E. E. Holmes and E. J. Ward

Case Studies for the MARSS routines

October 21, 2009

Mathematical Biology Program Northwest Fisheries Science Center

Holmes, E. E. and E. J. Ward. 2009. Case Studies for the MARSS routines. NOAA Fisheries, Northwest Fisheries Science Center, 2725 Montlake Blvd E., Seattle, WA 98112, unpublished documentation. Contact eli.holmes@noaa.gov

Contents

1 Instructions for running case studies using provided scripts					
	1.1	Set up: R	1		
	1.2	R pointers when doing case studies	1		
	1.3	Symbols reminders	2		
	1.4	Case Study Guide	2		
		1.4.1 First write down (on paper) the matrix form for $\mathbf{Z}, \mathbf{U},$			
		\mathbf{Q} , and \mathbf{R}	2		
		1.4.2 Translate the \mathbf{Z} , \mathbf{U} , \mathbf{Q} , and \mathbf{R} forms into KalmanEM			
		arguments	3		
	1.5	KalmanEM tips	3		
2	Cas	e Study 1: Count-based PVA for data with observation			
-	erro		5		
	2.1	The Problem	5		
	$\frac{-1}{2.2}$	Simulated data with process and observation error	6		
	2.3	Parameter estimation	8		
	2.0	2.3.1 Maximum-likelihood estimates for a model with	0		
		observation error	8		
		2.3.2 Maximum-likelihood estimates for a model with no	0		
		observation error	9		
	2.4	Probability of hitting a threshold $\Pi(r_{d}, t_{c})$	12^{-12}		
	2.5	Certain and uncertain regions \dots	15^{12}		
	2.6	More risk metrics and some real data	16		
	2.7	Confidence intervals	18		
	2.8	Other parameter estimation methods	18		
3	Cas	e study 2. Combining multi-site and subpopulation			
	dat	a to estimate trends and trajectories	21		
	3.1	The problem	21		
	3.2	First analysis: a single total Puget Sound population	23		
	0.2	3.21 The nonulation process X for analysis 1	23		
		$5.2.1$ The population process, \mathbf{x} , for analysis $1 \dots \dots \dots$	40		

		3.2.2 The observation process, Y , for analysis 1	24 25
		3.2.4 Fit the model for analysis 1 to the data	20
	22	Second analysis: constraining the observation variances	20
	3.5 3.4	Third analysis. Constraining the observation variances	20
	0.4	3.4.1 Specifying the KalmanEM arguments for analysis 3	29
	25	Analysis 4.7: other nonulation and observation error structures	30 21
	3.6	Discussion	33
	0.0		00
4	Cas	e Study 3: Using MARSS models to identify spatial	
	pop	ulation structure and covariance	37
	4.1	The problem	37
	4.2	Analysis for question 1: how many distinct subpopulations?	37
		4.2.1 Specify the design, \mathbf{Z} , matrices	38
		4.2.2 Specify the grouping arguments	39
		4.2.3 Fit models and summarize results	40
		4.2.4 Interpret results for question 1	40
	4.3	Analysis for question 2: Is Hood Canal separate?	41
		4.3.1 Specify the Z matrix and whichPop	41
		4.3.2 Specify which parameters are shared across which	
		subpopulations	42
		4.3.3 Fit the models and summarize results	42
		4.3.4 Interpret results for question 2	43
5	Cas	e Study 5: Using state-space models to analyze noisy	
-	anir	nal tracking data	45
	5.1	Background: a simple random walk model of animal movement	45
	5.2	The problem	45
	5.3	Using the Kalman-EM algorithm to estimate locations from	
		bad tag data	46
		5.3.1 Read in the data and load maps packages	46
		5.3.2 Use KalmanEM to estimate the position of Big Mama	47
		5.3.3 Compare KalmanEM estimates to the real positions	48
		5.3.4 Estimate speeds for each turtle	48
	5.4	Comparing turtle tracks to proposed fishing areas	50
	5.5	Using fields to get density plots of locations	50
	5.6	Using specialized packages to analyze tag data	51
6	Cod	le	53
7	Tex	tbooks and articles that use state-space modeling	55
Ref	feren	ces	59
Ind	ex		61

Instructions for running case studies using provided scripts

1.1 Set up: R

- Open up R
- Change your working directory to the folder MSSM_Workshop (File: change dir...).
- Type in dir() and you should see a list of the workshop files
- If you haven't already, install the packages MASS, date, maps, and mvtnorm. You only need to install once. If you are online, go to Packages: Install package(s).... It will then ask you to pick a local mirror; Next, scroll down to each package and install. If you are not online, then ask one of the instructors; they have the packages on a memory stick.
- Type in source("KalmanEM.r") at the *R* command line (the >). Now you should be ready.

1.2 R pointers when doing case studies

- Each case study comes with an associated script file: Case_Study_#.r with the code you need to do the basic analyses in the worksheets. It also contains pointers for doing extensions of the basic analyses.
- With the script open (File: open), you can select the bits of code to run, right-click (Windows), and select "Run selection". You can also chose "Run all" from the Edit menu.
- Use **?fun** to get information on a function (replace **fun** with the name of a function).
- Use help.search("this that") to do a search for functions with keywords 'this' and 'that'.
- If you want to save figures, you can right-click over the figure and copy as metafile or bitmap and paste into a document (like Word or Powerpoint or whatever).

2 1 Instructions for running case studies using provided scripts

1.3 Symbols reminders

- A Matrix of biases for the observation model. Individual elements are A_i .
- \mathbf{E}_t The process errors. They are Normal(0,**Q**).
- $\boldsymbol{\eta}_t$ The observation errors. For this workshop, we assume they are Normal $(0, \mathbf{R})$.
- η_j^2 and η_j The measurement error variance and sd for observation time series j. We drop the j, when the same variance is shared across all observation time series.
- **Q** Variance-covariance matrix of process errors (**E**_t). Variances are on the diagonal and termed σ_i^2 .
- **R** Variance-covariance matrix of observation errors $(\boldsymbol{\eta}_t)$. Variances are on the diagonal and termed η_i^2 .
- σ_i^2 and σ_i The process variance and sd for subpopulation process *i*. We drop the *i*, when the same variance is shared across all subpopulations
- **U** Matrix of growth rates (population models) or drift rates (movement); u_i are the elements of **U**.
- \mathbf{X}_t Matrix of the subpopulation processes at time t. Individual subpopulation processes are termed x_t .
- x_t An individual subpopulation process at time t. In a univariate state-space model or when there is only one state process, only x_t appears.
- \mathbf{Y}_t Matrix of the observations at time t. Individual observations are termed y_t .
- y_t An individual observation at time t. In a univariate state-space model, only y_t appears. processes.

1.4 Case Study Guide

1.4.1 First write down (on paper) the matrix form for Z, U, Q, and R.

- *n* is the number of observation time series. You don't control this. It is determined by the number of observation time series (columns of data).
- m is the number of subpopulations. You specify this.
- **Z** is *n* rows by *m* columns. It is 0s and 1s and specifies which observation time series (row) is associated with which subpopulation (column).
- **U** is *m* rows by 1 column. Each row *i* is the *u* for subpopulation *i*; show which are shared.
- **Q** is *m* rows by *m* columns. The diagonal is the process variance (σ_i^2) for each subpopulation *i*; shows which are shared.
- **R** is *n* rows by *n* columns. The diagonal is the non-process or observation variances (η_i^2) for each observation time series *j*; show which are shared.
- A Don't worry about A; KalmanEM controls that.

1.4.2 Translate the Z, U, Q, and R forms into KalmanEM arguments

- Z is specified by whichPop. It has n elements (one for each observation time series) and specifies which observation time series belongs to which subpopulation (there are m and they are numbered $1, 2, 3, \ldots, m$). Every subpopulation must have at least 1 observation time series.
- U is specified by U.groups. It is a vector with m elements. It specifies which u are different.
- Q is specified by Q.groups and varcov.Q. Q.groups is a vector with m elements. varcov.Q is text in quotes ("diagonal", "equalvarcov", or "unconstrained")
- **R** is specified by **R.groups** and **varcov.R. Q.groups** is a vector with *n* elements. **varcov.R** is text in quotes ("diagonal", "equalvarcov", or "unconstrained")

1.5 KalmanEM tips

- The output from KalmanEM tells you the structure of the multivariate statespace model (MSSM) that it fit (so you can check your specifications) and the number of iterations required versus max.iter. If you reached the maximum, re run with max.iter set higher.
- If you misspecify the model, KalmanEM post an error that should give you an idea of the problem. Remember, the number of columns in your data is *n* and the maximum number in whichPop is *m*.
- Running KalmanEM with no arguments except your data (KalmanEM(dat)) will fit an unconstrained MSSM with m = n and a diagonal **R** matrix.

Case Study 1: Count-based PVA for data with observation error

2.1 The Problem

Estimates of extinction and quasi-extinction risk are an important risk metric used in the management and conservation of endangered and threatened species. By necessity, these estimates are based on data that contain both variability due to real year-to-year changes in the population growth rate (process errors) and variability in the relationship between the true population size and the actual count (observation errors). Classic approaches to extinction risk assume the data have only process error, i.e. no observation error. In reality, observation error is ubiquitous both because of the sampling variability and also because of year-to-year (and day-to-day) variability in sightability.

In this case study, we are use a Kalman filter to fit a univariate (meaning one time series) state-space model to count data for a population. We will compute the extinction risk metrics given in Dennis et al. (1991), however instead of using a process-error only model (as is done in the original paper), we use a model with both process and observation error. The risk metrics and their interpretations are the same as in Dennis et al. (1991). The only real difference is how we compute σ^2 , the process error variance. However this difference has a large effect on our risk estimates, as you will see.

In this case study, we use a density-independent model. Density-independence is often a reasonable assumption when doing a PVA because we do such calculations for at-risk populations that are either declining or that are well below historical levels (and presumably carrying capacity). In an actual PVA, it is necessary to justify this assumption and if there is reason to doubt the assumption, one tests for density-dependence (Taper and Dennis, 1994) and does sensitivity analyses using state-space models with density-dependence (Dennis et al., 2006).

The univariate model is written:

$$x_t = x_{t-1} + u + e_t \qquad \text{where } e_t \sim Norm(0, \sigma^2) \tag{2.1}$$

$$y_t = x_t + \epsilon_t$$
 where $\epsilon_t \sim Norm(0, \eta^2)$ (2.2)

6 2 CS1: Count-based PVA for data with observation error

where y_t is the logarithm of the observed population size at time t, x_t is the unobserved state at time t, u is the growth rate, and σ^2 and η^2 are the process and observation error variances, respectively. In the R code to follow, σ^2 is denoted \mathbf{Q} and η^2 is denoted \mathbf{R} (because the functions we are using are also for multivariate state-space models and those models use \mathbf{Q} and \mathbf{R} for the respective variance-covariance matrices).

