Lotka-Volterra Bayesian Parameter Estimation Benchmarks

Vaibhav Dixit, Chris Rackauckas

April 6, 2020

0.1 Parameter Estimation of Lotka-Volterra Equation using DiffEqBayes.jl

```
using DiffEqBayes, CmdStan, DynamicHMC

using Distributions, BenchmarkTools
using OrdinaryDiffEq, RecursiveArrayTools, ParameterizedFunctions
using Plots

gr(fmt=:png)

Plots.GRBackend()

f = @ode_def LotkaVolterraTest begin
    dx = a*x - b*x*y
    dy = -c*y + d*x*y
end a b c d
```

```
(::Main.WeaveSandBox7.LotkaVolterraTest{Main.WeaveSandBox7.var"###Parameter
izedDiffEqFunction#665",Main.WeaveSandBox7.var"###ParameterizedTGradFunctio
n#666",Main.WeaveSandBox7.var"###ParameterizedJacobianFunction#667",Nothing
,Nothing,ModelingToolkit.ODESystem}) (generic function with 1 method)
```

```
u0 = [1.0,1.0]
tspan = (0.0,10.0)
p = [1.5,1.0,3.0,1,0]
```

```
5-element Array{Float64,1}:
 1.5
 1.0
 3.0
 1.0
 0.0
```

```
prob = ODEProblem(f,u0,tspan,p)
sol = solve(prob,Tsit5())
```

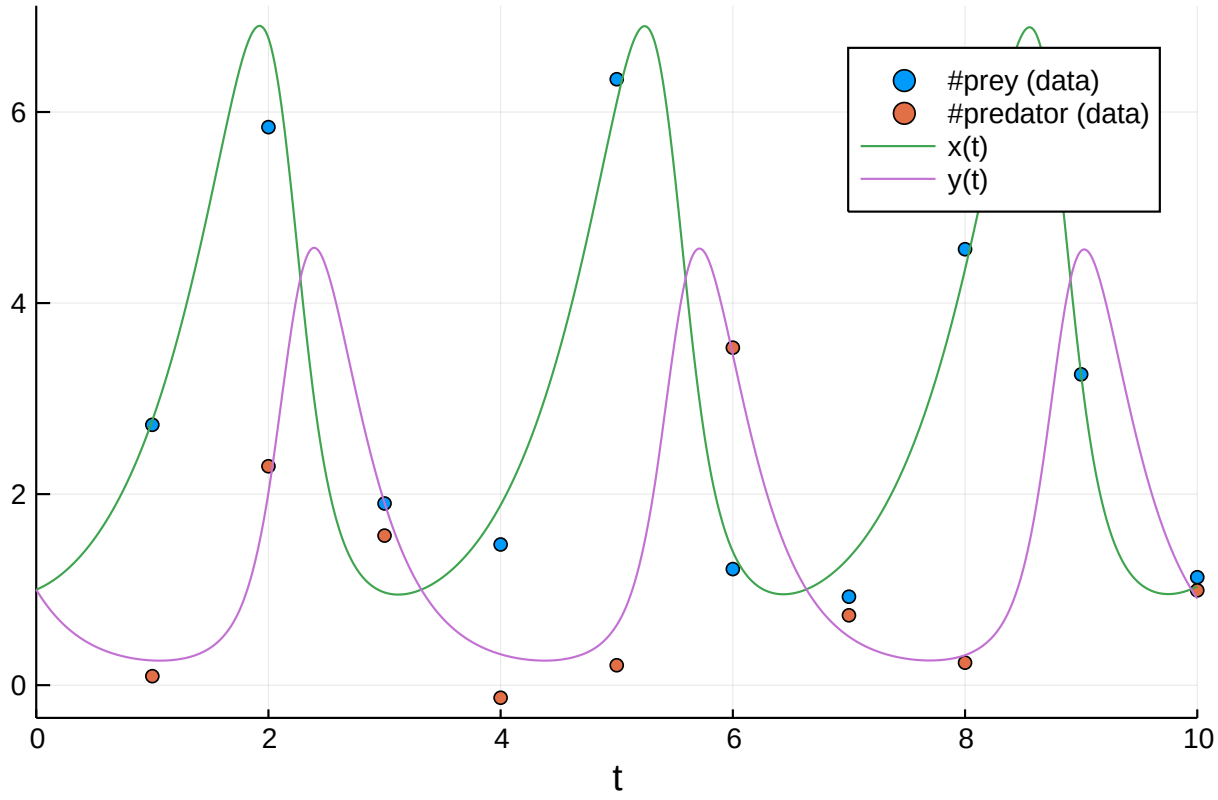
```
retcode: Success
Interpolation: specialized 4th order "free" interpolation
t: 34-element Array{Float64,1}:
  0.0
  0.0776084743154256
```

```
  0.23264513699277584
  0.4291185174543143
  0.6790821776882875
  0.9444045910389707
  1.2674601253261835
  1.6192913723304114
  1.9869755337814992
  2.264090367186479
  ⋮
  7.584862904164952
  7.978068388305894
  8.483164907244102
  8.719247868929038
  8.949206527971544
  9.200184813643565
  9.438028630962807
  9.711807852444823
 10.0
u: 34-element Array{Array{Float64,1},1}:
 [1.0, 1.0]
 [1.0454942346944578, 0.8576684823217127]
 [1.1758715885138267, 0.639459570317544]
 [1.4196809607170826, 0.4569962601282084]
 [1.8767193485546056, 0.32473343696185236]
 [2.5882499852859384, 0.26336255804531]
 [3.860708771268753, 0.2794458027885767]
 [5.750812903389158, 0.5220073140479389]
 [6.814978737433837, 1.917783300239219]
 [4.3929977807914105, 4.194671536988031]
 ⋮
 [2.614252575185928, 0.26416950055716665]
 [4.2410731685818694, 0.30512345857554246]
 [6.791122470590543, 1.1345265418479897]
 [6.26537352594436, 2.741690196017545]
 [3.78076791065078, 4.431164786168439]
 [1.8164212283793362, 4.0640577258289365]
 [1.1465027088171469, 2.791172606389902]
 [0.9557986534742364, 1.6235632025270912]
 [1.03375813933372, 0.9063703701433561]
```

```
t = collect(range(1,stop=10,length=10))
sig = 0.49
data = convert(Array, VectorOfArray([(sol(t[i]) + sig*randn(2)) for i in 1:length(t)]))
```

```
2×10 Array{Float64,2}:
 2.72574    5.84187  1.9023    1.47278   …   4.56358   3.25368  1.12983
 0.0941525  2.29158  1.56568  -0.131694      0.235526  5.17076  0.992696
```

```
scatter(t, data[1,:], lab="#prey (data)")
scatter!(t, data[2,:], lab="#predator (data)")
plot!(sol)
```




```
priors =
[Truncated(Normal(1.5,0.5),0.5,2.5),Truncated(Normal(1.2,0.5),0,2),Truncated(Normal(3.0,0.5),1,4),Trun
```

```
4-element Array{Truncated{Normal{Float64},Continuous,Float64},1}:
 Truncated(Normal{Float64}(μ=1.5, σ=0.5), range=(0.5, 2.5))
 Truncated(Normal{Float64}(μ=1.2, σ=0.5), range=(0.0, 2.0))
 Truncated(Normal{Float64}(μ=3.0, σ=0.5), range=(1.0, 4.0))
 Truncated(Normal{Float64}(μ=1.0, σ=0.5), range=(0.0, 2.0))
```

