

Operational characteristics of saemix, an R package implementing the SAEM algorithm

Emmanuelle Comets^{1★}, Audrey Lavenu² and Marc Lavielle³

¹ INSERM UMRI1137 IAME & Univ Denis Diderot, Paris, France ² Univ Rennes-I & INSERM CIC 1414, Rennes, France ³ INRIA Saclay & Ecole Polytechnique, France

Objective: assess the operational characteristics of saemix in terms of performance and scalability, using simulated data.

Introduction

- Stochastic Approximation Expectation Maximization (SAEM) algorithm
 - stochastic approximation version of the EM algorithm
 - quick and efficient convergence to the maximum likelihood estimators (MLE) [1]
 - increasingly widespread use over the last few years
- Implementation in MONOLIX [2], NONMEM and Matlab (statistical toolbox)
- R [3], a general statistical software
 - many packages for statistical analyses, including parameter estimation in linear and nonlinear mixed models (nlme, lmer, ...)
 - used by many modellers to handle data, prepare runs and evaluate results
 - flexible programming language including object-oriented concepts

Methods

Statistical models

Model for observation y_{ij}

$$y_{ij} = f(\psi_i, x_{ij}) + g(\psi_i, \gamma, x_{ij})\epsilon_{ij}$$

- subject i ($i = 1, \dots, N$), with n_i observations $\mathbf{y}_i = \{y_{i1}, \dots, y_{in_i}\}$ at times t_{ij} , and covariates $\mathbf{x}_i$
- f : structural model (analytical expression, see example)
- $g = g(a, b, c)$: residual error model (one of: constant, proportional, combined, exponential) with parameters $\gamma = \{a, b, c\}$
- individual parameters ψ_i
 - modelled parametrically as a function $\psi_i = h(\phi_i, \mathbf{z}_i) = h(\mu(\mathbf{z}_i), \eta_i)$ of fixed effects μ and random effects $\eta_i \sim \mathcal{N}(0, \Omega)$
 - in saemix, h can be the identity function (normal distribution for ψ),

the exponential function (log-normal distribution for ψ), or the logit or probit transformations

The SAEM algorithm

The SAEM algorithm computes the MLE of the unknown set of parameters $\theta = (\mu, \Omega, \gamma)$, by maximizing the likelihood of the observations $\ell(y; \theta)$. Given an initial estimate θ_0 , at iteration k :

- Simulation-step:** draw $\psi^{(k)}$ from the conditional distribution $p(\cdot | y; \theta_k)$
- Stochastic approximation:** update the conditional expectation $Q_k(\theta)$ of the complete likelihood $p(y; \psi^{(k)}; \theta)$ according to

$$Q_k(\theta) = Q_{k-1}(\theta) + \gamma_k (\log p(y, \psi^{(k)}; \theta) - Q_{k-1}(\theta)) \quad (1)$$

where (γ_k) is a decreasing sequence of positive numbers with $\gamma_1 = 1$.

- Maximization-step:** update θ_k according to

$$\theta_{k+1} = \underset{\theta}{\operatorname{Arg\,max}} Q_k(\theta).$$

- estimation problems with nlme and nlmer
 - low convergence rate, problems obtaining standard errors
 - performance of nlme still poor when altering initial estimates
 - even lower convergence rates with the scenarios in [4] simulated with sigmoidicity
- performances comparable to the results reported in [4] for several algorithms

Evaluation and comparison of saemix

Objective Evaluate saemix using a simulation study performed by Plan et al. [4] to compare different estimation software

Models:

- structural model: $E_{\max}$ model
- IIV: exponential model, correlation between $E_{\max}$ and EC_{50} ($\rho = 0.5$)
- residual error model: additive

Parameter	Value	Parameter	Value
E_0 (-)	5	$\omega_{E_0}^2$	0.09
$E_{\max}$ (-)	30	$\omega_{E_{\max}}^2$	0.49
EC_{50} (mg)	500	$\omega_{EC_{50}}^2$	0.49
σ (-)	2		

Table 1: Parameter values used in the simulation.