2.2 Simulated data with process and observation error

We'll start by using simulated data to see the difference between data and estimates from a model with process error only versus a model that also includes observation error. For our simulated data, we'll used a decline of 5% per year, process variability of 0.01 (typical for big mammals), and a observation variability of 0.05 (which is a bit on the high end). We'll randomly set 10% of the values as missing. Here's the code:

Set things up.

First generate the population sizes using equation 2.1:

```
 \begin{array}{l} x[1]=init\\ for(t \ in \ 2:nYr)\\ x[t] = x[t-1] + \ sim.u \ + \ rnorm(1, \ mean=0, \ sd=sqrt(sim.Q)) \end{array}
```

Add observation error and missing values to generate the observed data using equation 2.2:

```
for(t in 1:nYr)
  y[t]= x[t] + rnorm(1,mean=0,sd=sqrt(sim.R))
missYears =
    sample(years[2:(nYr-1)],floor(fracmissing*nYr),replace = F)
y[missYears]=-99
```

Now let's look at the simulated data. Stochastic population trajectories show much variation, so it is best to look at a few at once. In figure 2.1, nine simulations from the identical parameters (above) are shown.



Fig. 2.1. Plot of nine simulated population time series with process and observation error. Circles are observation and the dashed line is the true population size.

Exercise 1

A good way to get a feel for reasonable σ^2 values is to generate simulated data and look at the time series. As a biologist, you probably have a pretty good idea of what kind of year-to-year population changes are reasonable for your species. For example for most of the mammalian species I work with, the maximum population yearly increase would be around 50% (the population could go from 1000 to 1500 in one year), but some of the fish species could easily double or even triple in a really good year. Your observed data may bounce around a lot for many different reasons having to do with sightability, sampling error, age-structure, etc., but the underlying population trajectory is constrained by the kinds of year-to-year changes in population size that are biologically possible for your species. σ^2 describes those true population changes.

Run the Exercise 1 code (in Case_Study_1.r) several times using different parameter values to get a feel for how different the time series can look based on identical parameter values. Typical vertebrate σ^2 values are 0.002 to 0.02, and typical η^2 values are 0.005 to 0.1. A u of -0.01 translates to an average 1%

per year decline and a u of -0.1 translates to an average 10% per year decline (approximately).

2.3 Parameter estimation

2.3.1 Maximum-likelihood estimates for a model with observation error

We put the simulated data through the Kalman-EM algorithm in order to estimate the parameters, u, σ^2 , and η^2 , and population sizes. These are the estimates using a model with process and observation variability. The function call is kem = KalmanEM(data), where data is a vector of logged (base e) counts with missing values denoted by -99. After this call, the ML parameter estimates are kem\$U, kem\$Q and kem\$R. There are numerous other outputs from the KalmanEM function. To get a list of the outputs type in names(kem). Note that kem is just a name; I could have called the output foo. Here's some code to fit to the simulated time series. The silent=T keeps the algorithm from outputting some model information that we won't need until Case Study 2.

kem = KalmanEM(y,silent=T)

Let's look at the parameter estimates for the nine simulated time series in figure 2.1 to get a feel for the variation. I used the KalmanEM function on each time series to produce parameter estimate for each simulation. The estimates are followed by the mean (over the nine simulations) and the true values:

kem.params

	kem.U	kem.Q	kem.R
sim 1	-0.01616165	0.0152894742	0.04940696
sim 2	-0.04131631	0.0008442845	0.06965247
sim 3	-0.04785627	0.0009932629	0.05106351
sim 4	-0.02104032	0.0265155965	0.03500475
sim 5	-0.04934044	0.0006964074	0.06492995
sim 6	-0.04722872	0.0069335675	0.03282802
sim 7	-0.06114349	0.0004053795	0.03911615
sim 8	-0.07147773	0.0078744489	0.03702980
sim 9	-0.05901010	0.0075138430	0.04600571
mean sim	-0.04606389	0.0074518071	0.04722637
true	-0.05000000	0.010000000	0.05000000

As expected, the estimate parameters do not exactly match the true parameters, but the average should be fairly close (although 9 simulations is a small sample size). Also note that although we don't get u quite right, our estimates are usually negative. Thus our estimates usually indicate declining dynamics. The Kalman-EM algorithm also gives an estimate of the true population size with observation error removed. This is in kem\$states. Figure 2.2 shows the KalmanEM estimated true states of the population over time as a solid line. Note that the solid line is considerably closer to the actual true states (dashed line) than the observations. On the other hand with certain datasets, the Kalman filter can get it quite wrong as well!



Fig. 2.2. The circles are the observed population sizes with error. The dashed lines are the true population sizes. The solid thin lines are the estimates of the true population size from the Kalman-EM algorithm

2.3.2 Maximum-likelihood estimates for a model with no observation error

We used the Kalman-EM algorithm to estimate the mean population rate uand process variability σ^2 under the assumption that the count data have observation error. However, the classic approach to this problem, referred to as the "Dennis model" (Dennis et al., 1991), uses a model that assumes the data have no observation error; all the variability in the data is assumed to 10 2 CS1: Count-based PVA for data with observation error

result from process error. This approach works fine if the observation error in the data is low, but not so well if the observation error is high. We will next fit the data using the classic approach so that we can compare and contrast parameter estimates from the different methods.

Using the estimation method in (Dennis et al., 1991), our data need to be re-specified as the observed population changes (delta.pop) between censuses along with the time between censuses (tau). We re-specify the data as follows:

```
den.years = years[y!=-99] # the non missing years
den.y = y[y!=-99] # the non missing counts
den.n.y = length(den.years)
delta.pop = rep(NA, den.n.y-1) # population transitions
tau = rep(NA, den.n.y-1) # step sizes
for (i in 2:den.n.y ){
        delta.pop[i-1] = den.y[i] - den.y[i-1]
        tau[i-1] = den.years[i]-den.years[i-1]
        } # end i loop
```

Next, we regress the changes in population size between censuses (delta.pop) on the time between censuses (tau) while setting the regression intercept to 0. The slope of the resulting regression line is an estimate of u, while the variance of the residuals around the line is an estimate of σ^2 . The regression is shown in Figure 2.3. Here is the code to do that regression:

```
den91 <- lm(delta.pop ~ -1 + tau)
# note: the "-1" specifies no intercept
den91.u = den91$coefficients
den91.Q = var(resid(den91))</pre>
```

Here are the parameters values for the data in figure 2.2 using the processerror only model:

```
den91.params
```

```
den91.U
                        den91.Q
         -0.01781257 0.11853167
sim 1
sim 2
         -0.07421083 0.15309662
         -0.06988188 0.11837037
sim 3
sim 4
         -0.02988104 0.10035714
         -0.04549535 0.12155298
sim 5
         -0.05568473 0.07500778
sim 6
sim 7
         -0.08119574 0.10425296
sim 8
         -0.07582527 0.08848537
sim 9
         -0.07849467 0.09951930
mean sim -0.05872023 0.10879713
         -0.05000000 0.01000000
true
```

Notice that the u estimates are similar to those from the Kalman-EM algorithm, but the σ^2 estimate (Q) is much larger. That is because this approach



Fig. 2.3. The regression of $log(N_{t+\tau}) - log(N_t)$ against τ . The slope is the estimate of u and the variance of the residuals is the estimate of Q.

treats all the variance as process variance, so any observation variance in the data is lumped into process variance (in fact it appears as $2 \times$ the observation variance).

Exercise 2

The code for exercise 2 (in Case_Study_1.r) generates multiple (nsim) simulated data sets and then estimates parameter values for each. It compares the Kalman-EM estimates to the estimates using a process error only model (i.e. ignoring the observation error). Here is an example of the output from the code:

		kem.U	den91.U	kem.Q	kem.R	den91.Q
sim	1	-0.0540	-0.0562	0.021789	0.0266	0.0790
sim	2	-0.0365	-0.0233	0.006976	0.0265	0.0743
sim	3	-0.0237	-0.0543	0.031379	0.0602	0.1604
sim	4	-0.0638	-0.0620	0.010875	0.0526	0.1166
sim	5	-0.0312	-0.0238	0.025014	0.0565	0.1678

12 2 CS1: Count-based PVA for data with observation error

sim 6	-0.0152	-0.0169	0.000509	0.0316	0.0618
sim 7	-0.0542	-0.0683	0.021001	0.0417	0.1115
sim 8	-0.0551	-0.0392	0.001619	0.0632	0.1552
sim 9	-0.0423	-0.0436	0.010969	0.0428	0.1149
mean si	im -0.0418	-0.0431	0.014459	0.0446	0.1157
true	-0.0500	-0.0500	0.010000	0.0500	0.0100

- 1. Re-run the parameter estimation on new data sets a few times to see the performance of the estimates using a state-space model (kem.) versus the model with no observation error (den91).
- 2. Alter the observation variance, sim.R in the data generation step in order to get a feel for performance as observations are further corrupted. What happens as error is increased?
- 3. Decrease the number of years of data, nYr and re-run the parameter estimation. What changes?

If you find that the exercise code takes too long to run, reduce the number of simulations (by reducing nsim in the code).

2.4 Probability of hitting a threshold $\Pi(x_d, t_e)$

A common extinction risk metric is 'the probability that a population will hit a certain threshold x_d within a certain time frame $t_e - if$ the observed trends continue'. Under this definition, we can compute $\Pi(x_d, t_e)$ using the stochastic population model (equation 2.1) and our estimate of the parameters of that model. In practice, the threshold used is not $N_e = 1$, which would be true extinction. Often a 'functional' extinction threshold will be used ($N_e >> 1$). Other times a threshold of 'a p_d fraction of current levels' is used. The latter is used because we often have imprecise information about the relationship between the true population size and what we measure in the field; many population counts are index counts. In these cases, one must use 'fractional declines' as the threshold. Also, extinction estimates that use an absolute threshold (like 100 individuals) are quite sensitive to error in the estimate of true population size. In this workshop, we are going to use fractional declines as the threshold, specifically $p_d = 0.1$ which means a 90% decline below the population size at the last census.

