Variational inference for stochastic differential equations

Dennis Prangle

November 2018

Newcastle University, UK

Acknowledgements and reference

- Joint work with Tom Ryder, Steve McGough, Andy Golightly

- Supported by EPSRC cloud computing for big data CDT
- and NVIDIA academic GPU grant
- Published in ICML 2018
- http://proceedings.mlr.press/v80/ryder18a.html
- https://github.com/Tom-Ryder/VIforSDEs

Overview

- Background
    - Stochastic differential equations
    - Variational inference
- Variational inference for SDEs
- Example
- Conclusion

Stochastic differential equations (SDEs)

SDE definition

Stochastic differential equation:

$$dX_t = \alpha(X_t, \theta)dt + \sqrt{\beta(X_t, \theta)}dW_t, \quad X_0 = x_0.$$

Discretised form: (Euler-Maruyama)

$$x_{i+1} = x_i + \alpha(x_i, \theta)\Delta\tau + \sqrt{\beta(x_i, \theta)\Delta\tau}\, z_{i+1}$$

Notation:

- x_i – state vector at ith timestep
- α – drift vector
- β – diffusion matrix
- θ – unknown parameters
- x_0 – initial conditions
- $\Delta\tau$ – timestep size
- z_{i+1} – vector of independent $N(0, 1)$ draws

SDE applications

- Systems biology
- Ecology
- Epidemiology
- Finance/econometrics
- Physics
- ...

Simple examples later:

- Lotka-Volterra
- SIR epidemic model

Problem statement

- We observe states at several time points
- (Usually partial noisy observations)
- **Primary goal**
  - Infer parameters θ
  - e.g. their posterior distribution

Problem statement

- We observe states at several time points
- (Usually partial noisy observations)
- **Primary goal**
  - Infer parameters θ
  - e.g. their posterior distribution
- **Secondary goals**
  - Infer states x
  - Model choice/criticism

Posterior distribution

- Let $p(\theta)$ be prior density for parameters
- Posterior distribution is

$$p(\theta, x|y) \propto p(\theta)p(x|\theta)p(y|x, \theta)$$

(prior $\times$ SDE model $\times$ observation model)

Posterior distribution

- Let $p(\theta)$ be prior density for parameters
- Posterior distribution is

$$p(\theta, x|y) \propto p(\theta)p(x|\theta)p(y|x,\theta)$$

(prior $\times$ SDE model $\times$ observation model)

- where
  - $p(x|\theta)$ product of normal densities for state increments (i.e. $x_{i+1} - x_i$ values)
  - and $p(y|x,\theta)$ product of normal densities at observation times

Posterior distribution

- Let $p(\theta)$ be prior density for parameters
- Posterior distribution is

$$p(\theta, x|y) \propto p(\theta)p(x|\theta)p(y|x, \theta)$$

(prior $\times$ SDE model $\times$ observation model)

- where
    - $p(x|\theta)$ product of normal densities for state increments (i.e. $x_{i+1} - x_i$ values)
    - and $p(y|x, \theta)$ product of normal densities at observation times
- n.b. right hand side is unnormalised posterior $p(\theta, x, y)$

Posterior inference

- Likelihood tractable!
- Can use MCMC, SMC etc
    - Challenging as posterior high dimensional and lots of dependency
    - One approach is to use bridging (next slide)

Posterior inference

- Likelihood tractable!
- Can use MCMC, SMC etc
  - Challenging as posterior high dimensional and lots of dependency
  - One approach is to use bridging (next slide)
- ABC also possible (e.g. Picchini 2014)
  - But lots of hard-to-quantify approximation error

Bridge constructs

- Propose x via a bridge construct
- Use within Monte Carlo inference
- Bridge construct is approx state distribution conditioned on observations

  (usually conditioning just on next observation)

Bridge constructs

- Propose x via a bridge construct
- Use within Monte Carlo inference
- Bridge construct is approx state distribution conditioned on observations

  (usually conditioning just on next observation)
- Derived mathematically
- Various bridges used in practice
- Struggle with highly non-linear paths, large gaps between observations times, low observation variance

Bridge constructs

- Propose x via a bridge construct
- Use within Monte Carlo inference
- Bridge construct is approx state distribution conditioned on observations

  (usually conditioning just on next observation)
- Derived mathematically
- Various bridges used in practice
- Struggle with highly non-linear paths, large gaps between observations times, low observation variance
- Choosing bridges and designing new ones hard work!
- We automate this using **machine learning**
  - "Variational bridge"

Variational inference

Variational inference

- Goal: inference on posterior $p(\theta|y)$
  - Given unnormalised version $p(\theta, y)$

Variational inference

- Goal: inference on posterior $p(\theta|y)$
  - Given unnormalised version $p(\theta, y)$
- Introduce $q(\theta; \phi)$
  - Family of approximate posteriors
  - Controlled by parameters ϕ

Variational inference

- Goal: inference on posterior $p(\theta|y)$
  - Given unnormalised version $p(\theta, y)$
- Introduce $q(\theta; \phi)$
  - Family of approximate posteriors
  - Controlled by parameters ϕ
- Idea: find ϕ giving best approximate posterior

Variational inference

- Goal: inference on posterior $p(\theta|y)$
  - Given unnormalised version $p(\theta, y)$
- Introduce $q(\theta; \phi)$
  - Family of approximate posteriors
  - Controlled by parameters ϕ
- Idea: find ϕ giving best approximate posterior
- Converts Bayesian inference into optimisation problem
- n.b. produces approximation to posterior

Mean field approximation

- Simplest variational approximation
- Assumes $q(\theta; \phi)$ is $N(\mu, \Lambda)$ for diagonal Λ
- So $\phi = (\mu, \Lambda)$
- Helpful computationally
- Makes strong – often unrealistic – assumptions about posterior!

Variational inference

- VI finds ϕ minimising $KL(q(\theta;\phi)||p(\theta|y))$
- Equivalent to maximising ELBO (evidence lower bound),

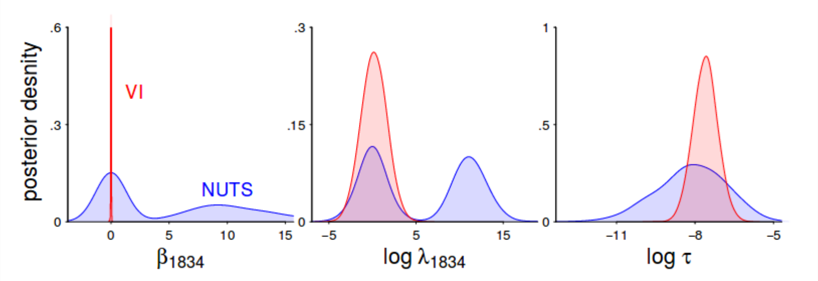
$$\mathcal{L}(\phi) = \mathbb{E}_{\theta \sim q(\cdot;\phi)}\left[\log \frac{p(\theta, y)}{q(\theta;\phi)}\right]$$

  (Jordan, Ghahramani, Jaakkola, Saul 1999)

- Computationally tractable choice

Variational inference

- Optimum q often finds posterior mode well
- But usually overconcentrated!
  (unless family of qs allows very good matches)



(source: Yao, Vehtari, Simpson, Gelman 2018)

Maximising the ELBO

- Several optimisation methods:
    - Variational calculus
    - Parametric optimisation (various flavours)

Maximising the ELBO

- "Reparameterisation trick"
  (Kingma and Welling 2014; Rezende, Mohamed and Wierstra 2014; Titsias and Lázaro-Gredilla 2014)
- Write $\theta \sim q(\cdot; \phi)$ as $\theta = g(\epsilon, \phi)$ where:
  - g inverible function
  - ϵ base random variable e.g. $N(0, \mathrm{I})$
- Mean field case: $\theta = \mu + \Lambda^{1/2}\epsilon$

Maximising the ELBO

- "Reparameterisation trick"
  (Kingma and Welling 2014; Rezende, Mohamed and Wierstra 2014; Titsias and Lázaro-Gredilla 2014)
- Write $\theta \sim q(\cdot; \phi)$ as $\theta = g(\epsilon, \phi)$ where:
  - g inverible function
  - ϵ base random variable e.g. $N(0, \mathrm{I})$
- Mean field case: $\theta = \mu + \Lambda^{1/2}\epsilon$
- Optimisation possible using:
  - Stochastic gradient descent
  - Automatic differentiation

Variational inference for SDEs

Variational inference for SDEs

- We want posterior $p(\theta, x|y)$ for SDE model
- Define
$$q(\theta, x; \phi) = q(\theta; \phi_\theta) q(x|\theta; \phi_x)$$
- We use mean-field approx for $q(\theta; \phi_\theta)$
- Leaves choice of $q(x|\theta; \phi_x)$

Variational inference for SDEs

- $q(x|\theta; \phi_x)$ should approximate $p(x|\theta, y)$
- SDE theory suggests this itself obeys a SDE
  (see e.g. Rogers and Williams 2013)
- But with different drift and diffusion to original SDE
- No nice tractable form, so we try to learn from simulations

Variational approximation to states

- We define $q(x|\theta;\phi_x)$ by a discretised SDE
- We let drift $\tilde{\alpha}$ and diffusion $\tilde{\beta}$ depend on:
  - Parameters θ
  - Most recent x and t values
  - Details of next observation
- To get flexible parametric functions we use neural network
- ϕ_x is neural network parameters (weights and biases)

