Supplementary material for Indirect gradient analysis by Markov-chain Monte Carlo. Plant Ecology

Steve Walker
scwalker@math.mcmaster.ca

January 20, 2015

Contents

1 Preliminaries .......................................................... 2
  1.1 Required R packages ............................................. 2
  1.2 Load in community data ......................................... 2
  1.3 Translating across fields ...................................... 3

2 One-dimensional model .............................................. 3
  2.1 Fitting the model ................................................ 3
  2.2 Working with the large amount of model output ............... 3
  2.3 Assessing and correcting the Markov chains .................... 4
    2.3.1 Computing fitted values .................................. 4
    2.3.2 Fixing sign ambiguities with orthogonal Procrustes rotation 5
    2.3.3 Sensitivity to prior variances .............................. 8
    2.3.4 Convergence diagnostics ................................... 10
  2.4 Analyze posterior ................................................ 11
    2.4.1 Posterior summaries ...................................... 11
    2.4.2 Correlations with direct gradients ....................... 12
    2.4.3 Posterior ordinations .................................... 12
    2.4.4 Posterior predictive checks .............................. 14

3 Direct gradient analysis (very short example) ................. 17
  3.1 Fitting the model ................................................ 17
  3.2 Analyze posterior ............................................... 17

4 Two-dimensional model (non-exhaustive illustration) .......... 18
  4.1 Fitting the model ............................................... 18
  4.2 Assessing and correcting the Markov chains .................... 19
    4.2.1 Computing fitted values .................................. 19
    4.2.2 Fixing sign ambiguities with orthogonal Procrustes rotation 19
    4.2.3 Convergence diagnostics ................................... 20
  4.3 Analyze posterior ............................................... 21
    4.3.1 Correlations with direct gradients ....................... 21
1 Preliminaries

This document was constructed using reproducible research tools. It describes the code used to produce the results of the main text. Occasionally I omit pieces of code to enhance readability, but this omitted code is provided in an automatically generated R script, which is also in the supplementary material.

1.1 Required R packages

The following R packages, and their dependencies, are required,

```r

invisible(lapply(pkgs, library, character.only = TRUE))
```

These first two packages contain convenience functions for conducting the analyses described in this paper, and can be obtain using the install_github function in the devtools package.

```r
library(devtools)
github_pkgs <- paste("stevencarlislewalker", pkgs[1:2], sep = "/")
l_ply(github_pkgs, install_github, dependencies = TRUE)
```

At the time of creating this document, the versions of these packages are,

```r
noquote(R.version.string); noquote(sapply(pkgs, packageDescription, fields = "Version"))
## ecoBayesIRT popmoments vegan MCMCpack coda plyr pryr
## 0.0 0.1 2.0-10 1.3-3 0.16-1 1.8.1 0.1
## abind logspline MASS reshape2 ggplot2 knitr ks
## 1.4-0 2.1.5 7.3-33 1.4 1.0.0 1.8 1.9.3
```

1.2 Load in community data

I load in the dune meadow data and convert them to a presence-absence matrix,

```r
data(dune)
Y <- 1*(as.matrix(dune) > 0)
```

I remove rare species because they contain essentially zero information, and store the numbers of sites and species,

```r
Y <- Y[,colSums(Y)>1]  # remove species present in only one site
n <- nrow(Y)  # number of sites
m <- ncol(Y)  # number of species
```

Finally, I reorder the rows of Y such that the site IDs are sorted,

```r
siteOrder <- order(as.numeric(rownames(Y)))
Y <- Y[siteOrder, ]
```

The R code described here may be used with other presence-absence community data, and therefore provides a template for conducting ecological item response analysis using the Markov chain Monte Carlo methods provided by the MCMCpack R package.
1.3 Translating across fields

Before actually analyzing the dune meadow data, I provide a table for translating the terms of item response theory between educational and ecological research.