Simulated datasets [4]:

- for each scenario, 100 simulated datasets with $N=100$ subjects
- two designs:
 - rich (RIA): 4 doses (0, 100, 300, 1000)
 - sparse (SIA): random selection of 2 doses per subject

Evaluation of saemix:

- bias and RMSE: compared to the parameters used for the simulation
- settings: 5 chains, $K_1=300$, $K_2=150$; real parameters used as CI
- comparison to the results obtained by nlme [5] and nlmer [6] (FOCE algorithm)

Sample saemix run

Input - data

Data for the first dataset read from disk, used to create a saemix data object through the saemixData function, and plotted:

```
dat<-saemixData(name.data="mc-sim-1-1.dat",
header=T,name.group="ID",
name.predictors="DOSE",
name.response="DV"),
units=list(x="mg",y="% response"))
```

plot(saemix.data)

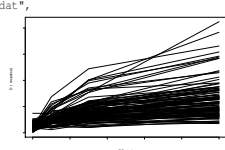


Figure 1: Typical simulated data.

Input - Model

Define the model function and create saemix model object through the saemixModel function:

```
modelmax<-function(psi,id,xidp) {
dose<-xidep[,1]
e0<-psi[id,1]
emax<-psi[id,2]
e50<-psi[id,3]
f<-e0+emax*dose/(e50+dose)
return(f)
}
omega.init<-matrix(c(0.09,0,0,0,0.49,0.245,0.49),ncol=3,byrow=T)
emax.model<-saemixModel(model=modelmax,description="Emax model",
psi0=matrix(c(5,30,500,0,0),ncol=3,byrow=T),
dimnames=list(NULL,c("E0","Emax","EC50")),transform.par=c(1,1,1),
covariance.model=matrix(c(1,0,0,0,1,0,1,1),ncol=3,byrow=T),
omega.init=omega.init)
```

Parameter estimation

```
saemix.options<-list(directory="output",seed=12345,nchains=5,
nriter.saemix=c(300,150))
saemix.fit<-saemix(emax.model,dat,saemix.options)
saemix.fit<-lglg.saemix(saemix.fit)
```

Parameter	Population estimate (SE%)	IIV Variance (SE%)
E_0 (%)	5.0 (5%)	0.09 (36%)
$E_{\max}$ (%)	27.8 (9%)	0.60 (16%)
EC_{50} (mg)	522.9 (12%)	0.69 (24%)
a (mg.L ⁻¹)	2.1 (6%)	-

Table 2: Parameters estimated by saemix for the simulated data (first dataset in scenario with rich sampling).

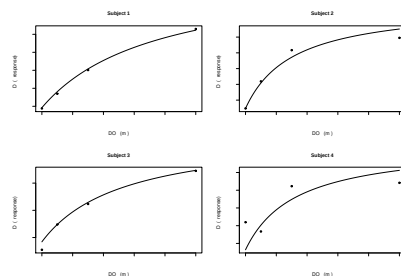


Figure 2: Individual plots for 4 subjects.

Results

Good parameter estimates for saemix

- fixed effects: less than 1% bias and RMSE around 5-10%
- variances: less than 5% bias and RMSE ranging from 20 to 40%

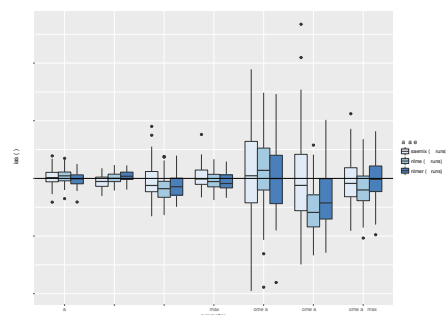


Figure 3: Estimation bias over 100 datasets with rich sampling.

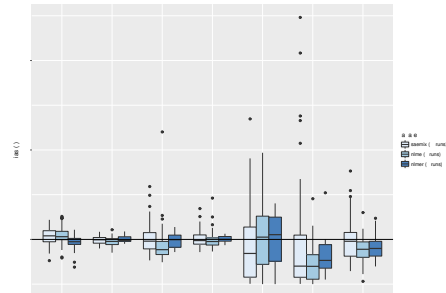


Figure 4: Estimation bias over 100 datasets with sparse sampling.