Variational approximation to states

- Drift and diffusion calculated from neural network
- Used to calculate x_1
- Fed back into same neural network to get next drift and diffusion
- ...
- **Recurrent neural network** structure
- V flexible (but tricky to scale up)

Variational approximation to states

- Typically we take

$$x_{i+1} = x_i + \tilde{\alpha}\Delta t + \sqrt{\tilde{\beta}\Delta t}\, z_{i+1}$$

Variational approximation to states

- Typically we take

$$x_{i+1} = x_i + \tilde{\alpha}\Delta t + \sqrt{\tilde{\beta}\Delta t} z_{i+1}$$

- Sometimes want to ensure non-negativity of xs
- So we use

$$x_{i+1} = h\left(x_i + \tilde{\alpha}\Delta t + \sqrt{\tilde{\beta}\Delta t} z_{i+1}\right)$$

- Where h outputs non-negative values e.g. softplus function

$$h(z) = \log(1 + e^z)$$

Algorithm summary

Initialise ϕ_θ, ϕ_x

Begin loop

Sample θ from $q(\theta; \phi_\theta)$ (independent normals)

Sample x from $q(x; \theta, \phi_x)$ (run RNN)

Update ϕ_θ, ϕ_x by stochastic optimisation

End loop

Algorithm summary

Initialise ϕ_θ, ϕ_x

Begin loop

Sample θ from $q(\theta; \phi_\theta)$ (independent normals)

Sample x from $q(x; \theta, \phi_x)$ (run RNN)

Update ϕ_θ, ϕ_x by stochastic optimisation

End loop

(n.b. can use larger Monte Carlo batch size)

Example

Lotka-Volterra example

- SDE model from Golightly and Wilkinson (2011)
- Prey population at time t is U_t
- Predator population at time t is V_t
- Drift
$$\alpha(X_t, \theta) = \begin{pmatrix} \theta_1 U_t - \theta_2 U_t V_t \\ \theta_2 U_t V_t - \theta_3 V_t \end{pmatrix}$$
- Diffusion
$$\beta(X_t, \theta) = \begin{pmatrix} \theta_1 U_t + \theta_2 U_t V_t & -\theta_2 U_t V_t \\ -\theta_2 U_t V_t & \theta_3 V_t + \theta_2 U_t V_t \end{pmatrix}$$
- Parameters:
    - θ_1 controls prey growth
    - θ_2 controls predator growth by consuming prey
    - θ_3 controls predator death

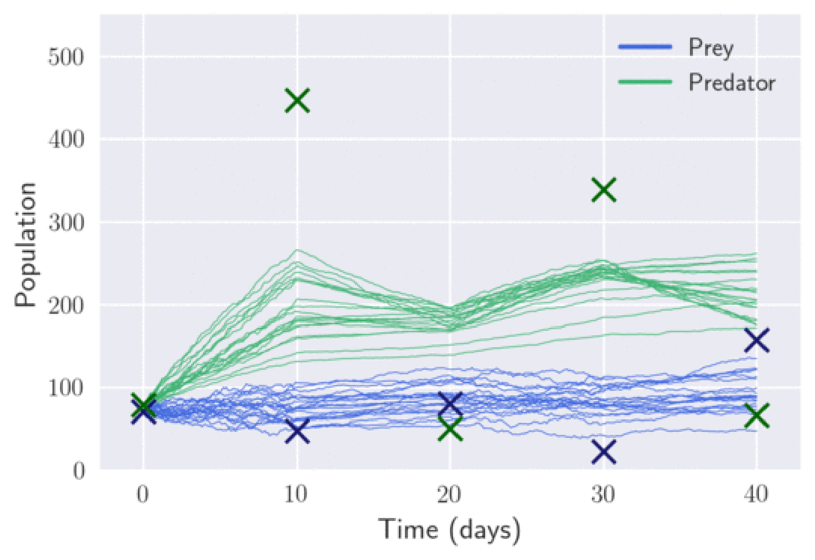
Settings - model

- Simulated data at times $0, 10, 20, 30, 40$
- IID priors: $\log\theta_i \sim N(0, 3^2)$ for $i = 1, 2, 3$
- Discretisation time step $\Delta\tau = 0.1$
- Observation variance $\Sigma = \mathrm{I}$ - small relative to typical population sizes
- Challenging scenario:
  - Non-linear state paths
  - Small observation variance
  - Long gaps between observations

Settings - variational inference

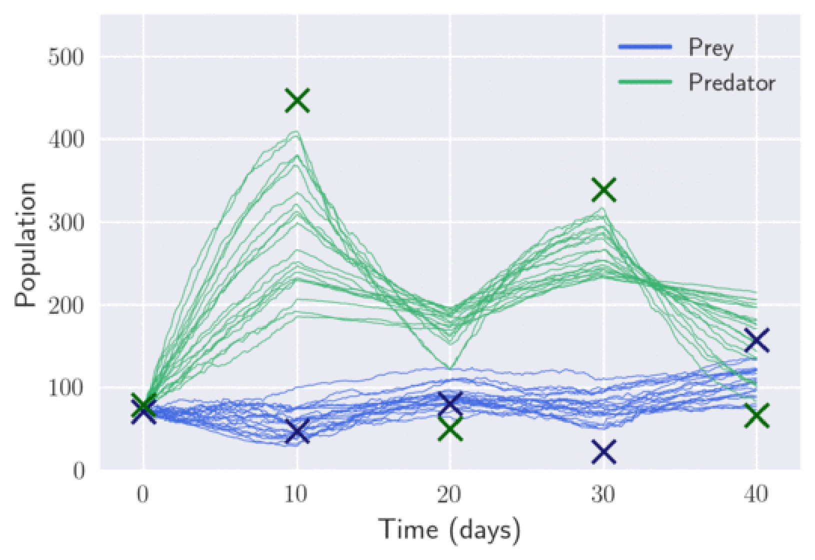
- Batch size 50 for gradient estimate
- 4 layer neural network (20 ReLU units / layer)
- Softplus transformation to avoid proposing negative population levels
- Various methods to avoid numerical problems in training

Results

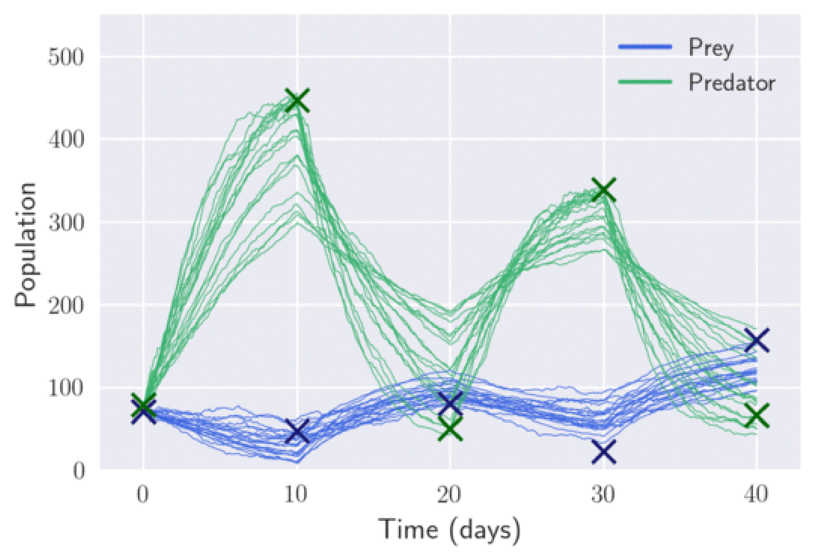


Prey
Predator
500
400
300
200
100
0
Population
0
10
20
30
40
Time (days)

Results

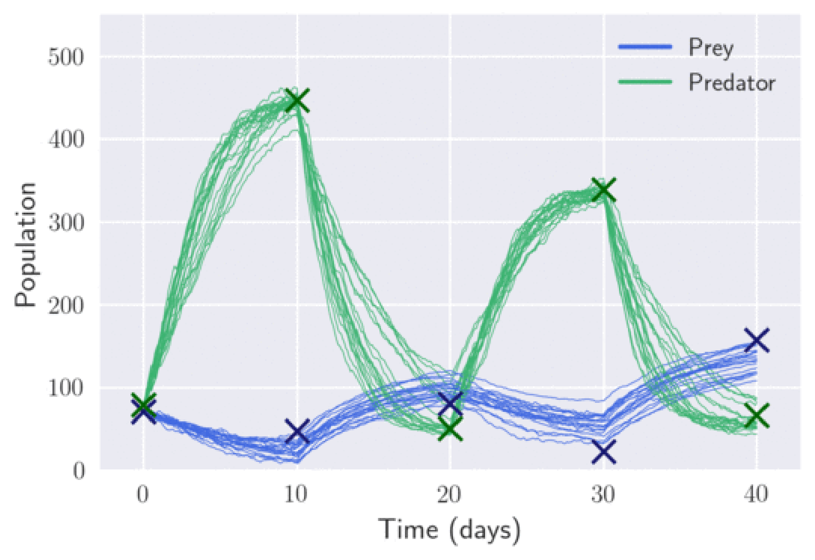

Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



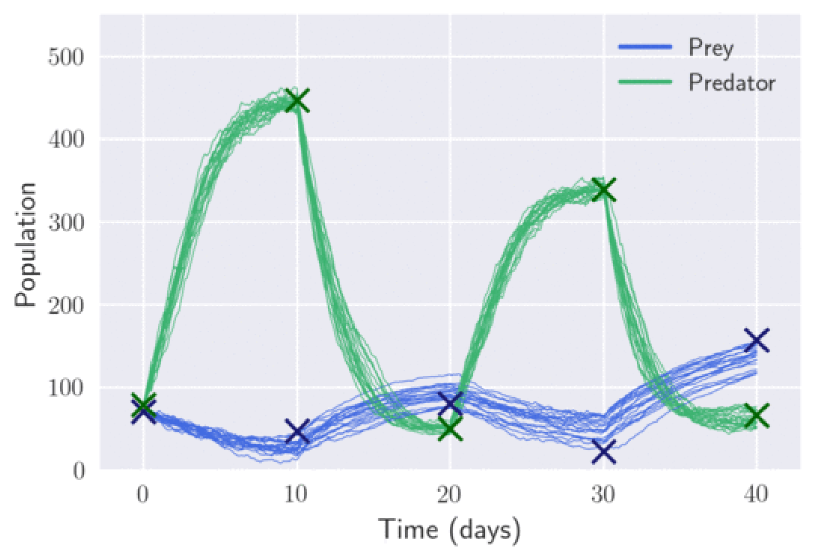
Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

