Management performance evaluation of state-space models for Pacific pink salmon stock-recruitment analysis  
(Equations)

Zhenming Su  
Ann Arbor, MI

Feb. 12, 2023

This document provides the full conditional posterior distributions used in the MCMC sampling algorithm for a state-space model proposed by Su (2023) “Management performance evaluation of state-space models for Pacific pink salmon stock-recruitment analysis” submitted for peer review and publication.

# The model

The state-space form of Ricker stock-recruitment model incorporates time-varying productivity, observation errors in spawners and catch data, a model of harvest rates, and process variability in recruitment process. Time-varying productivity is modeled by representing the Ricker productivity ‘*a*’ parameter as a random walk.

## Observation equation:

Let , , , .

## System equation:

Let log(recruitment) , = Ricker productivity, *k* = fixed maturity and return age.

## Initial conditions

## Parameters and states

## Data: observed escapement catch

Data collection started with *k* escapement (spawner abundance) observations from *t* = 1 to *k*: *E*1 to *Ek*. Returns and catch are available from *t* = *k* + 1 to *T*.

## Models and variables

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***t*** | ***year*** | ***Et*** | **St** | ***Rt*** | ***ht*** | ***t*** | ***Ct*** |  | ***at*** |
| 1 | 1960 | *E*1 | ***S*1** |  |  |  |  |  |  |
| 2 | 1961 | *E*2 | ***S*2** |  |  |  |  |  |  |
| 3 | 1962 | *E*3 | *S*3 | *R*3 | *h*3 | 3 | *C*3 |  |  |
| 4 | 1963 | *E*4 | *S*4 | *R*4 | *h*4 | 4 | *C*4 |  |  |
| 5 | 1964 | *E*5 | *S*5 | ***R*5** | *h*5 | 5 | *C*5 |  |  |
|  | ... | … | … |  |  |  |  |  |  |
| *t* |  | *Et* | *St* | *Rt* | *ht* | *t* | *Ct* |  |  |
|  | ... | … | … |  |  |  |  |  |  |
| *T*-2 |  | *ET-2* | *ST-2* |  |  |  |  |  |  |
| *T*-1 |  | *ET-1* | *ST-1* | *RT-*1 | *hT-*1 | *T-*1 | *CT-*1 |  |  |
| *T* |  | *ET* | *ST* | *RT* | *hT* | *T* | *CT* |  |  |
| *T+1* |  |  |  |  |  |  |  |  |  |

Note:

# Bayesian estimation

## Prior

## Posterior distribution

, , , , and

Also note that:

, , and

Posterior for all unknowns:

# Full conditionals distributions

## Full conditional of

### a) Inverse-Gamma prior on

Prior of

Full conditional density of based on an inverse-Gamma prior for :

### b) Uniform prior on

Prior of

Equivalently,

Full conditional density of :

## Full conditional of

### a) Uniform prior on

Equivalently:

Conditional posterior density of :

### b) Inverse-Gamma prior on

## Full conditional of

### a) Uniform prior on

Equivalently:

Conditional posterior density of :

### b) Inverse-Gamma prior on

## Full conditional of : independent

## Full conditional of for random walk

### a) Uniform prior on

Equivalently:

Conditional posterior density of :

### b) Inverse-Gamma prior on

## Full conditional of for hierarchical

### a) Uniform prior on

Equivalently:

Conditional posterior density of :

### b) Inverse-Gamma prior on

## Full conditional of

### a) Uniform prior on

Equivalently:

Conditional posterior density of :

### b) Inverse-Gamma prior on

## Full conditional of *rt*, *t* = *k*+1, …, *T*

Conditional distribution of

where:

Log conditional distribution of used in the Metropolis step

For

=

Gradient of :

For

Gradient of for

Metropolis step

A Metropolis step with a normal proposal distribution , , can be used to update each *rt* = log(*Rt*), where is the value of *rt* at the current iteration *i*, and the is a candidate value for *rt* drawn from the proposal distribution with a specified standard deviation *sdt*. In a Metropolis step, the is accepted as an update for *rt* with probability ; otherwise, it is rejected and the chain will remain in place .

The performance of the Metropolis-Hastings algorithm can be expressed by the acceptance rate of the candidate draws in the Metropolis-Hastings steps. Theoretical and empirical results show that the acceptance rate in the range 20% ~ 50% (depending on number of parameters) provides optimal performance (Gelman et al. 1995). For multilevel models, Browne and Draper (2000) proposed an acceptance rate of 40% ~ 60% for univariate updating.

To increase the efficiency of the Metropolis algorithm, we adopt an adaptive tuning step similar to that of Browne and Draper (2000) to tune the *sdt* before generating sample draws for inference. The goal of the tuning is to obtain a target acceptance rate around 50%. The adaptive step is stopped after a fixed number of iterations, after which the burn-in period (the pre-convergence period) and main monitoring run (the post-convergence period) is started.

*Independent chain*

In this algorithm, the density function *p*(*rtrt*-*k*) obtained from the transition equation is used as the proposal distribution for the general Metropolis-Hastings algorithm. In this case, *q*(*x*\**x*) = *q*(*x*\*) does not depend on the current value *x*(*i*), so the algorithm is called the independent chain. The *rt* can be updated using the general Metropolis-Hastings algorithm with the proposal *p*(*rtrt*- *k*),

where , , , and are the current values of *rt*, *rt-k* and at iteration *i*, denotes a candidate value for *rt*. To update each *rt* = ln(*Rt*), a is drawn from . Then is accepted as an update for *rt* with probability:

;

otherwise, it is rejected and .

## Full conditional of the initial states *s*1 to *sk*

Log conditional distribution of used in the Metropolis step

Gradient of :

*Metropolis step for updating s1 to sk*

A Metropolis step with a normal proposal distribution , *t* = *k*+1, …, *T*, is used to update each *st* = ln(*S*t), where is the value of *st* at current iteration *i*, and the is a candidate value for *st* drawn from the proposal distribution with a specified standard deviation *sdt*. In a Metropolis step, the is accepted as an update for *x*t with probability ; otherwise, it is rejected, and .

The performance of the Metropolis-Hastings algorithm can be expressed by the acceptance rate of the candidate draws in the Metropolis-Hastings steps. Theoretical and empirical results show that the acceptance rate in the range 20% ~ 50% (depending on number of parameters) provides optimal performance (Gelman et al. 1995). For multilevel models, Browne and Draper (2000) proposed an acceptance rate of 40% ~ 60% for univariate updating.

To increase the efficiency of the Metropolis algorithm, we adopt an adaptive tuning step similar to that of Browne and Draper (2000) to tune the *sdi* before generating sample draws for inference. The goal of the tuning is to obtain a target acceptance rate around 50%. The adaptive step is stopped after a fixed number of iterations, after which the burn-in period (the pre-convergence period) and main monitoring run (the post-convergence period) is started.

## Full conditional of : independent priors (fixed effects model)

Conditional distribution of

where:

Log conditional distribution of used in the Metropolis step

Gradient of :

## Full conditional of : hierarchical prior

Conditional distribution of

Log conditional distribution of used in the Metropolis step

Gradient of :

## Full conditional of : hierarchical prior

## Full conditional of : RW prior

Conditional distribution of used in the Metropolis step

For the initial at :

Gradient of :

For :

Gradient of :

For :

Gradient of :

## Full conditional of

### Conditional distribution of

### Full conditional of

For *t* = 1: :

Full conditional of :

### Full conditional of :

For *t* = 2, …, *T* – *k*:

Full conditional of :

## Full conditional of :

Prior

Full conditional of :