0.1.1 Stan.jl backend

The solution converges for tolerance values lower than 1e-3, lower tolerance leads to better accuracy in result but is accompanied by longer warmup and sampling time, truncated normal priors are used for preventing Stan from stepping into negative values.

```
@btime bayesian_result_stan =
stan_inference(prob,t,data,priors,num_samples=10_000,printsummary=false)
```

```
File /Users/vaibhav/DiffEqBenchmarks.jl/tmp/parameter_estimation_model.stan
 will be updated.


File /Users/vaibhav/DiffEqBenchmarks.jl/tmp/parameter_estimation_model.stan
 will be updated.


File /Users/vaibhav/DiffEqBenchmarks.jl/tmp/parameter_estimation_model.stan
 will be updated.
```

```
File /Users/vaibhav/DiffEqBenchmarks.jl/tmp/parameter_estimation_model.stan
 will be updated.


An error occurred while running the previously compiled Stan program.

Please check the contents of file parameter_estimation_model_run.log and th
e'command' field in the Stanmodel, e.g. stanmodel.command.

Error: Return code = -5
```

0.1.2 Turing.jl backend

```
@btime bayesian_result_turing =
turing_inference(prob,Tsit5(),t,data,priors,num_samples=10_000)
```

```
Error: InterruptException:
```

0.1.3 DynamicHMC.jl backend

```
@btime bayesian_result_dynamichmc =
dynamichmc_inference(prob,Tsit5(),t,data,priors,num_samples=10_000)
```

```
Error: InterruptException:
```