Results

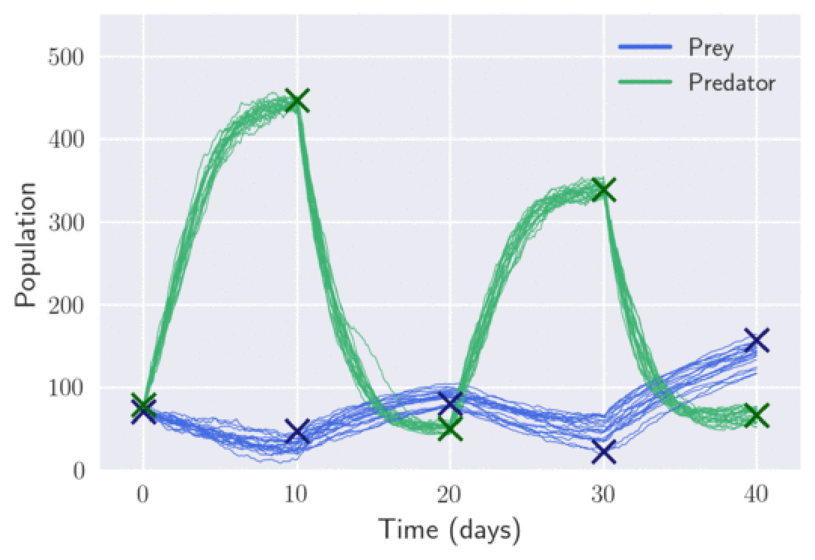


Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results

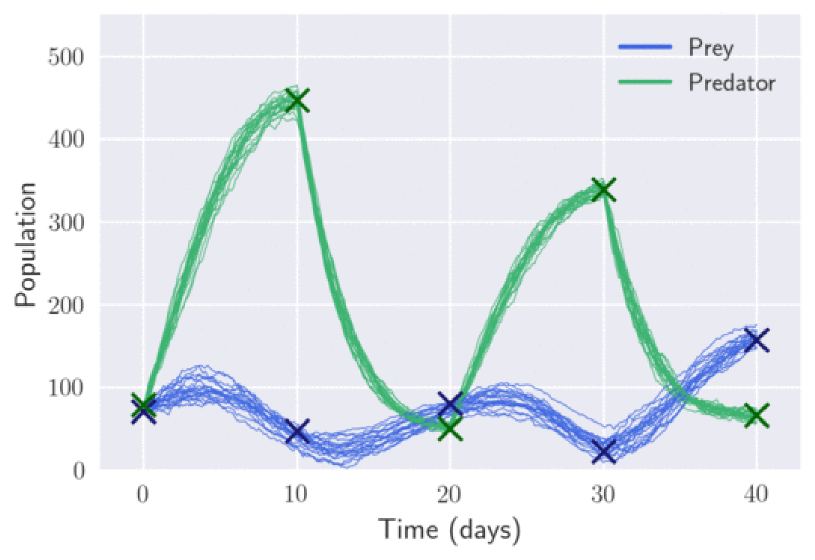

Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

Results



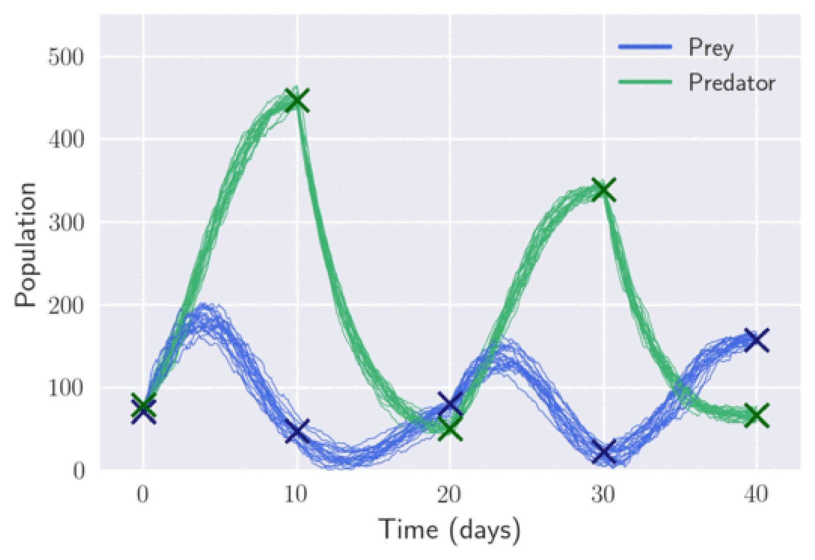
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



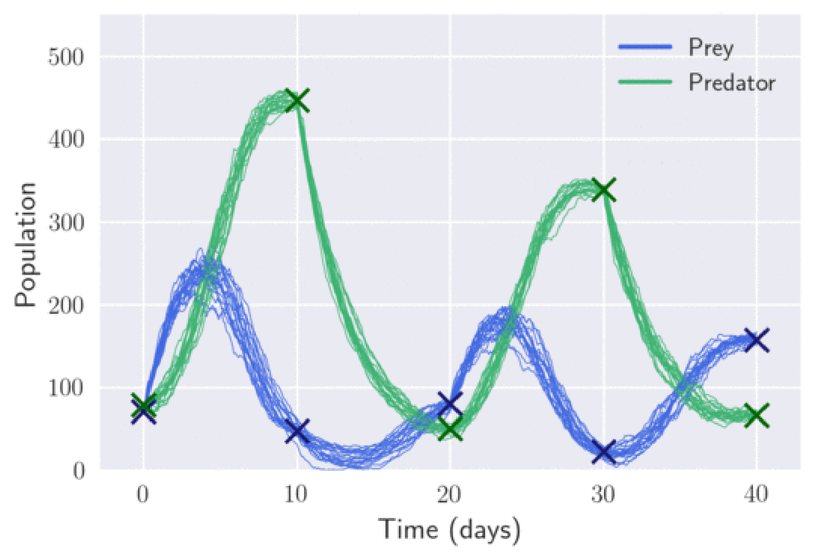
Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results

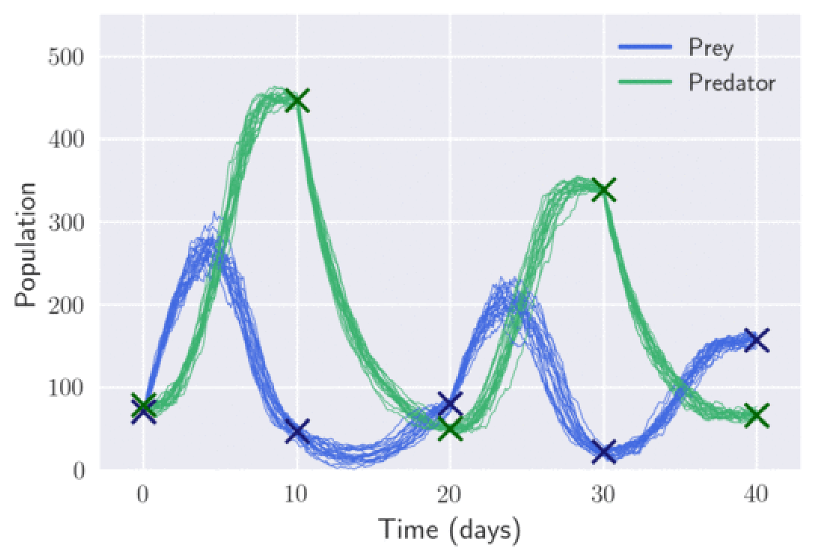


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

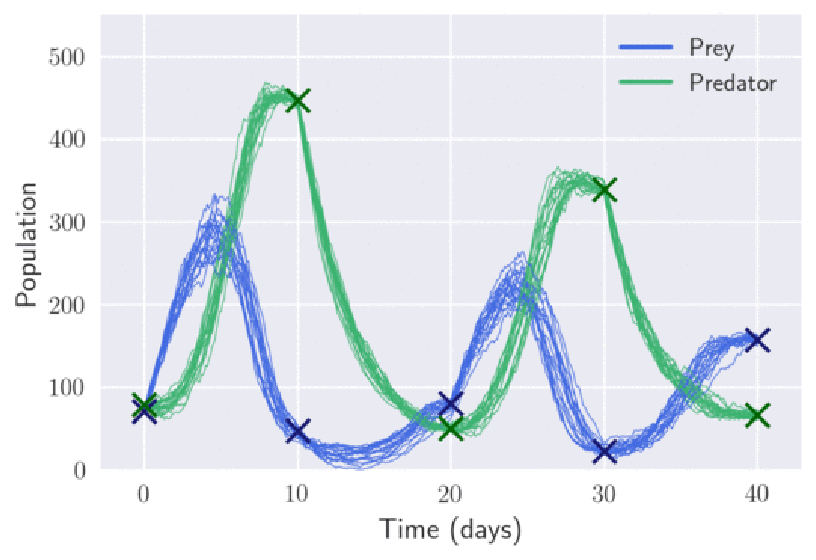
Results


Prey
Predator
500
400
300
200
100
0
Population
0
10
20
30
40
Time (days)

