

Predator–Prey Population Dynamics: the Lotka–Volterra model in Stan

Bob Carpenter

28 January 2018

- Data: Lynx and Hare Pelts in Canada
- Mechanistic Model: The Lotka–Volterra Equations
- Statistical Model: Prior Knowledge and Unexplained Variation
- Coding the Model: Stan Program
- Solving the Inverse Problem: Bayesian Inference in Stan
- Conclusion: What are the Population Dynamics?
- Exercises and Extensions

Abstract

Lotka (1925) and Volterra (1926) formulated parameteric differential equations that characterize the oscillating populations of predators and prey. A statistical model to account for measurement error and unexplained variation uses the deterministic solutions to the Lotka–Volterra equations as expected population sizes. Stan is used to encode the statistical model and perform full Bayesian inference to solve the inverse problem of inferring parameters from noisy data. The model is fit to Canadian lynx¹ and snowshoe hare² populations between 1900 and 1920, based

¹ *Predator*: Canadian lynx

on the number of pelts collected annually by the Hudson's Bay Company. Posterior predictive checks for replicated data show the model fits this data well. Full Bayesian inference may be used to estimate future (or past) populations.

Data: Lynx and Hare Pelts in Canada

The species of interest in this case study are

- snowshoe hares, an herbivorous cousin of rabbits, and
- Canadian lynxes, a feline predator whose diet consists largely of snowshoe hares.

Rogers (2011) writes about the long history of population dynamics,

The rise and fall in numbers of snowshoe hares and Canada lynx was observed more than two hundred years ago by trappers working for Hudson's Bay Company, which was once heavily involved in the fur trade. In the early 20th century, records of the number of lynx and hare pelts traded by Hudson's Bay were analyzed by biologist Charles Gordon Hewitt.

Following up on the original source, Hewitt (1921) provides plots of the number of pelts collected by the Hudson's Bay Company, the largest fur trapper in Canada, between the years of 1821 and 1914.³ Hewitt's discussion ranges over many natural factors affecting the



© 2009, Keith Williams, CC-BY 2.0

² Prey: snowshoe hare



© 2013, D. Gordon E. Robertson, CC-BY SA 3.0

³ Graph from (Hewitt 1921) showing the numbers of pelts captured by the Hudson's

population sizes, such as plagues, migrations, and weather-related events. Hewitt even discusses measurement confounders such as the fact that lynx are easier to trap when they are hungry and weak, which is correlated with a relative decline in the hare population. The models we consider here for illustrative purposes will not consider any of these factors, though they could be extended to do so through the usual strategy of the inclusion of covariates (see the exercises). Hewitt also discusses many other species; hares and lynxes occupy only a small part of a single chapter.

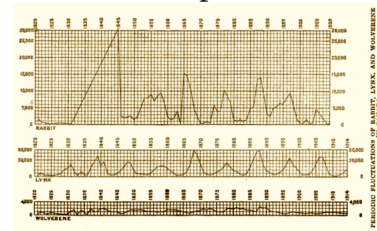
Howard (2009) provides numerical data for the number of pelts collected by the Hudson's Bay Company in the years 1900–1920, which we have included in comma-separated value (CSV) form in the source repository with the case study.

```
lynx_hare_df <-  
  read.csv("hudson-bay-lynx-hare.csv",  
           comment.char="#")
```

	year	species	pelts in thousands
1	1900	Lynx	4.0
2	1901	Lynx	6.1
21	1920	Lynx	8.6
22	1900	Hare	30.0
41	1919	Hare	16.2
42	1920	Hare	24.7

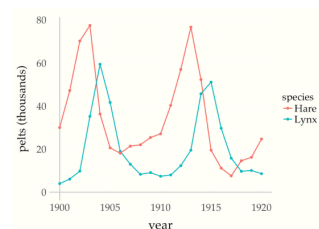
The plot makes it clear that the spikes in the lynx population lag those in the hare population. When the populations are plotted against one another over time, the population dynamics orbit in an apparently stable pattern. Volterra

Bay Company. The fluctuations are irregular and the linear growth in rabbits after 1830 appears to be an artifact of interpolation.



© Scribner's Sons 1921

Example rows (with their indexes) from the long-form data frame for number of pelts taken by the Hudson's Bay Company in the years 1900 to 1920 (in thousands).



Plot of the number of lynx and hare pelts (in thousands)

(1926) recognized that these population oscillations could be modeled with a pair of differential equations similar to that used to describe springs.

Mechanistic Model: The Lotka-Volterra Equations

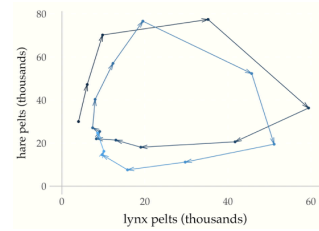
The Lotka-Volterra equations (Lotka 1925; Volterra 1926, 1927) are based on the assumptions that

- the predator population intrinsically shrinks,
- the prey population intrinsically grows,
- a larger prey population leads to a larger predator population, and
- a larger predator population leads to a smaller prey population.

More specifically, the rate of growth of the prey population is proportional to the size of the prey population, leading to exponential growth if unchecked. The prey population simultaneously shrinks at a rate proportional to the size of the product of the prey and predator populations. For the predator species, the direction of growth is reversed. The predator population shrinks at a rate proportional to its size and grows at a rate proportional to the product of its size and the prey population's size.

Together, these dynamics lead to a cycle of rising and falling populations. With a low lynx population, the hare population grows. As the

collected by the Hudson's Bay Company between 1900 and 1920.



Plot of the number of pelts collected for lynx versus hares from 1900 to 1920. This plot is similar to that of the dynamics of a spring in phase space (i.e., position vs. momentum).

hare population grows, it allows the lynx population to grow. Eventually, the lynx population is large enough to start cutting down on the hare population. That in turn puts downward pressure on the lynx population. The cycle then resumes from where it started.

