CHAPTER

Model Selection and Assessment

8

Our purpose in life is to analyze models. By that, we mean one or more of the following basic four tasks: (1) estimate parameters, (2) make predictions of unobserved random variables, (3) evaluate the relative merits of different models or choosing a best model (model selection), and (4) check whether a specific model appears to provide a reasonable description of the data or not (model checking, assessment, or “goodness-of-fit”). In previous chapters we addressed the problems of estimation of model parameters, and making predictions of latent variables, **s** or z, or functions of these variables such as density or population size. In this chapter, we focus on the last two of these basic inference tasks: model selection (which model or models should be favored), and model assessment (do the data appear to be consistent with a particular model).

We review basic strategies of model selection using both likelihood methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we review a number of standard methods of model selection that apply to “variable selection” problems, when our set of models consists of distinct covariate effects and they represent constraints of some larger model. For classical analysis based on likelihood, model selection by Akaike Information Criterion (AIC) is the standard approach (Burnham and Anderson, 2002). For Bayesian analysis we rely on a number of different methods. We demonstrate the use of the deviance information criterion (DIC) (Spiegelhalter et al., 2002) although it has deficiencies when applied to hierarchical models in some cases (Millar, 2009). We use the Kuo and Mallick indicator variable selection approach (Kuo and Mallick, 1998) which produces direct statements of posterior model probabilities which we think are the most useful, and leads directly to model-averaged estimates of density. There is a good review paper recently by O’Hara and Sillanpää (2009) that discusses these and many other related ideas for variable selection. In addition to O’Hara and Sillanpää (2009) we also recommend Link and Barker (2010, Chapter 7) for general information on model selection and assessment.

To check model adequacy in a Bayesian framework, or whether a specific model provides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value framework (Gelman et al., 1996). For assessing fit of SCR models, part of the challenge is coming up with good measures of model fit, and there does not appear much definitive guidance in the literature on this point. Following Royle et al. (2011a), we break the problem up into two components which we attack separately: (1) Conditional on the underlying point process, does the encounter model fit? (2) Do the uniformity and independence assumptions appear adequate for the point process

Spatial Capture-Recapture. http://dx.doi.org/10.1016/B978-0-12-405939-9.00008-6

model of activity centers? The latter component of model fit has a considerable precedence in the ecological literature as it is analogous to the classical problem of testing "complete spatial randomness" (Cressie, 1991; Illian et al., 2008).

We apply some of these methods to the wolverine camera trapping data first introduced in Chapter 5 to investigate sex specificity of model parameters and whether there is a behavioral response to encounter. We note that individuals are drawn to the camera trap devices by bait and therefore it stands to reason that once an individual discovers a trap, it might be more likely to return subsequently, a response termed "trap happiness". We evaluate whether certain models for encounter probability appear to be adequate descriptions of the data, and we evaluate the uniformity assumption for the underlying point process.

8.1 Model selection by AIC

Using classical analysis based on likelihood, model selection is easily accomplished using AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is simply twice the negative log-likelihood evaluated at the MLE, penalized by the number of parameters (np) in the model:

$$\text{AIC} = -2\log L(\hat{\boldsymbol{\theta}}|\mathbf{y}) + 2np.$$

Models with small values of AIC are preferred. It is common to use a modified ("corrected") AIC referred to as AIC$_c$ for small sample sizes which is

$$\text{AIC}_c = -2\log L(\hat{\boldsymbol{\theta}}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1},$$

where n is the sample size. Two important problems with the use of AIC and AIC$_c$ are that they don't apply directly to hierarchical models that contain random effects, unless they are computed directly from the marginal likelihood (for SCR models we can do this, see Chapter 6). Moreover, it is not clear what should be the effective sample size n in the calculation of AIC$_c$, as there can be covariates that affect individuals, that vary over time, or space. We do not offer strict guidelines as to when to use a small sample size adjustment.

The **R** package `secr` computes and outputs AIC automatically for each model fitted and it provides some capabilities for producing a model selection table (function `AIC`) and also doing model-averaging (function `model.average`), which we recommend for obtaining estimates of density from multiple models.

8.1.1 AIC analysis of the wolverine data

We provide an example of model selection for the wolverine camera trapping data using `secr`. We consider a model set with distinct models to accommodate various types of sex specificity of model parameters:

Model 0: Model SCR0 with constant density and constant encounter model parameters;

Model 1: Model SCR0 with constant encounter probability parameter values shared by both male and female wolverines but with sex-specific density only;
Model 2: Sex-specific density, sex-specific p_0 but constant σ;
Model 3: Sex-specific density, sex-specific σ but constant p_0;
Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ.

To model sex-specific abundance (density), we use the multi-session models provided by `secr` (introduced in Section 6.5.4), which allow one to model session-specific effects on density, baseline encounter probability, p_0 (labeled g_0 in `secr`), and the scale parameter σ of the encounter probability model. Using this formulation, we define the "Session" variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus *session*-specific parameters represent *sex*-specific parameters. For example, if we model session-specific density, D, then this corresponds to Model 1 in our list above. We note that "Model 0" in our list corresponds to a model where all of the encounter histories have the same session ID. This model is one of constant density, which implies that the population sex ratio is fixed at 0.5, i.e., $\psi_{sex} = 0.5$.

Although `secr` also uses the logit/log-linear predictors as the default for modeling covariates on baseline encounter probability and the scale parameter, respectively, `secr` does something different with the multi-session models. It reports estimates in a *session mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which arises from the use of dummy variables. We show this **BUGS** model description in Section 8.2.2.

To fit these models using `secr`, we load the wolverine data and do a slight bit of formatting to prepare the data objects for analysis by `secr`. The key difference from our analysis in Chapter 6 is, here, we use the wolverine sex information (`wolverine$wsex`) which is a binary 0/1 variable (1 = male) and we add 1 so that we can define a categorical "Session" variable (having values 1 or 2). We also have a function `scr2secr` which converts a standard trap deployment file (TDF) matrix into a `secr` object of class "`traps`." The **R** commands are as follows (contained in the help file `?secr_wolverine`):

```
> library(secr)
> library(scrbook)
> data(wolverine)
> traps <- as.matrix(wolverine$wtraps)

## Name variables as required by secr
> dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
## Convert trap information to a secr "traps" object
> trapfile <- scr2secr(scrtraps=traps,type="proximity")

## Grab the wolverine state-space grid (2km here)
> gr <- as.matrix(wolverine$grid2)
> dimnames(gr) <- list(NULL,c("x","y"))
> gr2 <- read.mask(data=gr)
```

```
## Grab the encounter data, and re-name variables
> wolv.dat <- wolverine$wcaps
> dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))

## Convert binary 0/1 sex variable to categorical 1/2 for "session"
> wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
> wolv.dat <- as.data.frame(wolv.dat)

## Convert to capthist object
> wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)
```