<table>
<thead>
<tr>
<th>Education research</th>
<th>Ecology</th>
</tr>
</thead>
<tbody>
<tr>
<td>item (e.g. test question)</td>
<td>species</td>
</tr>
<tr>
<td>subject (e.g. student)</td>
<td>site</td>
</tr>
<tr>
<td>item difficulty parameter (alpha)</td>
<td>species intercept (a = -alpha)</td>
</tr>
<tr>
<td>item discrimination parameter (beta)</td>
<td>species slope (b = beta)</td>
</tr>
<tr>
<td>subject ability parameter (theta)</td>
<td>gradient value (x = theta)</td>
</tr>
</tbody>
</table>

2 One-dimensional model

2.1 Fitting the model

I use the `MCMCirt1d` function from the `MCMCpack` R package to fit the model described by Section 2.3 in the main text, to the data, \( Y \). In order to ensure that the results are not dependent on the initial values of the parameters, I fit the model using three separate Markov chains that each start at different random initial conditions. We run each chain for 22000 iterations and drop the first 2000 as burnin in order to further reduce dependence on initial conditions.

```r
iters <- 20000; burnin <- 2000; priorPrecision <- 5; set.seed(1)
thetaInit <- replicate(3, scale(rnorm(n))[,], simplify = FALSE)
Ymcmc1d1 <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE, seed = 1, ABO = priorPrecision, theta.start = thetaInit[[1]])
Ymcmc1d2 <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE, seed = 2, ABO = priorPrecision, theta.start = thetaInit[[2]])
Ymcmc1d3 <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE, seed = 3, ABO = priorPrecision, theta.start = thetaInit[[3]])
Ymcmc1d <- as.mcmc.list(list(Ymcmc1d1, Ymcmc1d2, Ymcmc1d3))
```

The prior precision (= prior variance\(^{-1}\)) for the species parameters of 5 was chosen to put low prior probability density on extreme probabilities of occurrence (see Section 4.2.1 of the Supplementary Material and Section 2.3.2 of the main text).

2.2 Working with the large amount of model output

The large amount of memory required to store data from MCMC output poses a data management challenge. Not only does the model-fitting approach produce many MCMC iterations, but it is also a high-dimensional model and so the amount memory required to store each iteration is itself quite large. It is therefore important to store, access, and manipulate MCMC results thoughtfully.

The `Ymcmc1d` object is of the class `mcmc.list`, which is provided by the `coda` package. This class is convenient for comparing multiple MCMC runs, which we do below when diagnosing convergence of the chains (Section 2.3.4). After convergence is assessed, it is sometimes useful to combine the various chains in the `mcmc.list` object into a single `mcmc` object, which can be used to make various inferences. These objects are matrix-like with rows representing iterations and columns representing parameters.
YmcmcAll <- as.mcmc(as.matrix(Ymcmc1d))

This structure is useful as a standard for representing MCMC output from general statistical models. However, these matrix-like objects do not highlight the ecological structure hypothesized by the model. For example, it is a nuisance and distraction to have to explicitly keep track of which columns refer to which ecologically interpretable parameters. Therefore, it would be useful when exploring the model to have tools for extracting the gradient values, separately from the species scores. The ecoBayesIRT package contains a function, setIndex, which makes it easy to extract ecologically meaningful pieces of mcmc objects produced by MCMCirt1d and MCMCirtKd.

YmcmcAll <- setIndex(YmcmcAll)
Ymcmc1d <- setIndex(Ymcmc1d)

Note the getIndex function allows one to obtain this index.

The first task the index facilitates is to find the ecologically and statistically relevant dimensions of the model output.

dimIRT(YmcmcAll)
## iters sites species axes
## 60000 20 27 1

This printout indicates that YmcmcAll consists of 60000 Monte Carlo iterations, 20 sites, 27 species, and 1 latent gradient. These indices also allow one to use functions for easily extracting arrays containing the gradient values (with function x), the species slope parameters (with function b), and the species intercept parameters (with function a),

dim(x(YmcmcAll)); dim(b(YmcmcAll)); dim(a(YmcmcAll))
## [1] 60000 20 1
## [1] 60000 27 1
## [1] 60000 27