The Lotka–Volterra equations (Volterra 1926, 1927; Lotka 1925) are a pair of first-order, ordinary differential equations (ODEs) describing the population dynamics of a pair of species, one predator and one prey.⁴

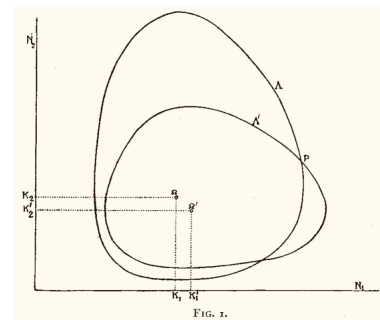
- $u(t) \geq 0$ is the population size of the prey species at time t , and
- $v(t) \geq 0$ is the population size of the predator species.

Volterra modeled the temporal dynamics of the two species (i.e., population sizes over times) in terms of four parameters, $\alpha, \beta, \gamma, \delta \geq 0$, as

$$\begin{aligned} \frac{d}{dt}u &= (\alpha - \beta v)u &= \alpha u - \beta uv \\ \frac{d}{dt}v &= (-\gamma + \delta u)v &= -\gamma v + \delta uv \end{aligned}$$

As usual in writing differential equations, $u(t)$ and $v(t)$ are rendered as u and v to simplify notation. The factor α is the growth rate of the prey population, whereas β is the rate of shrinkage relative to the product of the population sizes. The factor γ is the shrinkage rate of the predator population and δ is the growth rate of the predator population as a factor of the product of the population sizes.

⁴ Solutions to the Lotka-Volterra equations for predator and prey population sizes. The temporal trajectory of populations forms a stable orbit when plotted in two dimensions (Volterra 1926).



© 1926, Nature Publishing Group

Positivity constraint and extinction

As measures of population sizes, the values of $u(t)$ and $v(t)$ must be non-negative.⁵ Nothing in the differential equations here explicitly enforces positivity. Nevertheless, as long as the initial populations are non-negative, i.e., $u(0) \geq 0$ and $v(0) \geq 0$, the values $u(t)$ and $v(t)$ for other times t must also be non-negative. This is because the rate of change in each population is a factor of the population size itself.

Although a population following the Lotka-Volterra dynamics will never become extinct,⁶ it may become arbitrarily small. In reality, random events such as accidents can cause extinction events as can not so random events such as extensive hunting. Therefore, the Lotka-Volterra model should not be expected to adequately model the dynamics of small populations.

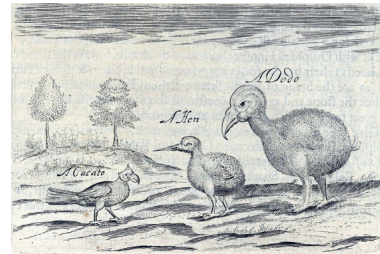
Behavior in the limit

One way to understand systems of equations is to consider their limiting behavior. In this case, there are four behaviors, falling in two broad classes.

1. If both population sizes are initially positive, the populations will oscillate in a fixed pattern indefinitely, remaining positive.
2. If both population sizes are initially zero, the population sizes will remain zero.
3. If the predator population size is zero

⁵ As in much of ecology, undead are frowned upon.

⁶ The dodo (right), in 1682, before the population became extinct.



1682, Sir Thomas Herbert, public domain.

and the prey population size positive, the predator population size remains zero and the prey population grows without bound.

4. If the predator population size is positive and the prey population size zero, the prey population size remains zero while the predator population shrinks toward zero size (i.e., it asymptotes at zero).

Statistical Model: Prior Knowledge and Unexplained Variation

Solving the inverse problem

For a given legal value of the model parameters and initial state, the Lotka-Volterra model predicts population dynamics into the future (and into the past). But given noisy data about population dynamics, how do we solve the inverse problem, that of inferring the values of model parameters consistent with the data? The general approach in Bayesian statistics is somewhat counterintuitive, as it involves formulating the forward model then using general principles to solve the inverse problem.

Specifically, a Bayesian model requires a mathematical model of what we know about the parameters (i.e., a prior) and a model of what we know about the data generating process given the parameters (i.e., a sampling distribution).⁷

Mathematically, a prior density $p(\theta)$ over the sequence of parameters θ encapsulates our

⁷ The choice of prior is no more or less “subjective” than the choice of sampling distribution—both are

knowledge of the parameters before seeing the data. A sampling distribution,⁸ which may have a continuous, discrete or mixed probability function, $p(y|\theta)$ characterizes the distribution of observable data y given parameters θ .

Bayes's rule gives us a general solution to the inverse problem, expressing the posterior $p(\theta|y)$ in terms of the prior $p(\theta)$ and likelihood $p(y|\theta)$.⁹

Stan provides a form of Markov chain Monte Carlo (MCMC) sampling that draws a sample $\theta^{(1)}, \dots, \theta^{(M)}$ from the posterior to use for computational inference.¹⁰ Posterior quantities of interest may be expressed as derived random variables using functions $f(\theta)$ of parameters. Such functions may be used for posterior expectation calculations such as parameter estimates that minimize expected square error when $f(\theta) = \theta$, or event probabilities such as the probability of the hare population falling below some fraction of the lynx population, when $f(\theta)$ is some indicator function.¹¹

Measurement Error and Unexplained Variation

The Lotka-Volterra model is deterministic. Given the system parameters and the initial conditions, the solutions to the equations are fully determined. Population measurements are not so well behaved that they exactly follow the model; there is residual, unexplained variation, as well as measurement error.

There are factors that impact predator and prey population size other than the current

mathematical approximations to some process of interest and as such must be validated for utility.

⁸ The sampling distribution $p(\theta|y)$ is called a likelihood function when considered as a function $\mathcal{L}(\theta)$ of the parameters θ for fixed data y .