 $\Pi(x_d, t_e)$ is typically presented as a curve showing the probabilities of hitting the threshold (y-axis) over different time horizons (t_e) on the x-axis. Extinction probabilities can be computed through Monte Carlo simulations or analytically using equation 16 in Dennis et al. (1991) (note there is a typo in equation 16; the last + is supposed to be -). We will use the latter method:

$$\Pi(x_d, t_e) = \pi(u) \times \Phi\left(\frac{-x_d + |u|t_e}{\sqrt{\sigma^2 t_e}}\right) + \exp(2x_d|u|/\sigma^2) \Phi\left(\frac{-x_d - |u|t_e}{\sqrt{\sigma^2 t_e}}\right)$$
(2.3)

where x_e is the threshold and is defined as $x_e = log(N_0/N_e)$, where N_0 is the current population estimate and N_e is the threshold. If we are using fractional declines then $x_e = log(N_0/(p_d \times N_0)) = -log(p_d)$. $\pi(u)$ is the probability that the threshold is eventually hit (by $t_e = \infty$). $\pi(u) = 1$ if u <= 0 and $\pi(u) = \exp(-2ux_d/\sigma^2)$ if u > 0. $\Phi()$ is the cumulative probability distribution of the standard normal (mean = 0, sd = 1). Here is the R code for that computation (using a fractional decline threshold):

```
pd = 0.1 #means a 90 percent decline
tyrs = 1:100
xd = -log(pd)
p.ever = ifelse(u<=0,1,exp(-2*u*xd/Q)) #Q=sigma2
for (i in 1:100){
    Pi[i] = p.ever * pnorm((-xd+abs(u)*tyrs[i])/sqrt(Q*tyrs[i]))
    + exp(2*xd*abs(u)/Q) *
    pnorm((-xd - abs(u)* tyrs[i])/sqrt(Q*tyrs[i]))
    }
```

Figure 2.4 shows the estimated probabilities of hitting the 90% decline for the nine 30-year times series simulated with u = -0.05, $\sigma^2 = 0.01$ and $\eta^2 =$ 0.05. The dashed line shows the estimates using the Kalman-EM parameter estimates and the solid line shows the estimates using a process-error only model (the Dennis91 estimates). The circles are the true probabilities. The difference between the estimates and the true probabilities is due to errors in \hat{u} . Those errors are due largely to process error – not observation error. As we saw earlier, by chance population trajectories with a u < 0 will increase, even over a 30-year period. In this case, \hat{u} will be positive when in fact u < 0.

Looking at the figure, it is obvious that the probability estimates are highly variable. However, look at the first panel. This is the average estimate (over 9 simulations). Note that on average (over 9 simulations), the estimates are good. If we had averaged over 1000 simulations instead of 9, you would see that the Kalman-EM line falls on the true line. It is an unbiased predictor. While that may seem a small consolation if estimates for individual simulations are all over the map, it is important for correctly specifying our uncertainty about our estimates. Second, rather than focusing on how the estimates and true lines match up, see if there are any forecasts that seem better than others. For example, are 20-year predictions better than 50 and are 100-yr better or worse. In Exercise 3, you'll remake this with different u. You'll discover from that that populations in the worst shape (smallest u) have better predictions.

Exercise 3

Use the code from exercise 2 to re-create new parameter estimates (if needed).

1. Change sim.R and rerun exercise 2 and then run exercise 3. When are the estimates using the process-error only model (den91) worse and in what way are they worse?



Fig. 2.4. Plot of the true and estimated probability of declining 90% in different time horizons (the x axis) for nine simulated population time series with observation error.

- 2. You might imagine that you should always use a model that assumes that the data contain observation error, since in practice observations are never perfect. However, there is a cost to estimating that extra variance parameter and the cost is a more variable σ^2 (Q) estimate. Play with shortening the time series and decreasing the sim.R values. Are there situations when the 'cost' of the extra parameter is greater than the 'cost' of ignoring observation error?
- 3. How does changing the extinction threshold (pd) change the extinction probability curves? (Do not remake the data, i.e. don't rerun exercise 2)
- 4. How does changing the rate of decline (sim.u) change the estimates of risk? Rerun exercise 2 using a lower u; this will create a new matrix of parameter estimates. Then run the exercise 3 code. Do the estimates seem better of worse for rapidly declining populations?
- 5. Rerun exercise 2 using fewer number of years (nYr smaller) and increase fracmissing. Rerun exercise 2 to create the new parameter estimates. Then run the exercise 3 code. The graphs will start to look peculiar. Why

do you think it is doing that? Hint: look at the estimated parameters using params.

2.5 Certain and uncertain regions

From exercise 3, you've observed one of the problems with estimates of the probability of hitting thresholds. Looking over the 9 simulations, your risk estimates will be on the true line sometimes and other times they are way off. So your estimates are variable. Using only the point estimates of the probability of 90% decline by themselves in a PVA should not be done. At the minimum, CIs need to be added (next section), but even with CIs, the probability of hitting declines often doesn't capture our certainty and uncertainty about our risk estimates.

From exercise 3, you might have also noticed that there are some time horizons (10, 20 years) for which the estimate are highly certain (not hitting the threshold), while for other time horizons (30, 50 years) the estimates are all over the map. Put another way, you may be able to say with high confidence that a 90% decline will NOT occur between years 1 to 20 and that by year 100 it most surely will have occurred. However, between the years 20 and 100, you are very uncertain about the risk. The point is that you can be certain about some forecasts while at the same time being uncertain about other forecasts.

One way to show this is to plot the uncertainty as a function of the forecast, where the forecast is defined in terms of the forecast length (number of years) and forecasted decline (percentage). Uncertainty is defined as how much of the 0-1 range your 95% CI covers. Ellner and Holmes (2008) show such a figure (their figure 1). Figure 2.5 shows a version of this figure that you can produce with the function TMUfigure(u= val, N= val, s2p= val). In the figure, I used u = -0.05 which is a 5% per year decline, N = 25 so 25 years between the first and last census, and $s_p^2 = 0.01$. The process variability for big mammals is typically in the range of 0.002 to 0.02.

Exercise 4

Use the code for exercise 4 (in Case_Study_1.r) to re-create Figure 2.5 and get a feel for when (what parameter ranges) risk estimates are more certain and when they are less certain.

par(mfrow=c(1,1))
TMUfigure(N=30, u=-0.05, s2p=0.01)

N are the number of years of data, **u** is the mean population growth rate, and **s2p** is the process variance.



nyrs = 30 mu = -0.05 s2.p = 0.01

Fig. 2.5. This figure shows your region of high uncertainty (dark grey). In this region, the minimum 95% CIs (meaning if you had no observation error) span 80% of the 0 to 1 probability. That is, you are uncertain if the probability of a specified decline is close to 0 or close to 1. The green (dots) shows where your upper 95% CIs does not exceed P=0.05. So you are quite sure the probability of a specified decline is less than 0.05. The red (dots) shows where your lower 95% CIs is above P=.95. So you are quite sure the probability is greater than P=0.95. The light grey is between these two certain/uncertain extremes.

2.6 More risk metrics and some real data

The previous sections have focused on the probability of hitting thresholds because this is an important and common risk metric used in PVA and it appears in IUCN Red List criteria. However, as you have seen, there is high uncertainty associated with such estimates. Part of the problem is that probability is constrained to 0 to 1, and it is easy to get estimates with CIs that span 0 to 1. Other metrics of risk, \hat{u} and the distribution of the time to hit a threshold (Dennis et al., 1991), don't have this problem and may be more informative. Figure 2.6 shows different risk metrics from Dennis et al. (1991) on a single plot. This figure is generated by the call

riskfigure(datafile)

The datafile is the name of the data file, with column 1 = years and column 2 = population count. riskfigure() has a number of arguments that can be passed in to change the default behavior. te is the forecast length (default is 100 years), threshold is the extinction threshold either as an absolute number, if absolutethresh=T, or as a fraction of current population count, if absolutethresh=F. The default is absolutethresh=F and threshold=0.1. datalogged=T means the data are already logged; this is the default.



Fig. 2.6. Risk figure using data for the critically endangered African Wild Dog (data from Ginsberg et al. 1995). This population went extinct after 1992.

18 2 CS1: Count-based PVA for data with observation error

Exercise 5

Use the code for exercise 5 (in Case_Study_1.r) to re-create Figure 2.6. I've included some other data for you to run: prairechicken.txt from the endangered Attwater Praire Chicken, graywhales.txt from Gerber et al. (1999), and grouse.txt from the Sharptailed Grouse (a species of U.S. federal concern) in Washington State. If you have other textfiles of data, you can run those too. Just replace the datafile name and ensure that the data are in the same format as wilddogs.txt.

2.7 Confidence intervals

The figures produced by riskfigure() have confidence (95% and 75%) on the probabilities in the top right panel. The standard way to produce these CIs is via parametric bootstrapping. Here are the steps in a parametric bootstrap:

- You estimate u and σ^2 and η^2
- Then you simulate time series using those estimates and equations 2.1 and 2.2
- Then you re-estimate your parameters from the simulated data (using say KalmanEM(simdata)
- Repeat for 1000s of time series simulated using your estimated parameters. This gives you a large set of bootstrapped parameter estimates
- For each bootstrapped parameter set, compute a set of extinction estimates (you use equation 2.3 and code from exercise 3)
- The α% ranges on those bootstrapped extinction estimates gives you your α CIs on your probabilities of hitting thresholds

Look at the code in riskfigure.r to see how to do this in R.

For the workshop, producing our parameter estimates by estimating them from the simulated data would be far too slow. Therefore I used approximate CIs on the parameters using the inverse of a numerically estimated Hessian matrix. This uses an estimate of the variance-covariance matrix of the parameters from the inverse of a numerically estimated Hessian matrix. The function <code>riskfigure()</code> has an option you can set CI.method=c("hessian", "paramboot", "nonparamboot", "none") which tells it how to compute the CIs. For the workshop, I set CI.method="hessian". Using an estimated Hessian matrix to compute CIs is a handy trick that can be used for all sorts of maximum-likelihood parameter estimates. Look at the code in <code>riskfigure()</code> to see how to use the <code>nlme</code> package in *R* to do this very easily.

2.8 Other parameter estimation methods

Restricted maximum-likelihood algorithms are also available for state-space models, both univariate and multivariate (Staples et al., 2004; Hinrichsen,

2009). REML can give parameter estimates with lower variance than the Kalman-EM algorithm. Also the REML algorithm is much easier to code than the Kalman-EM algorithm (see code provided with the cited papers). However, the algorithms for REML when there are missing values are not currently available, so you are limited to data with no missing values (at the moment). Data with cycles, from age-structure or predator-prey interactions, are difficult to analyze and both REML and Kalman-EM will give poor estimates for this type of data. The slope method (Holmes, 2001), while more ad-hoc, is robust to those problems. Holmes et al. (2007) used the slope method in a large study of data from endangered and threatened species. Ellner and Holmes (2008) showed that the slope estimates are close to the theoretical minimum uncertainty. However estimates using the slope method are not easily extended to multi-site data. If you wish, you can run the slope method on the data in this case study by using the function slopemethod(logged.data); replace logged.data with your time series of data. The function will output u, σ^2 , and η^2 . See the reference list on the workshop website for a bibliography of papers on maximum-likelihood estimation of state-space models for ecological data.

My research is focused on Kalman-based and REML algorithms because of they are true maximum-likelihood methods, and the research I do on model selection requires that. However if I am doing a PVA and have a single time series with fewer than 25 years of data. I will often use the slope method because that method is less data-hungry. I am using the Kalman-based methods in this workshop because they allow one to easily study multi-site data and we don't have to worry about lots of missing values. One reason the EM algorithm is popular is that it is quite simple conceptually and if coded correctly, must increase in likelihood at each iteration. However, the EM algorithm is slower, sometimes much, much slower, than Newton-based methods. For any but the simplest model structures with few missing values, we have not had success getting Newton-based methods to work via the optim function in R. We have not tried creating a customized Newton method for our problems, in part because we are trying to write code for general model structures and in part, because it seemed hard. However, if you need a very fast algorithm, you should look into the research on Newton methods for state-space models. For our purposes, the Kalman-EM algorithm is fast enough and it is quite robust and likely to work on any data students might bring to our workshops.

