Bayesian Dynamic Models

Models for Socio-Environmental Data

Chris Che-Castaldo, Mary B. Collins, N. Thompson Hobbs

June 13, 2019
Roadmap

- Overview
- Model types with examples
  - discrete time
    - single state
    - multiple states
  - continuous time (briefly)
- Autocorrelation
- Forecasting
- Coding tips
Dynamic hierarchical models (aka state space models)

Also called “state space” models

\[
\begin{align*}
[y_t | \theta_d, z_t] \\
[z_t | \theta_p, z_{t-1}]
\end{align*}
\]

The idea is simple. We have a stochastic model of an unobserved, true state \((z_t)\) and a stochastic model that relates our observations \((y_t)\) to the true state.
A broadly applicable approach to modeling dynamic processes in ecology and social science

\[
[z, \theta_{\text{process}}, \theta_{\text{data}} | y] \propto \\
\prod_{t=2}^{T} [y_t | \theta_{\text{data}}, z_t] [z_t | \theta_{\text{process}}, z_{t-1}] [\theta_{\text{process}}, \theta_{\text{data}}, z_1]
\]
Sources of uncertainty in state space models

**Process uncertainty**
- Failure to perfectly represent process
- Propagates in time
- Decreases with model improvement
- Basis for forecasting

**Observation uncertainty**
- Failure to perfectly observe process
- Does not propagate
- Sampling uncertainty decreases with increased sampling effort.
- Observation (calibration) uncertainly decreases with improved instrumentation, calibration, etc.
Components of observation uncertainty

- Observation (aka calibration) \( [y|h(z, \theta_d), \sigma_o^2] \)
- Sampling \( [y|z, \sigma_s^2] \)
When can we separate process variance from observation variance?

- Replication of the observation for the latent state with sufficient $n$
- Calibration model with properly estimate prediction variance
- Strongly differing “structure” in process and observation models
- We may not need to separate them—sometimes the observed state and the true state are the same.
General joint and posterior distribution for single state model

\[
[z, \theta_{\text{process}}, \theta_{\text{data}}, \sigma_p^2, \sigma_d^2 | y] \propto \prod_{t=2}^{T} \left[ y_t | \theta_{\text{data}}, z_t, \sigma_d^2 \right] \\
\times \left[ z_t | g(\theta_{\text{process}}, z_{t-1}, x_{t-1}), \sigma_d^2 \right] \\
\times [\theta_{\text{process}}, \theta_{\text{data}}, \sigma_p^2, \sigma_d^2, z_1]
\]

Deterministic model = \( g(\theta_{\text{process}}, z_{t-1}, x_{t-1}) \)
How does rainfall influence density dependence?

\[ g(\beta, z_{t-1}, x_{t-1}) = z_{t-1} e^{(\beta_0 + \beta_1 z_{t-1} + \beta_2 x_{t-1} + \beta_3 z_{t-1} x_{t-1})\Delta t} \]

- \( z_t \) = true population size
- \( x_{t-1} \) = standardized, annual dry season rainfall during time \( t-1 \) to \( t \).
- \( \beta_0 = r_{max} \) = intrinsic, per-capita rate of increase at average rainfall
- \( \beta_1 \) = strength of density dependence, \( \frac{r}{K} \) at average rainfall.
- \( \beta_2 \) = change in rate of increase per standard deviation change in rainfall
- \( \beta_3 \) = effect of rainfall on strength of density dependence
Serengeti wildebeest model

\[
g (\beta, z_{t-1}, x_{t-1}) = z_{t-1} e^{(\beta_0 + \beta_1 z_{t-1} + \beta_2 x_{t-1} + \beta_3 z_{t-1} x_{t-1}) \Delta t}
\]

\[
[z, \beta, \sigma_p^2 | y] \propto \prod_{t \in y} \left[ y_t \mid z_t, y.sdt \right]
\]

\[
\times \prod_{t=2}^{48} \left[ z_t \mid g (\beta, z_{t-1}, x_{t-1}), \sigma_p^2 \right] \times \left[ \beta_0 \right] \left[ \beta_1 \right] \left[ \beta_2 \right] \left[ \beta_3 \right] \left[ \sigma_p^2 \right] \left[ z_1 \right]
\]