⁹ Bayes's rule for parameters θ and observed data y is

$$\begin{aligned} p(\theta | y) &= \frac{p(\theta, y)}{p(y)} \\ &= \frac{p(y|\theta) p(\theta)}{p(y)} \\ &= \frac{p(y|\theta) p(\theta)}{\int_{\Theta} p(y|\theta) p(\theta) d\theta} \\ &\propto p(y|\theta) p(\theta). \end{aligned}$$

¹⁰ The matrix of $\theta^{(m)}$ values (parameter by draw) is what is returned by the `extract()` function in Stan.

¹¹ The convergence result (as well as error bounds) follows from the MCMC central limit theorem when $\theta^{(m)}$ are drawn according to $p(\theta|y)$ with an appropriate MCMC algorithm,

$$\begin{aligned} \mathbb{E}[f(\theta) | y] &= \int_{\Theta} f(\theta) p(\theta|y) d\theta \\ &= \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M f(\theta^{(m)}) \\ &\approx \frac{1}{M} \sum_{m=1}^M f(\theta^{(m)}) \quad \text{for so} \end{aligned}$$

population size. There are variable environmental effects, such as weather, which will vary from season to season and year to year and affect population sizes. Infectious diseases occasionally spread through a population, reducing its size (Hewitt 1921). There are also more long-term environmental factors such as carrying capacity.¹²

In addition to factors that affect population which are not modeled, there is typically noise in the measurements. In this particular case, we cannot even measure the population directly. Instead, pelts are used as a noisy proxy for population sizes.¹³ Hewitt (1921) further noted that animals are easier to trap when food is scarce, making the population measurements dependent on the population size. Even the exact number of pelts taken is likely to be only approximate, as they were collected in a range of locations over an entire season. Here, the number of pelts is rounded to the closest 100, building in measurement error on the order of 1%.¹⁴

A linear regression analogy

Like in a simple linear regression, we will proceed by treating the underlying deterministic model as providing an expected population value around which there will be variation due to both measurement error and simplifications in the scientific model. Consider the typical formulation of a linear regression, where y_n is an observable scalar outcome, x_n is a row vector of unmodeled predictors (aka covariates, features),

¹² Carrying capacity is roughly the maximum population that an environment can sustain. It is often modeled in the system dynamics as an asymptote on population size.

¹³ It would be possible to combine Lotka-Volterra dynamics with population models based on mark-recapture data, which are able to estimate actual population sizes, not just relative changes in proportions.

¹⁴ Rounding can be modeled explicitly by allowing the true value to be a parameter; for instance if rounding is to the closest 100, the parameter can be modeled as uniform over the range ± 50 from the measured value or even greater if the measurement on which the rounding was based may itself be in error.

β is a coefficient vector parameter, and $\sigma > 0$ is the error scale,

$$y_n = x_n \beta + \epsilon_n$$

$$\epsilon_n \sim \text{Normal}(0, \sigma)$$

The deterministic part of the equation is the linear predictor $x_n \beta$ with predictor x_n (row n of the data matrix x) and coefficient (column) vector β . The stochastic error term, ϵ_n , is assigned a normal distribution located at zero with scale parameter $\sigma > 0$.¹⁵ We typically formulate this model without the latent error variable ϵ_n as follows,¹⁶

$$y_n \sim \text{Normal}(x_n \beta, \sigma).$$

Lotka-Volterra error model

Solutions to the Lotka-Volterra equations replace the linear predictor $x_n \beta$, but we maintain the error term to compensate for measurement error and unexplained variation in the data. In the case of population dynamics, the data y_n consists of measurements of the prey $y_{n,1}$ and predator $y_{n,2}$ populations at times t_n .¹⁷

The true population sizes at time $t = 0$ are unknown—we only have measurements y_1^{init} and y_2^{init} of them. The true initial population sizes at time $t = 0$ will be represented by a parameter z^{init} , so that

$$z_1^{\text{init}} = u(t = 0)$$

$$z_2^{\text{init}} = v(t = 0).$$

Next, let z_1, \dots, z_N be the solutions to the

¹⁵ Gauss initially noted that the maximum likelihood estimate derived from the normal error model is identical to the least square error estimate derived by minimizing the sum of squared errors, $e^\top e$. With Markov, Gauss further proved that it was the lowest variance unbiased estimator.

¹⁶ The latent error variable may be defined in terms of x , y , and β as

$$\epsilon_n = y_n - x_n \beta$$

.

¹⁷ This model makes the assumption that the underlying population sizes $z_{n,k}$ and measurements of it $y_{n,k}$ are continuous. This is a very tight approximation to counts when the numbers are in the thousands.

Lotka–Volterra differential equations at times t_1, \dots, t_N given initial conditions $z(t = 0) = z^{init}$ and parameters $\theta = (\alpha, \beta, \gamma, \delta)$. Each z_n is a pair of prey and predator population sizes at the specified times,

$$\begin{aligned} z_{n,1} &= u(t_n) \\ z_{n,2} &= v(t_n). \end{aligned}$$

The z_n are random variables, but they are deterministic functions of the random variables for the initial state z^{init} and system parameters $\alpha, \beta, \gamma, \delta$.

The observed data is in the form of measurements y^{init} of the initial population of prey and predators, and subsequent measurements y_n of the populations at times t_n , where y^{init} and the y_n consist of a pair of measured population sizes, for the prey and predator species.

In summary, the measurements, y^{init} and y_n , are drawn indepently from a normal distribution centered at the underlying population sizes, z^{init} and z_n , with noise scales σ . Each quantity here, including the noise scale, is a pair consisting of values for prey and predator.

Multiplicative error and the lognormal distribution

It is common to log transform positive-only parameters so that they are no longer constrained to be positive. On the log scale, we can then take the error to be unconstrained and additive, just as in linear regression.

$$\log y_{n,k} = \log z_{n,k} + \epsilon_{n,k}$$

$$\epsilon_{n,k} \sim \text{Normal}(0, \sigma_k)$$

where the z_n are the solutions to the Lotka-Volterra equations at times t_1, \dots, t_N given initial populations z^{init} . The prey and predator populations have error scales (on the log scale) of σ_1 and σ_2 .

