

So you want to fit ODEs to data ... *⟨BMB: title??⟩*

Sang Woo Park

Ben Bolker

David J. D. Earn *

August 15, 2019

Target audience: students and researchers with some degree of mathematical sophistication who aren't necessarily great with statistics. They know a reasonable amount about ODEs, probably a little bit about numerical integration, maybe next to nothing about maximum likelihood/process vs observation error/state space models/MCMC/etc..

Target journal: R Journal? JSS? Bull Math Biol? Methods in Ecol & Evol?

1 Introduction

Mathematical models based on ordinary differential equations (ODEs) are ubiquitous in a vast array of application areas. *⟨DJDE: refs? maybe not necessary.⟩* *⟨BMB: "in science and engineering?"⟩* Researchers studying ODE models are often highly skilled in dynamical systems theory, but many lack experience estimating the parameters of their models from data. When confronted with observations from the real world, how should one go about fitting an ODE model to the data?

In principle, the answer is straightforward: examine all possible solutions of the ODEs (considering all possible parameter values and initial states) and identify the solution curve that is closest to the observed data. This approach is sometimes feasible, for simple enough problems. But more often than not, fitting an ODE to data is a challenging statistical and computational problem. In particular:

- The dimension of the combined of parameter and initial states is rarely less than 4 and often much larger, making a naïve search for the best fitting solution computationally overwhelming.
- Different measures of goodness of fit may identify different "best" solutions.
- Multiple solutions, associated with different regions of parameter space, may fit the data equally well, or nearly equally well.
- Statistical inference — testing for clear evidence of the operation of a particular process, attaching a measure of confidence to a parameter estimates, or putting confidence intervals on predictions — requires additional knowledge or assumptions about the stochastic processes that generate the variation of observed data around the best-fit solution.

*Department of Mathematics & Statistics, McMaster University, Hamilton, Ontario, Canada

As one attempts to address these issues, more subtle challenges arise and one can quickly reach the limits of the state of the art. Making meaningful mechanistic inferences from ODE fits to data can be difficult, especially when fitting complex models to small, noisy data sets.

At this point you may want to give up and work on something else, but we hope you won't. The situation isn't *that* bad. Our goal here is to provide—and to present a pedagogical introduction to—an R package that makes fitting many ODEs to data relatively painless. Simultaneously, we aim to explain the uncertainties and limitations inherent in ODE-fitting, and provide convenient tools and references for users who wish to delve more deeply into the subject.

(DJDE: The general style I'm imagining is similar to Numerical Recipes (Press et al., 1992), i.e., provide tools that work but also explain clearly and concisely why they work, what they are doing, and under what circumstances they can be expected to fail.)

2 Trajectory matching: from simple to complex

2.1 Least-squares fitting

The natural (“intuitive”?) approach. Suppose we have (1) a set of gradient functions; (2) a method for numerical integration; (3) a set of observations (for simplicity, from a single time series); (4) a method for numerical minimization of an objective function. Define a trajectory (a solution of the ODE for specified parameters β [and ICs] and time points t_i) as $\hat{y}(\beta, t_i)$ *(BMB: notation???)*

$$\hat{\beta} = \arg \min_{\beta} \sum_i (y(t_i) - \hat{y}(\beta, t_i))^2 \quad (1)$$

gradient functions should be straightforward (you know what you want to model, right? ...)

observations as noted above, we're assuming a single time series for now. Regularity of sampling isn't much of a problem. It's OK to measure a subset of the state variables (e.g. just prevalence of infected in an SIR model, or just predators in a pred-prey model). May need some care to make sure you're comparing observations with ODE properly (e.g. do observations represent measurements of state variables, rates, or integrals of rates? Measuring cumulative properties may have tricky statistical consequences later ...)

integrator for simple problems, anything should do. Open question as to whether simple/stable (e.g. RK4) is better than something more sophisticated (e.g. LSODA) *(BMB: in general, the easier the problem, the less the details matter!)*

minimizer similar to integrator, i.e. details don't matter if the problem is easy. Derivative-free methods such as Nelder-Mead are robust but slower; derivative-based methods are effective *if sensitivity equations are used* (see below). Some minimizers such as

Levenberg-Marquardt are particularly tuned to least-squares (although derivs still matter)

initial conditions if all variables are observed, we can pretend the initial observations as known without error and set $\hat{y}(0) = y(0)$. Otherwise we have to treat ICs as auxiliary parameters.

Least-squares fits provide solutions that are in some sense “optimal” (although the Gauss-Markov theorem only applies to *linear* models), but don’t provide any scope for inference. That said, by avoiding inference, one also avoids having to think much about the characteristics of the observations (independence, distribution, ...)

EXAMPLE? (this would look a bit funny with `fitode` as we’d have to use a fake-Gaussian)

2.2 Gaussian errors

If we’re willing to assume that the residuals (observed-expected) are *independent, identically distributed* Gaussian random variables, then we can start making statistical inferences: confidence intervals on parameters, confidence and prediction intervals on predictions, etc.. Assuming independence of successive residuals typically means we are assuming that the noise in the data is *observation error*; that is, the underlying process is deterministic (and smooth), and all the noise in the data comes from (independent) imperfect observations of the process.