These extraction functions can be used as replacement functions as well. To illustrate this facility, we flip the sign of the species intercepts for each of the three MCMC chains and the combined chain.

for(i in 1:3) a(Ymcmc1d[[i]]) <- -a(Ymcmc1d[[i]])
a(YmcmcAll) <- -a(YmcmcAll)

In education research, the intercept is subtracted from the effect of the other parameters. With this approach the intercept may be interpreted as a difficulty parameter, which measures how challenging various test questions are. However, in ecology, it makes more sense to add the species intercepts to the linear prediction equation, so that they may be used to measure species’ prevalence at intermediate gradient values.

2.3 Assessing and correcting the Markov chains

2.3.1 Computing fitted values

One of the most useful elements of the model-based approach to ordination is the ability to check the fit of the model to the data. There are several types of fitted values, which I clarify here. The first type is the linear predictor, \(\eta\) (thick solid line in the upper panel of Figure 1 in the main text),

\[ \eta_{ij} = a_j + b_j x_i \] (1)
The second type is the observability, $\delta_{ij}$, which is the linear predictor plus the residual variation (shading in the upper panel of Figure 1 in the main text),

$$\delta_{ij} = \eta_{ij} + \epsilon_{ij}$$

The third and final type is the probability of occurrence (thick solid curve in the lower panel of Figure 1 in the main text),

$$p_{ij} = \Phi(\eta_{ij})$$

where $\Phi$ is the cumulative distribution function of the standard normal distribution. Each of these types of fitted values depends on the products, $x_i b_j$. The facilities of ecoBayesIRT may be used to compute the posterior of these products.

```r
xb <- apply2arrays(x(YmcmcAll),
                   aperm(b(YmcmcAll), c(1, 3, 2)),
                   1, 1, "%*%"

# Fixing sign ambiguities with orthogonal Procrustes rotation
```

The `apply2arrays` function is like the `apply` function, but can be used with two arrays, rather than a single array. In effect, this computation applies a matrix product, `%*%`, to the matrix of gradient values and matrix of species slopes for each MCMC iteration.

To calculate the $\eta$-scale fitted values, one sweeps out the intercepts. With these values, one may compute the other types of fitted values.

```r
eta <- sweep(xb, c(1, 3), a(YmcmcAll), "+")
p <- pnorm(eta)
```

To assess the choice of prior precision for the species parameters, I compute the range of posterior mean probabilities of occurrence. Both overly moderate and overly extreme fitted values will cross-validate poorly, due to under- and over-fitting respectively. The range of posterior mean probability of occurrence for our choice of precision is,

```r
range(pMean <- apply(p, c(2, 3), mean))
## [1] 0.05582799 0.92321728
```

This range seems reasonable given that it does not include highly extreme probabilities. However, as we find in posterior predictive checks below, this range is very conservative in that it largely trades off model fit in favour of generality (i.e. the range is narrow).

### 2.3.2 Fixing sign ambiguities with orthogonal Procrustes rotation

One unique challenge with fitting indirect gradient analysis models using MCMC is that there is that the gradient values and species scores can be multiplied by $-1$ without changing the fitted values (main text Section 2.3.3). This is similar to the axis flipping problem in traditional ordination analysis, but is more difficult because it must be dealt with in a consistent manner for every MCMC sample. Furthermore, as we will see, this problem becomes more difficult with more than one indirect gradient. However, there is an easy test for the problem, which manifests itself as bimodality in the posterior distributions of the gradient values. For example, the joint posterior density of the gradient values for sites two and five illustrates the nature of the bimodality.
The important feature of these graphs is that at both modes, the gradient values for the two sites have the same (left) or opposite (right) signs. Sites with identical signs are predicted by the model to be similar in the species that are likely to be present—a similar statement holds for sites with opposite signs. Put differently, the model makes probabilistically identical predictions regardless of which of the two modes the gradient values are closer to. That is, the data alone cannot be used to identify what mode should be used to infer the gradient values.