With additive errors on the log transformed scales, the result of transforming back to the natural scale (by exponentiation) leads to multiplicative errors,

$$\begin{aligned} y_{n,k} &= \exp(\log z_{n,k} + \epsilon_{n,k}) \\ &= z_{n,k} \exp(\epsilon_{n,k}) \end{aligned}$$

Because $\exp(\epsilon_{n,k})$ is positive and the underlying population size $z_{n,k}$ is positive, their product will be positive. In other words, rather than the measurement being plus or minus some value, it's plus or minus some percentage of the total.

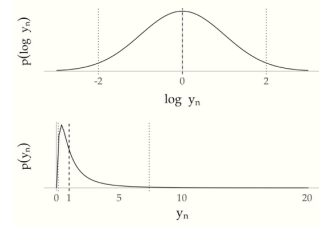
This transform and its associated fractional error is so common that they are jointly known as the lognormal distribution, so that we can simplify the above notation as we did with linear regression and write

$$y_{n,k} \sim \text{LogNormal}(z_{n,k}, \sigma_n).$$

whenever

$$\log y_{n,k} \sim \text{Normal}(z_{n,k}, \sigma_n).$$

The LogNormal density accounts for the non-linear change of variables through a Jacobian adjustment.¹⁸



The top figure shows the density of a standard normal variate, i.e., location (mean) 0 and scale (standard deviation) 1. The bottom figure shows the corresponding standard lognormal variate, i.e., log location 0 and log scale 1. The central 99.5% interval is shown in both cases with a dashed vertical line indicating the median and dotted lines the boundaries of the central 95% interval. The skew in the lognormal plot is due to the exponentiation—the range (2, 3) in the standard normal plot, containing only 2% of the total probability mass, maps to (7, 20) in the lognormal plot and still contains only 2% of the probability mass. If the horizontal and vertical axes were drawn to the same scale, the areas under the two curves would be identical.

Weakly informative priors

Our only remaining task is to formulate what we know about the parameters as priors. In general, we recommend at least weakly informative priors on parameters.¹⁹ In practice, weakly informative priors inform the scale of the answer, but have only a negligible effect on the exact value; that is, they are not very sensitive by design. While not being very sensitive, they help enormously to ease computation, as Stan might otherwise work very hard to explore the outskirts of gnarly, unregularized distributions. In order to formulate such priors, we need to know the rough scale on which to expect an answer. Because this model is well understood and widely used, as is the basic behavior of predator and prey species such as lynxes and hares, the parameter ranges leading to stable and realistic population oscillations are well known.

Priors for system parameters

For the parameters, recall that α and γ are multipliers of u and $-v$ respectively in the state equations, whereas β and δ are multipliers of the product $u v$. Combined with the fact that the scale of u and v are roughly 10 as the data has been encoded, we can formulate similar weakly informative priors,

$$\alpha, \gamma \sim \text{Normal}(1, 0.5)$$

$$\beta, \delta \sim \text{Normal}(0.05, 0.05)$$

Prior for noise scale

chapter on changes of variables works through this particular example.

¹⁹ The Stan Development Team, in a Wiki on prior choice recommendations.

The noise scale is proportional, so the following prior should be weakly informative, as a value smaller than 0.05 or larger than 3 would be unexpected.²⁰ Because values are positive, this prior adopts the lognormal distribution.²¹

$$\sigma \sim \text{LogNormal}(-1, 1)$$

Another reasonable prior choice would be a broad half normal.²²

For the initial population of predator and prey, the following priors are weakly informative.

$$z_1^{\text{init}}, z_2^{\text{init}} \sim \text{LogNormal}(\log(10), 1)$$

Coding the Model: Stan Program

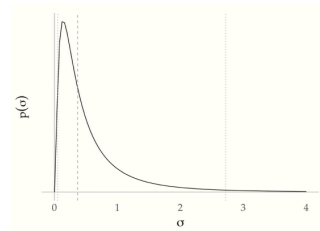
Coding the system dynamics

Whenever a general system of differential equations is involved, the system equations must be coded as a Stan function. In this case, the model is relatively simple as the state is only two dimensional and there are only four parameters. Stan requires the system to be defined with exactly the signature defined here for the function `dz_dt()`, even if not all of them are used. The first argument is for time, which is not used here because the rates in the Lotka-Volterra equations do not depend on time. The second argument is for the system state, coded as an array $z = (u, v)$. The third argument is for the parameters of the equation, of which the Lotka-Volterra equations have four, which are coded as $\theta = (\alpha, \beta, \gamma, \delta)$. The fourth and fifth argument are for data constants, but none are needed here,

²⁰ The 95% interval is approximately the mean plus or minus two standard deviations, which here is $\exp(-1 - 2) \approx 0.05$ and $\exp(-1 + 2) \approx 3$

²¹ A lognormal prior on σ is not consistent with zero values of σ , but we do not expect the data to be consistent with values of σ near zero because the model will is not particularly accurate. It makes well calibrated predictions, but they are not very sharp.

²² An interval prior with hard boundaries is not recommended as it may bias results if the data is compatible with results near the boundary.