Bayesians: Bayesian applications using state-space models to analyze population data are also well developed. See the reference list on the website for a summary of this literature. The MathBio group at Northwest Fisheries Science Center is actively developing and using Bayesian approaches also. You can find links to this code and research at my website: http://faculty.washington.edu/eeholmes.

Case study 2: Combining multi-site and subpopulation data to estimate trends and trajectories

3.1 The problem

In this example, we will use multivariate state-space models to combine surveys from multiple sites into one estimate of the average long term population growth rate and the year-to-year variability in that growth rate. Note this is not quite the same as estimating the 'trend'; 'trend' often means what population change happened, whereas the long-term population growth rate refers to the underlying population dynamics. We will use as our example a dataset from harbor seals in the Puget Sound, Washington, USA.

We have five regions where harbor seals were censused from 1978-1999 while hauled out of land¹. During the period of this dataset, harbor seals were recovering steadily after having been reduced to low levels by hunting prior to protection. The methodologies were consistent throughout the 20 years of the data but we do not know what fraction of the population that each region represents nor do we know the observation-error variance for each region. Given differences between behaviors of animals in different regions and the numbers of haul-outs in each region, the observation errors may be quite different. The regions have had different levels of sampling; the best sampled region has only 4 years missing while the worst has over half the years missing.

Figure 3.1 shows the data. The numbers on each line denote the different regions:

1 Str.JF 2 SJ.Islands 3 E.Bays 4 Puget.Snd 5 Hood.Canal

¹ Jeffries et al. 2003. Trends and status of harbor seals in Washington State: 1978-1999. Journal of Wildlife Management 67(1):208-219



Puget Sound Harbor Seal Surveys

Fig. 3.1. Plot of the of the count data from the five harbor seal regions (Jeffries et al. 2003). Each region is an index of the total harbor seal population, but the bias (the difference between the index and the true population size) for each region is unknown.

For this example, we will assume that the underlying population process is a stochastic exponential growth process with rates of increase that were not changing through 1978-1999. However, we are not sure if all five regions sample a single "total Puget Sound" population or if there are independent subpopulations. You are going to estimate the long-term population growth rate using different assumptions about the population structures (1 big population versus multiple smaller ones) and observation error structures to see how your assumptions change your estimates.

The data for this case study are stored in a comma-delimited file, Case_Study_2_data.csv and have already been log transformed. Read the data into R with the following commands:

d <- read.csv("Case_Study_2_data.csv",header=TRUE)
years = d[,1] #[,1] means all rows, column 1
dat = d[,2:ncol(d)]
n = ncol(dat)</pre>

The years (years) are in column 1 and the logged data (dat) are in the rest of the columns. The number of observation time series (n) is the number of columns in dat. Let's look at the first few years of data:

```
print(d[1:4,], digits=3)
```

	Years	$\operatorname{Str.JF}$	SJ.Islands	E.Bays	Puget.Snd	Hood.Canal
1	1978	6.03	6.75	6.63	5.82	6.6
2	1979	-99.00	-99.00	-99.00	-99.00	-99.0
3	1980	-99.00	-99.00	-99.00	-99.00	-99.0
4	1981	-99.00	-99.00	-99.00	-99.00	-99.0

The -99's in the data are missing values. The algorithm will ignore those values when estimating $x_{1:T}$.

3.2 First analysis: a single total Puget Sound population

The first step in a state-space modeling analysis is to specify the population structure and how the regions relate to that structure. The general state-space model is

$$\mathbf{X}_{t} = \mathbf{B}\mathbf{X}_{t-1} + \mathbf{U} + \mathbf{E}_{t}, \text{ where } \mathbf{E}_{t} \sim \mathrm{MVN}(0, \mathbf{Q})$$
(3.1)

$$\mathbf{Y}_t = \mathbf{Z}\mathbf{X}_t + \mathbf{A} + \boldsymbol{\eta}_t, \text{ where } \boldsymbol{\eta}_t \sim \mathrm{MVN}(0, \mathbf{R})$$
(3.2)

where all the bolded symbols are matrices. To specify the structure of the population and observations, we will specify what those matrices look like.

3.2.1 The population process, X, for analysis 1

For our first analysis, we assume that there is one population. When we are looking at trends over a large geographic region, we might make this assumption if we think animals are moving sufficiently that the whole area (multiple regions together) acts like a single population. We then write a model of the population abundance as:

$$n_t = exp(u + e_t)n_{t-1}, (3.3)$$

where n_t is the total count in year t, u is the mean population growth rate, and e_t is the deviation from that average in year t. We then take the log of both sides and write the model in log space:

$$x_t = x_{t-1} + u + e_t. (3.4)$$

 $x_t = \log n_t$. When there is one effective population, there is one x, there for \mathbf{X}_t is a 1×1 matrix. There is one population growth rate (u) and there is one process variance (σ^2) . Thus **U** and **Q** are 1×1 matrices.

24 3 CS2: Combining multi-site and subpopulation data

3.2.2 The observation process, Y, for analysis 1

For analysis 1, we assume that all five regional time series are observing this one population trajectory but they are scaled up or down relative to that trajectory. In effect, we think that animals are moving around a lot and our regional samples are some fraction of the population. There is year-to-year variation in the fraction in each region, just by chance. Notice that under this analysis, we don't think the regions represent independent subpopulations but rather independent observations of one population.

Our model for the data, $\mathbf{Y}_t = \mathbf{Z}\mathbf{X}_t + \mathbf{A} + \boldsymbol{\eta}_t$, is written out as:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \\ y_{4,t} \\ y_{5,t} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x_t + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \epsilon_{3,t} \\ \epsilon_{4,t} \\ \epsilon_{5,t} \end{bmatrix}$$
(3.5)

Each y_i is the time series for a different region (the names for the numbered regions are given on page 2). The A's are the bias between the regional sample and the total population. The A's are scaling (or intercept-like) parameters that are not important for trend estimation². We will ignore them ³. We allow that each region could have a unique observation variance and that the observation errors are independent between regions. Lastly, we assume that the observations errors on log(counts) are normal and thus the errors on (counts) are log-normal.⁴

We specify independent observation errors with unique variances by $\epsilon_t \sim MVN(0, \mathbf{R})$, where

$$\mathbf{R} = \begin{bmatrix} \eta_{1,t} & 0 & 0 & 0 & 0\\ 0 & \eta_{2,t} & 0 & 0 & 0\\ 0 & 0 & \eta_{3,t} & 0 & 0\\ 0 & 0 & 0 & \eta_{4,t} & 0\\ 0 & 0 & 0 & 0 & \eta_{5,t} \end{bmatrix}$$
(3.6)

Z is specifying which observation time series, $y_{i,1:T}$, is associated with which population trajectory, $x_{j,1:T}$. **Z** is like a look up table with 1 row for each of the *n* observation time series and 1 column for each of the *m* population

 $^{^2\,}$ To get rid of the A's, we scale multiple observation time series against each other; thus one A will be fixed at 0

³ Estimating the bias between regional indices and the total population is important for getting an estimate of the total population size. However, the time series analysis that we are doing for this workshop is not useful for estimating A's. Instead one uses some type of mark-recapture data. For trend estimation, the A's are not important. The regional observation variance captures increased variance due to a regional being a smaller sample of th total population.

⁴ The assumption of normality is not unreasonable since these regional counts are the sum of counts across multiple haul-outs.

trajectories. A 1 in row *i* column *j* means that observation time series *i* is measuring state process *j*. Otherwise the value in $\mathbf{Z}_{ij} = 0$. Since we have only 1 population trajectory, all the regions must be measuring that one population trajectory. Thus \mathbf{Z} is $n \times 1$.

3.2.3 Set the arguments for KalmanEM for analysis 1

Now that we have specified our state-space model, we set the arguments that will tell the function KalmanEM the structure of our model. First we need to tell the KalmanEM function that Z is a column vector of 1s (as in equation 3.5). We do this using the argument whichPop. whichPop is a $1 \times n$ vector where the *i*-th element specifies which population trajectory the *i*-th observation time series belongs to. Since there is only one population trajectory in analysis 1, whichPop is just a vector of 5 1's. Every observation time series is measuring the first, and only, population trajectory. In later analyses, you'll see how whichPop changes when we have subpopulations.

whichPop = c(1, 1, 1, 1, 1)

Next we specify that the \mathbf{R} variance-covariance matrix only has terms on the diagonal (the variances) and set the off-diagonals (the covariances) to zero.⁵

```
varcov.R = "diagonal"
```

That's it. KalmanEM has a number of other arguments we could set, but for this example, we only need to set these two.

3.2.4 Fit the model for analysis 1 to the data

We will send the data to the function KalmanEM and put the result in kem1. When we run KalmanEM, it will print information on the structure of the model it is fitting and how many iterations it took to run. If you haven't already, you need to source the KalmanEM function file by typing in source("KalmanEM.R"). After you have read in the data, type in the following to fit the model

```
kem1 = KalmanEM(dat, whichPop=c(1,1,1,1,1), varcov.R="diagonal")
Model Structure is
m: 1 state process(es)
n: 5 observation time series
whichPop: Observation time series assigned to state processes as 1 1 1 1 1
R: Observation errors are uncorrelated and have a diagonal var-cov matrix.
R.groups: Observation variances assigned to groups as 1 2 3 4 5
x00 is treated as fixed but unknown (estimated). V00=0 (but set larger for the EM algorithm
Finished in 15 interations. Max.iter was 5000.
```

⁵ For the EM function that we wrote for this workshop, the measurement errors must be uncorrelated if there are missing values in the data.

26 3 CS2: Combining multi-site and subpopulation data

The function will output some information about the model structure you are fitting. kem1 is a list of objects and names(kem1) shows the objects in it (this is a partial list; if you do it from R, you'll see the full list but watch out, it is long):

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [1,] "states" "States.se" "A" "B" "Q" "R" "U" "Kt"

kem1\$states are the maximum-likelihood estimates of "total harbor seal population" scaled to the first observation data series, and kem1\$states.se are the standard errors on those estimates. To get 95% CIs, use kem1\$states +/-1.96*kem1\$states.se. One of the biases, the As, cannot be estimated and arbitrarily KalmanEM choses $A_1 = 0$, so the population estimate is scaled to the first observation time series. Since we are only trying to estimate the trend, u, the unknown bias is unimportant. Figure 3.2 shows a plot of kem1\$states with its 95% CIs over the data. Because kem1\$states has been scaled relative to the first time series, it is on top of that time series.

