The Journal of Open Source Software

simode: R Package for Statistical Inference of Ordinary Differential Equations using Separable Integral-Matching

Rami Yaari[1, 2] **and Itai Dattner**[1]

1 Department of Statistics, University of Haifa, Haifa, 34988, Israel **2** Bio-statistical and Bio-mathematical Unit, The Gertner Institute for Epidemiology and Health Policy Research, Chaim Sheba Medical Center, Tel Hashomer, 52621, Israel

DOI: 10.21105/joss.01850

Software

- Review
- Repository
- Archive

Editor: Matthew Sottile

Reviewers:

- @cscherrer
- @jgoldfar
- @osorensen

Submitted: 29 September 2019
Published: 22 December 2019

Summary

Systems of ordinary differential equations (ODEs) are commonly used for mathematical modeling of the rate of change of dynamic processes in areas such as mathematical biology (Edelstein-Keshet, 2005), biochemistry (Voit, 2000) and compartmental models in epidemiology (Anderson & May, 1992), to mention a few. Inference of ODEs involves the 'standard' statistical problems such as studying the identifiability of a model, estimating model parameters, predicting future states of the system, testing hypotheses, and choosing the 'best' model. However, dynamical systems are typically very complex: nonlinear, high dimensional and only partialy measured. Moreover, data may be sparse and noisy. Thus, statistical learning (inference, prediction) of dynamical systems is not a trivial task in practice. In particular, numerical application of standard estimators, like maximum-likelihood or least-squares, may be difficult or computationally costly. It typically requires solving the system numerically for a large set of potential parameters values, and choosing the optimal values using some nonlinear optimization technique. Starting from a random initial guess, the optimization can take a long time to converge to the optimal solution. Furthermore, there is no guarantee the optimization will converge to the optimal solution at all.

`simode` is an R package for conducting statistical inference for ordinary differential equations that aims to ease the optimization process and provide more robust solutions to parameter estimation problems. The package implements a 'two-stage' approach. In the first stage, fast estimates of the ODEs' parameters are calculated by way of minimization of an integral criterion function while taking into account separability of parameters and equations (if such a mathematical feature exists). In the second stage, a regular nonlinear least-squares optimization is performed starting from the estimates obtained in the first stage, in order to try and improve these estimates.

The statistical methodologies applied in the package are based on recent publications that study theoretical and applied aspects of smoothing methods in the context of ordinary differential equations (Dattner, 2015; Dattner & Gugushvili, 2018; Dattner & Klaassen, 2015; Dattner et al., 2017; Yaari et al., 2018). In that sense `simode` is close in spirit to the `CollocInfer` R package of (Hooker, Ramsay, & Xiao, 2015) and the `episode` R package of (Mikkelsen & Hansen, 2017). Unlike `CollocInfer`, `simode` does not involve penalized estimation but focuses on integral-matching criterion functions instead. Unlike `episode` that also uses integral-matching criteria, `simode` uses a minimization procedure that takes advantage of the mathematical structure of the ODEs (i.e., separability of parameters from equations).

Statistical Methodology

A system of ODEs is given by

$$(1) \quad x'(t) = F(x(t);\theta),\ t \in [0,T], x(0) = \xi$$

where $x(t)$ takes values in R^d, ξ in $\Xi \subset R^d$, and θ in $\Theta \subset R^p$. Let $x(t;\theta,\xi), t \in [0,T]$ be the solution of the initial value problem (1) given values of ξ and θ. We assume measurements of x are collected at discrete time points

$$(2) \quad Y_j(t_i) = x_j(t_i;\theta,\xi) + \epsilon_{ij}, \quad i = 1,\ldots,nj = 1,\ldots,d$$

where the random variables ϵ_{ij} are independent measurement errors (not necessarily Gaussian) with zero mean and finite variance. By integration, equation (1) yields the system of integral equations

$$(3) \quad x(t) = \xi + \int_0^t F(x(s);\theta)\,ds\ \ t \in [0,T].$$

Here $x(t) = x(t;\theta,\xi)$ is the true solution of the ODE. Let $\hat{x}(t)$ stand for a nonparametric estimator (e.g., smoothing the data using splines or local polynomials) of x given the observations (2). The criterion function of an integral-matching approach for a fully observed systems of ODEs takes the form

$$(4) \quad \int_0^T \parallel \hat{x}(t) - \zeta - \int_0^t F(\hat{x}(s);\eta)\,ds \parallel^2 dt$$

where $\|\cdot\|$ denotes the standard Euclidean norm. The estimator of the parameter will be the minimizer of the criterion function (3), with respect to ζ and η. As its name suggests, integral-matching avoids the estimation of derivatives of the solution x as done in other smooth and match applications and hence is more stable (Dattner & Klaassen, 2015).