Plot of $\text{lognormal}(-1, 1)$ prior. The median, 0.4, is highlighted with a dashed line and the central 95% interval, (0.05, 2.7), with dotted lines. This prior density is not consistent with values of sigma approaching zero.

so these arguments are unused.

```
real[] dz_dt(real t, real[] z, real[] theta
              real[] x_r, int[] x_i) {
  real u = z[1];
  real v = z[2];

  real alpha = theta[1];
  real beta = theta[2];
  real gamma = theta[3];
  real delta = theta[4];

  real du_dt = (alpha - beta * v) * u;
  real dv_dt = (-gamma + delta * u) * v;
  return { du_dt, dv_dt };
}
```

After unpacking the variables from their containers, the derivatives of population with respect to time are defined just as in the mathematical specification. The return value uses braces to construct the two-element array to return, which consists of the derivatives of the system components with respect to time,

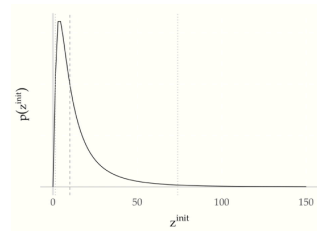
$$\frac{d}{dt}z = \frac{d}{dt}(u, v) = \left(\frac{d}{dt}u, \frac{d}{dt}v \right).$$

The data variables are coded following their specifications,

```
data {
  int<lower = 0> N;           // num measur
  real ts[N];                // measuremen
  real y_init[2];            // initial me
  real<lower = 0> y[N, 2];    // measured p
}
```

as are the parameters,

```
parameters {
  real<lower = 0> theta[4];    // theta = {
  real<lower = 0> z_init[2];   // initial po
```



Plot of $\text{lognormal}(\log(10), 1)$ prior. The median, 2.3, is highlighted with a dashed line and the 95% interval, (1.4, 75), with dotted lines. This prior suffices for both predator and prey population sizes.

```

    real<lower = 0> sigma[2];    // error scal
}

```

The solutions to the Lotka–Volterra equations for a given initial state z^{init} are defined as transformed parameters. This will allow them to be used in the model and inspected in the output (as we do in later graphs). It also makes it clear that they are all functions of the initial population and parameters (as well as the solution times).²³

```

transformed parameters {
  real z[N, 2]
    = integrate_ode_rk45(dz_dt, z_init, 0,
                        rep_array(0.0, 0),
                        1e-6, 1e-5, 1e3);
}

```

The required real and integer data arguments in the second line are both given as constant size-zero arrays. The last line provides relative and absolute tolerances, along with the maximum number of steps allowed in the solver before rejecting.²⁴ For further efficiency, the tolerances for the differential equation solver are relatively loose for this example; usually tighter tolerances are required (smaller numbers).

With the solutions in hand, the only thing left are the prior and likelihood. As with the other parts of the model, these directly follow the notation in the mathematical specification.

```

model {
  theta[{1, 3}] ~ normal(1, 0.5);
  theta[{2, 4}] ~ normal(0.05, 0.05);
  sigma ~ lognormal(-1, 1);
  z_init ~ lognormal(log(10), 1);
}

```

²³ Stan provides two solvers, a Runge–Kutta 4th/5th-order solver (`_rk45`) and a backwards-differentiation formula solver (`_bdf`) for stiff systems. The RK45 solver is twice as fast as the BDF solver for this data. The symptom of stiffness is slow iterations that may appear to be hanging if a large number of iterations is permitted. If the solver runs into stiffness and more careful initialization or prior choice does not mitigate the problem, the BDF solver may be used to adjust for the ill-conditioned Jacobians at the cost of increased computation time.

²⁴ Rejections behave like zero density points from which no progress may be made; in C++, the behavior is to throw an exception, which is caught and displayed as an error message by the interfaces.

```

    for (k in 1:2) {
      y_init[k] ~ lognormal(log(z_init[k]), s
      y[, k] ~ lognormal(log(z[, k]), sigma[
    }
  }
}

```

The multiple indexing `theta[{1, 3}]` indicates that both `theta[1]` and `theta[3]` get the prior `normal(1, 0.5)`; it is both more efficient and more compact to represent them this way. The second sampling statement involving `y[, k]` is vectorized and amounts to the same thing as an additional loop,

```

for (n in 1:N)
  y[n, k] ~ lognormal(log(z[n, k]), sigma[k]

```

As with the prior, the vectorized form used in the actual model block is much more efficient. The `y_init` sampling statement could be pulled out of the loop, but there is not much efficiency gain as there are only two elements, so it is left in this form for clarity.

Solving the Inverse Problem: Bayesian Inference in Stan

Fitting the Hudson's Bay Company data

First, the data is munged into a form suitable for Stan.

```

N <- length(lynx_hare_df$Year) - 1
ts <- 1:N
y_init <- c(lynx_hare_df$Hare[1], lynx_hare
y <- as.matrix(lynx_hare_df[2:(N + 1), 2:3]
y <- cbind(y[, 2], y[, 1]); # hare, lynx
lynx_hare_data <- list(N, ts, y_init, y)

```

Next, the model is translated to C++ and compiled.

```
model <- stan_model("lotka-volterra.stan")
```

Finally, the compiled model and data are used for sampling.²⁵ Stan's default settings are sufficient for this data set and model.

```
fit <- sampling(model, data = lynx_hare_dat
```

The output can be displayed in tabular form, here limited to the median (0.5 quantile) and 80% interval (0.1 and 0.9 quantiles), and restricted to the parameters of interest.

```
print(fit, pars=c("theta", "sigma", "z_init",
  probs=c(0.1, 0.5, 0.9), digits = 3)
```

```
Inference for Stan model: lotka-volterra.
4 chains, each with iter=2000; warmup=1000;
post-warmup draws per chain=1000, total pos
```

	mean	se_mean	sd	10%	50%
theta[1]	0.549	0.002	0.065	0.469	0.54
theta[2]	0.028	0.000	0.004	0.023	0.02
theta[3]	0.797	0.003	0.091	0.684	0.79
theta[4]	0.024	0.000	0.004	0.020	0.02
sigma[1]	0.248	0.001	0.045	0.198	0.24
sigma[2]	0.252	0.001	0.044	0.201	0.24
z_init[1]	33.960	0.056	2.909	30.363	33.87
z_init[2]	5.949	0.011	0.533	5.294	5.92

Samples were drawn using NUTS(diag_e) at Su
For each parameter, n_eff is a crude measure of
and Rhat is the potential scale reduction factor for
convergence, Rhat=1).

There were no divergent transitions²⁶ reported. The \hat{R} values are all near 1, which is consistent with convergence. The effective sample size estimates for each parameter are sufficient for inference.²⁷ Thus we have reason to trust that Stan has produced an adequate approximation of the posterior.