- \( y.i \) is a vector of years with non-missing census data
- \( y_t \sim \text{normal}(z_t, y.sdt) \)
- \( z_t \sim \text{lognormal} \left( \log(g(\beta, z_{t-1}, x_{t-1})), \sigma_p^2 \right) \)
- \( \beta_0 \sim \text{normal}(0.234, 0.136^2) \) informative prior
- \( \beta_{i \in 1,2,3} \sim \text{normal}(0, 1000) \)
- \( \sigma_p^2 \sim \text{gamma}(0.01, 0.01) \)
- \( z_1 \sim \text{normal}(y_1, y.sdt_1) \)
Deterministic matrix model

Process model:

\[
\begin{pmatrix}
z_1 \\
z_2 \\
z \\
\vdots \\
z_n
\end{pmatrix}_t = \Theta \begin{pmatrix}
z_1 \\
z_2 \\
z_3 \\
\vdots \\
z_n
\end{pmatrix}_{t-1}
\]  

(1)

where \( \Theta \) is an \( n \times n \) matrix governing the transitions among states. The product \( \Theta z_t \) defines a system of \( n \) linked, difference equations. We can learn a great deal about the dynamics of the system from analyzing the properties of \( \Theta \), its eigenvalues, eigenvectors, characteristic polynomials, etc. We can make inference on these using derived quantities.
Posterior and joint distribution

\[ [z, \Theta, \theta_{data} \mid Y] \propto \prod_{t=2}^{T} \left[ y_t \mid \theta_{data}, z_t \right] \left[ z_t \mid \Theta, z_{t-1} \right] \left[ \Theta, \theta_{data}, z_1 \right] \]
## Example: Raiho matrix model

<table>
<thead>
<tr>
<th>state</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1$</td>
<td>The number of juvenile deer, aged 6 months on their first census</td>
</tr>
<tr>
<td>$n_2$</td>
<td>The number of adult female deer, aged 18 months and older</td>
</tr>
<tr>
<td>$n_3$</td>
<td>The number of adult male deer, aged 18 months and older</td>
</tr>
</tbody>
</table>
\( f \) number of recruits per female surviving to census
\( \phi_j \) probability that a juvenile (aged 6 months) survives to 18 months
\( \phi_d \) annual survival probability of adult females
\( \phi_b \) annual survival probability of adult males
\( m \) proportion of juveniles surviving to adults that are female

\[
A = \begin{pmatrix}
0 & \phi_d f & 0 \\
\phi_j & \phi_d & 0 \\
(1 - m) & \phi_j & \phi_b
\end{pmatrix}
\]

\[
n_t = A n_{t-1}.
\]
The posterior and joint distribution

\[
\begin{bmatrix}
\phi, m, f, N, \sigma_p, \rho \\
\text{elements of } \Sigma
\end{bmatrix}
\propto
\prod_{t=2}^{T} \text{multivariate normal}(\log(n_t) | \log(A_t n_{t-1}), \Sigma)
\begin{cases}
\times \prod_{t=2}^{T} \text{normal}(y_t \text{census.mean} | \sum_{i=1}^{3} n_{i,t}, y_t \text{census.sd}) \\
\times \text{multinomial}(y_t \text{classification} | \left(\sum_{i=1}^{3} y_{i,t}, \frac{n_{1,t}}{\sum_{i=1}^{3} n_{i,t}}, \frac{n_{2,t}}{\sum_{i=1}^{3} n_{i,t}}, \frac{n_{3,t}}{\sum_{i=1}^{3} n_{i,t}}\right)')
\end{cases}
\times \text{priors}
\]
Systems of differential equations

\[
\frac{dz_1}{dt} = k_1 z_1 - k_2 z_1 z_2 \\
\frac{dz_2}{dt} = -k_3 z_1 + \alpha k_2 z_1 z_2 \\
\frac{dz_3}{dt} = \frac{k_4 z_3}{k_5 + z_3}
\]

Implementing the process model usually needs a numerical solver to align the states with the data.
Continuous time models

- Must deterministically update states at discrete intervals to match with data

- To estimate states:
  - Use analytical solutions to ODE system if available.
  - For models without analytical solutions:
    - STAN has superb ODE solver. ¹
    - R’s Nimble package ² allows you to embed functions in JAGS. A sturdy ODE solver (Runge-Kutta IV) can be written in 6-8 lines of code.
    - Write your own MCMC sampler with embedded numerical solver (e.g. lsoda() in R). ³

²https://r-nimble.org/
The problem:

Assume for simplicity that the state is observed perfectly. The simplest model of the change in state with time is

\[ y_t = \alpha y_{t-1} + \varepsilon_t \]  

(2)

where \( E(y_t) = 0 \) and \( \varepsilon_t \sim \text{normal}(0, \sigma^2) \). We might introduce effects of predictor variables using

\[ y_t = g(\theta, x_t) + \alpha y_{t-1} + \varepsilon_t. \]  

(3)

What if \( \varepsilon_t \) depends on previous errors, that is, \( e_t = h(e_{t-1}) \)? In this case, there is structural variation in the data, also called temporal dependence. The assumptions of independent errors does not hold. We have two choices:

1. Improve \( g(\theta, x_t) \) so that the deterministic model accounts for the temporal dependence via the covariates.

2. Model the temporal dependence in the errors directly.
Detecting temporal dependence

The empirical autocorrelation function (ACF):

\[
\rho_g = \frac{\sum_{i=1}^{n-g} (\varepsilon_i - \bar{\varepsilon})(\varepsilon_{i+g} - \bar{\varepsilon})}{\sum_{i=1}^{N} (\varepsilon_i - \bar{\varepsilon})^2}
\]

where \( n \) is the number of steps in the time series and \( g \) is the “lag,” the number of steps examined for temporal dependence, \(-1 \leq \rho_g \leq 1\).

The notation ACF(\( g \)) means the correlation between points separated by \( g \) time periods.
ACF plots
ACF in MCMC

\( \mu_t = g(\theta, z_{t-1}, x_{t-1}) \)

1. Compute residuals at each MCMC iteration, 
\( e_{(k)}^t = y_t - \mu_{(k)}^t \)

2. Compute \( \rho_{g_{(k)}}^t \) at each MCMC iteration and plot posterior means of \( \rho_{g_{(k)}}^t \) as a function of \( g \).

3. Or, better and easier, sample from MCMC output for \( e_{(k)}^t \), use \texttt{acf()} \ function in R to find posterior distributions of \( \rho_g \).

Make statements like “Mean autocorrelation was .21 (BCI = .23,.18) at lag 3, revealing minimal temporal dependence in the residuals.”
Bayesian forecasting future states $z'$

$$\mathbb{E}(z'_{T+1} | y) = \text{predictive process distribution}$$

$$\int_{\theta_1} ... \int_{\theta_P} \int_{z_1} ... \int_{z_T} \mathbb{E}(z'_{T+1} | z, \theta_{\text{process}}, y) \mathbb{E}(z, \theta_{\text{process}}, \theta_{\text{data}} | y) \, dz \cdots dz_t \, d\theta_1 \cdots d\theta_P$$

posterior distribution
## Predictive process distribution

The MCMC output:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\theta_1$</th>
<th>$\theta_1$</th>
<th>$\theta_3$</th>
<th>$z_{1,1}$</th>
<th>$z_{1,2}$</th>
<th>\cdots</th>
<th>$z_{1,T}$</th>
<th>$z'_{1,T+1}$</th>
<th>$z'_{1,T+2}$</th>
<th>\cdots</th>
<th>$z'_{1,T+F}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.42</td>
<td>3.3</td>
<td>20.3</td>
<td>$z_{1,1}$</td>
<td>$z_{1,2}$</td>
<td>\cdots</td>
<td>$z_{1,T}$</td>
<td>$z'_{1,T+1}$</td>
<td>$z'_{1,T+2}$</td>
<td>\cdots</td>
<td>$z'_{1,T+F}$</td>
</tr>
<tr>
<td>2</td>
<td>.41</td>
<td>2.3</td>
<td>18.5</td>
<td>$z_{2,1}$</td>
<td>$z_{2,2}$</td>
<td>\cdots</td>
<td>$z_{2,T}$</td>
<td>$z'_{2,T+1}$</td>
<td>$z'_{2,T+2}$</td>
<td>\cdots</td>
<td>$z'_{2,T+F}$</td>
</tr>
<tr>
<td>3</td>
<td>.46</td>
<td>3.1</td>
<td>16.6</td>
<td>$z_{3,1}$</td>
<td>$z_{3,2}$</td>
<td>\cdots</td>
<td>$z_{3,T}$</td>
<td>$z'_{3,T+1}$</td>
<td>$z'_{3,T+2}$</td>
<td>\cdots</td>
<td>$z'_{3,T+F}$</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>$n$</td>
<td>.39</td>
<td>3.4</td>
<td>22.1</td>
<td>$z_{n,1}$</td>
<td>$z_{n,2}$</td>
<td>\cdots</td>
<td>$z_{n,T}$</td>
<td>$z'_{n,T+1}$</td>
<td>$z'_{n,T+2}$</td>
<td>\cdots</td>
<td>$z'_{n,T+F}$</td>
</tr>
</tbody>
</table>