The `simode` package is especially useful for a class of ODE systems that are linear in the parameters θ, which means the system can be expressed as

$$(5) \quad F(x(t);\theta) = g(x(t))\theta$$

This ODE system is separable in the linear parameter vector θ. In this case, minimizing the integral criterion function (4) with respect to ζ and η results in the direct estimators

$$(6) \quad \hat{\xi} = \left(TI_d - \hat{A}\hat{B}^{-1}\hat{A}^\top\right)^{-1} \int_0^T \left(I_d - \hat{A}\hat{B}^{-1}\hat{G}^\top(t)\right)\hat{x}(t)\,dt,$$

$$(7) \quad \hat{\theta} = \hat{B}^{-1} \int_0^T \hat{G}^\top(t)\left(\hat{x}(t) - \hat{\xi}\right)dt$$

where I_d denotes the $d \times d$ identity matrix and where

$$\hat{G}(t) = \int_0^t g(\hat{x}(s))\,ds\,, \quad t \in [0,T], \hat{A} = \int_0^T \hat{G}(t)\,dt, \hat{B} = \int_0^T \hat{G}^\top(t)\hat{G}(t)\,dt.$$

Therefore, in this case, the complex task of nonlinear optimization reduces to the least squares solutions (6) and (7) which can be calculated directly, leading to a substantial computational improvement. An ODE system can also be semi-linear in the parameters, meaning that some of the parameters are separable and some are not. Formally, this can be described as

$$(8) \quad F(x(t);\theta) = g(x(t);\theta_{NL})\theta_L$$

where θ_{NL} are the nonlinear parameters and θ_L are the linear parameters. In this case, the first stage of inference using `simode` will involve minimization of the integral criterion function

(4), where the nonlinear parameters are obtained using nonlinear optimization and the linear parameters are calculated directly using solutions (6) and (7) given the estimates of the nonlinear parameters at each iteration of the optimization. If the ODE system has no separable parameters, the first stage of inference using `simode` will simply perform minimization of the integral criterion function (4) using nonlinear optimization. It is not mandatory for the user of the package to know which parameters are linear and which are not. By default, all parameters are assumed to be linear. If this is not the case, the user will be notified which parameters should be set as nonlinear. This feature makes it very useful for handling ODEs with linear features in case the mathematical knowledge for characterizing them is lacking.

Additional Features

`simode` implements several features supporting various modeling setups and requirements, including:

- external input functions - inference of systems that employ external time-related data or function.
- user-defined likelihood functions - inference using a user-defined likelihood function.
- partially observed systems - inference of partially observed systems is supported when the unobserved variables can be reconstructed using estimates of the system parameters.
- multiple subjects - inference using observations of multiple subjects (experiments), where some parameters are assumed to be the same for all subjects while other parameters are specific to an individual subject.
- system decoupling - estimation of each equation's parameters separately using data smoothing to replace variables appearing in that equation. As (Voit & Almeida, 2004) have shown, this may lead to better reconstruction of the underlying dynamic system.
- parallel Monte-Carlo simulations - fitting in parallel sets of observations from Monte Carlo simulations.
- confidence intervals - calculation of confidence intervals for the parameters estimates using profile likelihood.

Example

Consider the following simple biochemical system taken from Chapter 2, Page 54 of (Voit, 2000):

$$(9) \qquad \begin{array}{l} x_1'(t) = 2x_2(t) - 2.4x_1(t)^{0.5}, \\ x_2'(t) = 4x_1(t)^{0.1} - 2x_2(t) \end{array}$$

This system is a special case of an S-system (Voit, 2000) defined as

$$x_j'(t) = \alpha_j \Pi_{k=1}^{d} x_k^{g_{jk}}(t) - \beta_j \Pi_{k=1}^{d} x_k^{h_{jk}}(t), \quad j = 1, \ldots, d.$$

Here, α_j, β_j are rate constants and g_{jk}, h_{jk} are kinetic orders that reflect the strength and directionality of the effect a variable has on a given influx or efflux. The system is linear in α_j, β_j but nonlinear in g_{jk}, h_{jk}. For example, (9) can be written in the form of (8), as:

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \begin{pmatrix} x_1^{g_{11}}(t)x_2^{g_{12}}(t), -x_1^{h_{11}}(t)x_2^{h_{12}}(t) \\ x_1^{g_{21}}(t)x_2^{g_{22}}(t), -x_1^{h_{21}}(t)x_2^{h_{22}}(t) \end{pmatrix} \theta_L,$$

where $\theta_L = (\alpha_1, \beta_1, \alpha_2, \beta_2)^\top = (2, 2.4, 4, 2)^\top$ and
$\theta_{NL} = (g_{11}, g_{12}, h_{11}, h_{12}, g_{21}, g_{22}, h_{21}, h_{22})^\top = (0, 1, 0.5, 0, 0.1, 0, 0, 1)^\top$.

Here we demonstrate how to define this system symbolically, in order to be used with `simode`:

```
R> pars <- c('alpha1','g12','beta1','h11', 'alpha2','g21','beta2','h22')
R> vars <- paste0('x', 1:2)
R> eq1 <- 'alpha1*(x2^g12)-beta1*(x1^h11)'
R> eq2 <- 'alpha2*(x1^g21)-beta2*(x2^h22)'
R> equations <- c(eq1,eq2)
R> names(equations) <- vars
R> theta <- c(2,1,2.4,0.5,4,0.1,2,1)
R> names(theta) <- pars
R> x0 <- c(2,0.1)
R> names(x0) <- vars
```

The following code solves the model and generates observations according to the statistical model defined in equation (2), where the distribution of the measurement error is Gaussian with standard deviation of 0.05. It uses 'solve_ode' in `simode` that wraps the 'ode' function of `deSolve` package (Soetaert, Petzoldt, & Setzer, 2010) and accepts symbolic objects:

```
R> library("simode")
R> set.seed(1000)
R> n <- 50
R> time <- seq(0,10,length.out=n)
R> model_out <- solve_ode(equations,theta,x0,time)
R> x_det <- model_out[,vars]
R> sigma <- 0.05
R> obs <- list()
R> for(i in 1:length(vars)) {
+    obs[[i]] <- x_det[,i] + rnorm(n,0,sigma)
+  }
R> names(obs) <- vars
```

Now that we have setup the system of ODEs in a symbolic form and generated observations from the statistical model, we can use the `simode` package to estimate model parameters. We define the linear parameters $\theta_L = (\alpha_1, \beta_1, \alpha_2, \beta_2)^\top$ and the nonlinear parameters $\theta_{NL} = (g_{12}, h_{11}, g_{21}, h_{22})$. For the nonlinear parameters we need to provide initial guess values for optimization. In this example, we generate random initial guess values in the vicinity of the true nonlinear parameters. The call to 'simode' returns an object of class `simode`, containing the parameters estimates obtained using integral-matching (im_est) as well as those obtained using nonlinear least-squares optimization starting from the integral-matching estimates (nls_est).

```
R> lin_pars <- c('alpha1','beta1','alpha2','beta2')
R> nlin_pars <- setdiff(pars,lin_pars)
R> nlin_init <- rnorm(length(theta[nlin_pars]),theta[nlin_pars],
+                     0.1*theta[nlin_pars])
R> names(nlin_init) <- nlin_pars
R> est <- simode(
+    equations=equations, pars=pars, fixed=x0, time=time, obs=obs,
+    nlin_pars=nlin_pars, start=nlin_init)
R> summary(est)

call:
simode(equations = equations, pars = pars, time = time, obs = obs,
    nlin_pars = nlin_pars, fixed = x0, start = nlin_init)

equations:
```

```
                                  x1                                   x2
"alpha1*(x2^g12)-beta1*(x1^h11)" "alpha2*(x1^g21)-beta2*(x2^h22)"

initial conditions:
 x1  x2
2.0 0.1

parameter estimates:
     par       type      start im_est nls_est
1 alpha1     linear         NA 1.8740  2.0580
2    g12 non-linear 0.86305878 1.0040  0.9637
3  beta1     linear         NA 2.2270  2.4490
4    h11 non-linear 0.50815084 0.5136  0.4877
5 alpha2     linear         NA 3.5260  3.6870
6    g21 non-linear 0.09886774 0.1057  0.1019
7  beta2     linear         NA 1.5840  1.7160
8    h22 non-linear 1.08597553 1.1330  1.0840

im-method:  separable

im-loss:  0.1142

nls-loss:  0.239
```