Results

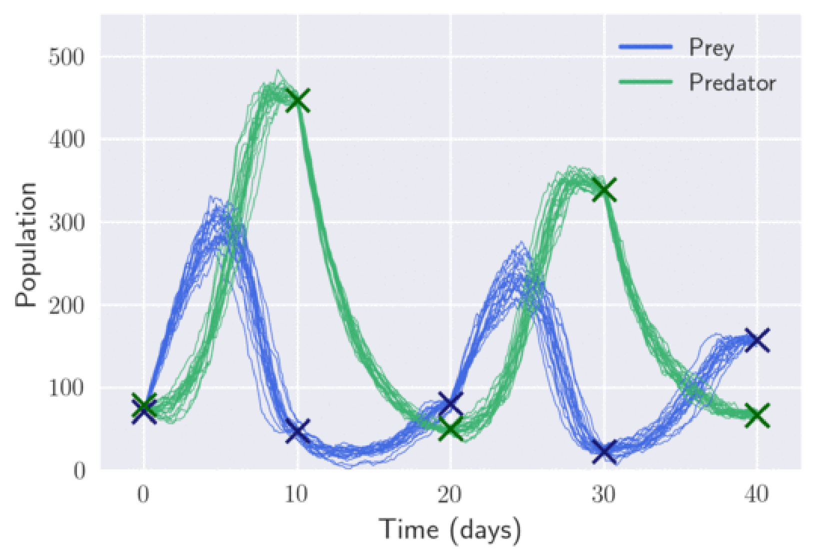

Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

Results

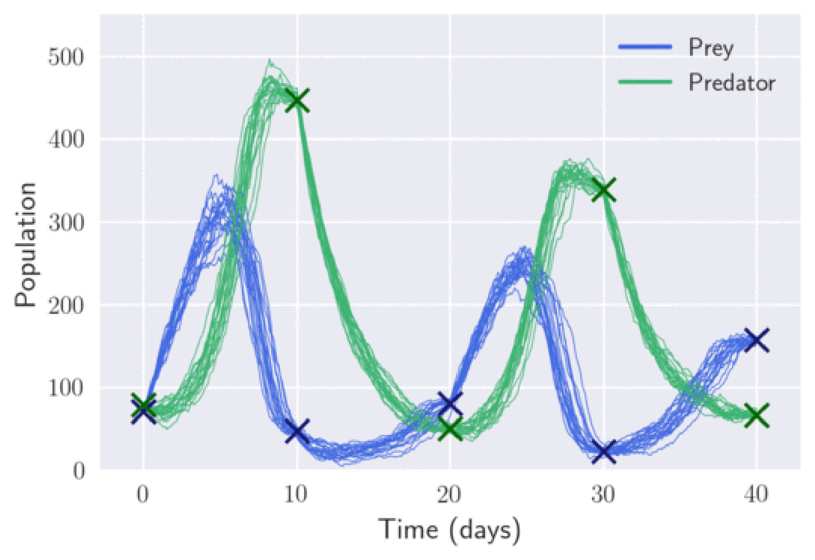


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

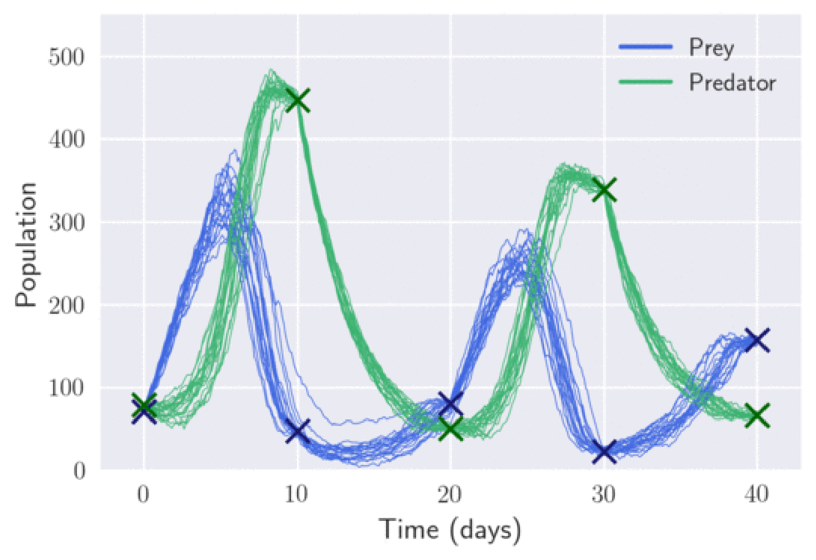
Results


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

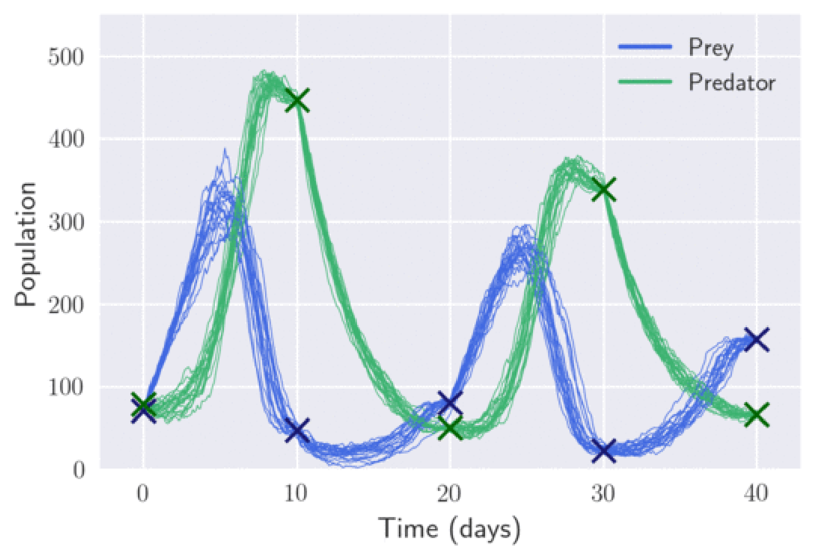
Results


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

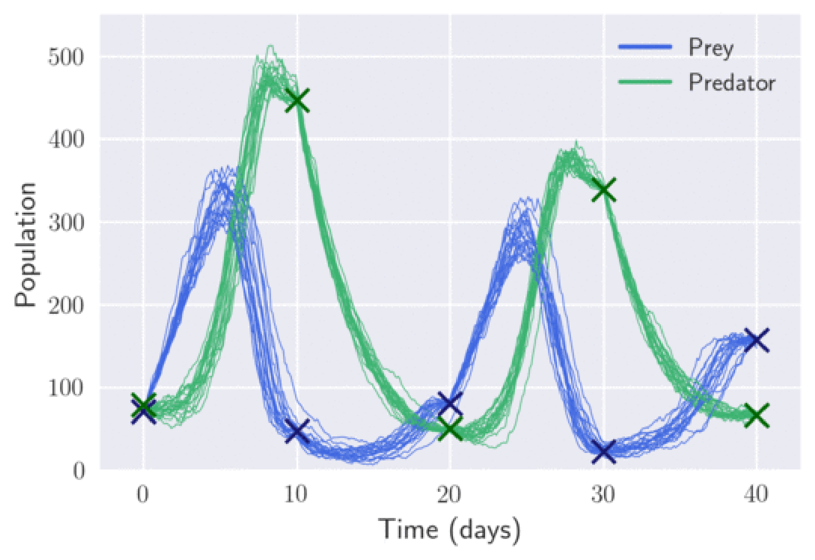

Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

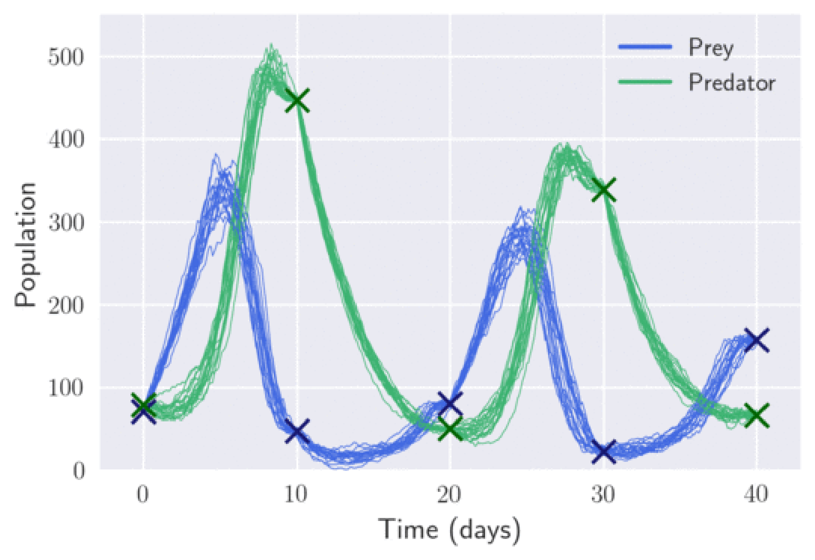


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

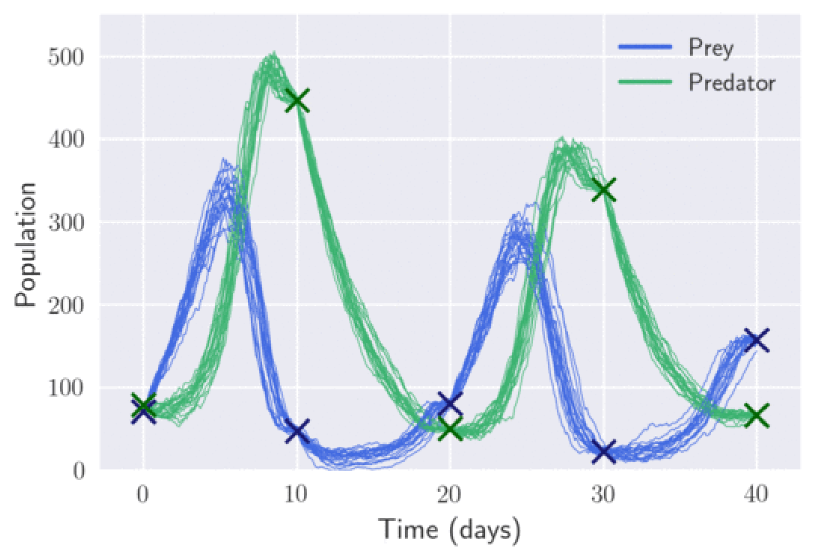

Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results



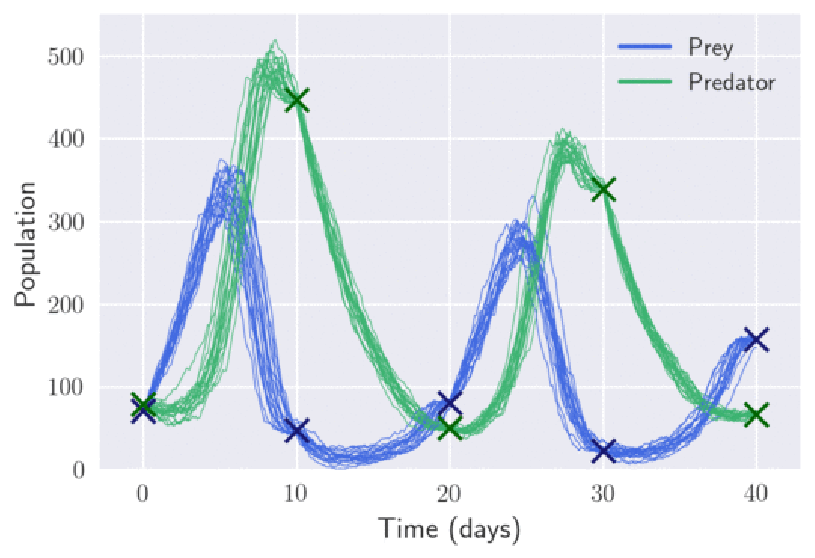
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