To get the estimated long term population growth rate, type in

kem1\$U

Multiply by 100 to get the percent increase per year. The estimated process variance is given by

kem1\$Q

The log-likelihood of this model is

kem1\$loglike

We estimated 1 initial x (t = 0), 1 process variance, 1 U, 4 A's, and 5 observation variances's. So K = 12 parameters. The AIC of this model is $-2 \times loglike + 2K$, which we can show by typing

kem1\$AIC

After you do the analysis, add the estimates to the table at the end of this case study write-up.

3.3 Second analysis: constraining the observation variances

The variable kem1R contains the estimates of the observation error variances. It is a matrix. Here is **R** from analysis 1:

kem1\$R

 1:1:1
 1:2:1
 1:3:1
 1:4:1
 1:5:1

 1:1:1
 0.0317712
 0.0000000
 0.0000000
 0.0000000
 0.0000000

 1:2:1
 0.0000000
 0.03456110
 0.0000000
 0.0000000
 0.0000000



Observations and total population estimate

Fig. 3.2. Plot of the estimate of "In total harbor seals in Puget Sound" (minus the unknown bias for time series 1) against the data. The estimate of the total seal count has been scaled relative to the first time series. The 95% CIs on the population estimates are the dashed lines. These are not the CIs on the observations and the observations (the numbers) should not fall between the CI lines.

Notice that the variances along the diagonal are all different-we estimated 5 unique observation variances. We might be able to improve the fit (relative to the number of estimated parameters) by assuming that the observation variance is equal across regions but the errors are independent. This means we estimate 1 observation variance instead of 5. This is a fairly standard assumption for data that come from the same survey methodology⁶.

To impose this constraint, we set the argument R.groups for KalmanEM to

R.groups=c(1,1,1,1,1)

⁶ This is not a good assumption for these data since the number haul-outs in each region varies and the regional counts are the sums across all haul-outs in a region. We'll see that this is a poor assumption when we look at the AIC values.

28 3 CS2: Combining multi-site and subpopulation data

This tells KalmanEM that all the η^2 's along the diagonal in **R** are the same (the default is R.groups = c(1,2,3,4,5) which tells KalmanEM that all the η^2 's are different). To fit the model for analysis 2 to the data:

```
kem2 = KalmanEM(dat, whichPop=c(1,1,1,1,1),
varcov.R="diagonal", R.groups=c(1,1,1,1,1))
Model Structure is
m: 1 state process(es)
n: 5 observation time series
whichPop: Observation time series assigned to state processes as 1 1 1 1 1
R: Observation errors are uncorrelated and have a diagonal var-cov matrix.
R.groups: Observation variances assigned to groups as 1 1 1 1 1
x00 is treated as fixed but unknown (estimated). V00=0 (but set larger for the EM algorithm
Finished in 15 interations. Max.iter was 5000.
```

The new parameter estimates and log likelihood for this model are

[1] 3.528793

We estimated 1 initial x, 1 process variance, 1 U, 4 A's, and 1 observation variance. So K = 8 parameters. The AIC for this new model compared to the old model with 5 observation variances is:

c(kem1\$AIC,kem2\$AIC)

[1] -9.238935 8.942415

A smaller AIC means a better model. The difference between the 1 observation variance versus the unique observation variances is >10, suggesting that the unique observation variances model is better. Go ahead and type in the R code. Then add the parameter estimates to the table at the back.

One of the key diagnostics when you are comparing fits from multiple models, it to examine whether the model is flexible enough to fit the data. You do this by looking for temporal trends in the the residuals between the estimated population states (e.g. kem2\$states) and the data. In Fig. 3.3, the residuals for analysis 2 are shown. Ideally , these residuals should not have a temporal trend. They should look cloud-like. The fact that the residuals for analysis 2 have a strong temporal trend is an indication that our 1 population model is too restrictive for the data⁷.



Fig. 3.3. Analysis 2 residuals. The plots of the residuals should not have trends with time, but they do... This is an indication that the 1 population model is inconsistent with the data. The code to make this plot is given in the script file for case study 2.

3.4 Third analysis: North and South subpopulations

For the third analysis, we will change our assumption about the structure of the population. We will assume that there are 2 subpopulations, North and South, and that regions 1 and 2 (Strait of Juan de Fuca and San Juan) fall in

⁷ When comparing models via AIC, it is important that you only compare models that are flexible enough to fit the data. Fortunately, inadequate models will usually have very high AICs and fall out of the mix.

30 3 CS2: Combining multi-site and subpopulation data

the north subpopulation and regions 3, 4 and 5 fall in the south subpopulation. For this analysis, we will assume that these two subpopulations share their growth parameter, u, and process variance, σ^2 , since they share a similar environment and prey base. However we postulate that because of fidelity to natal rookeries for breeding, animals do not move much year-to-year between the north and south and the two subpopulations are independent.

We need to write the state-space model to reflect this population structure. There are two subpopulations, x_n and x_s , and they have the same growth rate u:

$$\begin{bmatrix} x_{n,t} \\ x_{s,t} \end{bmatrix} = \begin{bmatrix} x_{n,t-1} \\ x_{s,t-1} \end{bmatrix} + \begin{bmatrix} u \\ u \end{bmatrix} + \begin{bmatrix} e_{n,t} \\ e_{s,t} \end{bmatrix}$$
(3.7)

We specify that they are independent by specifying that their year-to-year population fluctuations (their process error) come from a multivariate normal with no covariance:

$$\begin{bmatrix} e_{n,t} \\ e_{s,t} \end{bmatrix} \sim MVN \left(mean = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, varcov = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \right)$$
(3.8)

For the observation process, we use a matrix to associate the regions with their respective x_n and x_s values:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \\ y_{4,t} \\ y_{5,t} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{n,t} \\ x_{s,t} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \epsilon_{3,t} \\ \epsilon_{4,t} \\ \epsilon_{5,t} \end{bmatrix}$$
(3.9)

3.4.1 Specifying the KalmanEM arguments for analysis 3

We need to change whichPop to specify that there are 2 subpopulations (north and south), and that regions 1 and 2 are in the north subpopulation and regions 3,4 and 5 are in the south subpopulation. We can write whichPop either with numbers or strings:

We want to specify that the *u*'s are the same for each subpopulation and that the σ^2 's are the same. To do this, we pass the arguments U.groups and Q.groups to KalmanEM. To specify that both our subpopulations share their parameters, we set

$$U.groups = c(1,1)$$

$$Q.groups = c(1,1)$$

This says that there is 1 u and σ^2 parameter and both subpopulations share it (if we wanted the u's to be different, we would use U.groups=c(1,2)).

3.5 Analyses 4-7: other population and observation error structures 31

We also want to specify the the \mathbf{Q} matrix is diagonal (no covariances). This means that the subpopulations are temporally independent (good and bad years aren't correlated):

varcov.Q = "diagonal"

Now we fit this model to the data:

```
kem3 = KalmanEM(dat, whichPop=c(1,1,2,2,2), U.groups=c(1,1),
varcov.Q="diagonal", Q.groups=c(1,1), R.groups=c(1,1,1,1,1),
varcov.R="diagonal")
Model Structure is
m: 2 state process(es)
n: 5 observation time series
whichPop: Observation time series assigned to state processes as 1 1 2 2 2
U.groups: State process growth rates assigned to groups as 1 1
Q: Process errors are uncorrelated and have a diagonal var-cov matrix.
Q.groups: State process variances assigned to groups as 1 1
R: Observation errors are uncorrelated and have a diagonal var-cov matrix.
R.groups: Observation variances assigned to groups as 1 1 1 1
x00 is treated as fixed but unknown (estimated). V00=0 (but set larger for the EM algorithm
Finished in 14 interations. Max.iter was 5000.
```

The output confirms the model structure and tells us how long it took to fit the model. We estimated 2 initial x's, 1 process variance, 1 U, 3 A's, and 1 observation variance. So K = 8 parameters. The Kalman filter requires an initial condition (t = 0) for each x time series. When m < n, the number of A's estimated is n - m since one of the A's for each state process will be set to 0. The AIC is 2*8 - 2*kem3\$loglike.

As before, the parameter estimates and AIC can be output by typing in kem3U, kem3Q, kem3R, kem3loglike, and kem3AIC. Go ahead and use R to output the parameter values and AICs. Then add the values to the table at the back of this hand-out.

Fig. 3.4 shows the residuals for the 2 subpopulations case. The residuals look better (more cloud-like) but the Hood Canal residuals are still temporally correlated.

3.5 Analyses 4-7: other population and observation error structures

Now work through a number of different structures and fill out the table at the back of this worksheet. At the end you'll see how your estimation of the mean population growth rate varies under different assumptions about the population and the data. All these analyses assume that the observation variances are unique at each site.

Each call to the KalmanEM algorithm is the same:



Fig. 3.4. The residuals for analysis 3. The plots of the residuals should not have trends with time. Compare with the residuals for analysis 2.

```
kem = KalmanEM(dat, whichPop=Z, varcov.Q="diagonal",
varcov.R="diagonal", U.groups=U.groups, Q.groups=Q.groups,
R.groups=R.groups)
```

Analysis 4: There are five subpopulations, and each site is sampling one of them. However, each subpopulation shares the same population parameters, u and σ^2 . How to set the arguments for that case. The run the code above. You'll want to give the output a separate name, e.g. kem4=KalmanEM(....

Z=c(1,2,3,4,5) U.groups=c(1,1,1,1,1) Q.groups=c(1,1,1,1,1) R.groups=c(1,2,3,4,5)

You can pass in R.groups=c(1,1,1,1,1) to make all the observation variances equal.

Analysis 5: The Strait of Juan de Fuca and San Juan Islands represent a Northern Puget Sound subpopulation, while the other three are sampling from a Southern Puget Sound subpopulation. But each site trajectory is allowed to have different population parameters, u and σ^2 . Write down the specification for Z, U.groups, and Q.groups. If you get stuck (or want to check your work), look in the R code for case study 2.

Analysis 6: Two subpopulations with different parameters, but the divisions are Hood Canal versus everywhere else.

Analysis 7: Three subpopulations with different parameters, but the divisions are North, South, Hood Canal.

Other things to try. You can set

```
varcov.Q = "unconstrained"
```

to allow the subpopulations to covary in time (i.e. not be temporally independent). You will need to delete Q.groups from your KalmanEM call because you can't have shared values in Q and all values different at the same time.

You can also set

varcov.Q = "equalvarcov"

to make all the subpopulations covary in time but with equal covariances and variances. Again remove Q.groups from your KalmanEM call if you use this setting.

You can set

R.groups = c(1,1,2,2,2)

to make regions 1 and 2 share an observation variance and regions 3,4,5 share a different observation variance.

3.6 Discussion

Case Study 2 shows you how to combine multiple datasets that are measuring the same underlying process and fit those data using a multivariate state-space framework. This allows you to combine data sets and use all the available data. You can also combine data that are discontinuous; that is data that don't overlap in time. For example, if you have data from one type of monitoring program in one set of years and then data from a different program starting in some later years, you can still easily estimate the population dynamics parameters using both sets of data.