- Gaussian errors are equivalent to *scaled* least-squares errors
- we can fit via explicit Gaussian (with or without normalization terms), profiled Gaussian (set $\hat{\sigma} = \text{RSS}/n$), or quasi-likelihood (fit with $\sigma = 1$, then estimate $\hat{\sigma}$ from final value of RSS)
- once we have a Gaussian fit, we can use standard methods (Fisher information/Wald) to estimate parameter CIs [still assuming iid Gaussian]
- for CIs on predictions or forecasts, use delta method or [whatever it’s called when we sample from a MVN sampling distribution of parameters; par. bootstrap? PPI?] or importance sampling or ...
- now we are assuming independence, so measuring cumulative variables will result in incorrect inference (cf. ODE papers in epidemiology that do this carelessly??)

EXAMPLE?

2.3 non-Gaussian errors

Once we’ve gotten this far, it’s fairly easy to extend the model to cover conditional distributions other than Gaussian. For example, we often have count data that would naturally be modeled by a Poisson or negative binomial distribution.

EXAMPLE? (does `fitode` have an update method? that would be a cute way to make compact examples)

This doesn’t raise very many additional problems.

3 Increasing robustness and efficiency

So far we've assumed that the numerical methods should Just Work. People often have really crappy data (e.g., very short time series, complex models). In R, we're typically depending on derivatives computed by finite differences.

3.1 Sensitivity equations

Don't forget to cite Raue et al. (2013).

3.2 Link functions

aka, transforming parameters to unconstrained scales. Don't know if this deserves its own section, but it's a good idea. Generally improves scaling, applicability of Wald approximations, robustness.

The only caveat (besides increased complexity) is that if a parameter ever ends up on the boundary of its feasible space, then transforming to an unconstrained space will push the optimum to infinity, leading to problems ... box constraints may be better in this case, although Wald-based inference will break in any case.

3.3 Self-starting models

fitode doesn't do this at present; demote to an "other" paragraph?

3.4 Compiled models

fitode doesn't do this at present; demote to an "other" paragraph? See deSolve vignette on compiled gradient functions: also cf. nimble, odeintr, cOde, dMod, rodeo, others?

Speed is important when using simulation-based methods (SMC, ABC, etc.; see below) or when fitting a large number of series. (Sensitivity-based methods can help improve speed, but not as much as compilation.)

3.5 Multi-start methods

Slow (hence faster integration would be nice) but deals with multi-modality. Again cf. Raue et al. (2013): Latin hypercube, Sobol, etc.. They discuss diagnosis of multi-modality as well.

3.6 Other

Should we include regularization here? Box constraints are a cheaper/easier possibility. Regularization would make a nice lead-up to Bayesian methods.

4 Beyond frequentist models

(BMB: hard to decide on the best order: beyond-freq (i.e. Bayesian) then beyond-trajectories or vice versa? Reasons you would need these do overlap, although not completely)

Discuss advantages of Bayesian approach (priors, integration over uncertainty). Need to think about sampling (Gibbs, Metrop-Hastings, HMC ...)

Stan, boms (??), deBIInfer

5 Beyond trajectory-matching

5.1 gradient matching etc. (process error only?)

CollocInfer, papers by Hooker/Ellner/etc.; Kim McAuley

5.2 state-space models

Huge range here. Sequential Monte Carlo (pomp etc.). Kalman/extended Kalman filters?

Heydari et al. (2014) (SDEs + Bayesian + Kalman filter)

Mention ABC/synthetic likelihood?

5.3 random effects/mixed models

What do we want to say here? Easier in Bayesian-toolbox contexts such as Stan. nlmeODE;

Tornøe et al. (2004) ...

6 Package comparison

7 Miscellaneous

- Appropriate citations to PKPD literature, e.g. RxODE, Wang et al. (2016) (although this looks like it is a simulation rather than fitting tool?). In many cases they use particular models with nonlinear but known solutions, e.g. two-compartment linear models.
- cite that mixed-ODE paper we reviewed.
- Discuss identifiability/estimability?

References

- Heydari, J., C. Lawless, D. A. Lydall, and D. J. Wilkinson (2014, August). Fast Bayesian parameter estimation for stochastic logistic growth models. *Bio Systems* 122, 55–72.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C: The Art of Scientific Computing* (2nd ed.). New York: Cambridge University Press.
- Raue, A., M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, U. Klingmüller, and J. Timmer (2013, September). Lessons Learned from Quantitative Dynamical Modeling in Systems Biology. *PLOS ONE* 8(9), e74335.
- Tornøe, C. W., H. Agersø, E. N. Jonsson, H. Madsen, and H. A. Nielsen (2004, October). Non-linear mixed-effects pharmacokinetic/pharmacodynamic modelling in NLME using differential equations. *Computer Methods and Programs in Biomedicine* 76(1), 31–40.
- Wang, W., K. M. Hallow, and D. A. James (2016). A Tutorial on RxODE: Simulating Differential Equation Pharmacometric Models in R. *CPT: Pharmacometrics & Systems Pharmacology* 5(1), 3–10.