²⁵ The independent steps of compiling the Stan model and running it are highlighted here. These two steps may be composed into a single call as `stan("lotka_volterra.stan", data = lynx_hare_dat, seed = 123)`. Calling `rstan_options(auto_write = TRUE)` configures `stan_model()` and `stan()` to reuse a compiled model if it hasn't changed.

²⁶ Divergences occur when Stan's Hamiltonian solver diverges from the true Hamiltonian, which must be conserved, because of numerical problems in the stepwise gradient-based approximation of the

Comparing the fitted model to data

Using a non-statistically motivated error term and optimization, Howard (2009, Figure 2.10) provides the following point estimates for the model parameters based on the data.

$$\alpha^* = 0.55, \quad \beta^* = 0.028, \quad \gamma^* = 0.84, \quad \delta^* = 0.026$$

Our model produced the following point estimates based on the posterior mean,²⁸

$$\hat{\alpha} = 0.55, \quad \hat{\beta} = 0.028, \quad \hat{\gamma} = 0.80, \quad \hat{\delta} = 0.024,$$

which are very close to Howard's estimates.²⁹ The posterior intervals, which are quite wide here, may be interpreted probabilistically,

$$\Pr[0.47 \leq \alpha \leq 0.63] = 0.8$$

$$\Pr[0.23 \leq \beta \leq 0.33] = 0.8$$

$$\Pr[0.69 \leq \delta \leq 0.91] = 0.8$$

$$\Pr[0.020 \leq \gamma \leq 0.029] = 0.8$$

The effect of these estimates is plotted later with simulated population orbits.

Error scales for both populations have the same posterior mean estimate,

$$\hat{\sigma}_1 = \hat{\sigma}_2 = 0.25.$$

and both have the same posterior 80% interval, (0.20, 0.31).³⁰

Inference for population sizes

One inference we would like to make from our

curvature of the log density.

²⁷ With effective sample sizes of roughly one thousand, standard errors are roughly one thirtieth the size of posterior standard deviations, being in an inverse square root relation.

²⁸ The posterior mean minimizes expected squared error, whereas posterior medians minimize expected absolute error. Here, the mean and median are the same to within MCMC standard error.

²⁹ Discrepancies are to be expected in that we are finding a posterior mean whereas Howard is finding a posterior mode. We do not suspect the priors have a strong influence here, but this could be checked by varying them and comparing results, i.e., performing a sensitivity analysis.

³⁰ This suggests they may be completely pooled and modeled using a single parameter.

data is the size of the lynx and hare populations over time. Technically, we can only estimate the expected sizes of future pelt collections and must assume the population is somehow directly related to the numbers of pelts collected. Howard (2009) plugs in optimization-derived point estimates to derive population predictions.

Rather than plugging in point estimates to get point predictions, we will follow the fully Bayesian approach of adjusting for uncertainty. This uncertainty takes two forms in inference. First, there is estimation uncertainty, which is fully characterized by the joint posterior density $p(\alpha, \beta, \gamma, \delta, z^{init}, \sigma \mid y)$.³¹

The second form of uncertainty stems from measurement error and unexplained variation, which are both rolled into a single sampling distribution, $\log y_n \sim \text{Normal}(\log z_n, \sigma)$. As in the Stan implementation, $z_n = (u_n, v_n)$ is the solution to the differential equation conditioned on the parameters $\theta = (\alpha, \beta, \gamma, \delta, \sigma)$ and initial state z^{init} .

Posterior predictive checks

We use posterior predictive checks to evaluate how well our model fits the data from which it was estimated.³²

The basic idea is to take the posterior for the fitted model and use it to predict what the data should've looked like. That is, we will be replicating new y values that parallel the actual observations y . Because they are replicated values, we write them as y^{rep} . The

³¹ In well-behaved models in which the data is broadly consistent with the prior and likelihood, estimation uncertainty is reduced by larger samples; as more data is available, it will overwhelm the fixed priors and the posterior will concentrate around the true values of the parameters.

³² This is “testing on the training data” in machine learning parlance, and while we would not trust it for final evaluation, it is an easy way to spot inconsistencies in the implementation of misspecification in the model.

distribution of these replicated values is given by the posterior predictive distribution,

$$p(y^{\text{rep}} | y) = \int p(y^{\text{rep}} | \theta) p(\theta | y) d\theta,$$

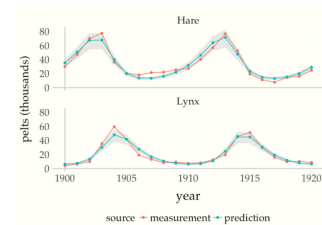
where $\theta = (\alpha, \beta, \gamma, \delta, z^{\text{init}}, \sigma)$ is the vector of parameters for the model. Our two forms of uncertainty are represented in the two terms in the integral. The first is the sampling distribution for the replications, $p(y^{\text{rep}} | \theta)$, which is the distribution of observations y^{rep} given parameters θ . This term encapsulates the unexplained variance and measurement error. The second term is the posterior $p(\theta | y)$, which encapsulates our uncertainty in our parameter estimates θ given the observations y . Here, the integral takes a weighted average of the sampling distribution, with weights given by the posterior. In statistical terms, we are calculating an expectation of a function of the parameters, $f(\theta) = p(y^{\text{rep}} | \theta)$, over the posterior $p(\theta | y)$, which can be written concisely as a conditional expectation,

$$p(y^{\text{rep}} | y) = \mathbb{E}[p(y^{\text{rep}} | \theta) | y].$$

Stan code for posterior predictive checks

Stan defines predictive quantities in the generated quantities block, which is executed once per iteration.³³ The code declares variables at the top of the block, then defines them in a loop over the species, then over the times.

```
generated quantities {
  real y_init_rep[2];
  real y_rep[N, 2];
```



Posterior predictive checks, including posterior means and 50% intervals along with the measured data. If the model is well calibrated, as this one appears to be, 50% of the points are expected to fall in their 50% intervals.