Once the data have been prepared in this way, we use the `secr` function `secr.fit` to fit the different models, and then the function `AIC` to package the models together and summarize them in the form of an AIC table, with rows of the table ordered from best to worst. The function `model.average` performs AIC-based model-averaging of the parameters specified by the `realnames` variable (below this is demonstrated for the parameter density, D). Because this function defaults to averaging by AIC$_c$, we slightly modified this function (called `model.average2`) to do model-averaging by either AIC or AIC$_c$ as specified by the user. The model fitting commands look like this for Model 0 and Model 1:

```
> model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
             buffer=20000)
> model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
             buffer=20000)
```

Next we use the function `AIC`, passing the fit objects from all five models, and that produces the following output (abbreviated horizontally to fit on the page):

```
> AIC(model0,model1,model2,model3,model4)
             model       ... npar   logLik     AIC      AICc    dAICc  AICwt
model0 D~1 g0~1 sigma~1 ...   3   −627.2603 1260.521 1261.932 0.000 0.5831
model2        ..         ...   5   −624.9051 1259.810 1263.810 1.878 0.2280
model1        ..         ...   4   −627.2365 1262.473 1264.973 3.041 0.1275
model4        ..         ...   6   −624.6632 1261.326 1267.326 5.394 0.0393
model3        ..         ...   5   −627.2358 1264.472 1268.472 6.540 0.0222
```

By default, the `AIC` functions orders the models by AIC$_c$. Model-averaging the results is done as follows:

```
> model.average(model0,model1,model2,model3,model4,realnames="D")
               estimate    SE.estimate            lcl            ucl
session=1   2.707190e-05   7.913577e-06   1.544474e-05   4.745224e-05
session=2   2.927423e-05   8.270402e-06   1.700631e-05   5.039193e-05
```

As usual, estimates and standard errors of the individual model parameters can be obtained from the `secr.fit` summary output of any of the `modelX` objects shown above. The default output of estimated density is in individuals per ha, so we have to scale this up to something more reasonable. To get into units of per 1000 km^2, we need to first multiply by 100 to get to units of km^2 and then multiply by 1000.

Table 8.1 Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1,000 km^2. Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

				Female			**Male**		
Model	**npar**	**AIC**	**AICc**	**D**	p_0	σ	**D**	p_0	σ
No Habitat Mask									
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03
With Habitat Mask									
2: sex/sex/1	5	1268.1	1272.1	3.64	0.07	6382.88	4.73	0.03	6382.88
4: sex/sex/sex	6	1268.7	1274.7	3.87	0.07	5859.40	4.41	0.03	7039.09
0: 1/1/1	3	1271.2	1272.6	4.18	0.05	6282.62	4.18	0.05	6282.62
1: sex/1/1	4	1273.1	1275.6	3.98	0.05	6282.65	4.38	0.05	6282.65
3: sex/1/sex	5	1275.1	1279.1	3.93	0.05	6357.26	4.41	0.05	6220.22

This produces an estimated density of about 2.71 for `session=1` (females) and 2.93 for `session = 2` (males). We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain specific information about the MLEs on the natural scale.

We don't necessarily agree with the use of AIC$_c$ here and think its better to use AIC, in general. This is because, as noted previously, it is not clear what the effective sample size is for most capture-recapture problems. While we have 21 individuals in the data set, most of the model structure has to do with encounter probability samples and for that there are hundreds of observations. We do note that the AIC and AIC$_c$ results are not entirely consistent. By looking at the best model by AIC (Table 8.1), we find that the model with sex-specific density and sex-specific baseline encounter probability, p_0, is preferred (Model 2). This is just slightly better than the null model (Model 0) with no sex effects at all which has an implied fixed sex ratio of $\psi_{sex} = 0.50$.

We fit the same models but now using a modified state-space which excludes the ocean (this is called a habitat mask in `secr`). Results are shown in Table 8.1 (bottom half of the table). We see AIC values are smaller for the model without the mask. It is probably acceptable to compare these different fits (with and without habitat mask) by AIC because we recognize the mask as having the effect of modifying the random effects distribution (i.e., of the activity centers, **s**) and the results should be sensitive to choice of the distribution for **s**. That said, we tend to prefer the mask model because it makes sense to exclude the areas of open water from the state-space of **s**. For females the model-averaged density is 3.88 individuals per 1,000 km^2 and for males the model-averaged density estimate is 4.46 individuals per 1,000 km^2 as we see here:

```
> model.average(model0b,model1b,model2b,model3b,model4b,realnames="D")
               estimate      SE.estimate           lcl            ucl
session=1    3.876615e-05    1.189102e-05    2.153795e-05    6.977518e-05
session=2    4.459658e-05    1.323696e-05    2.523280e-05    7.882022e-05
```

This is quite a bit higher than that based on the rectangular state-space (i.e., not specifying a habitat mask). This is not surprising given that **the state-space is part of the model** and the specific state-space modification we made here, which reduces the area from the rectangular state-space, should be extremely important from a biological standpoint.

8.2 Bayesian model selection

Model selection is somewhat less straightforward as a Bayesian, and there is no canned all-purpose method like AIC. As such we recommend a pragmatic approach, in general, for all problems, based on a number of basic considerations:

1. For a small number of fixed effects we think it is reasonable to adopt a conventional "hypothesis testing" approach—i.e., if the posterior for a parameter overlaps zero substantially, then it is probably reasonable to discard that effect from the model.
2. Calculation of posterior model probabilities: In some cases we can implement methods which allow calculation of posterior model probabilities. One such idea is the indicator variable selection method from Kuo and Mallick (1998). For this, we introduce a latent variable $w \sim \text{Bern}(.5)$ and expand the model to include the variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w \times \alpha_1 \times C_{ijk}.$$

   The importance of the covariate C is then measured by the posterior probability that $w = 1$.
3. The Deviance Information Criterion (DIC): Bayesian model selection is now routinely carried out using DIC (Spiegelhalter et al., 2002), although its effectiveness in hierarchical models depends very much on the manner in which it is constructed (Millar, 2009). We recommend using it if it leads to sensible results, but we think it should be calibrated to the extent possible for specific classes of models. This has not yet been done in the literature for SCR models, to our knowledge.
4. Logical argument: For something like sex specificity of certain parameters, it seems to make sense to leave an extra parameter in the model no matter what because, biologically, we should expect a difference (e.g., home range size). In some cases failure to apply logical argument leads to meaningless tests of gratuitous hypotheses (Johnson, 1999).

In all modeling activities, as in life itself, the use of logical argument should not be underutilized.

8.2.1 Model selection by DIC

The availability of AIC makes the use of likelihood methods convenient for problems where likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-purpose equivalent, at least for a brief period of time after its invention. However, there seem to be many variations of DIC, and a consistent version is not always reported across computing platforms. Even statisticians don't have general agreement on practical issues related to the use of DIC (Millar, 2009). Despite this, it is still widely reported. We think DIC is probably reasonable for certain classes of models that contain only fixed effects, or for which the latent variable structure is the same across models so that only the fixed effects are varied (this covers many SCR model selection problems). However, it would be useful to see some calibration of DIC for some standardized model selection problems.