Before fixing the problem, it is instructive to explore its extent. This exploration begins by estimating the marginal posterior densities of the gradient values for each site, using the \texttt{logspline} package.

\begin{verbatim}
  densitiesX <- apply(x(YmcmcAll), 2:3, logspline)
  I have written the \texttt{findModes} function in the \texttt{ecoBayesIRT} package to explore bimodality over all sites.

  sapply(sapply(densitiesX, findModes), length)
  ## [1] 1 2 1 2 2 2 1 1 2 1 1 2 2 2 1 1 1 2

  These are the numbers of detected modes for each site's gradient values. The fairly large number of sites that are flagged as bimodal implies a lack of identifiability. To visualize this issue, consider the most extreme case, which is the site with the lowest posterior density at zero.

  denAtZero <- matrix(sapply(densitiesX, dlogspline, q = 0), dim(x(YmcmcAll))[2:3])
  (extremeSites <- apply(denAtZero, 2, which.min))
  ## [1] 5

  The graph below of the density for this extreme site, with its two modes highlighted using the \texttt{findModes} function, helps to illustrate the problem.

  plot(densitiesX[extremeSites, 1][[1]], las = 1, cex.lab = 0.9,
    xlab = "Extreme site score", ylab = "Marginal posterior density")
  abline(v = findModes(densitiesX[extremeSites, 1][[1]]))
\end{verbatim}
Again, a lack of identifiability is visible from this plot by the two modes, which are entirely an artifact of the different starting conditions of the three chains leading to flipped (but ecologically and statistically equivalent) gradient values.

The first step to fixing this non-identifiability is to recognize the aspects of the model that are identifiable. In this case, unlike the gradient values themselves, the model predictions do not suffer from these multi-modality and identifiability issues. This is similar to the situation in principal component analysis in which the underlying eigendecomposition is underdetermined with respect to sign changes of the eigenvectors.

One may use this fact to eliminate the non-identifiability of gradient values using the following steps:

- Note that the posterior of the matrix containing the products, $x_ib_j$, is identifiable.
- Compute the posterior mean of this matrix, which is justifiable because it is identifiable (i.e. not multi-modal).

```r
Exb <- aapply(xb, 2:3, mean)
```
- Calculate the singular vectors of this matrix, in particular the singular vectors associated with the sites.

```r
svdExb <- svd(Exb)$u[, 1:axesIRT(YmcmcAll), drop = FALSE]
```
- Match each posterior sample of the matrix containing the gradient values as closely as possible to these singular vectors, using orthogonal Procrustes analysis.

```r
xList <- lapply(Ymcmc1d, x)
Qlist <- lapply(xList, aapply, 1, orthProcrustesRotMat, Z = svdExb, .drop = FALSE)
for(i in 1:3) {
  x(Ymcmc1d[i]) <- apply2arrays(x(Ymcmc1d[i]), Qlist[i], 1, 1, /ts1, /ts1)
  b(Ymcmc1d[i]) <- apply2arrays(b(Ymcmc1d[i]), Qlist[i], 1, 1, /ts1, /ts1)
}
```

```r
class(Ymcmc1d) <- "mcmc.list"
YmcmcAll <- setIndex(as.mcmc(as.matrix(Ymcmc1d)))
```
This procedure both (1) leaves the model predictions unchanged, and (2) eliminates the non-identifiable multi-modality of the gradient values.

I now refit new densities.

```r
densitiesXrefit <- apply(x(YmcmcAll), 2:3, logspline)
sapply(sapply(densitiesXrefit, findModes), length)
```

And finally plot the density with the flipped version of the axis.

```r
plot(densitiesXrefit[extremeSites, 1][[1]], las = 1, cex.lab = 0.9,
     xlab = "Extreme site score", ylab = "Marginal posterior density")
abline(v = findModes(densitiesXrefit[extremeSites, 1][[1]])
```

