# Cross validation using `spOccupancy`

Jeffrey W. Doser

October 04, 2021

## Contents

**1 Introduction** **1**
1.1 Example data set: Ovenbird at Hubbard Brook . . . . . . . . . . . . . . . . . . . . . . . . . . 1

**2 k-fold Cross-validation** **2**
2.1 Performing k-fold cross-validation using `PGOcc` . . . . . . . . . . . . . . . . . . . . . . . . . . 2

**References** **6**

# 1 Introduction

This vignette displays how to perform k-fold cross-validation using `spOccupancy` model objects. Model assessment and selection among a series of candidate models is often a key task in statistical analyses. In `spOccupancy` we provide the function `waicOcc` that computes the Widely Applicable Information Criterion (WAIC) to distinguish between different models and select a model for a final analysis. However, a k-fold cross-validation approach is typically a very adequate way to assess the predictive performance of a model (Hooten and Hobbs 2015), especially given recent criticisms of the WAIC for hierarchical models often used in wildlife ecology (Link, Sauer, and Niven 2020). In this vignette we will show how cross-validation can be easily achieved using `spOccupancy` model functions. For simplicity, we will display cross-validation approaches using a single species occupancy model with the `PGOcc` function. We first load the `spOccupancy` package along with the `coda` package for use with MCMC objects.

```
library(spOccupancy)
library(coda)
```

## 1.1 Example data set: Ovenbird at Hubbard Brook

We will use data on the Ovenbird (OVEN) collected from point counts at Hubbard Brook Experimental Forest (HBEF) in New Hampshire, USA. Specific details on the data set are available on the [Hubbard Brook website]((https://portal.edirepository.org/nis/mapbrowse?scope=knb-lter-hbr&identifier=178) and Doser et al. (2021). The data are provided in the `spOccupancy` package and are loaded with `data(hbef2015)`. Below, we subset the data portion of the `hbef2015` object to only include data on OVEN.

```
data(hbef2015)
str(hbef2015)
```

```
List of 4
 $ y       : num [1:12, 1:373, 1:3] 0 0 0 1 0 1 1 0 0 0 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:12] "AMRE" "BAWW" "BHVI" "BLBW" ...
  .. ..$ : chr [1:373] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "1" "2" "3"
 $ occ.covs: num [1:373, 1:2] -0.889 -0.765 -0.413 -0.14 -0.13 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Elevation" "Elevation.2"
 $ det.covs:List of 3
  ..$ day  : num [1:373, 1:3] -1.62 -1.62 -1.62 -1.62 -1.62 ...
  ..$ tod  : num [1:373, 1:3] -1.565 -1.378 -1.084 -0.79 -0.549 ...
  ..$ day.2: num [1:373, 1:3] 2.61 2.61 2.61 2.61 2.61 ...
 $ coords  : num [1:373, 1:2] 280000 280000 280000 280001 280000 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:373] "1" "2" "3" "4" ...
  .. ..$ : chr [1:2] "Easting" "Northing"
```

```r
sp.names <- attr(hbef2015$y, "dimnames")[[1]]
oven.hbef <- hbef2015
oven.hbef$y <- oven.hbef$y[sp.names == 'OVEN', , ]
```

# 2 k-fold Cross-validation

Suppose $J$ denotes the number of sites in our data sets where we have replicated detection-nondetection data for a species of interest. A k-fold cross validation approach requires fitting a model $k$ times, where each time the model is fit using $J/k$ data points. Each time the model is fit, it uses a different portion of the data and then predicts the remaining $J - J/k$ hold out values. Because the data are not used to fit the model, this yields true samples from the posterior predictive distribution that we can use to assess the predictivey capability of the model.

As a measure of out-of-sample predictive performance, we used the deviance as a cross-validation score following Hooten and Hobbs (2015). For K-fold cross-validation, our scoring function is computed as

$$-2\sum_{k=1}^{K}\log\left(\frac{\sum_{q=1}^{Q}\text{Bernoulli}(\boldsymbol{y}_k \mid \boldsymbol{p}^{(q)}\boldsymbol{z}_k^{(q)})}{Q}\right), \tag{1}$$

where $\boldsymbol{p}^{(q)}$ and $\boldsymbol{z}_k^{(q)}$ are MCMC samples of detection probability and latent occurrence, respectively, arising from a model that is fit without the observations $\boldsymbol{y}_k$. $Q$ is the total number of posterior samples from the MCMC sampler. The -2 is used so that smaller values indicate better model fit, which aligns with most information criteria used for model fit (like the WAIC implemented using `waicOcc`).