³³ The log density and its gradient are typically

```

for (k in 1:2) {
  y_init_rep[k] = lognormal_rng(log(z_init
  for (n in 1:N)
    y_rep[n, k] = lognormal_rng(log(z[n,
  }
}

```

The uncertainty due to parameter estimation is rolled into the values of z_{init} , z , and σ . The uncertainty due to unexplained variation and measurement error is captured through the use of the lognormal pseudorandom number generator, `lognormal_rng`. The additional noise in the measurements y over that of the underlying population predictions z is visualized in the plots.

Conclusion: What are the Population Dynamics?

Even with the strong assumption that the number of pelts collected is proportional to the population, we only know how the relative sizes of the populations change, not their actual sizes.

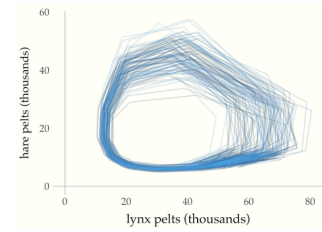
Predicted population cycles

In the same way as Volterra (1926) plotted the cycles of predator and prey populations, we can select draws of z^{rep} from the posterior and plot them. The variation here is due to posterior uncertainty in the value of the system parameters $\alpha, \beta, \gamma, \delta$ and the initial population z^{init} .

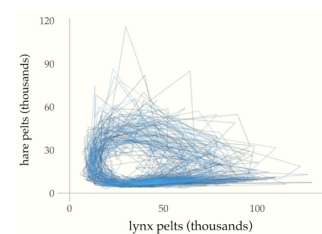
Predicted measurements

In addition to the estimation uncertainty

evaluated many times per iteration to follow the Hamiltonian trajectory of the parameters given some initial momenta.



Plot of expected population orbit for one hundred draws from the posterior. Each draw represents a different orbit determined by the differential equation system parameters. Together they provide a sketch of posterior uncertainty for the *expected* population dynamics. If the ODE solutions were extracted per month rather than per year, the resulting plots would appear fairly smooth.



Plot of expected pelt collection orbits for one hundred draws of system parameters from the posterior. Even if plotted at more fine-grained time intervals, error would remove any apparent smoothness. Extreme draws as seen here are typical when large values have high error on the multiplicative scale.

discussed in the previous section, there is also the general error due to measurement error, model misspecification, etc. In order to simulate the number of pelts that are reported collected (which may itself have error relative to the actual number of pelts), we must additionally consider the general error term. That is already rolled into the variables y^{rep} , so we plot those here.

Exercises and Extensions

The Lotka-Volterra model is easily extended for realistic applications in several obvious ways. I ran out of steam before turning this case study into a book on Bayesian modeling. I leave the next steps for this model to the dedicated reader.³⁴ Even if you don't plan to do the exercises, they provide a concise description of where this model can be taken.

1. *Simulation-based calibration.* Write a Stan model to simulate data from this model. First simulate parameters from the prior (or pick ones consistent with the priors). Then simulate data from the parameters. Finally, fit the model in Stan and compare the coverage as in the last plot in the case study.³⁵
2. *Forecasting and backcasting.* Extend predictions another 50 years into the future and plot as in the last plot. This can be done by extending the solution points in the transformed parameters, but is more efficiently done in the generated quantities block. Next, extend the predictions 50 years into the past and plot.³⁶ Is there anything

³⁴ If you complete a few of these exercises and write them up, please let me know. I'd be happy to extend this case study and add a co-author or publish a follow-on case study.

³⁵ This basic validation technique was extended and converted to a statistical test of MCMC algorithm fit by Cook, Gelman and Rubin (2006). Their test of Bayesian calibration relies on data simulated from the

suspicious about the long-term uncertainty measures?

3. *Missing data.* Suppose that several of the measurements are missing. Write a Stan program that uses only the observed measurements. This will require coding the data in long form³⁷ as

```
int J;                // num observ
int spec[J];          // species fo
real tt[J];           // time for o
real<lower = 0> yy[J]; // yy[j] == y
```

and coding the model as

```
for (j in 1:J)
  yy[j] ~ normal(zz[tt[j], spec[j]],
```

Only this part of the model changes; the ODE is set up and fit as before with the complete set of time points. What happens to the computation and posterior inferences as increasing amounts of data are missing? How can the missing data points be imputed using the generated quantities block?³⁸

4. *Error model.* Replace the lognormal error with a simple normal error model. What does this do to the z estimates and to the basic parameter estimates? Which error model fits better?
5. *Sensitivity analysis and prior choice.* Perform a sensitivity analysis on the prior choices made for this model. When the prior means or scales are varied, how much does the posterior vary? Does the model become easier or

model being calibrated by construction in full Bayesian posterior inference. As extra credit, apply this test to validate the model.

³⁶ *Hint:* the initial time will have to be changed in the call to `ode_integrate`.

³⁷ Long form is known as “melted” in R’s Tidyverse.

³⁸ There is a chapter on missing data with more details in the Stan manual.

harder to fit (in terms of effective sample size per unit time or divergent transitions) with different prior choices? What does this imply about the number of digits with which we report results and thus the effective sample sizes necessary for most inferences?

6. *Model misspecification*. Swap the coding of the lynx and hare in the input so that the predator is modeled as prey and vice-versa. How well does it fit the data? How does this provide evidence for the folk theorem?³⁹
7. *Multiple species*. Extend the model to deal with more than one species. Treat each species as as having a mixture of predators and a mixture of prey affecting its growth. Simulate data and fit to your model. Do the equations become stiff as more species and interactions are added? How much data is required to identify the model as the number of species and their mixing increases?⁴⁰
8. *Covariates*. Suppose we have measured covariates such as temperature, water, and plant abundance. Set up a model where these have further effects on populations modeled as unconstrained multiplies of the existing species size. Simulate data using your covariates. What does the existence of covariates do to the uncertainty estimates compared to using a model without covariates?
9. *Joint mark-recapture modeling*. Suppose that we also have mark-recapture data for each species over the same time period. Find or simulate such data.