0.2 Conclusion

Lotka-Volterra Equation is a "predator-prey" model, it models population of two species in which one is the predator (wolf) and the other is the prey (rabbit). It depicts a cyclic behaviour, which is also seen in its Uncertainity Quantification Plots. This behaviour makes it easy to estimate even at very high tolerance values (1e-3).

```
using DiffEqBenchmarks
DiffEqBenchmarks.bench_footer(WEAVE_ARGS[:folder],WEAVE_ARGS[:file])
```

0.3 Appendix

These benchmarks are a part of the DiffEqBenchmarks.jl repository, found at: https://github.com/JuliaDi

To locally run this tutorial, do the following commands:

```
using DiffEqBenchmarks
DiffEqBenchmarks.weave_file("ParameterEstimation","DiffEqBayesLotkaVolterra.jmd")
```

Computer Information:

```
Julia Version 1.4.0
Commit b8e9a9ecc6 (2020-03-21 16:36 UTC)
Platform Info:
  OS: macOS (x86_64-apple-darwin18.6.0)
  CPU: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
```

Package Information:

```
Status: `/Users/vaibhav/DiffEqBenchmarks.jl/Project.toml`
[28f2ccd6-bb30-5033-b560-165f7b14dc2f] ApproxFun 0.11.10
[a134a8b2-14d6-55f6-9291-3336d3ab0209] BlackBoxOptim 0.5.0
[a93c6f00-e57d-5684-b7b6-d8193f3e46c0] DataFrames 0.20.2
[2b5f629d-d688-5b77-993f-72d75c75574e] DiffEqBase 6.25.2
[ebbdde9d-f333-5424-9be2-dbf1e9acfb5e] DiffEqBayes 2.9.1
[eb300fae-53e8-50a0-950c-e21f52c2b7e0] DiffEqBiological 4.2.0
[f3b72e0c-5b89-59e1-b016-84e28bfd966d] DiffEqDevTools 2.18.0
[c894b116-72e5-5b58-be3c-e6d8d4ac2b12] DiffEqJump 6.5.0
[1130ab10-4a5a-5621-a13d-e4788d82bd4c] DiffEqParamEstim 1.13.0
[a077e3f3-b75c-5d7f-a0c6-6bc4c8ec64a9] DiffEqProblemLibrary 4.6.4
[ef61062a-5684-51dc-bb67-a0fcdec5c97d] DiffEqUncertainty 1.4.1
[0c46a032-eb83-5123-abaf-570d42b7fbaa] DifferentialEquations 6.12.0
[7073ff75-c697-5162-941a-fcdaad2a7d2a] IJulia 1.21.1
[7f56f5a3-f504-529b-bc02-0b1fe5e64312] LSODA 0.6.1
[76087f3c-5699-56af-9a33-bf431cd00edd] NLopt 0.5.1
[c030b06c-0b6d-57c2-b091-7029874bd033] ODE 2.6.0
[54ca160b-1b9f-5127-a996-1867f4bc2a2c] ODEInterface 0.4.6
[09606e27-ecf5-54fc-bb29-004bd9f985bf] ODEInterfaceDiffEq 3.6.0
[1dea7af3-3e70-54e6-95c3-0bf5283fa5ed] OrdinaryDiffEq 5.32.2
[2dcacdae-9679-587a-88bb-8b444fb7085b] ParallelDataTransfer 0.5.0
[65888b18-ceab-5e60-b2b9-181511a3b968] ParameterizedFunctions 5.0.3
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 0.29.9
[b4db0fb7-de2a-5028-82bf-5021f5cfa881] ReactionNetworkImporters 0.1.5
[f2c3362d-daeb-58d1-803e-2bc74f2840b4] RecursiveFactorization 0.1.0
[9672c7b4-1e72-59bd-8a11-6ac3964bc41f] SteadyStateDiffEq 1.5.0
[c3572dad-4567-51f8-b174-8c6c989267f4] Sundials 3.9.0
[a759f4b9-e2f1-59dc-863e-4aeb61b1ea8f] TimerOutputs 0.5.3
[44d3d7a6-8a23-5bf8-98c5-b353f8df5ec9] Weave 0.9.4
[b77e0a4c-d291-57a0-90e8-8db25a27a240] InteractiveUtils
[d6f4376e-aef5-505a-96c1-9c027394607a] Markdown
[44cfe95a-1eb2-52ea-b672-e2afdf69b78f] Pkg
[9a3f8284-a2c9-5f02-9a11-845980a1fd5c] Random
```