Operational characteristics

Objective evaluate the scalability of the algorithm through 2 simulations

- effect of dataset size
 - using the previous $E_{\max}$ model
 - varying number of subjects (30, 100, 200, 500) and samples per subject (3, 6, 12)
- effect of model complexity
 - three models with one, two and three exponential terms (respectively 2, 4 and 6 parameters)
 - settings: 100 subjects and 6 sample times (optimised by the PFIM software ([7]) to ensure reasonable identifiability of the most complex model)

Runtimes for the fit of 100 datasets generated for each setting, collected through the Rprof() function in R.

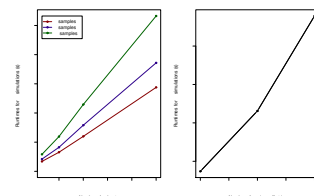


Figure 5: Evolution of runtimes (in seconds, on a personal laptop) with the number of subjects, the number of samples, and model complexity.

Scaling results:

- generation of random effects at each iteration: depends on number of subjects and model complexity
- computation of model predictions: depends mostly on number of samples (and model complexity)
- note: adjustment of the algorithm with more chains when less than 50 subjects in the dataset

Profiling results to be used to identify computational bottleneck in further developments of saemix.

Conclusion

- saemix package for R [8]
 - implementation of the SAEM algorithm for parameter estimation in nonlinear mixed effect models
 - good performance (also adequate type I error for both LRT (with LL computed by importance sampling) and Wald test [9])
 - available on the CRAN
- S4 class system of R: user-friendly input and output system
 - generic methods such as plot(), summary(), ...
 - extensive diagnostic graphs
- under active development
 - hidden Markov models [10]
 - statistical criteria [11]
 - model diagnostics [12] and improved plots with ggplot2
 - covariates and model selection [13]

REFERENCES

- [1] B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Ann Stat*, 27:94–128, 1999.
- [2] M. Lavielle. *Mixed Effects Models for the Population Approach - Models, Tasks, Methods and Tools*. Chapman & Hall/CRC Biostatistics Series, CRC Press, Boca Raton, Florida, USA, 2014.
- [3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.
- [4] E. Plan, A. Maloney, F. Meunier, M.O. Karlsson, and J. Bertrand. Performance comparison of various maximum likelihood nonlinear mixed-effects estimation methods for dose-response models. *AAPS J*, 14:420–432, 2012.
- [5] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, and the R Core team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2009. R package version 3.1-96.
- [6] D. Bates, M. Maechler, and B. Bolker. *Ineq4: Linear Mixed-Effects Models Using S4 Classes*, 2013. R package version 0.999999-2.
- [7] S. Renou, E. Comets, A. Samson, and F. Meunier. Design in nonlinear mixed effects models: Optimization using the Fedorov-Wynn algorithm and power of the Wald test for binary covariates. *Stat Med*, 26:5162–79, 2007.
- [8] E. Comets, A. Lavenu, and M. Lavielle. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. *J Stat Soft*, (Accepted), 2016.
- [9] E. Comets, A. Lavenu, and M. Lavielle. saemix, an R version of the SAEM algorithm. *20th meeting of the PMGE*, Athens, Greece, 2011. Abstr 2173.
- [10] M. Delattre. Inference in hidden markov models and applications to medical studies. *J Soc Fam Stat*, 151:90–105, 2010.
- [11] M. Delattre, M. A. Pourast, and M. Lavielle. A note on BIC in mixed-effects models. *Elec J Stat*, 8:456–475, 2014.
- [12] E. Comets, K. Brendel, and F. Meunier. Computing normalised prediction distribution errors to evaluate nonlinear mixed-effect models: the npde add-on package for R. *Comput Meth and Prog Biomed*, 90:154–66, 2008.
- [13] J. Bertrand and D. Balading. Multiple single nucleotide polymorphism analysis using penalized regression mixed-effect pharmacokinetic models. *Pharmacogenet Genomics*, 23:167–74, 2013.

Inserm **Presenting author**
email: emmanuelle.comets@inserm.fr