An implementation of the generic plot function for `simode` objects can be used to plot the fits obtained using these estimates (Figure 1). In this case, it is also possible to plot the fit against the true curves, since the true values of the parameters that were used to generate the observations are known.

```
R> plot(est, type='fit', pars_true=theta[lin_pars],
+       mfrow=c(1,2), legend=T)
```

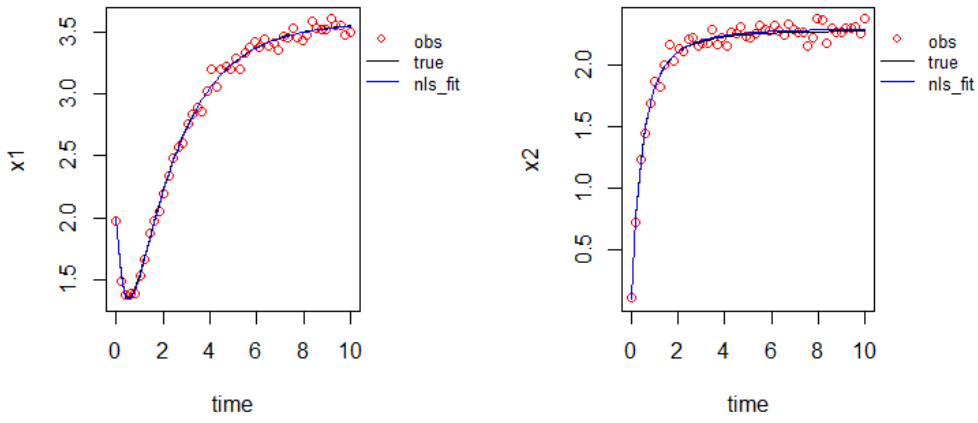

Figure 1: True and estimated solutions x_1 and x_2 of the biochemical system of equation (9)

More examples of usages of `simode`, including examples with Lotka-Volterra, FitzHugh-Nagumo spike potential equations and SIR (Susceptible-Infected-Recovered) systems, can be found in the package demos and the package manual (Yaari & Dattner, 2018).

References

Anderson, R. M., & May, R. M. (1992). *Infectious diseases of humans: Dynamics and control*. Oxford University Press.

Dattner, I. (2015). A model-based initial guess for estimating parameters in systems of ordinary differential equations. *Biometrics*, *71*(4), 1176–1184. doi:10.1111/biom.12348

Dattner, I., & Gugushvili, S. (2018). Application of one-step method to parameter estimation in ode models. *Statistica Neerlandica*, *72*(2), 126–156. doi:10.1111/stan.12124

Dattner, I., & Klaassen, C. A. J. (2015). Optimal rate of direct estimators in systems of ordinary differential equations linear in functions of the parameters. *Electron. J. Statist.*, *9*(2), 1939–1973. doi:10.1214/15-EJS1053

Dattner, I., Miller, E., Petrenko, M., Kadouri, D. E., Jurkevitch, E., & Huppert, A. (2017). Modelling and parameter inference of predator–prey dynamics in heterogeneous environments using the direct integral approach. *Journal of The Royal Society Interface*, *14*(126). doi:10.1098/rsif.2016.0525

Edelstein-Keshet, L. (2005). *Mathematical models in biology. Classics in applied mathematics* (Vol. 46). Society for Industrial; Applied Mathematics.

Hooker, G., Ramsay, J. O., & Xiao, L. (2015). CollocInfer: Collocation inference in differential equation models. *Journal of Statistical Software*. doi:10.18637/jss.v075.i02

Mikkelsen, F. V., & Hansen, N. R. (2017). Learning large scale ordinary differential equation systems. *arXiv preprint arXiv:1710.09308*.

Soetaert, K., Petzoldt, T., & Setzer, R. W. (2010). Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, *33*(9), 1–25. doi:10.18637/jss.v033.i09

Voit, E. O. (2000). *Computational analysis of biochemical systems: A practical guide for biochemists and molecular biologists*. Cambridge University Press.

Voit, E. O., & Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, *20*(11), 1670–1681. doi:10.1093/bioinformatics/bth140

Yaari, R., & Dattner, I. (2018). *simode: R package for statistical inference of ordinary differential equations using separable integral-matching*. Retrieved from https://cran.r-project.org/web/packages/simode/vignettes/R_package_simode.pdf

Yaari, R., Dattner, I., & Huppert, A. (2018). A two-stage approach for estimating the parameters of an age-group epidemic model from incidence data. *Statistical Methods in Medical Research*, *27*(7), 1999–2014. doi:10.1177/0962280217746443