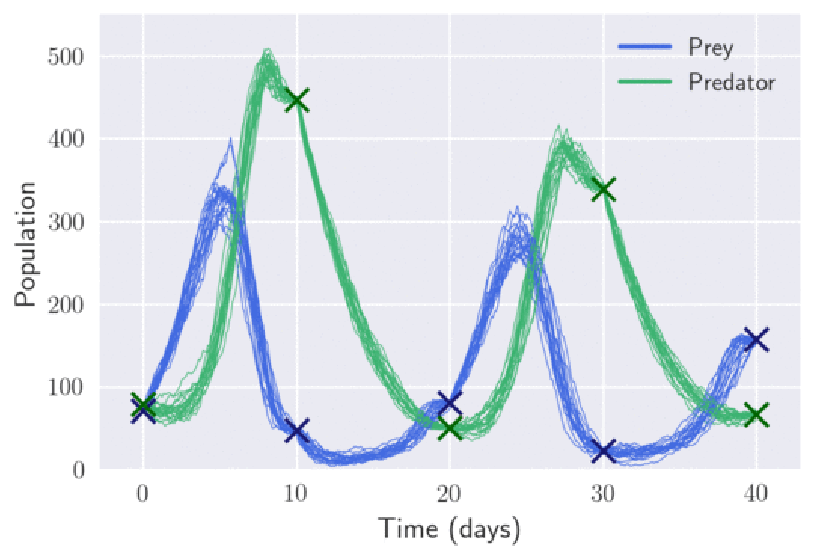


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

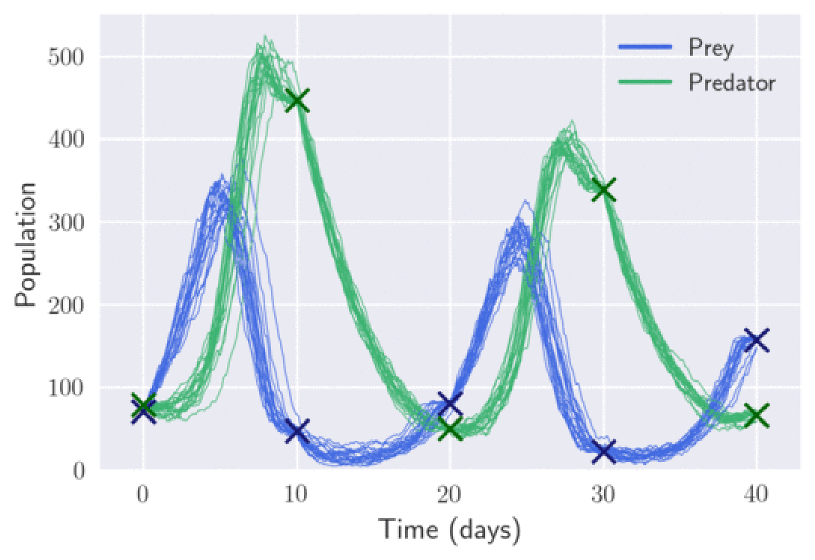
Results


Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

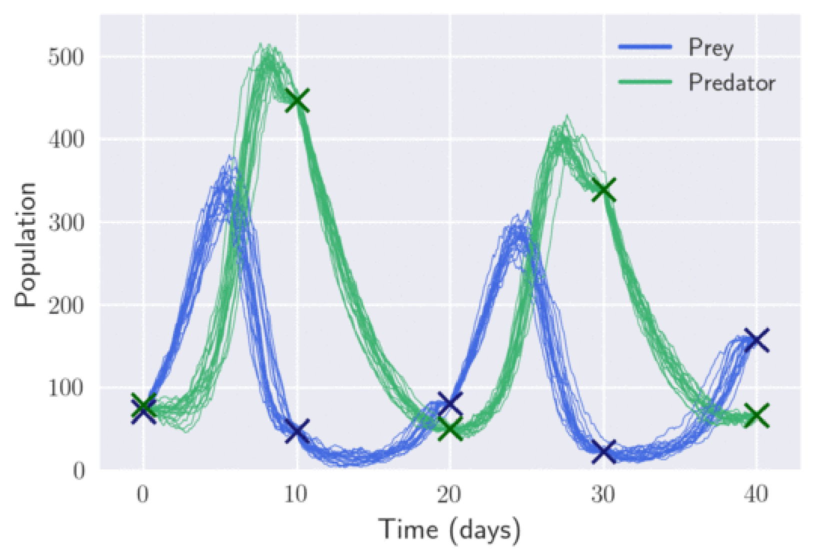
Results


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

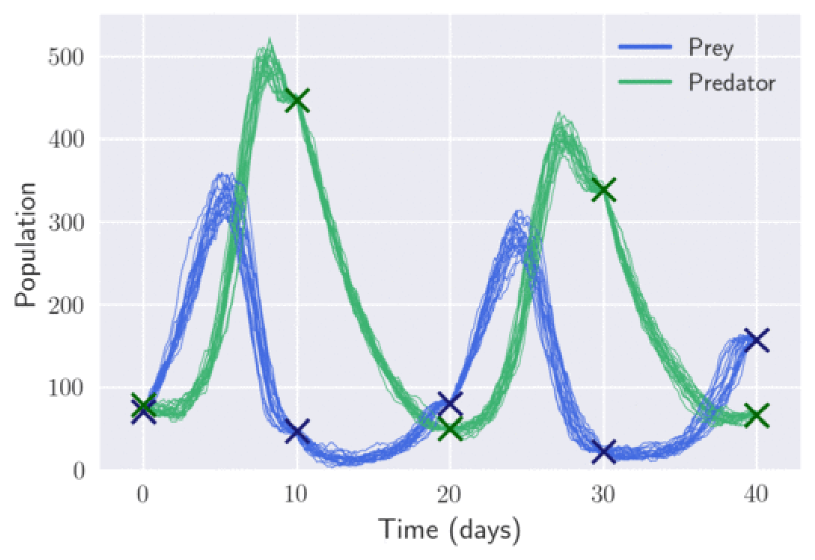

Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results



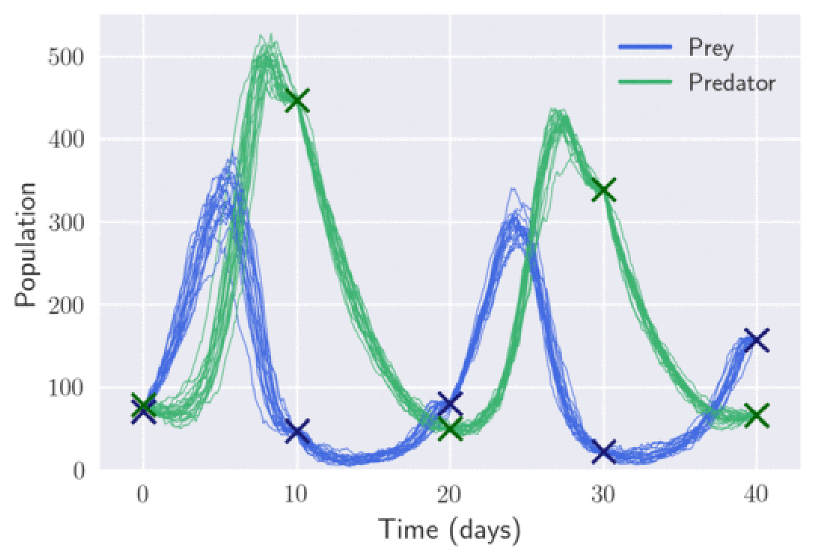
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