Model deviance is defined as negative twice the log-likelihood; i.e., for a given model with parameters θ: $\text{Dev}(\theta) = -2\log L(\theta|\mathbf{y})$. The DIC is defined as the posterior mean of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D:

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D.$$

The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta}),$$

where the second term is the deviance evaluated at the posterior mean of the model parameter(s), $\bar{\theta}$. The p_D here is interpreted as the effective number of parameters in the model. Gelman et al. (2004) suggest a different version of p_D based on one-half the posterior variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

This is what is produced from **WinBUGS** and **JAGS** if they are run from `R2WinBUGS` or `R2jags`, respectively. It is less easy to get DIC summaries from `rjags`, so we used `R2jags` in our analyses below.

8.2.2 DIC analysis of the wolverine data

We repeated the analysis of the wolverine models with sex specificity, but this time doing a Bayesian analysis paralleling the likelihood analysis we did above in `secr`, using the logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy variables. Thus, we can express models allowing for sex specificity using a dummy variable `Sex` and new parameters ($\alpha_{sex}, \beta_{sex}$) which represent the *effect* of `Sex` at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex}\texttt{Sex}_i$$

and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex}\texttt{Sex}_i.$$

In these expressions, the sex variable $\texttt{Sex}_i$ is a binary variable where $\texttt{Sex}_i = 0$ corresponds to female, and $\texttt{Sex}_i = 1$ corresponds to male.

Unlike the multi-session model in `secr`, we carry out the analysis of the sex-specific model here by putting all of the data into a single data set, and explicitly accounting for the covariate "sex" in the model by assigning it a Bernoulli prior distribution with ψ_{sex} being the proportion of males in the population. In this case, we produce "Model 0" above, the model with no sex effect on density, by setting the population proportion of males at one-half: $\psi_{sex} = 0.5$ (see also Section 7.2.4). As usual, handling of missing values of the sex variable is done seamlessly which might be a practical advantage of Bayesian analysis in situations where sex is difficult to record in the field which may lead to individuals of unknown sex (i.e., missing values).

The **BUGS** model specification for the most complex model, Model 4, is shown in Panel 8.1. This model has a sex-specific intercept, scale parameter, σ, and density. We provide an **R** script named `wolvSCR0ms` in the `scrbook` package which fits each model. The function uses **JAGS** by default for the fitting, with the `R2jags`

```
alpha.sex ~ dunif(-3,3)            ## Prior distributions
beta.sex  ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                   ## Data augmentation parameter
psi.sex  ~ dunif(0,1)              ## Probability of ‘‘male’’

for(i in 1:M){                     ## DA loop
  wsex[i] ~ dbern(psi.sex)         ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)                ## DA variables, activity centers, etc..
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2)  + pow(s[i,2] - traplocs[j,2],2)
    p[i,j]  <-  p0[i]*exp( - alpha1[i]*dd[i,j] )
   }
 }
```

PANEL 8.1

Part of the **BUGS** specification for complete sex specificity of model parameters. This is a simplified version of the model contained in the `wolvSCR0ms` script, which allows users to run any of the models presented in Table 8.2.

Table 8.2 DIC results for the five models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on three chains of length 61,000 yielding 180,000 posterior samples.

Model	Meandev	p_D	DIC	Rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.50	519.28	3
Model 2	440.12	78.44	518.56	2
Model 3	443.31	79.47	522.79	5
Model 4	441.24	80.07	521.32	4

package. The template of this function is the model specification in Panel 8.1, which gets modified depending on the model we wish to fit using a command line option `model`. For example, `model = 1` fits the model with constant parameter values for males and females, but sex-specific population sizes (`model = 0` constrains the male probability parameter, ψ_{sex}, to be 0.5). The **R** function fits each of the five models using a binary indicator variable to turn "on" or "off" each effect. Here is how we obtain the MCMC output for each of the five models:

```
> wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
> wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
> wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
> wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
> wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

We fitted the five models to the wolverine data and summarize the DIC computation results in Table 8.2. The model rank has Model 0, Model 2, Model 1, Model 4, and Model 3. Interestingly, this is the same order as the models based on AIC$_c$ which we found above (see Table 8.1). The posterior mean and SD of model parameters for each model are given in Table 8.3.

8.2.3 Bayesian model-averaging with indicator variables

A convenient way to deal with model selection and averaging problems in Bayesian analysis by MCMC is to use the method of model indicator variables (Kuo and Mallick, 1998). Using this approach, we expand the model to include a set of prescribed models as specific reductions of a larger model. This has been demonstrated in some specific capture-recapture models in Royle and Dorazio (2008, Section 3.4.3) and Royle (2009b) and in the context of SCR by Tobler et al. (2012). A useful aspect of this method is that model-averaged parameters are produced by default. We emphasize the need to be careful of reporting model-averaged parameters that don't have a common interpretation in the different models because they are meaningless (averaging apples and oranges). For example, if a regression parameter is in a specific model then the posterior is informed by the data and a specific MCMC draw is from the

Table 8.3 Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, Model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21,000 iterations, 1,000 burn-in, 3 chains for a total of 60,000 posterior samples.

Parameter	**Model 0**		**Model 1**		**Model 2**		**Model 3**		**Model 4**	
	Mean	***SD***	***Mean***	***SD***	***Mean***	***SD***	***Mean***	***SD***	***Mean***	***SD***
N	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
D	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
α_0	−2.81	0.18	−2.82	0.17	−2.44	0.25	−2.82	0.18	−2.43	0.25
α_{sex}	0.00	1.73	0.00	1.73	−0.75	0.34	0.00	1.73	−0.79	0.36
σ_0	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
β_{sex}	0.00	1.73	−0.01	1.73	0.01	1.74	−0.01	0.17	0.10	0.18
ψ_{sex}	0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11
ψ	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07
deviance	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	pD = 77.1		pD = 77.5		pD = 78.4		pD = 79.5		pD = 80.1	
	DIC = 518.1		DIC = 519.3		DIC = 518.6		DIC = 522.8		DIC = 521.3	

appropriate posterior distribution. On the other hand, if the regression parameter is not in the model then the MCMC draw is obtained directly from the prior distribution, and so we need to think carefully about whether it makes sense to report an average of such a thing (in the vast majority of cases the answer is no). But some parameters like N or density, D, do have a consistent interpretation and we support producing model-averaged results of those parameters.

To implement the Kuo and Mallick approach, we expand the model to include the latent indicator variables, say w_m, for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m, \\ 0 & \text{linear predictor does not include covariate } m. \end{cases}$$

We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

for each variable $m = 1, 2, \ldots,$ in the model. For example, with two variables, the expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk},$$

where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a behavioral response covariate which is individual, trap, and occasion-specific. We can assume a parallel model specification on the parameter σ which is liable to vary by individual-level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

Using this indicator variable formulation of the model selection problem we can characterize unique models by the sequence of w variables. In this case, each unique sequence (w_1, w_2, w_3) represents a model, and we can tabulate the posterior frequencies of each model by postprocessing the MCMC histories of (w_1, w_2, w_3), as we demonstrate shortly. This method then evaluates all possible combinations of covariates or 2^m models.

Conceptually, analysis of this expanded model within the data augmentation framework does not pose any additional difficulty. One broader, technical consideration is that posterior model probabilities are well known to be sensitive to priors on parameters (Aitkin, 1991; Link and Barker, 2006; Tenan et al., 2013). See also Royle and Dorazio (2008, Section 3.4.3) and Link and Barker (2010, Section 7.2.5). What might normally be viewed as vague or non-informative priors are not usually innocuous or uninformative when evaluating posterior model probabilities. The use of AIC seems to avoid this problem largely by imposing a specific and perhaps undesirable prior that is a function of the sample size (Kadane and Lazar, 2004). One solution is to compute posterior model probabilities under a model in which the prior for parameters is fixed at the posterior distribution under the full model (Aitkin, 1991). The Gibbs Variable Selection (GVS) method (Dellaportas et al., 2000) resolves this by prescribing distinct priors depending on whether a variable is in the model or not (Tenan et al., 2013). At a minimum, one should evaluate the sensitivity of posterior model probabilities to different prior specifications.

8.2.3.1 *Analysis of the wolverine data*

The **R** script `wolvSCR0ms` in the package `scrbook` provides the model indicator variable implementation for the fully sex-specific SCR model. It is run by setting `model=5` in the function call. We note again that it is not very useful to report most parameter estimates from this model because their marginal posterior is a mixture from the prior (when a value of the indicator variable of 0 is sampled) and draws informed by the data (i.e., from the posterior, when a 1 is drawn for the indicator variable w). On the other hand, the parameters N and density D should be reported and they represent marginal posteriors over all models in the model set. In effect, model-averaging is done as part of the MCMC sampling. The variable "mod" contains the two binary indicator variables (w above) which multiply the "sex" term in each of the p_0 and σ model components, like this:

$$\text{logit}(p_{0,i}) = \alpha_0 + \texttt{mod}[1]\alpha_{sex}\texttt{sex}_i$$

and

$$\log(\sigma_i) = \log(\sigma_0) + \texttt{mod}[2]\beta_{sex}\texttt{sex}_i.$$

The third element of `mod` determines whether the ψ_{sex} parameter is estimated or fixed at $\psi_{sex} = 0.5$ which is accomplished with the line of **BUGS** code as follows:

```
sex.ratio <- psi.sex*mod[3] +.5*(1-mod[3]).
```

The MCMC output for "mod" was post-processed to obtain the model weights using the following **R** commands:

```
> mod <- wolv5$BUGSoutput$sims.list$mod
> mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
>
> table(mod)
mod
  000     001     010     011     100     101     110     111
17181    4935    1057     296   25211    8337    2275     708
> round(table(mod)/length(mod) , 3)
mod
  000     001     010     011     100     101     110     111
0.286   0.082   0.018   0.005   0.420   0.139   0.038   0.012
```

This results in a comparison of all eight possible models (based on $m = 3$ covariates) instead of just the five models we originally proposed. We see that the best model is that labeled (1,0,0) which, according to our construction above, has mod[1]=1, mod[2]=0, and mod[3]=0. This is the model having sex-specific baseline encounter probability p_0, and $\psi_{sex} = 0.5$. This model has posterior model probability 0.420. The model with no sex specificity at all (i.e., model (0,0,0)) has posterior probability 0.286, and the remaining posterior mass is distributed over the other six models. We could arrive at a qualitatively similar conclusion about which effects should be in the model using a more ad hoc approach based on looking at the posterior mass for each parameter under the full model (Model 4; see Table 8.3, in part). Considering the sex-specific intercept, it appears to be very important as its posterior mass is mostly away from zero. On the other hand, the coefficient on log-sigma is concentrated around 0, and the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior standard deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$ based on classical thinking-like-a-hypothesis-testing-person and settle for the model with a sex-specific intercept in the encounter probability model. This is consistent with our indicator variable approach which found that model $(1, 0, 0)$ has posterior probability of 0.420.

We can obtain model-averaged estimates from the indicator variable approach, which produces direct model-averaged estimates of N and D:

```
  mu.vect sd.vect   2.5%    25%    50%    75%  97.5%  Rhat n.eff
D   5.695   1.133  3.759  4.916  5.591  6.362  8.193 1.002  3600
N  59.077  11.758 39.000 51.000 58.000 66.000 85.000 1.002  3600
```

We obtain a model-averaged estimate (posterior mean) for density of $\hat{D} = 5.695$ which is hardly different from our model-specific estimates (Table 8.3) and, in particular, from Model 2 which has only a sex-specific intercept.

8.2.4 Choosing among detection functions

Another approach to implementing model indicator variables is to introduce a categorical "model identity" variable which is itself a parameter of the model. Using this approach, then each distinct model is associated with a unique set of covariates or other set of model features. This is convenient especially when we cannot specify the linear predictor as some general model that reduces to various alternative submodels simply by switching binary variables on or off. In the context of SCR models, choosing among different encounter probability models would be an example. For this case we do something like this `mod ~ dcat(probs[])` where `probs` is a vector with elements $1/(\#models)$, and the encounter probability matrix is filled-in depending on the value of `mod`. In particular, instead of a two-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \ldots, M$ models. An example in the **BUGS** language with three distinct models is:

```
 mod  ~ dcat(probs[])
##
## Using a double loop construction fill-in p[i,j,m] for each model:
##
 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]

 mu[i,j] <- z[i]*p[i,j,mod]
 y[i,j] ~ dbin(mu[i,j],K[j])
```

As before the posterior probabilities can be highly sensitive to priors on the different model parameters and sometimes mixing is really poor and, in general, we've experienced mixed success trying to carry out model selection using this construction. We do provide a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example of choosing among three different encounter probability models: The Gaussian encounter probability, Gaussian hazard, and logistic model with the square of distance (defined in Section 7.1). The key things to note are that there are three intercepts and three different "`alpha1`" parameters (the coefficient on distance). The parameters should not be regarded as equivalent across the models, so it is important to have them separately defined (and estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1) for the intercept parameter (either log or logit scale of baseline encounter probability p_0) and a Uniform(0,5) prior for one-half the inverse of the coefficient on distance squared. In the **BUGS** model specification the priors look like this:

```
for(i in 1:3){
 alpha0[i] ~ dnorm(0,.1)
 sigma[i] ~ dunif(0,5)
 alpha1[i] <- 1/(2*sigma[i]*sigma[i])
}
```

Then, we create a probability of encounter for each individual, trap *and* model so that the holder object "`p`" in the model description is a three-dimensional array (sometimes this would have to be a four or five-dimensional array in more complex models with time effects, etc.), so that construction of the encounter probability models looks like this:

```
p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
```

where

```
logit(p0[1]) <- alpha0[1]
log(p0[2]) <- alpha0[2]
p0[3]  <- alpha0[3]
```

You can experiment with the `wolvSCR0ms2` script to investigate the importance of different models of encounter probability and whether they have an affect on the inferences.

8.3 Evaluating goodness-of-fit

In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch of models and report estimates from all of them or a model-averaged summary of density. An important question is: Is our model worth anything? In other words, does the model appear to be an adequate description of our data? Formal assessment of model adequacy or goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical challenges to evaluating model fit, such as, if we do model-averaging then should all of the models have to fit? Or should the averaged model have to fit? What if none of the models fit? We don't know the answers to these questions and we won't try to answer them. Instead, we will provide what guidance we can on taking the first steps to evaluating fit, of a single model, as if it were a cherished family heirloom of great importance. We suggest that if you have a model that you really like, a single model, then it is a sensible thing to check that the model provides a good fit to your data. If it does not, we do not think the model is useless but just that some thought should be put into why the model doesn't fit so that, perhaps, some remediation might happen as future data are collected. After all, you may have spent 2, 3, or many more years of your life collecting that data set, perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to do some estimation and analysis of the model regardless of model fit. You can still learn something from a model that does not pass some technical test of model fit.

For either Bayesian or classical inference, the basic strategy to assessing model fit is to come up with a fit statistic that depends on the parameters and the data set, which we denote by $T(\mathbf{y}, \theta)$, and then compute this for the observed data set, and compare its value to that computed for perfect data sets simulated under the correct model. In the case of classical inference, we will often rely on the standard practice of parametric bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE $\hat{\theta}$ and compare realizations with what we've observed. The **R** package `unmarked` (Fiske and Chandler, 2011) contains generic bootstrapping methods for certain hierarchical models, including distance sampling (e.g., see

Sillett et al., 2012, for an application). In simple cases, using classical inference methods, it is sometimes possible to identify a test statistic of theoretical merit, perhaps with a known asymptotic distribution. For examples from capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and Chapter 5 of Cooch and White (2006). For Bayesian analysis we use the Bayesian *p*-value method (Gelman et al., 1996) (we introduced the Bayesian *p*-value in Section 3.9.1). Using this approach, data sets are simulated based on a posterior sample of the model parameters, θ, and some fit statistic for the simulated data sets, usually based on the discrepancy of the observed data from its expected values, is compared to that for the actual data. In most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the same: We compare data sets from the model we're interested in with the data set we have in hand. If they appear to be consistent with one another, then our faith in the model increases, at least to some extent, and we say "the model fits."

To date, there have been few suggestions on how to evaluate fit of SCR models from a classical perspective (but see Borchers and Efford, 2008, Section 6.5). For Bayesian analysis of SCR models, there has not been a definitive or general proposal for a fit statistic or even a class of fit statistics, although a few specialized implementations of Bayesian *p*-values have been provided (Royle, 2009b; Gardner et al., 2010a; Royle et al., 2011a; Gopalaswamy et al., 2012a,b; Russell et al., 2012). While we universally adopt the Bayesian *p*-value approach, and suggest some fit statistics in the following text, we caution that there is no general expectation to support how well they should do. As such, one might consider doing some kind of custom evaluation or calibration when using such methods, to gage the power of the test (ability to reject under specific departures from the model).

8.4 The two components of model fit

For most SCR models, there are at least two distinct components of model fit, and we propose to evaluate these two distinct components individually. First, we can ask, are the data consistent with the *observation* model, conditional on the underlying point process? We can evaluate this based on the encounter frequencies of individuals *conditional* on (posterior samples of) the underlying point process $\mathbf{s}_1, \ldots, \mathbf{s}_N$. We discuss some potential fit statistics for addressing this in Section 8.4.2. Second, we can evaluate whether the data appear consistent with the *state* process model (i.e., the "uniformity" assumption of the point process). For the simple model of independence and uniformity, this is similar to the assumption of *complete spatial randomness* (CSR) which we consider in Section 8.4.1 below. Actually, this is not strictly the assumption of CSR because of the binomial assumption on N under data augmentation, so we instead use the term *spatial randomness*.

8.4.1 Testing uniformity or spatial randomness

Historically, especially in ecology, there has been an extraordinary amount of interest in whether a realization of a point process indicates "complete spatial randomness," i.e., that the points are distributed uniformly and independently in space. Two good references for such things are Cressie (1991, Chapter 8) and Illian et al. (2008).[1] In the context of animal capture-recapture studies, the spatial randomness hypothesis is manifestly false, purely on biological grounds. Typically individuals will be clustered, or more regular (for territorial species), than expected under spatial randomness and heterogeneous habitat will generate the appearance of clustering even if individuals are distributed independently of one another. While we recommend modeling spatial structure explicitly when possible (Chapters 11–13), the uniformity assumption may be an adequate description of data sets in some situations. Further, we find that it is generally flexible enough to reflect non-uniform patterns in the data, because we do observe some direct information about some of the point locations.

The basic technical framework for evaluating the spatial randomness hypothesis is based on counts of activity centers in cells or bins. For that we use any standard goodness-of-fit test statistic, based on griding (i.e., binning) the state-space of the point process into $g = 1, 2, \ldots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the number of activity centers in bin g, centered at coordinate $\mathbf{x}_g$. Specifically, let $B(\mathbf{x})$ indicate a bin centered at coordinate $\mathbf{x}$, then[2] $N(\mathbf{x}) = \sum_{i=1}^{N} I(\mathbf{s}_i \in B(\mathbf{x}))$ is the population size of bin $B(\mathbf{x})$. In Section 5.11.1, we used the summaries $N(\mathbf{x})$ for producing density maps from MCMC output. Here, we use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of this form:

$$T(\mathbf{N}, \theta) = \sum_{g} \left(\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)} \right)^2,$$

where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional assessment of fit is based on the following statistic: Conditional on N, the total number of activity centers in the state-space $\mathcal{S}$, the bin counts N_g should have a binomial distribution. It will usually suffice to approximate the binomial cell counts by Poisson cell counts, in which case we can use the classical "index-of-dispersion" test (Illian et al., 2008, p. 87), based on the variance-to-mean ratio:

$$ID = (G - 1)*s^2/\bar{N},$$

where s^2 is the sample variance of the bin counts and $\bar{N}$ is the sample mean. When the point process realization is *observed*, as in classical point pattern modeling (but not in SCR), this statistic has approximately a Chi-square distribution on $(G - 1)$ degrees of freedom under the spatial randomness hypothesis. If $s^2/\bar{N} > 1$, clustering is suggested whereas, $s^2/\bar{N} < 1$ suggests the point process is too regular.

Whatever statistic we choose as our basis for assessing spatial randomness, the important technical issue is that we don't observe the point process and so the standard

[1] We also like Tony Smith's lecture notes (Univ. of Penn. ESE 502), which can be found at, http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part_I/3_Testing_Spatial_Randomness.pdf, accessed January 24, 2013.

[2] $I(arg)$ is the indicator function which evaluates to 1 if arg is true, otherwise 0.

statistics for evaluating spatial randomness cannot be computed directly. However, using Bayesian analysis, we do have a posterior sample of the underlying point process and so we suggest computing the posterior distribution of any statistic in a Bayesian p-value framework. For a given posterior draw of all model parameters, we can obtain a posterior sample of $N(\mathbf{x})$ by taking all the output for MCMC iterations $m = 1, 2, \ldots,$ and computing:

$$N(\mathbf{x})^{(m)} = \sum_{z_i^{(m)}=1} I\left(\mathbf{s}_i^{(m)} \in B(\mathbf{x})\right).$$

Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \ldots,$ is the Markov chain for the derived parameter $N(\mathbf{x})$.

In addition to computing the bin counts for each iteration of the MCMC algorithm, at the same time we generate a realization of the activity centers $\mathbf{s}_i$ under the spatial randomness model, and we obtain bin counts for these "new" data, $\tilde{N}(\mathbf{x})$. For each of the posterior samples—that of the real data, and that of the posterior simulated data, we compute the fit statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x \left(\sqrt{N(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})}\right)^2,$$

whereas the fit statistic based on a simulated realization of points under the spatial randomness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x \left(\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})}\right)^2.$$