There are a number of corners that we cut in order to have an example that runs quickly for a workshop:

• We ran the code starting from one initial condition. For a real analysis, you should start from a large number of random initial conditions and use the one that gives the highest likelihood. Since KalmanEM.r is a "hill-climbing" algorithm, this ensures that it does not get stuck on a local maxima. KalmanEM.r will do this for you if you pass it the argument MonteCarloInit = TRUE.

- 34 3 CS2: Combining multi-site and subpopulation data
- We assume independent observation and process errors. Depending on your system, observation errors may is driven by large-scale environmental factors (temperature, tides, prey locations) that would cause your observation errors to covary across regions. If your observation errors strongly covary between regions and you treat them as independent, this could be bad for your analysis. The current KalmanEM code will not handle covariance in **R** when there are missing data, but even it did, separating covariance across observation versus process errors will require much data (to have any power). In practice, the first step is to think hard about what drives sightability for your species and what are the relative levels of process and observation variance. You may be able to change your data in a way that will make the observation errors independent–for example, using data from different months or defining your "regions"
- We left the default tolerance, tol=0.01. You'll want to set this lower, e.g. tol=0.0001, for a real analysis. You'll need to up the max.iter argument correspondingly.
- We used the large-sample computation for AIC instead of an AIC that is designed to correct for small sample size in state-space models. The better metric, AICb, takes a long time to run. We could have shown AICc, which is the small-sample size corrector for non-state-space models. Type kem\$AICc to get that.

Finally, in a real (maximum-likelihood) analysis, one needs to be careful not to dredge the data. The temptation is to look at the data and pick a population structure that will fit that data. This can lead to including models in your analysis that have no biological basis. In practice, we spend a lot of time discussing the population structure with biologists working on the species and review all the biological data that might tell us what are reasonable structures. From that, a set of model structures to use are selected. Other times, a particular model structure needs to be used because the population structure is not in question rather it is a matter of using that (given) structure and all the data to get parameter estimates for forecasting (U, Q, R). Finally, other times, one wants to have a measure of the support the observed data give to all possible different population structures. That is a Bayesian question $(P(\Theta|data))$ and we would fit a model where **Q** is unconstrained and look at the posterior distribution of the elements in **Q**.

Results table

		pop. growth	process	num.	log-like	
An.		rate	variance	params	kem\$	AIC
		kem\$U	kem\$Q	kem\$K	loglike	kem\$AIC
1	one population					
	different obs. vars					
	uncorrelated					
2	one population					
	identical obs vars					
	uncorrelated					
3	N+S subpops					
	identical obs vars					
	uncorrelated;					
4	5 subpops					
	unique obs vars					
	u 's + σ^2 's identical					
5	N+S subpops					
	unique obs vars					
	u 's + σ^2 's identical					
6	PS + HC subpops					
	unique obs vars					
	u 's + σ^2 's unique					
7	N + S + HC subpops					
	unique obs vars					
	u 's + σ^2 's unique					

For AIC, only the relative differences matter. A difference of 10 between two AIC means substantially more support for the model with lower AIC. A difference of 30 or 40 between two AICs is very large.

Questions

- Do different assumptions about whether the measurement error variances are all identical versus different affect your estimate of the trend? You may want to rerun cases 3-7 with the R.groups specification changed. R.groups=c(1,2,3,4,5) means measurement variances all different versus R.groups=c(1,1,1,1,1).
- 2. Do assumptions about the underlying structure of the population affect your estimates of trend? Structure here means number of subpopulations and which areas are in which subpopulation. Try changing 'state parameters differ' to 'state params identical' for analyses 5-7.

- 36 3 CS2: Combining multi-site and subpopulation data
- 3. The CIs for the first two analyses are very tight because the estimate process variance was very small, kem1\$Q. Why do you think σ^2 was forced to be so small? [Hint: We are forcing there to be 1 and only 1 true process and all the observation time series have to fit that one time series. Look at the AICs too.]

Case Study 3: Using MARSS models to identify spatial population structure and covariance

4.1 The problem

Some of our previous case studies this morning have utilized pieces of the harbor seal (*Phoca vitulina*) dataset; in this example we use time series of observations from 9 sites to examine spatial structure for the entire west coast population of harbor seals (Jeffries et al., 2003).

Harbor seals are distributed along the west coast of the US. The populations in Oregon and Washington have been surveyed for > 25 years at a number of haul-out sites (Figure 4.1). In general, these populations have been increasing steadily since the 1972 (Marine Mammal Protection Act). It remains unknown whether they are at carrying capacity. For management purposes, 2 stocks are recognized: the coastal stock consists of 4 sites (Northern/Southern Oregon, Coastal Estuaries, Olympic Peninsula), and the inland WA stock consists of the remaining 5 sites (Figure 4.1). Subtle differences exist in the demographics across sites (e.g. pupping dates), however mtDNA analyses and tagging studies have suggested that these sites may be structured on a much larger scale. Harbor seals are known for strong site fidelity, but at the same time travel large distances to forage. Our goal for this case study is to address the following questions about spatial structure: 1) Does population abundance data support the existing management boundaries, or are there alternative groupings that receive more support? and 2) Does the Hood Canal site represent a distinct subpopulation?

4.2 Analysis for question 1: how many distinct subpopulations?

For this analysis, we will analyze the support for five hypotheses about the population structure. These do not represent all possible structures but instead represent those that are considered most biologically plausible given the geography and the behavior of harbor seals.

38 4 CS3: Using MARSS models to identify spatial population structure and covariance



Fig. 4.1. Map of spatial distribution of 9 harbor seal sites in Washington and Oregon.

Hypothesis 1 Sites are grouped by stock (m = 2), unique process errors Hypothesis 2 Sites are grouped by stock (m = 2), same process error Hypothesis 3 Sites are grouped by state (m = 2), unique process errors Hypothesis 4 Sites are grouped by state (m = 2), same process error Hypothesis 5 All sites are part of the same paramictic population (m = 1)

Aerial survey methodology has been relatively constant across time and space, and we will assume that all sites have the same constant (and independent) observation error variance for all sites.

4.2.1 Specify the design, Z, matrices

Write down the **Z** matrices for the hypotheses. Hint: Hypothesis 1 and 2 have the same **Z** matrix, Hypothesis 3 and 4 have the same **Z** matrix and Hypothesis 5 is a column of 1s.



4.2 Analysis for question 1: how many distinct subpopulations? 39

Next you need to specify whichPop so that KalmanEM knows the structure of your $\mathbf{Z}\xspace{'s.space{-1.5}}$ so that KalmanEM knows the structure of your $\mathbf{Z}\xspace{'s.space{-1.5}}$

- Hypothesis 1 and 2: whichPop=
- Hypothesis 3 and 4: whichPop=
- Hypothesis 5: whichPop=

4.2.2 Specify the grouping arguments

For this case study, we will assume that subpopulations share the same growth rate. What should U.groups look like for each hypothesis? Recall that U.groups is length m and specifies which subpopulations share their u parameter. Written in R it takes the form c(#,#,...)

- Hypothesis 1-4: U.groups=
- Hypothesis 5: U.groups=

What about Q.groups? Q.groups is also length m and specifies which subpopulations share their process variance parameter. To specify Q.groups, look at each hypothesis (above).

- Hypothesis 1: Q.groups=
- Hypothesis 2: Q.groups=
- Hypothesis 3: Q.groups=
- Hypothesis 4: Q.groups=
- Hypothesis 5: Q.groups=

Lastly, specify R.groups. As we mentioned above, we will assume that the observation variance is the same across sites. R.groups is length n.

• Hypothesis 1-5: R.groups=

40 4 CS3: Using MARSS models to identify spatial population structure and covariance

4.2.3 Fit models and summarize results

Fit each model for each hypothesis to the seal data (look at the script Case_Study_3.r for the code to load the data). Each call to KalmanEM will look like

kem = KalmanEM(sealData, varcov.Q = "diagonal", varcov.R = "diagonal", whichPop = whichPop, U.groups = U.groups, Q.groups = Q.groups, R.groups = R.groups)

We set both varcov.Q and varcov.R to diagonal so that there is no covariance between process errors and between measurement errors.

Fill in the following table, by fitting the five state-space models – that you have defined for the five hypotheses – to the harbor seal data (using KalmanEM). Use the Case_Study_3.r script so you don't have to type in all the commands.

	pop. growth	process	obs.	num.	log-	
$ \mathbf{H} $	rate	variance	variance	params	like.	AIC
	kem\$U	kem\$Q	kem\$R	kem\$K	kem\$loglike	kem\$AIC
1						
┝						
2						
3						
4						
┝						
5						
ľ						

4.2.4 Interpret results for question 1

What do these results indicate about the process error grouping, and spatial grouping? A lower AIC means a more parsimonious model (highest likelihood given the number of parameters). A difference of 10 between AICs is large, and means the model with the higher AIC is unsupported relative to the model with lower AIC.

Extra analysis (if you have time): Do your results change if you assume that observation errors are independent but have unique variances? The 9 sites have different numbers of haul-outs and so the observation variances might be different. Repeat the analysis with unique observation variances for each site (this means changing R.groups). You can also try the analysis with temporally co-varying subpopulations (good and bad years correlated) by setting varcov.Q="unconstrained".

4.3 Analysis for question 2: Is Hood Canal separate?

The Hood Canal site may represent a distinct population, and has recently been subjected to a number of catastrophic events (hypoxic events, possibly leading to reduced prey availability, and several killer whale predation events, removing up to 50% of animals per occurrence). Build four models, assuming that each site (other than Hood Canal) is assigned to its current management stock, but Hood Canal is allowed to be a different subpopulation (m = 3). Again, assume observation error is independent and constant across sites.

Hypothesis 1 Subpopulations have a shared process error and shared growth rate

Hypothesis 2 Each subpopulation has a unique process error and growth rate

Hypothesis 3 Hood Canal has the same process error, but different growth rate

Hypothesis 4 Hood Canal has unique process error and unique growth rate

4.3.1 Specify the Z matrix and whichPop

The **Z** matrix for each hypothesis is the same. The coastal subpopulation consists of 4 sites (Northern/Southern Oregon, Coastal Estuaries, Olympic Peninsula), the Hood Canal subpopulation is the Hood Canal site, and the inland WA subpopulation consists of the remaining 4 sites. Thus m = 3 and **Z** is a 9×3 matrix:



42 4 CS3: Using MARSS models to identify spatial population structure and covariance

Then write down whichPop for this **Z**.

4.3.2 Specify which parameters are shared across which subpopulations

U.groups specifies which u are shared across subpopulations. Look at the hypothesis descriptions above which will specify whether subpopulations share their population growth rate or have unique population growth rates.

- Hypothesis 1: U.groups=
- Hypothesis 2: U.groups=
- Hypothesis 3: U.groups=
- Hypothesis 4: U.groups=

Once you have more than 2 subpopulations, it can get hard to keep straight which U.groups= number goes to which subpopulation. It is best to sketch your Z matrix (which tells you which site in the rows corresponds to which subpopulation in the columns). Then remember that the elements of U.groups correspond 1 to 1 with the columns of Z:

U.groups=c(col 1 Z, col 2 Z, col 3 Z, ..).