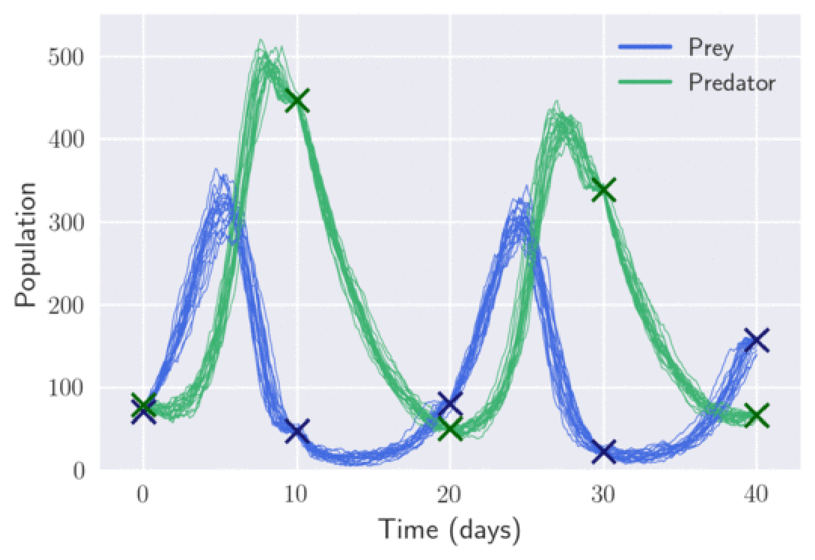


Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

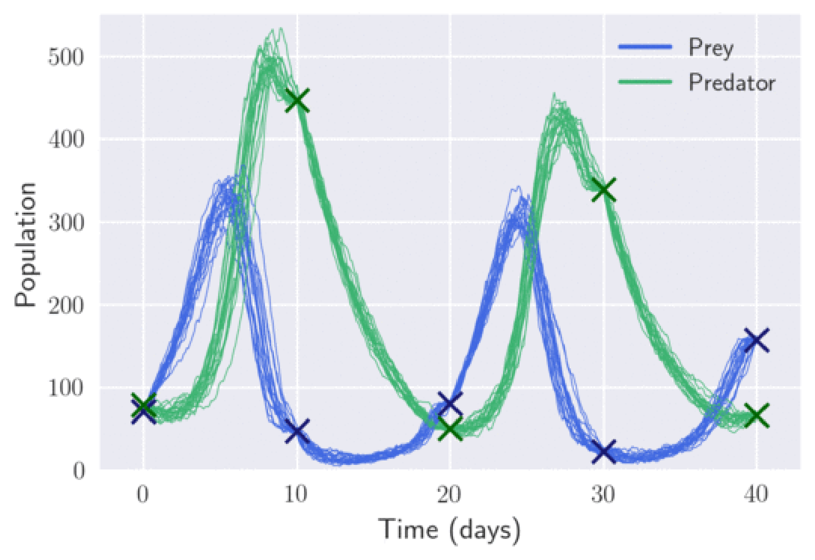

Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results



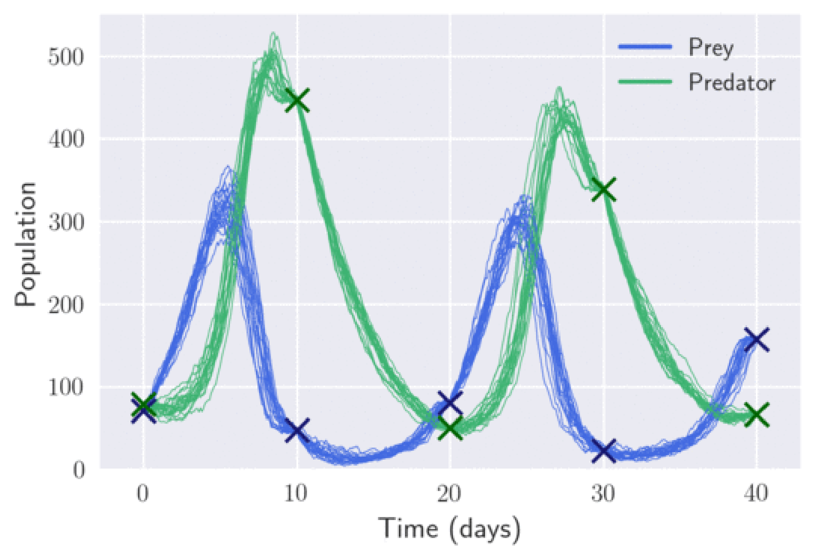
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



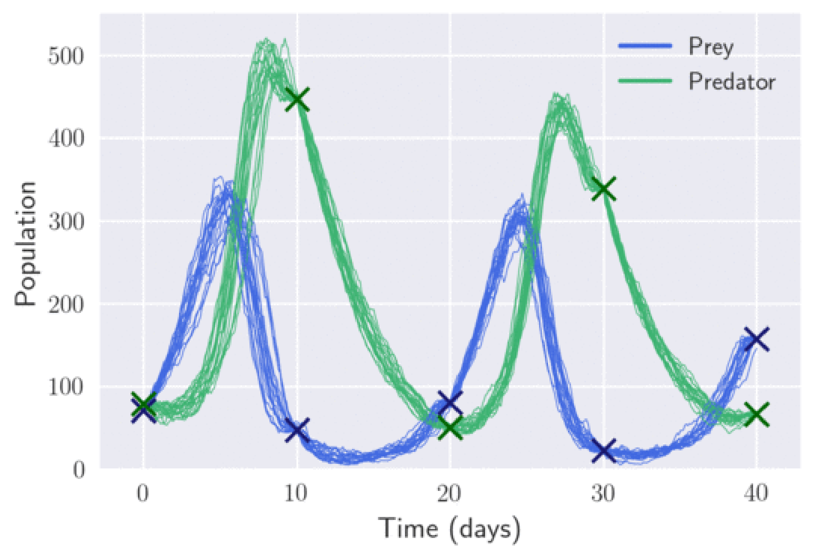
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



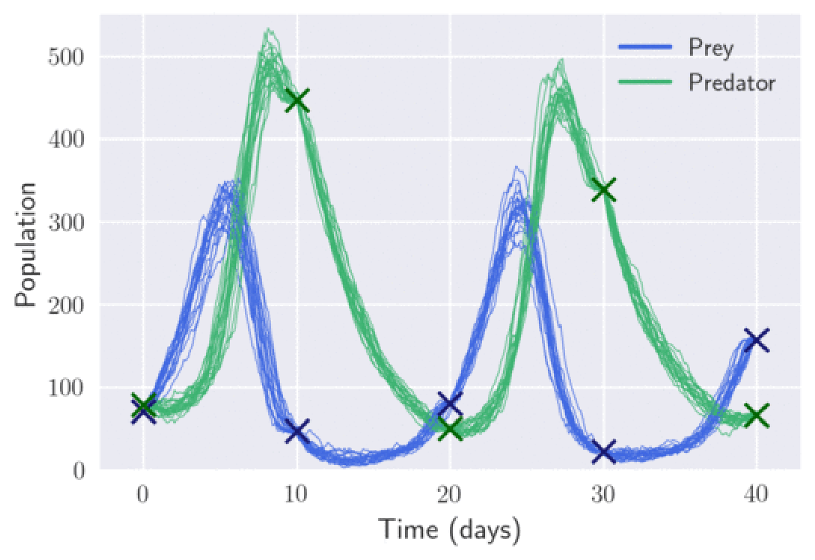
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



