

Parameter estimation for nonlinear dynamical models.

GSoC 2018 Proposal

Vaibhav Kumar Dixit

February 25, 2018

1 Synopsis

Most of the scientific community deals with the basic problem of trying to mathematically model the reality around them, this often involves dynamic systems. The general trend to model these complex dynamical systems is through the use of Differential Equations. These differential equation models often have non-measurable parameters. The popular “forward-problem” of simulation consists of solving the differential equations for a given set of parameters, the “inverse problem” to simulation, known as parameter estimation, is the process of utilizing data to determine these model parameters. This area is yet to be covered to the same extent and thus it presents a good avenue for further research primarily because of its applications in systems biology, HIV-AIDS study, and drug dosage estimation.

Parameter estimation aims to find the unknown parameters of the model which give the best fit to a set of experimental data. In this way, parameters which cannot be measured directly will be determined in order to ensure the best fit of the model with the experimental results. The purpose of this project is to utilize the growing array of statistical, optimization, and machine learning tools in the Julia ecosystem to build library functions with primary focus on Bayesian Inference. These will make it easier for scientists to perform this parameter estimation with the most high-powered and robust methodologies.

Currently JuliaDiffEq has DiffEqParamEstim.jl for the Optimization Methods and DiffEqBayes.jl for Bayesian Methods of parameter estimation. DiffEqParamEstim.jl supports the optimization algorithms available in BlackBoxOptim.jl, NLOpt.jl and Optim.jl. DiffEqBayes.jl currently supports Stan.jl, Turing.jl and DynamicHMC.jl as the backends for the MCMC step. I propose the following.

- To add Stochastic Approximation Expectation-Maximization and Maximum a Posteriori Estimation methods (#62) for parameter estimation.
- I would also work on extending support for parameter estimation in Stochastic Differential equations by adding first differences distribution to generalized Log-Likelihood (#64).
- Also there isn't much work done on a good heuristic to measure the performance in Parameter Estimation problems, this would be another avenue to pursue as benchmarking is

among the most important aspects of such packages and therefore some concrete way of comparative analysis will be required.

- Further during the course of the summer I would be actively involved in creating new benchmarks using the current methods and for the methods to be developed by me (#30).

2 The Project

2.1 Background

Julia already has among the best Differential Equations suites with a very active community. It also has a great array of optimization and statistical packages, therefore most of the basic infrastructure required for the development of new parameter estimation methodologies, are already in place. The primary task would be to implement the mentioned methods using these tools in a generalised manner so that all the problem types can be covered.

2.2 Goals

2.2.1 Theoretical description

1. SAEM

We start off with a problem of fitting linear and nonlinear ODEs with random effects in the structural parameters under situations in which the initial conditions of the ODEs may be unknown. The parameters can be considered a mix of fixed β and random effects b , the standard Expectation-Maximization (EM) procedures involves obtaining the Maximum Likelihood Estimate by cycling iteratively through an expectation (E)-step which involves analytically computing terms that appear in a pseudo-loglikelihood function given by the conditional expectation of the complete-data log-likelihood function with respect to the distribution $\Pr(b \mid Y; \theta)$ followed by the maximization (M)-step in which the parameter estimates are updated by using analytic formulas that serve to maximize the pseudo-loglikelihood function.

SAEM differs from conventional E-M algorithm because we use a stochastic approximation procedure in the E-step, coupled with a gradient-type updating procedure (e.g. the Newton-Raphson) in the M-step. In the E-step we handle the intractable integrals of expectations by instead using Markov Chain Monte Carlo sampling techniques to obtain summary statistics of $\Pr(b \mid Y; \theta)$ and thus replacing the analytical expectation by an "approximation", then in the M-step a Newton-Raphson algorithm is used to obtain MLEs of θ , here a sequence of gain constants γ is used as a weightage factor for the degree to which the parameter estimates are updated in the subsequent iterations.

To use any gradient-type algorithm such as the Newton-Raphson we require the score vector and the information matrix of the loglikelihood function, denoted as $s_Y(\theta; Y)$ and $I_Y(\theta; Y)$, respectively. These are then used to update the parameter vector as defined by

$$\theta^m = \theta^{m-1} + [I_Y(\theta^{(m-1)}; Y)]^{-1} s_Y(\theta^{(m-1)}; Y)$$

Thus in the E step the samples obtained from MCMC on $\Pr(b \mid Y; \theta)$ are then used to obtain the score and Fisher information matrix. Then in the M Step updated parameter estimates are obtained using the above obtained score and information and the gain constant γ which is useful in moderating the estimation algorithm from settling too quickly into local minima in earlier

iterations and also helps speed convergence toward a final set of parameter estimates during the later iterations. The SAEM algorithm alternates between the E-step and the M-step until some predefined convergence criteria have been met.

2. MAP

The usual Maximum a posteriori algorithm involves prediction of parameters of a model, given the data. We assume a prior distribution for the parameter θ with a probability density function, say, $g(\theta)$ and the distribution of data y given θ as $f(Y | \theta)$. Then θ is estimated as the mode of the posterior distribution obtained using the Bayes Theorem from the prior distribution and the likelihood as follows.

$$\theta_{MAP}(x) = \operatorname{argmax}_{\theta}(f(x | \theta)g(\theta))$$

In our case of Dynamical Models, the parameters, as mentioned in the SAEM description are considered to be a mixture of fixed and random effects and we assume the parameter vector as θ and the random effect as u . Thus the observed log-likelihood for the subject i as

$$L_{\theta}^i = \log\left(\int_{R^p} L_{y^i|u^i}^{\theta}(Y^i | u)\phi(u)du\right)$$

We denote $L_{\theta} = L_{\theta}^i + \dots L_{\theta}^n$. Now further we get

$$\theta_{MAP} = \operatorname{Argmax}_{\theta}[L_{\theta}^{MAP}] \quad \text{where } L_{\theta}^{MAP} = L_{\theta} + \log(\pi(\theta))$$

This gives us our parameter estimate.

3. First differences distribution to loglikelihood

The idea here is to provide an option for using a cost function that uses the likelihood of the first differences of the data against a distribution as the loss. In the the current implementation the distribution of the data points is measured at every time point but this can lead to problems in some cases where the SDE's solution is constant, here we lose some information because wrong parameters can get that correct too.

4. New benchmarks and heuristic for comparative analysis of parameter estimation methods

There is a lot of scope for expanding the current set of benchmarks with new problems, this will require some familiarity with academic literature to obtain new dynamical models and then to convert them to DifferentialEquations.jl's form. Also currently the parameter estimation benchmarking present a not so straightforward problem of comparison between the various available optimization and bayesian approaches available, this arises because there isn't a concrete set of properties that can be evaluated to make that comparison and also because the results are very sensitive to intialization both in accuracy and the time taken for obtaining the result. Hence using some standard problems with similar intializations of the parameters and obtaining more informative results from the algorithms will be required for future work, but this would require some theoretical guarantees to ensure reproducibility and correctness.