, 20, 25)$, $n_1 = 11$,
 $S = (5\text{mg/L})^2$, $k = 0.08\text{min}^{-1}$, $A_0 = 1000\text{mg}$, $V=10\text{L}$
- Prepare to
 - estimate S and k
 - A_0 (dose) and V fixed

Example Model-list

```
• mod1 <- list()

mod1$ModelPar = function(THETA) {
  A0 <- 1000
  V <- 10
  list(theta = list(k=THETA['k'],
    S = THETA['S'], V = V, A0 = A0), OMEGA=NULL)
}

mod1$Matrices = function(phi) {
  list(
    matA=matrix(c(-1*phi$k),ncol=1,nrow=1),
    matC=matrix(c(1/phi$V),ncol=1,nrow=1)
  )
}
```

Example

Model-list - continued

```
• mod1$X0 = function(Time,phi,U) {  
  A0 <- phi$A0  
  matrix(c(A0),ncol=1,nrow=1)  
}  
mod1$S = function(phi) {  
  matrix(phi$S,ncol=1,nrow=1)  
}  
mod1$SIG = function(phi) {  
  matrix(0,ncol=1,nrow=1)  
}  
  
> names(mod1)  
[1] "ModelPar" "Matrices" "X0" "S" "SIG"
```

Example Simulation

- `TimeVec <- c(0,1,2,3,5,7,10,13,16,20,25)`

```
tmpData = list( list(Time = TimeVec) )    #length(tmpData)=1  
MyTHETA = c(k = 0.08, S = 25)           #named vector!
```

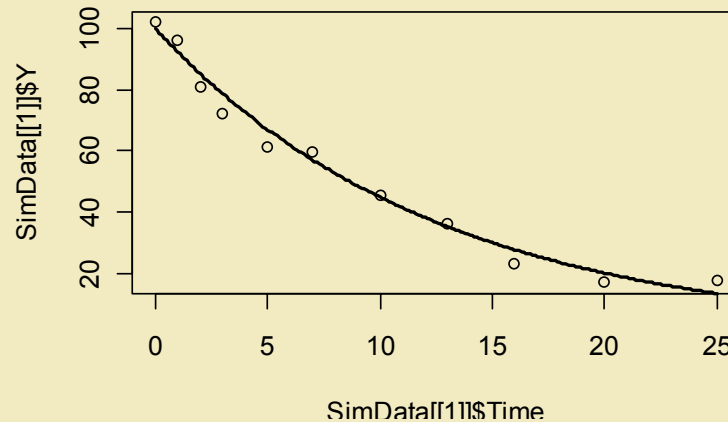
```
SimData <- PSM.simulate(mod1,PrepData,  
                        MyTHETA,deltaTime = .1)
```

```
> names(SimData[[1]])  
[1] "X"  "Y"  "Time" "U"  "eta" "longX" "longTime"
```

Example

Plot simulation

- `plot(SimData[[1]]$Time, SimData[[1]]$Y )`
`lines(SimData[[1]]$longTime,`
`SimData[[1]]$longX[1,]/10 ) % V = 10`



Example

Estimate and plot

- `par <- list(LB = c(k=.0005, S=.1),
 Init = c(k=.005, S=10),
 UB = c(k=.5, S=100))`
- `fit <- PSM.estimate(mod1, SimData, par, CI=TRUE)`

```
>fit[1:3]
  $NegLogL
[1] 33.54679

  $THETA
           k           S
0.07528455 26.08504477

  $CI
           k           S
Lower CI95 0.06732354 11.41457
MLE         0.07528455 26.08504
Upper CI95 0.08419406 59.40889
```

Example

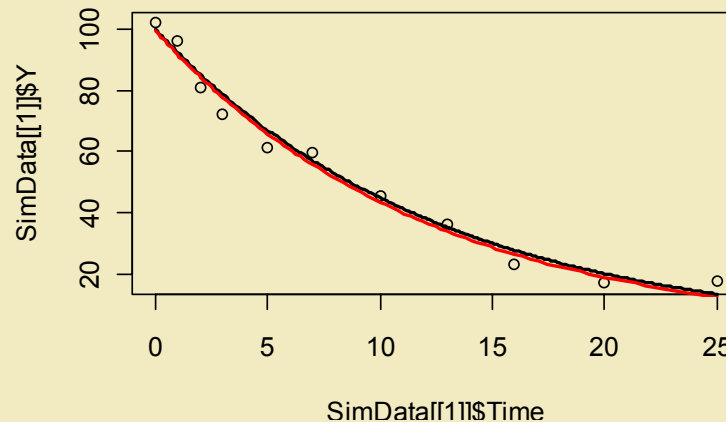
Estimate model

- `sm <- PSM.smooth(mod1, SimData, fit$THETA, subs=10)`

```
> names(sm[[1]])
```

[1]	"Time"	"Xs"	"Ps"	"Ys"	"Xf"	"Pf"
"Xp"	[8]	"Pp"	"Yp"	"R"	"eta"	"negLogL"

```
lines(sm[[1]]$Time, sm[[1]]$Ys, col=2, lwd=2)
```



Exercise 1

- **1.1** R Basic operations
- **1.2** Simulation: Bolus 1-Compartment
- **1.3** Simulation: Oral Absorption
- **1.4** Simulation: Infusion 1-Compartment