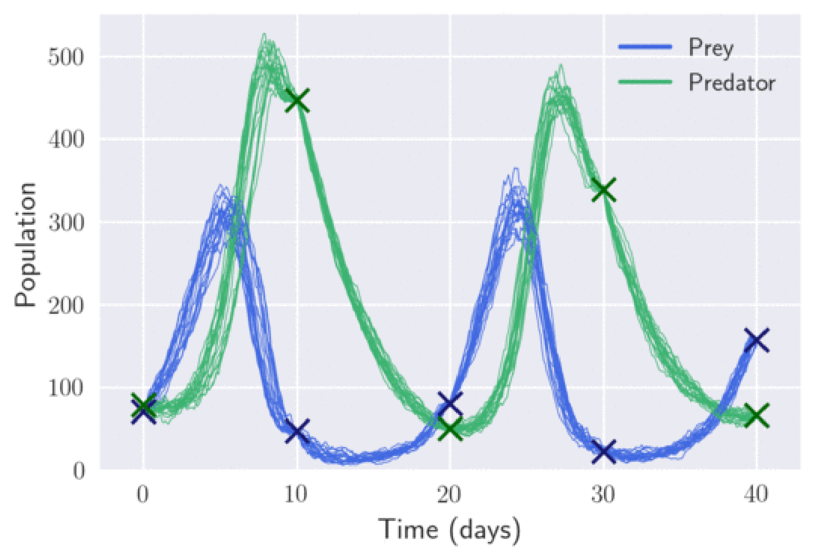
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



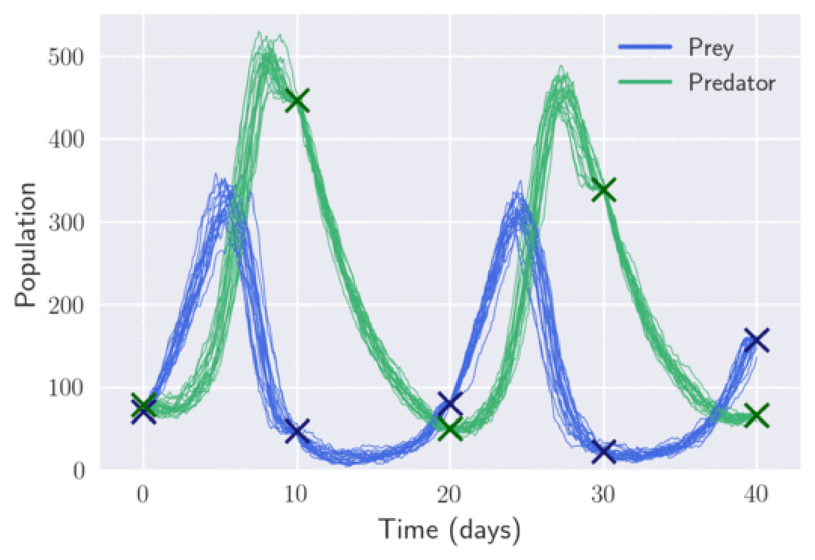
Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

Results



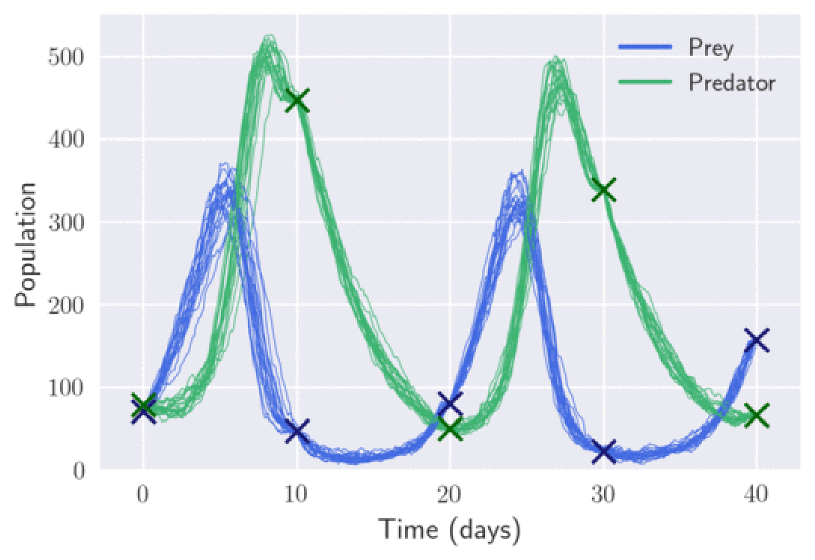
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



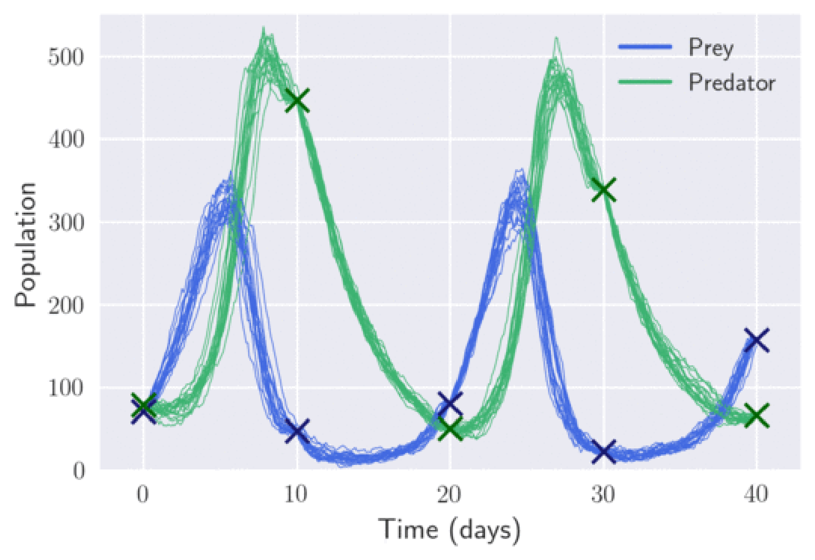
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



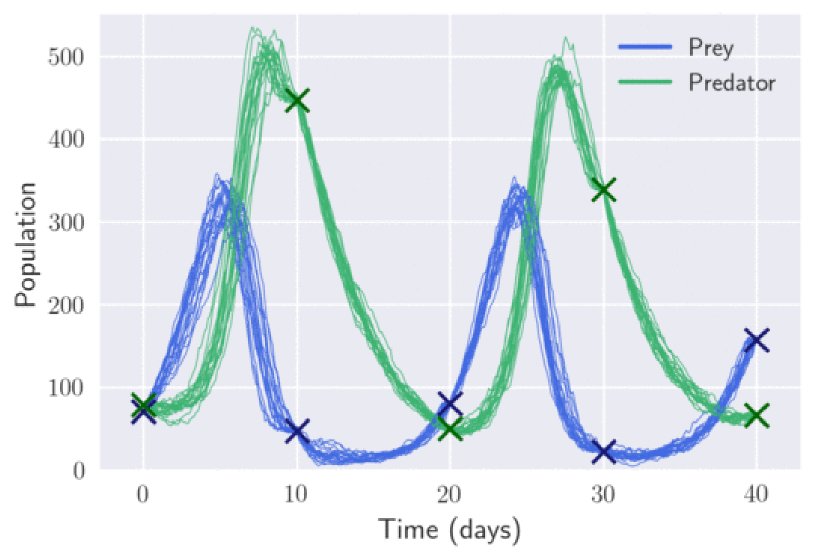
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



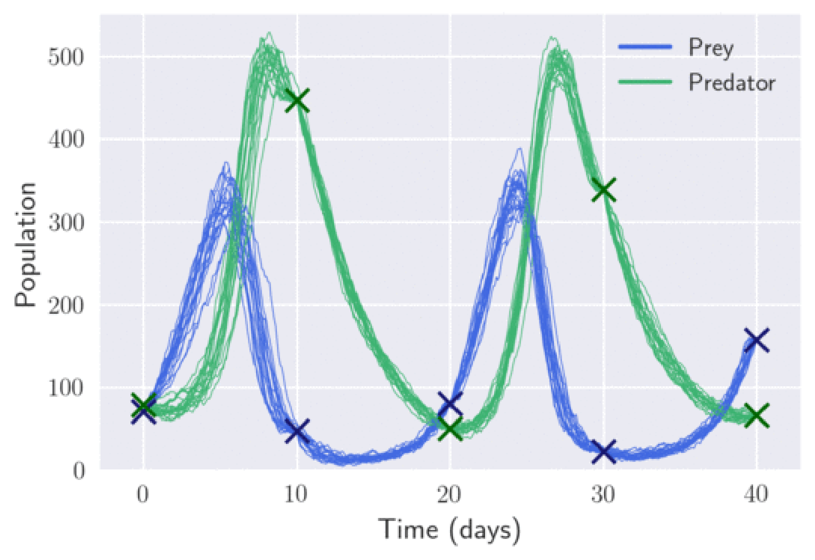
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