\[ n \quad \text{number of iterations} \]
\[ T \quad \text{final time with data} \]
\[ F \quad \text{number of forecasts beyond data} \]
Posterior and joint distribution with forecasts

\[ \mu_t = g(\theta_{process}, z_{t-1}, x_{t-1}) \]

\[ [z, \theta_{process}, \theta_{data} | y] \propto \prod_{t=2}^{T} [y_t | \theta_{data}, z_t] \prod_{t=2}^{T+F} [z_t | \mu_t] [\theta_{process}, \theta_{data}, z_1] \]
Posterior and joint distribution with missing data

\[ \mu_t = g(\theta_{process}, z_{t-1}, x_{t-1}) \]

\[
[z, \theta_{process}, \theta_{data}|y] \propto \\
\prod_{\forall t \in y} [y_t|\theta_{data}, z_t] \prod_{t=2}^{T} [z_t|\mu_t] [\theta_{process}, \theta_{data}, z_1]
\]

Can put NA’s in data for all missing values or use the indexing trick shown below.
The fundamental problem of management:
What actions can we take today that will allow us to meet goals for the future?
Predictive process distribution of $z'$
Objective: reduce state below a target

Future state \( z' \)
Objective: maintain state within acceptable range

Future state  $z'$
Objective: increase state above a target

Future state $z'$
Action: do nothing

Objective

Future state $z'$

Probability density
Action: implement management

Future state of system, $z'$
Net effect of management

Future state $z'$

Objective
Net effect of management

Future state $z'$

Probability density

Objective

Overview
Discrete time models
Continuous time models
Autocorrelation
Forecasting
Coding tips
Papers using forecasting relative to goals


More on forecasting

- Workshop July 28 - August 2
  https://ecoforecast.wordpress.com/summer-course/
**JAGS code for posterior and joint distributions**

\[
[z, \beta, \sigma^2_p | y] \propto \prod_{\forall t \in y} \left[ y_t \left| z_t, y, sd_t \right. \right]
\]

\[
\times \prod_{t=2}^{48} \left[ z_t \left| g(\beta, z_{t-1}, x_{t-1}), \sigma^2_p \right. \right] \times \left[ \beta_0 \right] \left[ \beta_1 \right] \left[ \beta_2 \right] \left[ \beta_3 \right] \left[ \sigma^2_p \right] \left[ z_1 \right]
\]

model{
  # Priors
  b[1] ~ dnorm(.234,1/.136^2)
  for(j in 2:n.coef){
    b[j] ~ dnorm(0,.0001)
  }
  tau.p ~ dgamma(.01,.01)
  sigma.p <- 1/sqrt(tau.p)
  z[1] ~ dnorm(N.obs[1],tau.obs[1])  # this must be treated as prior so that you have z[t-1]
  ## Process model
  for(t in 2:(T+F)){
    mu[t] <- log(z[t-1]*exp(b[1] + b[2]*z[t-1] + b[3]*x[t] +b[4]*x[t]*z[t-1]))
    z[t] ~ dlnorm(mu[t], tau.p)
  }
  # Data model
  for(j in 2:n.obs){
    N.obs[j] ~ dnorm(z[index[j]],tau.obs[j])  # index to match z[t] with data
  }
} # end of model
Posterior predictive checks for time series data

Test statistic:

$$\frac{1}{T-1} \sum_{t=2}^{T} | y_t - y_{t-1} |$$

Conventional statistics are also used (mean, CV, discrepancy statistic for the $y_t$).

# Derived quantities for model evaluation

```r
for(i in 1:n.obs){
    # for autocorrelation test
    epsilon.obs[i] <- N.obs[i] - z[index[i]]
    # simulate new data
    N.new[i] ~ dnorm(z[index[i]],tau.obs[i])
    sq[i] <- (N.obs[i] - z[index[i]])^2
    sq.new[i] <- (N.new[i] - z[index[i]])^2
}
fit <- sum(sq[])
fit.new <- sum(sq.new[])
pvalue <- step(fit.new-fit)
```