Specify **Q.groups** showing which subpopulations share their process variance parameter.

- Hypothesis 1: Q.groups=
- Hypothesis 2: Q.groups=
- Hypothesis 3: Q.groups=
- Hypothesis 4: Q.groups=

R.groups is the same as for Question 1; the observation variances are the same for each site.

4.3.3 Fit the models and summarize results

Fit each model for each hypothesis to the seal data (look at the script Case_Study_3.r for the code to load the data). Each call to KalmanEM will look like

```
kem = KalmanEM(sealData, varcov.Q = "diagonal", varcov.R = "di-
agonal", whichPop = whichPop, U.groups = U.groups, Q.groups = Q.groups,
R.groups = R.groups)
```

	pop. growth			num.	log-like	
H	rate	proc. variance	obs. variance	params	kem\$	AIC
	kem\$U	kem\$Q	kem\$R	kem\$K	loglike	kem\$AIC
1						
2						
3						
4						

4.3 Analysis for question 2: Is Hood Canal separate?

4.3.4 Interpret results for question 2

How do the residuals for the Hood Canal site compare from these models relative to the best model from Question 1? Hint: If you have the vector of estimated population states (Xpred = t(kem\$states)) and the data (Xobs = sealData), the residuals for site i can be plotted in R as:

```
Xpred = t(kem$states)
Xobs = sealData
plot(Xpred[, whichPop[i]] - Xobs[,i],ylab="Predicted-Observed Data")
```

In R, if you have a matrix Y[1:numYrs, 1:n], you can extract column j by writing Yj = Y[, j].

Relative to the previous models from Question 1, do these scenarios have better or worse AIC scores (smaller AIC is better)? If you were to provide advice to managers, would you recommend that the Hood Canal population is a source or sink? What implications does this have for population persistence?

43

Case Study 5: Using state-space models to analyze noisy animal tracking data

5.1 Background: a simple random walk model of animal movement

A simple random walk model of movement with drift but no correlation is

$$x_{1,t} = x_{1,t-1} + u_1 + e_{1,t}, \quad e_{1,t} \sim Normal(0,\sigma_1^2)$$

$$x_{2,t} = x_{2,t-1} + u_2 + e_{2,t}, \quad e_{2,t} \sim Normal(0,\sigma_2^2)$$
(5.1)

where $x_{1,t}$ is the location at time t along one axis (in our case study, longitude) and $x_{2,t}$ is for another, generally orthogonal, axis (in our case study, latitude). We add measurement errors to our observations of location:

$$y_{1,t} = x_{1,t} + a_1 + \epsilon_{1,t}, \quad \epsilon_{1,t} \sim Normal(0, \eta_1^2)$$

$$y_{2,t} = x_{2,t} + a_2 + \epsilon_{2,t}, \quad \epsilon_{2,t} \sim Normal(0, \eta_2^2),$$
(5.2)

Together Equations 5.2 and 5.3 are an MSSM (now written in matrix form):

$$\mathbf{X}_t = \mathbf{X}_{t-1} + \mathbf{U} + \mathbf{E}_t, \ \mathbf{E}_t \sim MVN(0, \mathbf{Q})$$
(5.3)

$$\mathbf{Y}_t = \mathbf{X}_t + \mathbf{A} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim MVN(0, \mathbf{R}).$$
(5.4)

5.2 The problem

Loggerhead sea turtles (*Caretta caretta*) are listed as threatened under the United States Endangered Species Act of 1973. Over the last ten years, a number of state and local agencies have been deploying ARGOS tags on loggerhead turtles on the east coast of the United States. We have data on eight individuals over that period. However, we have a "Bad Tag" problem. Our latitude and longitude data has been corrupted by errors (Figure 5.1) and it 46 5 CS5: Analyzing animal tracking data

would appear that our sea turtles are becoming land turtles (at least part of the time).

For this case study, we will use KalmanEM to estimate true positions and speeds from the corrupted data. We will a couple extra package to plot and analyze the results: the maps and mvtnorm packages. If you have not already, install these packages by selecting the 'Packages' menu and then 'Install packages' and then select the two packages. If you are on a Mac, remember to select "binaries" for the package type.



Fig. 5.1. Plot of the of the tag data from the turtle Big Mama. Errors in the location data make it seem that Big Mama has been moving overland.

5.3 Using the Kalman-EM algorithm to estimate locations from bad tag data

5.3.1 Read in the data and load maps packages

Our noisy data are in the file loggerhead_noisy.csv. They consist of daily readings of location (longitude/latitude).

5.3 Using the Kalman-EM algorithm to estimate locations from bad tag data 47

loggerhead = read.csv("loggerhead_noisy.csv",header=T) # read in the data

The data are recorded daily and KalmanEM requires an entry for each day. The data look like so

```
loggerhead[1:6,]
```

	turtle	$\tt month$	day	year	lon	lat
1	BigMama	5	28	2001	-81.45989	31.70337
2	BigMama	5	29	2001	-80.88292	32.18865
3	BigMama	5	30	2001	-81.27393	31.67568
4	BigMama	5	31	2001	-81.59317	31.83092
5	BigMama	6	1	2001	-81.35969	32.12685
6	BigMama	6	2	2001	-81.15644	31.89568

And the file has data for 8 turtles:

```
levels(loggerhead$turtle)
```

[1] "BigMama" "Bruiser" "Humpty" "Isabelle" "Johanna"
[6] "MaryLee" "TBA" "Yoto"

We will first analyze the position data for "Big Mama". We put the data for "Big Mama" into variable dat:

```
turtlename="BigMama"
dat = loggerhead[which(loggerhead$turtle==turtlename),5:6]
```

5.3.2 Use KalmanEM to estimate the position of Big Mama

We will begin specifying the structure of the MARSS model used for animal movement and then use KalmanEM to fit that model to the data for each individual. There are two state processes (one for latitude and the other for longitude). There is one observation time series for each so

whichPop=c(1,2)

We'll assume that the errors are independent (you can try something different if you want) and that there are different drift rates, process variances and measurement variances for latitude and longitude (again you can try something different):

```
varcov.Q="diagonal"
varcov.R="diagonal"
Q.groups=c(1,2)  # separate process errors to do scale diffs
R.groups=c(1,2)  # separate measurement errors
U.groups=c(1,2)  # separate directional drifts
```

Fit the model to the data:

```
kem = KalmanEM(dat, varcov.Q=varcov.Q, varcov.R=varcov.R,
whichPop=whichPop, U.groups=U.groups, Q.groups=Q.groups,
R.groups=R.groups, max.iter=10000, silent=T)
```

48 5 CS5: Analyzing animal tracking data

5.3.3 Compare KalmanEM estimates to the real positions

The real locations (from which loggerhead_noisy.csv was produced by adding noise) are in loggerhead.csv. In Figure 5.2, we compare the tracks estimated from the noisy data with the original, good, data (see the *R* script, Case_Study_5.r for the code to make this plot. There are only a few data points for the real data because the real tag data has lots of missing days.



Fig. 5.2. Plot of the estimated track of the turtle Big Mama versus the good location data (before we corrupted it with noise).

5.3.4 Estimate speeds for each turtle

Turtle biologists designated one of these loggerheads "Big Mama," presumably for her size and speed. For each of the 8 turtles, estimate the average miles traveled per day. To calculate the distance traveled by a turtle each day, you use the estimate (from KalmanEM) of the lat/lon location of turtle at day tand at day t - 1. To calculate distance traveled in miles from lat/lon start and finish locations, we will use the function GCDF defined at the beginning of the R script, Case_Study_5.r): 5.3 Using the Kalman-EM algorithm to estimate locations from bad tag data 49

distance[i-1]=GCDF(pred.lon[i-1],pred.lon[i],pred.lat[i-1],pred.lat[i])

pred.lon and pred.lat are the predicted longitudes and latitudes from KalmanEM. To calculate the distances for all days, we put this through a for loop:

```
distance = array(-99, dim=c(dim(dat)[1]-1,1))
for(i in 2:dim(dat)[1])
    distance[i-1]=GCDF(pred.lon[i-1],pred.lon[i],pred.lat[i-1],pred.lat[i])
```

Take the mean (mean(distance) to get the average distance per day. We can also make a histogram of the distances traveled per day:

```
par(mfrow=c(1,1))
hist(distance) #make a histogram of distance traveled per day
```

(1)

Histogram of distance

Fig. 5.3. Histogram of the miles traveled per day for Big Mama.

Compare this to the estimate of miles traveled per day if you had not accounted for measurement errors (using the Kalman-EM algorithm). See the script file, Case_Study_5.r, for the code.

50 5 CS5: Analyzing animal tracking data



If you were given the opportunity to race these turtles, would you bet on Big Mama being the fastest?

5.4 Comparing turtle tracks to proposed fishing areas

One of the greatest threats to the long term viability of loggerhead turtles is incidental take by net/pot fisheries. After returning from ESA this year, you rave to your advisor/boss about how great state space models are. A week later, she promptly volunteers you to serve on a review team, providing advice to sea turtle managers about the potential impact of two potential new fishery areas on sea turtle bycatch. To add the fishing areas, to your turtle plots:

Given that only one area can be chosen as a future fishery, what do your predicted movement trajectories for our 8 turtles tell you?

5.5 Using fields to get density plots of locations

If you are comfortable programming in R, load the fields package. Make 3D density plots of predicted sea turtle locations. Which two areas appear to be most visited?

Include the confidence interval estimates for each location in this analysis. For this part of the exercise, we will assume that the confidence intervals are roughly the same as the probability intervals (Bayesian). We can assume that the error in latitude is independent from error in longitude. The **fields** package includes a couple different functions. One that might be useful here is **Tps**(), like in the example (**?fields**). To call **fields**, we need predictor variables (X), which can be random lon/lat pairs randomly drawn within the range of the data. The other requirement for **Tps**() is the response, y. If we think of each predicted state being a bivariate normal density, the response for each of our random pairs can be the density across all of the predicted states. There is code to help you get started in the R file.

5.6 Using specialized packages to analyze tag data

If you have tag data to analyze, you should use a state-space modeling package that has all the bells and whistles for that kind of data. There a number of R packages available for this. These are a couple we have come across:

UKFSST http://www.soest.hawaii.edu/tag-data/tracking/ukfsst/ KFTRACK http://www.soest.hawaii.edu/tag-data/tracking/kftrack/

kftrack is a full-featured toolbox for analyzing tag data with extended Kalman filtering. It incorporates a number of extensions that are important for analyzing track data: barriers to movement such as coastlines and non-Gaussian movement distributions. With kftrack, you can use the real tag data which has big gaps, i.e. days with no location. KalmanEM will struggle with these data because it will estimate states for all the unseen days; kftrack only fits to the seen days.