In `spOccupancy`, we implement k-fold cross-validation directly in the functions used to fit occupancy models. The following discussion is in the context of single species occupancy models (`PGOcc`), but the exact same implementation is included in `spOccupancy` for all other model fitting functions (i.e., `spPGOcc`, `msPGOcc`, `spMsPGOcc`, `intPGOcc`, `spIntPGOcc`).

## 2.1 Performing k-fold cross-validation using `PGOcc`

The final three arguments (`k.fold`, `k.fold.threads`, `k.fold.seed`) in `PGOcc` control whether or not k-fold cross validation is performed following the complete fit of the model using the entire data set. The `k.fold` argument indicates the number of $k$ folds to use for cross-validation. If `k.fold` is not specified, cross-validation is not performed and `k.fold.threads` and `k.fold.seed` are ignored. The `k.fold.threads` argument indicates the number of threads to use for running the $k$ models in parallel across multiple threads. Parallel processing is accomplished using the R packages `foreach` and `doParallel`. Specifying `k.fold.threads > 1` can substantially increase run time since it allows for models to be fit simultaneously on different threads rather than sequentially. The `k.fold.seed` indicates the seed used to randomly split the data into $k$ groups. This is by default set to 100.

Below we fit an occupancy model for OVEN with linear and quadratic elevation as occurrence predictors and day of year (linear and quadratic) and time of day (linear) as detection predictors. We set `k.fold = 4` to perform 4-fold cross-validation and `k.fold.threads = 4` to run the model across 4 threads.

```r
# Number of detection and occurrence parameters (including intercept)
p.det <- 4
p.occ <- 3
oven.starting <- list(alpha = rep(0, p.det),
                      beta = rep(0, p.occ),
                      z = apply(oven.hbef$y, 1, max, na.rm = TRUE))
oven.priors <- list(alpha.normal = list(mean = rep(0, p.det),
                                        var = rep(2.72, p.det)),
                    beta.normal = list(mean = rep(0, p.occ),
                                       var = rep(2.72, p.occ)))
n.samples <- 20000
n.burn <- 10000
n.thin <- 20
out.full <- PGOcc(occ.formula = ~Elevation + Elevation.2,
                  det.formula = ~day + tod + day.2,
                  data = oven.hbef,
                  starting = oven.starting,
                  n.samples = n.samples,
                  priors = oven.priors,
                  n.omp.threads = 1,
                  verbose = TRUE,
                  n.report = 5000,
                  n.burn = n.burn,
                  n.thin = n.thin,
          k.fold = 4,
          k.fold.threads = 4)
```

```
----------------------------------------
    Preparing the data
----------------------------------------
----------------------------------------
    Model description
----------------------------------------
Occupancy model with Polya-Gamma latent
variable fit with 373 sites.

Number of MCMC samples: 20000
Burn-in: 10000
Thinning Rate: 20
Total Posterior Samples: 500


Source compiled with OpenMP support and model fit using 1 thread(s).

Sampling ...
Sampled: 5000 of 20000, 25.00%
-------------------------------------------------
```

```
Sampled: 10000 of 20000, 50.00%
-------------------------------------------------
Sampled: 15000 of 20000, 75.00%
-------------------------------------------------
Sampled: 20000 of 20000, 100.00%
----------------------------------------
    Cross-validation
----------------------------------------

Performing 4-fold cross-validation using 4 thread(s).
```

```r
names(out.full)
```

```
 [1] "beta.samples"    "alpha.samples"   "z.samples"       "psi.samples"
 [5] "y.rep.samples"   "X"               "X.p"             "y"
 [9] "n.samples"       "call"            "n.post"          "n.thin"
[13] "n.burn"          "pRE"             "psiRE"           "k.fold.deviance"
[17] "run.time"
```

The cross-validation metric (model deviance) is stored in the `k.fold.deviance` tag of the resulting model object.

```r
out.full$k.fold.deviance
```

```
[1] 1524.573
```