And we compute the Bayesian p-value by tallying up the proportion of times that $T(\tilde{\mathbf{N}}, \theta)$ is larger than $T(\mathbf{N}, \theta)$, as an estimate of: $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$. The **R** function `SCRgof` in our package `scrbook` will do this, given the output from **JAGS** (Section 8.4.3).

8.4.1.1 *Sensitivity to bin size*

Evaluating fit based on bin counts in point process models is sensitive to the number of bins (Illian et al., 2008, pp. 87–88). This is related to the classical problem of fit testing in binary regression because, in a point process model, as the number of grid cells gets small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance or Pearson residuals) are known not to be very useful. There is some good discussion of this in McCullagh and Nelder (1989, Section 4.4.5). What it boils down to is, using the example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the fit statistic is exactly a deterministic function of the sample size only, which clearly should not be regarded as useful for model fit. This is why, in order to do a check of model fit when you have a binary response, one must always aggregate the data in some fashion. In the context of testing spatial randomness, computing the test statistic we described above has us chop up the region $\mathcal{S}$ into bins, and tally up N_g, the frequency of activity centers in each bin g. Suppose we choose the bin size to be extremely small such that $\mathbb{E}(N_g)$ tends to N/G (N being the number of activity centers). Further, N_g tends to a binary outcome. Therefore the fit statistic has N components that have value $N_g = 1$, and it has $G - N$ components that have value $N_g = 0$. Therefore, the fit statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g = 1}^{N} (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g = 0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

(here $\ni$ means "such that"). If G is huge relative to N, then we see that $T(\mathbf{N}, \theta)$ tends to about $2 * N$, which does not provide any meaningful assessment of model fit. So if you look at this in the limit in which the bin counts become binary, the fit statistic loses all its variability to the specific model used and is just a deterministic function of N. As a practical matter, it probably makes sense to restrict the number of bins to *fewer* than the number of observed individuals in the sample size. In typical SCR applications this will therefore result, usually, in very large (and few) bins, and presumably not much power.

There are some extensions that help resolve the issue of sensitivity to bin size. We can construct fit statistics based not just on quadrat counts but also the neighboring quadrat counts—this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a myriad of "distance methods" for evaluating point process models, and we believe that many of these can (and will) be adapted to SCR models. Again the main feature is that the point process on which inference is focused is completely latent in SCR models—so this makes the fit assessment slightly different than in classical point analysis.

8.4.1.2 Sensitivity to state-space extent

An issue that we have not investigated is that any model assessment that applies to a *latent* point process is probably sensitive to the size of the state-space. As the size of the state-space increases then the cell counts (far away from the data) *are* independent binomial counts with constant density, and so we can overwhelm the fit statistic with extraneous "data" simulated from the posterior, which is equal to the prior as we move away from the data, and therefore uninformed by the observed data that live in the vicinity of the trap array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity of the trap array only. Perhaps, as an ad hoc rule of thumb, less than the average trap spacing from the rectangle enclosing the trap array. For example, if the average trap spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity centers should not extend any further from the traps than 5 km. This should be a matter of future research.

8.4.2 Assessing fit of the observation model

In evaluating the spatial randomness hypothesis, we were able to draw on well-established ideas from point process modeling. On the other hand, it is less clear how to approach goodness-of-fit evaluation of the observation model. For many SCR problems, we have a three-dimensional data array of *binary* observations, y_{ijk} for individual i, trap j, and sample occasion k. As discussed in the previous section, we need to construct fit statistics based on observed and expected frequencies that are aggregated in some fashion. In practice, the data will be too sparse to have much power, unless the data are highly aggregated. We recommend focusing on summary statistics that represent aggregated versions of y_{ijk} over 1 or 2 of the dimensions. We describe three such fit statistics below. We recognize that, depending on the model, some information about model fit will be lost by summarizing the data in this way. For example if there is a behavioral response and we aggregate over time to focus on the individual and trap level summaries then some information about lack-of-fit due to temporal structure in the data is lost.

Fit statistic 1: individual x trap frequencies. We summarize the data by individual-and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard "dot notation" to represent summed quantities, we express that as: $y_{ij.} = \sum_{k=1}^{K} y_{ijk}$. Conditional on $\mathbf{s}_i$, the expected value under any encounter model is:

$$\mathbb{E}(y_{ij.}) = p_{ij}K$$

(or K_j if the traps are operational for variable periods). If there is time-varying structure to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij.}) = \sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j \left(\sqrt{y_{ij.}} - \sqrt{\mathbb{E}(y_{ij.})} \right)^2,$$

where we use θ here to represent the collection of all parameters in the model. This is conditional on $\mathbf{s}$ as well as on the data augmentation variables $\mathbf{z}$. We compute this statistic for *each* iteration of the MCMC algorithm for the observed data set and for a new data set simulated from the posterior distribution, say $\tilde{\mathbf{y}}$.

We could also use a similar fit statistic derived from summarizing over traps to obtain an `nind` $\times K$ matrix of count statistics. We imagine that either summary of the data will probably be too disaggregated (have mostly values of 0) in most practical settings to have much power.

Fit statistic 2: Individual encounter frequencies. SCR models represent a type of model for heterogeneous encounter probability, like model M_h, but with an explicit factor (space) that explains part of the heterogeneity. For model M_h, the individual encounter frequencies are the sufficient statistic for model parameters. Therefore, it makes intuitive sense to provide some kind of omnibus fit assessment of the core heuristic that the SCR model is adequately explaining the heterogeneity using a model M_h-like statistic based on individual encounter frequencies. So, we build a fit statistic based on the individual total encounters (Russell et al., 2012), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value is a similar summary over traps and occasions: $\mathbb{E}(y_{i\cdot\cdot}) = \sum_j \sum_k p_{ijk}$. Then, we define statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i \left(\sqrt{y_{i\cdot\cdot}} - \sqrt{\mathbb{E}(y_{i\cdot\cdot})}\right)^2.$$

We imagine this test statistic should provide an omnibus test of extra-binomial variation and should therefore capture some effect of variable exposure to encounter of individuals, although we have not carried out any evaluations of power under specific alternatives. Obviously, in using this statistic, we lose information on departures from the model that might only be trap- or time-specific.

Fit statistic 3: Trap frequencies. We construct an analogous statistic based on aggregating over individuals and replicates to form trap-encounter frequencies: $y_{\cdot j\cdot} = \sum_i \sum_k y_{ijk}$ (Gopalaswamy et al., 2012b) and the expected value is a similar summary over individuals and occasions: $\mathbb{E}(y_{\cdot j\cdot}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j \left(\sqrt{y_{\cdot j\cdot}} - \sqrt{\mathbb{E}(y_{\cdot j\cdot})}\right)^2$$

This seems like a sensible fit statistic because we can think of SCR models as spatial models for counts (Chandler and Royle, 2013). Therefore, we should seek models that provide good predictions of the observable spatial data, which are the trap totals. In this context, it might even make sense to pursue cross-validation-based methods for model selection. Cross-validation is a standard method of evaluating models such as in kriging or spline smoothing, so we could as well develop such ideas based on the trap-specific frequencies.

8.4.3 Does the SCR model fit the wolverine data?

We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR model to the wolverine camera trapping data. We consider first whether the simple model of spatial randomness of the activity centers is adequate. We think that the encounter model shouldn't have a large effect on whether the spatial randomness assumption is adequate or not, so we fit "Model 0" (in which parameters are *not* sex-specific) using an **R** script provided in the function `wolvSCR0gof` which will default to fitting the model in **JAGS**. This is the same script as `wolvSCR0ms` except that it saves the MCMC output for the activity centers **s** and the data augmentation variables z, which are required in order to compute the Bayesian p-value test of spatial randomness.

The MCMC output is processed with the **R** function `SCRgof` which computes the test of spatial randomness based on bin counts, using the Bayesian p-value calculation. The function `SCRgof` requires a few things as inputs: (1) the output from a **BUGS** run (in particular, the activity center coordinates and the data augmentation variables); (2) the number of bins to use for computing activity center counts; (3) the trap locations, and (4) the buffer around the trap array to use for making the bins and computing the bin counts. This buffer could be that used in defining the state-space for the model fitting, but we think it should be relatively tighter to the trap array than the state-space used in model fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km) and a 20-km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for computing the fit statistic. The **R** commands to fit the model and obtain the goodness-of-fit result are as follows:

```
> wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)

> bugsout <- wolv1$BUGSoutput$sims.list

> traplocs <- wolverine$wtraps[,2:3]
> traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
> traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
> traplocs <- traplocs/10000

> set.seed(2013)   # set seed so Bayesian p-value is the same each time

> SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)

Cluster index observed: 1.099822
Cluster index simulated: 1.000453
P-value index of dispersion: 0.408
P-value2 freeman-tukey: 0.6842667
```

The output produced by SCRgof is the index of dispersion based on the ratio of the variance to the mean (see above), which is computed as the posterior mean index of dispersion for the latent point process, and the average value for simulated data. If this value is >1 then clustering is suggested, which we see a (very) minor amount of evidence for here. Two Bayesian *p*-values are produced: the first is based on the cluster index, and the second is based on the Freeman-Tukey statistic calculated as described in Section 8.4.1. Because our *p*-values aren't close to 0 or 1, we judge that the model of spatial randomness provides an adequate description of the data. You can verify that a similar result is obtained if we use the model with fully sex-specific parameters (Model 4).

Next, we did a Bayesian *p*-value analysis of the observation component of the model, using the three fit statistics described in Section 8.4.2. These statistics can be calculated as part of the **BUGS** model specification or by postprocessing the MCMC output returned from a **BUGS** run. The **R** script wolvSCR0gof contains the relevant calculations. For example, to compute fit statistic T_1, we have to add some commands to the **BUGS** model specification such as the following (note: this is only a fraction of the model specification):

```
.......
for(j in 1:ntraps){
  mu[i,j] <- w[i]*p[i,j]

  y[i,j]~dbin(mu[i,j],K[j])
  ynew[i,j]~dbin(mu[i,j],K[j])

  err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
```

```
   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
}

T1obs <- sum(err[,])
T1new <- sum(errnew[,])
.......
```

Similar calculations are carried out to obtain the posterior samples of test statistics 2 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian *p*-value calculations produce:

```
> mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
[1]  0

> mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
[1]  0.17

> mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
[1]  0.02066667
```

Based on statistic T_2, we might conclude that the model is adequate for explaining individual heterogeneity although the other two statistics suggest a general lack-of-fit of the observation model. A similar result is obtained using the fully sex-specific model. We note that one individual was captured eight times in one trap, which is pretty extreme under a model which assumes independent Bernoulli trials. We surmise that the trap counts simply are not well explained by this model.

In attempt to resolve this problem, we extended the model to include a local (trap-specific) behavioral response (following Royle et al., 2011b) which can be fitted using the sample **R** script `wolvSCRMb`. To fit a model using **WinBUGS**, and then compute the Bayesian *p*-values we do this:

```
> wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)

> mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
[1]  0.9666667

> mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
[1]  0.3644667

> mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
[1]  0.4990667
```

This model seems to fit better, and so we might prefer reporting estimates from it, which we do in Table 8.4. (the behavioral response parameter is labeled α_2 in the table). Estimated density is about 1 individual higher per 1,000 km^2 compared with the various models that lack a behavioral response. It might be useful to try these fit assessment exercises using the habitat mask as described in Section 5.10. That takes an extremely long time to run in **BUGS** though, especially for the behavioral response model.

Table 8.4 Posterior summary statistics for local (trap-specific) behavioral response model M_b fitted to the wolverine camera trapping data using **WinBUGS**. The parameter α_2 is the local (trap-specific) behavioral response parameter. $T_x()$ are the posterior summaries of fit statistics $x = 1,2,3$ used in the Bayesian p-value analysis (see text for definitions). Results are based on three chains, each with 6,000 iterations (first 1,000 discarded) for a total of 15,000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	71.32	19.07	42.00	69.00	114.02	1.00	2100
D	6.87	1.84	4.05	6.65	10.99	1.00	2100
σ	0.88	0.13	0.68	0.86	1.17	1.00	730
p_0	0.01	0.00	0.01	0.01	0.02	1.01	530
α_1	0.69	0.19	0.37	0.67	1.10	1.00	730
α_2	2.50	0.27	1.99	2.50	3.04	1.00	700
ψ	0.36	0.10	0.20	0.35	0.58	1.00	2600
T_1^{obs}	54.71	6.12	43.69	54.39	67.47	1.00	3900
T_1^{new}	64.73	7.62	50.93	64.39	80.96	1.00	3900
T_2^{obs}	13.93	4.07	7.25	13.53	23.04	1.00	5700
T_2^{new}	12.65	3.35	6.93	12.36	20.07	1.00	2000
T_3^{obs}	12.80	1.74	9.80	12.64	16.61	1.00	2400
T_3^{new}	12.94	3.05	7.77	12.67	19.58	1.00	15000

8.5 Quantifying lack-of-fit and remediation

Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic occupancy models similar to that which we suggested above. They constructed a fit statistic based on aggregating the data over replicate samples (k), to obtain the total detections per site i and year j. They used a Bayesian p-value analysis based on a Chi-squared test statistic (also see Kéry and Schaub, 2012, Chapter 12). Their analysis suggested a model that didn't fit, and, so they computed the "lack-of-fit ratio" (see Kéry and Schaub, 2012, Section 12.3)—the ratio of the fit statistic computed for the observed data to that of the replicate (simulated) data sets. They interpret this analogous to the overdispersion coefficient in generalized linear models (McCullagh and Nelder, 1989), usually called the c-hat statistic in capture-recapture literature (see Cooch and White, 2006, Chapter 5). Molinari-Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which suggests a minor lack-of-fit, compared to perfect data having a value of 1, because the posterior standard deviations will be too small by a factor of $\sqrt{1.14} = 1.07$. In classical capture-recapture applications of goodness-of-fit assessment, inference for non-fitting models is dealt with by inflating the resulting SEs (of the non-fitting model), by the square root of c-hat. We believe that these ideas related to quantifying lack-of-fit and understanding its effect could also be applied to SCR models, although we have not yet explored this.

8.6 Summary and outlook

In this chapter, we offered some general strategies for model selection and model checking, or assessment of model fit. We think the strategies we outlined for model selection are fairly standard and can be effectively applied to many SCR modeling problems. Some technical issues of Bayesian analysis need to be addressed (in general) before Bayesian methods are more generally useful and accessible. For one thing, Bayesian model selection based on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow even for small data sets, and so improved computation will improve our ability to do Bayesian model selection in practical situations. Also, and most importantly, sensitivity to prior distributions is an important issue. Further research and practice might identify preferred prior configurations for SCR that provide a good calibration in relevant model selection problems. Finally, we believe that cross-validation should prove to be a useful method in model assessment and selection, as SCR models are a form of spatial model of counts, and so it is natural to pick models that predict the observable spatial counts (i.e., at trap locations) well. Bayesian model selection is an active area of research and development, and we expect more general and efficient methods will be developed, and even exist now for application. For example, Tenan et al. (2013) present an evaluation of two additional methods, including the product space method (Carlin and Chib, 1995) and Gibbs variable selection (GVS) (Dellaportas et al., 2000), which could provide improvement over the methods we've used in SCR applications. These ideas merit consideration.

For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework based on independent testing of the spatial model of independence and uniformity, and testing fit of the observation model conditional on the underlying point process. These ideas are based on mostly ad hoc attempts in a number of published applications (e.g., Royle et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012). While we think this general strategy should be fruitful, we know of no studies on the power to detect various model departures, and so the ideas should be viewed as experimental. We have not discussed assessment of model fit for SCR models using likelihood methods, although we imagine that standard bootstrapping ideas should be effective, perhaps based on the fit statistics (or similar ones) we suggested here for computing Bayesian p-values.

Clearly there is much research to be done on assessment of model fit in SCR models. Other approaches from spatial point process modeling should be pursued including nearest-neighbor methods or distance-based methods. In addition, studies to evaluate the power to detect relevant departures from the standard assumptions, and the robustness of inferences about N or density, need to be conducted. If the spatial randomness model appears inadequate, it is possible to fit models that allow for a non-uniform distribution of points (see Chapter 11) and even point process models that allow for interactions among points (Reich et al., in review). On the other hand, we expect that most of these Bayesian p-value tests will have low power in typical data sets consisting of a few to a few dozen individuals. As such, failure to detect a lack-of-fit may not be that meaningful. But, on the other hand, it may not make a difference in terms of density estimates either. We think inference about density should be relatively insensitive to departures from spatial randomness, because we get to observe direct information on some component of the population. For those activity centers, the assumed model of the point process should exert little influence on the placement of the activity centers. Conversely, as is the case with classical closed population models (Otis et al., 1978; Dorazio and Royle, 2003; Link, 2003), inferences may be somewhat more sensitive to bad-fitting models for the observation process.

This page is intentionally left blank

Non Print Items

Abstract: In this chapter, we focus on two basic inference tasks: model selection (which model or models should be favored), and model assessment (do the data appear to be consistent with a particular model). Researchers commonly want to compare competing models to determine “the best” model for their data. For example, this might include comparisons of models with and without a behavior response or models with and without time-dependent encounter rates. To assist readers in making such comparisons, we review some basic strategies of model selection using both likelihood methods (as implemented in the the R package secr) and Bayesian analysis. Specifically, we review a number of methods for “variable selection” problems, when our set of models consists of distinct covariate effects and they represent constraints of some larger model. For classical analysis based on likelihood, model selection by Akaike Information Criterion (AIC) is the standard approach. For Bayesian analysis there are a number of different methods—we cover the use of the deviance information criterion (DIC), the Kuo and Mallick indicator variable selection approach, and for model adequacy, we cover the Bayesian p-value method for assessing goodness-of-fit. We also discuss sensitivity to state space resolution and extent and how to quantify lack-of-fit. As a demonstration of the various approaches that are outlined, we work with data from the wolverine camera trapping study to investigate sex specificity of model parameters and whether there is a behavioral response to encounter. We evaluate whether certain models for encounter probability appear to be adequate descriptions of the data, and we evaluate the uniformity assumption for the underlying point process.

Keywords: AIC, Bayesian *p*-value, DIC, Goodness-of-fit, Indicator variable selection, Model checking, Model selection, Sensitivity, Spatial randomness, Wolverine study