To use kftrack to fit the turtle data, type

To look at what the kftrack model consists of, type

model

Code

The code in the function KalmanEM() was written by Elizabeth Holmes and Eric Ward, who are research scientists with the Northwest Fisheries Science Center, part of NOAA Fisheries. You are welcome to use the code and adapt it with attribution. It may not be used in any commercial applications. This code is an amalgamation of a series of functions in an R package we are developing for fitting state-space models via maximum-likelihood and Bayesian approaches. Links to lots more code can be found by following the links at E. Holmes' website http://faculty.washington.edu/eeholmes Links to our papers that use these methods can also be found at the same website. The function TMUfigure is based on code by Steve Ellner and Elizabeth Holmes. The function riskfigure was written by Elizabeth Holmes.

Textbooks and articles that use state-space modeling

Textbooks Describing the Estimation of Process and Non-process Variance

There are many textbooks on Kalman filtering and estimation of state-space models. The following are a sample of books that are probably more accessible for those interested in population modeling.

Shumway, R. H., and D. S. Stoffer. 2000. Time series analysis and its applications. Springer-Verlag, New York, New York, USA.

Harvey, A. C. 1989. Forecasting, structural time series models and the Kalman filter. Cambridge University Press, Cambridge, UK.

Durbin, J., and S. J. Koopman. 2001. Time series analysis by state space methods. Oxford University Press, Oxford.

King, R., G. Olivier, B. Morgan, and S. Brooks. 2009. Bayesian Analysis for Population Ecology.

Giovanni, P., S. Petrone, and P. Campagnoli. 2009. Dynamic Linear Models in R.

Pole, A., M. West, and J. Harrison. 1994. Applied Bayesian Forecasting and Time Series Analysis.

Bolker, B. 2008. Ecological Models and Data in R. Last chapters.

Maximum-likelihood papers

This is just a sample of the papers from the population modeling literature.

de Valpine, P. 2002. Review of methods for fitting time-series models with process and observation error and likelihood calculations for nonlinear, non-Gaussian state-space models. Bulletin of Marine Science 70:455-471.

de Valpine, P. and A. Hastings. 2002. Fitting population models incorporating process noise and observation error. Ecological Monographs 72:57-76. 56 7 Textbooks and articles that use state-space modeling

de Valpine, P. 2003. Better inferences from population-dynamics experiments using Monte Carlo state-space likelihood methods. Ecology 84:3064-3077.

de Valpine, P. and R. Hilborn. 2005. State-space likelihoods for nonlinear fisheries time series. Canadian Journal of Fisheries and Aquatic Sciences 62:1937-1952.

Dennis, B., J.M. Ponciano, S.R. Lele, M.L. Taper, and D.F. Staples. 2006. Estimating density dependence, process noise, and observation error. Ecological Monographs 76:323-341.

Ellner, S.P. and E.E. Holmes. 2008. Resolving the debate on when extinction risk is predictable. Ecology Letters 11:E1-E5.

Hinrichsen, R.A. and E.E. Holmes. 2009. Using multivariate state-space models to study spatial structure and dynamics. In Spatial Ecology (editors Robert Stephen Cantrell, Chris Cosner, Shigui Ruan). CRC/Chapman Hall.

Hinrichsen, R.A. 2009. Population viability analysis for several populations using multivariate state-space models. Ecological Modelling 220:1197-1202.

Holmes, E.E. 2001. Estimating risks in declining populations with poor data. Proceedings of the National Academy of Sciences of the United States of America 98:5072-5077.

Holmes, E.E. and W.F. Fagan. 2002. Validating population viability analysis for corrupted data sets. Ecology 83:2379-2386.

Holmes, E.E. 2004. Beyond theory to application and evaluation: diffusion approximations for population viability analysis. Ecological Applications 14:1272-1293.

Holmes, E.E., W.F. Fagan, J.J. Rango, A. Folarin, S.J.A., J.E. Lippe, and N.E. McIntyre. 2005. Cross validation of quasi-extinction risks from real time series: An examination of diffusion approximation methods. U.S. Department of Commerce, NOAA Tech. Memo. NMFS-NWFSC-67, Washington, DC.

Holmes, E.E., J.L. Sabo, S.V. Viscido, and W.F. Fagan. 2007. A statistical approach to quasi-extinction forecasting. Ecology Letters 10:1182-1198.

Kalman, R.E. 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82:35-45.

Lele, S.R. 2006. Sampling variability and estimates of density dependence: a composite likelihood approach. Ecology 87:189-202.

Lele, S.R., B. Dennis, and F. Lutscher. 2007. Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods. Ecology Letters 10:551-563.

Lindley, S.T. 2003. Estimation of population growth and extinction parameters from noisy data. Ecological Applications 13:806-813.

Ponciano, J.M., M.L. Taper, B. Dennis, S.R. Lele. 2009. Hierarchical models in ecology: confidence intervals, hypothesis testing, and model selection using data cloning. Ecology 90:356-362.

Staples, D.F., M.L. Taper, and B. Dennis. 2004. Estimating population trend and process variation for PVA in the presence of sampling error. Ecology 85:923-929.

Bayesian papers

This is a sample of the papers from the population modeling and animal tracking literature.

Buckland, S.T., K.B. Newman, L. Thomas and N.B. Koestersa. 2004. State-space models for the dynamics of wild animal populations. Ecological modeling 171:157-175.

Calder, C., M. Lavine, P. Müller, J.S. Clark. 2003. Incorporating multiple sources of stochasticity into dynamic population models. Ecology 84:1395-1402.

Chaloupka, M. and G. Balazs. 2007. Using Bayesian state-space modelling to assess the recovery and harvest potential of the Hawaiian green sea turtle stock. Ecological Modelling 205:93-109.

Clark, J.S. and O.N. Bjørnstad. 2004. Population time series: process variability, observation errors, missing values, lags, and hidden states. Ecology 85:3140-3150.

Jonsen, I.D., R.A. Myers, and J.M. Flemming. 2003. Meta-analysis of animal movement using state space models. Ecology 84:3055-3063.

Jonsen, I.D, J.M. Flemming, and R.A. Myers. 2005. Robust state-space modeling of animal movement data. Ecology 86:2874-2880.

Meyer, R. and R.B. Millar. 1999. BUGS in Bayesian stock assessments. Can. J. Fish. Aquat. Sci. 56:1078-1087.

Meyer, R. and R.B. Millar. 1999. Bayesian stock assessment using a statespace implementation of the delay difference model. Can. J. Fish. Aquat. Sci. 56:37-52.

Meyer, R. and R.B. Millar. 2000. Bayesian state-space modeling of agestructured data: fitting a model is just the beginning. Can. J. Fish. Aquat. Sci. 57:43-50.

Newman, K.B., S.T. Buckland, S.T. Lindley, L. Thomas, and C. Fernández. 2006. Hidden process models for animal population dynamics. Ecological Applications 16:74-86.

Newman, K.B., C. Fernández, L. Thomas, and S.T. Buckland. 2009. Monte Carlo inference for state-space models of wild animal populations. Biometrics 65:572-583

Rivot, E., E. Prévost, E. Parent, and J.L. Baglinière. 2004. A Bayesian state-space modelling framework for fitting a salmon stage-structured population dynamic model to multiple time series of field data. Ecological Modeling 179:463-485.

Schnute, J.T. 1994. A general framework for developing sequential fisheries models. Canadian J. Fisheries and Aquatic Sciences 51:1676-1688.

Swain, D.P., I.D. Jonsen, J.E. Simon, and R.A. Myers. 2009. Assessing threats to species at risk using stage-structured state-space models: mortality trends in skate populations. Ecological Applications 19:1347-1364.

58 7 Textbooks and articles that use state-space modeling

Thogmartin, W.E., J.R. Sauer, and M.G. Knutson. 2004. A hierarchical spatial model of avian abundance with application to cerulean warblers. Ecological Applications 14:1766-1779.

Trenkel, V.M., D.A. Elston, and S.T. Buckland. 2000. Fitting population dynamics models to count and cull data using sequential importance sampling. J. Am. Stat. Assoc. 95:363-374.

Viljugrein, H., N.C. Stenseth, G.W. Smith, and G.H. Steinbakk. 2005. Density dependence in North American ducks. Ecology 86:245-254.

Ward, E.J., R. Hilborn, R.G. Towell, and L. Gerber. 2007. A state-space mixture approach for estimating catastrophic events in time series data. Can. J. Fish. Aquat. Sci., Can. J. Fish. Aquat. Sci. 644:899-910.

Wikle, C.K., L.M. Berliner, and N. Cressie. 1998. Hierarchical Bayesian space-time models. Journal of Environmental and Ecological Statistics 5:117-154

Wikle, C.K. 2003. Hierarchical Bayesian models for predicting the spread of ecological processes. Ecology 84:1382-1394.

References

- Dennis, B., Munholland, P. L., and Scott, J. M. (1991). Estimation of growth and extinction parameters for endangered species. *Ecological Monographs*, 61:115–143.
- Dennis, B., Ponciano, J. M., Lele, S. R., Taper, M. L., and Staples, D. F. (2006). Estimating density dependence, process noise, and observation error. *Ecological Monographs*, 76(3):323–341.
- Ellner, S. P. and Holmes, E. E. (2008). Resolving the debate on when extinction risk is predictable. *Ecology Letters*, 11:E1–E5.
- Gerber, L. R., Master, D. P. D., and Kareiva, P. M. (1999). Grey whales and the value of monitoring data in implementing the u.s. endangered species act. *Conservation Biology*, 13:1215âĂŞ1219.
- Hinrichsen, R. (2009). Population viability analysis for several populations using multivariate state-space models. *Ecological Modelling*, 220(9-10):1197– 1202.
- Holmes, E. E. (2001). Estimating risks in declining populations with poor data. Proceedings of the National Academy of Sciences of the United States of America, 98(9):5072–5077.
- Holmes, E. E., Sabo, J. L., Viscido, S. V., and Fagan, W. F. (2007). A statistical approach to quasi-extinction forecasting. *Ecology Letters*, 10(12):1182âĂŞ1198.
- Jeffries, S., Huber, H., Calambokidis, J., and Laake, J. (2003). Trends and status of harbor seals in washington state 1978-1999. Journal of Wildlife Management, 67(1):208âĂŞ219.
- Staples, D. F., Taper, M. L., and Dennis, B. (2004). Estimating population trend and process variation for pva in the presence of sampling error. *Ecol*ogy, 85(4):923–929.
- Taper, M. L. and Dennis, B. (1994). Density dependence in time series observations of natural populations: estimation and testing. *Ecological Monographs*, 64(2):205–224.

Index

animal tracking, 45

confidence intervals, 18 Hessian approximation, 18 parametric bootstrap, 18

density-independent, 5 diagnostics, 28

error observation, 6 process, 5, 6 estimation, 8 Dennis method, 9 KalmanEM, 8 maximum-likelihood, 8, 9 REML, 18 extinction, 5 diffusion approximation, 12 uncertainty, 15

kftrack, 51

state-space model multivariate, 21, 37, 45 univariate, 5