Next, we run the same analysis but this time we only use 1 thread for cross validation.

```r
out.slow <- PGOcc(occ.formula = ~Elevation + Elevation.2,
                  det.formula = ~day + tod + day.2,
                  data = oven.hbef,
                  starting = oven.starting,
                  n.samples = n.samples,
                  priors = oven.priors,
                  n.omp.threads = 1,
                  verbose = TRUE,
                  n.report = 5000,
                  n.burn = n.burn,
                  n.thin = n.thin,
          k.fold = 4,
          k.fold.threads = 1)
```

```
----------------------------------------
    Preparing the data
----------------------------------------
----------------------------------------
    Model description
----------------------------------------
Occupancy model with Polya-Gamma latent
variable fit with 373 sites.

Number of MCMC samples: 20000
Burn-in: 10000
Thinning Rate: 20
Total Posterior Samples: 500


Source compiled with OpenMP support and model fit using 1 thread(s).
```

```
Sampling ...
Sampled: 5000 of 20000, 25.00%
-------------------------------------------------
Sampled: 10000 of 20000, 50.00%
-------------------------------------------------
Sampled: 15000 of 20000, 75.00%
-------------------------------------------------
Sampled: 20000 of 20000, 100.00%
----------------------------------------
    Cross-validation
----------------------------------------

Performing 4-fold cross-validation using 1 thread(s).
```

```
# Compare run times
out.full$run.time
```

```
   user  system elapsed
 45.303   1.614  23.932
```

```
out.slow$run.time
```

```
   user  system elapsed
 54.532   1.484  53.286
```

We see running the model using four threads cuts the run time in half. For more computationally intensive spatial models and multispecies models, running cross-validation using multiple threads can lead to significant decreases in run time.

Next, we vary the occurrence portion of the occupancy model and compare the results of the cross-validation.

```
# Linear elevation only
p.occ <- 2
oven.starting$beta <- rep(0, p.occ)
oven.priors$beta.normal <- list(mean = rep(0, p.occ),
                var = rep(2.72, p.occ))
out.1 <- PGOcc(occ.formula = ~ Elevation,
               det.formula = ~day + tod + day.2,
               data = oven.hbef,
               starting = oven.starting,
               n.samples = n.samples,
               priors = oven.priors,
               n.omp.threads = 1,
               verbose = FALSE,
               n.report = 5000,
               n.burn = n.burn,
               n.thin = n.thin,
           k.fold = 4,
           k.fold.threads = 4)
# Intercept only
p.occ <- 1
oven.starting$beta <- rep(0, p.occ)
oven.priors$beta.normal <- list(mean = rep(0, p.occ),
                var = rep(2.72, p.occ))
out.2 <- PGOcc(occ.formula = ~ 1,
               det.formula = ~day + tod + day.2,
               data = oven.hbef,
               starting = oven.starting,
               n.samples = n.samples,
               priors = oven.priors,
               n.omp.threads = 1,
               verbose = FALSE,
               n.report = 5000,
               n.burn = n.burn,
               n.thin = n.thin,
           k.fold = 4,
           k.fold.threads = 4)
out.full$k.fold.deviance
```

```
[1] 1524.573
```

```
out.1$k.fold.deviance
```

```
[1] 1521.326
```

```
out.2$k.fold.deviance
```

```
[1] 1539.662
```

Looking at the cross-validation deviance scores, we see support for including elevation in the model given the intercept only model has the highest deviance. However, the score for the linear only effect of elevation is lower than that of the full model with linear and quadratic elevation, indicating that including a quadratic parameter for elevation may not be necessary. Let's compare this to the results from the WAIC.

```
waicOcc(out.full)
```

```
       elpd          pD        WAIC
-697.824832    5.833331 1407.316326
```

```
waicOcc(out.1)
```

```
       elpd          pD        WAIC
-701.197470    5.116707 1412.628355
```

```
waicOcc(out.2)
```

```
       elpd          pD        WAIC
-766.006449    3.996356 1540.005610
```

We see the WAIC is lowest for the model with both linear and quadratic elevation. This is not all that surprising, as different forms of model assessment will potentially provide different results (Hooten and Hobbs 2015). This clearly indicates we should carefully consider the results from model assessments using both WAIC and k-fold cross-validation, and assess their results in the context of the goals of the analysis. We recommend using both WAIC and cross-validation when performing a more formal analysis that involves the comparison of multiple competing models, as well as potentially exploring alternative approaches not currently supported by `spOccupancy` (see Hooten and Hobbs (2015) for an overview of Bayesian model selection in ecology).

# References

Doser, Jeffrey W., Wendy Leuenberger, T. Scott Sillett, Michael T. Hallworth, and Elise F. Zipkin. 2021. "Integrated Community Occupancy Models: A Framework to Assess Occurrence and Biodiversity Dynamics Using Multiple Data Sources." *arXiv Preprint arXiv:2109.01894*.

Hooten, Mevin B, and N Thompson Hobbs. 2015. “A Guide to Bayesian Model Selection for Ecologists.” *Ecological Monographs* 85 (1): 3–28.

Link, William A, John R Sauer, and Daniel K Niven. 2020. “Model Selection for the North American Breeding Bird Survey.” *Ecological Applications* 30 (6): e02137.