³⁹ *Folk Theorem* (Gelman) When you have computational problems, often there's a problem with your model.

⁴⁰ *Hint*: see Michael Betancourt's case study on mixture models on the Stan web site for general advice on mixture modeling.

Build a mark-recapture model such as Cormack-Jolly-Seber and jointly model the population in terms of the CJS model and the Lotka-Volterra model.⁴¹ How can you scale the population dynamics model in order to work at the same scale as the mark-recapture model? Does this reduce uncertainty in the population dynamics? Does the joint model improve on fitting the mark-recapture model or Lotka-Volterra models separately?

⁴¹ The Stan manual chapter on latent discrete parameters describes how to code the Cormack-Jolly-Seber model for mark-recapture data.

10. *Measurement error.* Suppose the number of lynx pelts collected is affected by the size of the population. In particular, suppose a larger proportion of lynx are captured when they are hungry. How could this be included as a component of the model? What parameter would be introduced? Simulate data and see if the true populations can be recovered with the model.
11. *Carrying capacity.* How can a carrying capacity (upper bound on population size) be incorporated into the Lotka-Volterra model? Formulate the model in Stan and fit it to simulated data. If the data is simulated near the carrying capacity, how does it fit using the simpler Lotka-Volterra model?
12. *Cross-validation and predictive calibration.* Cross validation as such doesn't make much sense for time series. Nevertheless, it is possible to fit data to initial sequences of a time series and predict the remaining sequence. How could this be achieved here?⁴² Do you need to write more than one Stan

⁴² *Hint:* Consider using the `loo` package for leave-

model?

one-out cross-validation (Vehtari, Gelman, and Gabry 2017).

References

- Cook, S. R., Gelman, A., & Rubin, D. B. (2006) Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3), 675–692.
- Hewitt, C. G. (1921) *The Conservation of the Wild Life of Canada*. Charles Scribner's Sons.
- Howard, P. (2009). Modeling basics. Lecture Notes for Math 442, Texas A&M University.
- Lotka, A. J. (1925). *Principles of physical biology*. Baltimore: Waverly.
- Rogers, K. (2011) The rise and fall of the Canada lynx and snowshoe hare. *Encyclopedia Britannica Blog*.
- Stan Development Team (2017) *Stan Modeling Language Users Guide and Reference Manual*, Version 2.17, <http://mc-stan.org>.
- Vehtari, A., Gelman, A. & Gabry, J. (2017) Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Journal of Statistics and Computing* 27(5):1413–1432.
- Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118(2972), 558–560.
- Volterra, V. (1927). *Variazioni e fluttuazioni del numero d'individui in*

Source code

All of the source code, data, text, and images for this case study are available on GitHub at

- `stan-dev/example-models/knitr/lotka-volterra`

Acknowledgements

Thanks to Simon Woodward, Arya Pourzanjani, and Leo Lahti for helpful comments on earlier drafts.

Complete Stan program

Here is the complete Stan program for solving the inverse problem for the Lotka-Volterra model, including posterior predictive checks.

```
functions {
  real[] dz_dt(real t,      // time
               real[] z,    // system stat
               real[] theta, // parameters
               real[] x_r,   // unused data
               int[] x_i) {
    real u = z[1];
    real v = z[2];

    real alpha = theta[1];
    real beta = theta[2];
    real gamma = theta[3];
    real delta = theta[4];

    real du_dt = (alpha - beta * v) * u;
    real dv_dt = (-gamma + delta * u) * v;
```

```

    return { du_dt, dv_dt };
  }
}
data {
  int<lower = 0> N;          // number of m
  real ts[N];               // measurement
  real y_init[2];           // initial mea
  real<lower = 0> y[N, 2];   // measured po
}
parameters {
  real<lower = 0> theta[4];   // { alpha, b
  real<lower = 0> z_init[2]; // initial po
  real<lower = 0> sigma[2];   // measuremen
}
transformed parameters {
  real z[N, 2]
    = integrate_ode_rk45(dz_dt, z_init, 0,
                        rep_array(0.0, 0),
                        1e-5, 1e-3, 5e2);
}
model {
  theta[{1, 3}] ~ normal(1, 0.5);
  theta[{2, 4}] ~ normal(0.05, 0.05);
  sigma ~ lognormal(-1, 1);
  z_init ~ lognormal(10, 1);
  for (k in 1:2) {
    y_init[k] ~ lognormal(log(z_init[k]), s
    y[ , k] ~ lognormal(log(z[, k]), sigma[
  }
}
generated quantities {
  real y_init_rep[2];
  real y_rep[N, 2];
  for (k in 1:2) {
    y_init_rep[k] = lognormal_rng(log(z_ini
    for (n in 1:N)
      y_rep[n, k] = lognormal_rng(log(z[n,
  }
}

```

Session information

```
sessionInfo()
```

R version 3.3.2 (2016-10-31)

Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X Yosemite 10.10.5

locale:

[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/e

attached base packages:

[1] stats graphics grDevices utils

other attached packages:

[1] tufte_0.3 rstan_2.17.3 S
[4] reshape_0.8.7 knitr_1.18 g
[7] ggplot2_2.2.1

loaded via a namespace (and not attached):

[1] Rcpp_0.12.8 magrittr_1.5 muns
[5] rlang_0.1.4 highr_0.6 stri
[9] tools_3.3.2 parallel_3.3.2 grid
[13] htmltools_0.3.6 yaml_2.1.14 lazy
[17] digest_0.6.10 tibble_1.3.4 code
[21] evaluate_0.10 rmarkdown_1.5 labe
[25] scales_0.4.1 backports_1.0.5 stat

Licenses

Code © 2017–2018, Trustees of Columbia University
in New York, licensed under BSD-3.

Text © 2017–2018, Bob Carpenter, licensed under
CC-BY-NC 4.0.