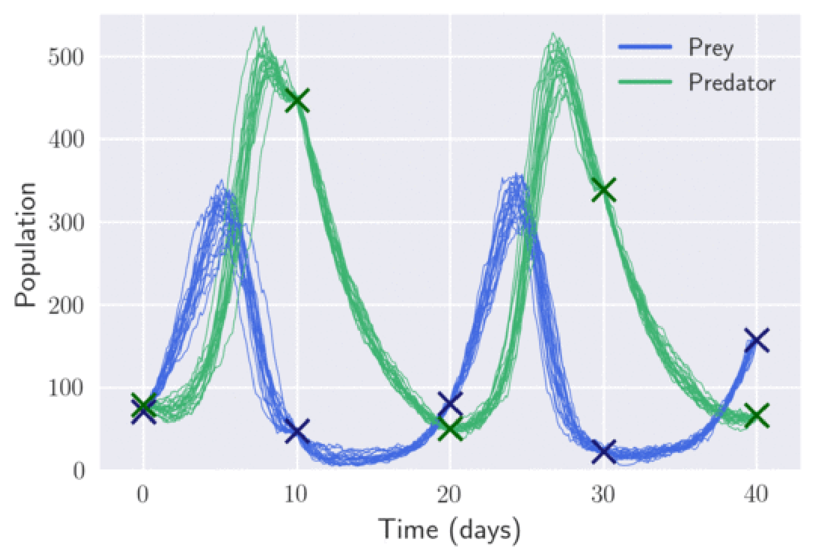
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results



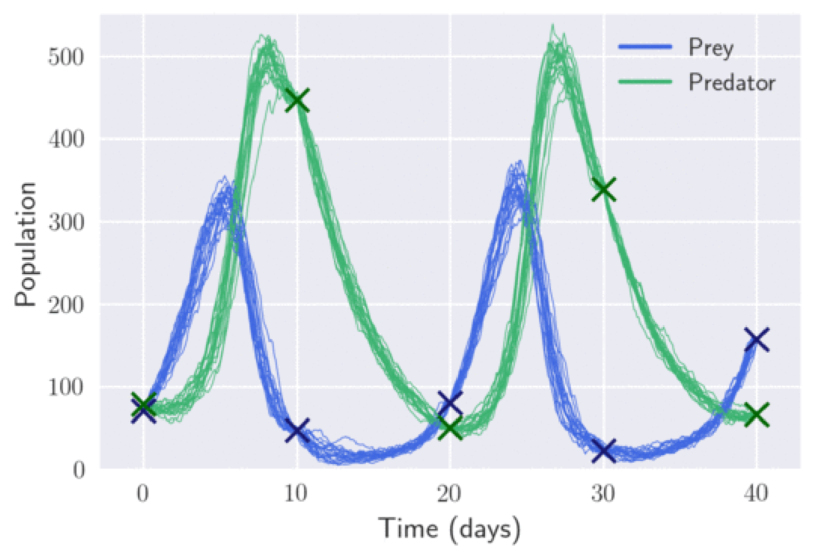
Prey
Predator
Population
Time (days)
0
100
200
300
400
500
0
10
20
30
40

Results

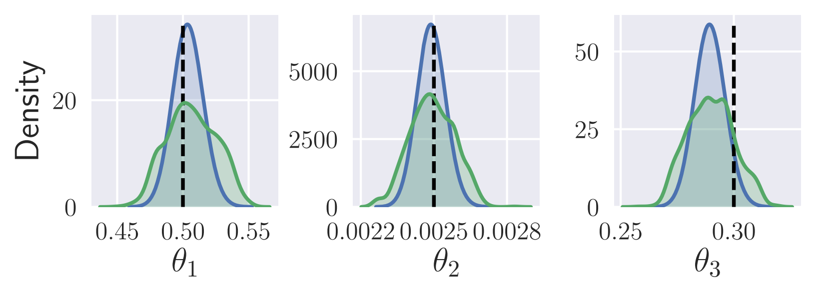


Prey
Predator
Population
500
400
300
200
100
0
0
10
20
30
40
Time (days)

Results


Prey
Predator
Population
0
100
200
300
400
500
0
10
20
30
40
Time (days)

Parameter inference results



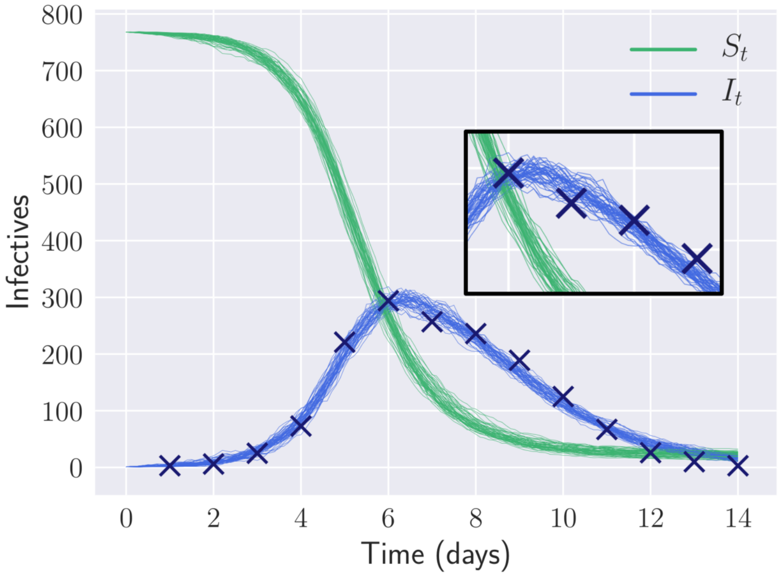
- Black: true parameter values
- Blue: variational output
- Green: importance sampling (shows over-concentration)

Computing time: ≈ 2 hours on a desktop PC

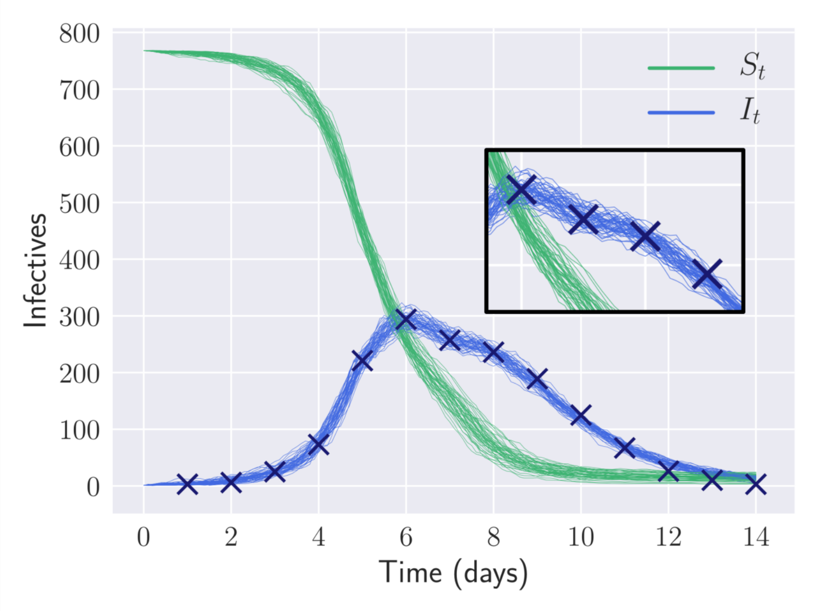
Epidemic example

- Boarding school data
- We look at:
    - SIR SDE model of Fuchs (2013)
    - Version with time-varying infection rate

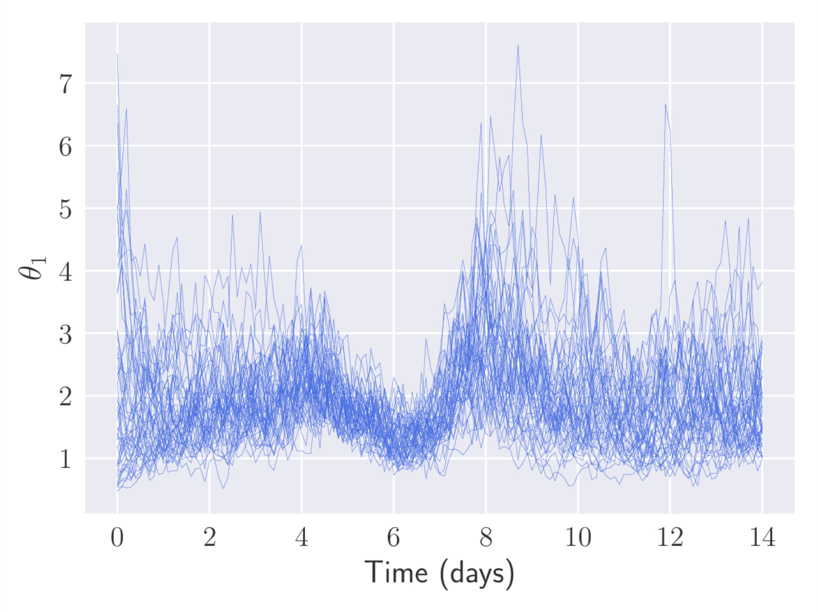
Epidemic example - constant infection rate



Epidemic example - varying infection rate



Epidemic example - varying infection rate



Conclusion

Summary

- Variational approach to approx Bayesian inference for SDEs
- Modest tuning requirements (compated to MCMC)
- Results in a few hours on a desktop PC
- Good estimation of posterior mode

Current/future work

- Faster inference (using alternatives to RNNs)
- Big data - long or wide
- Other models e.g. state space models, MJPs
- Model comparison/improvement
- Real applications - suggestions very welcome!