This distribution is indeed unimodal, as are the posterior distributions for all other sites (code but not graphs shown). Note that when using this code, the uncorrected densities are in black for comparison with the corrected red lines, and therefore that the black densities are not necessarily expected to be unimodal.

```r
for(i in 1:20) {
  plot(densitiesXrefit[[i]], xlim = range(x(YmcmcAll)), col = "red")
  plot(densitiesX[[i]], add = TRUE)
  readline("press enter to continue")
}
```

### 2.3.3 Sensitivity to prior variances

Here I fit another model that differs only from the original model in that it has a higher prior precision. In particular, I shift the prior precision of the species parameters from 5 to 10. This shifts from higher prior variance (placing more weight on the data) to lower prior variance (placing more weight on the prior).

```r
iters <- 20000; burnin <- 2000; priorPrecisionSen <- 10
Ymcmc1dsen <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE,
                        seed = 10, ABO = priorPrecisionSen,
                        theta.start = thetaInit[[1]])
Ymcmc1d2sen <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE,
                        seed = 100, ABO = priorPrecisionSen,
```
theta.start = thetaInit[[2]])

Ymcmc1d3sen <- MCMCirt1d(Y, mcmc = iters, burnin = burnin, store.item = TRUE,
seed = 1000, AB0 = priorPrecisionSen,
theta.start = thetaInit[[3]])

Ymcmc1dsen <- setIndex(as.mcmc.list(list(Ymcmc1d1sen, Ymcmc1d2sen, Ymcmc1d3sen)))

YmcmcAllsen <- setIndex(as.mcmc(as.matrix(Ymcmc1dsen)))

As before I address identifiability issues (code in `suppmat.R`), to produce `Ymcmc1dsen` and `YmcmcAllsen` objects, which are analogous to the objects above. The most important difference induced by this change in prior precision is evident in the fitted values.

etaSen <- sweep(xbSen, c(1, 3), a(YmcmcAllsen), "+")
pSen <- pnorm(etaSen)

In particular, the model with higher precision exhibits a much narrower range of posterior mean probability of occurrence.

range(apply(pSen, c(2, 3), mean)); range(apply(p, c(2, 3), mean))

## [1] 0.1240499 0.8551029
## [1] 0.05582799 0.92321728

However, the ordering of the posterior mean gradient values, and species parameters, are not greatly influenced by the prior precision. That is to say, the inferred ordination is essentially unchanged. Note that the measurement scale of the gradient is unchanged, as all points in the first panel are near to the 1:1 line.
2.3.4 Convergence diagnostics

The chains seem to have converged to the same stationary distribution, which I formally assess using Gelman-Rubin convergence diagnostics provided by the `coda` package.

```r
gelman.diag(Ymcmc1d)
```

<table>
<thead>
<tr>
<th>Point</th>
<th>est.</th>
<th>Upper C.I.</th>
<th>beta.Runfla</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta.1</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Hyprad</td>
<td>1</td>
</tr>
<tr>
<td>theta.2</td>
<td>1</td>
<td>1.00</td>
<td># beta.Hyprad</td>
<td>1</td>
</tr>
<tr>
<td>theta.3</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Leoaut</td>
<td>1</td>
</tr>
<tr>
<td>theta.4</td>
<td>1</td>
<td>1.00</td>
<td># beta.Leoaut</td>
<td>1</td>
</tr>
<tr>
<td>theta.5</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Potpal</td>
<td>1</td>
</tr>
<tr>
<td>theta.6</td>
<td>1</td>
<td>1.00</td>
<td># beta.Potpal</td>
<td>1</td>
</tr>
<tr>
<td>theta.7</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Poapra</td>
<td>1</td>
</tr>
<tr>
<td>theta.8</td>
<td>1</td>
<td>1.00</td>
<td># beta.Poapra</td>
<td>1</td>
</tr>
<tr>
<td>theta.9</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Calcul</td>
<td>1</td>
</tr>
<tr>
<td>theta.10</td>
<td>1</td>
<td>1.00</td>
<td># beta.Calcul</td>
<td>1</td>
</tr>
<tr>
<td>theta.11</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Tripra</td>
<td>1</td>
</tr>
<tr>
<td>theta.12</td>
<td>1</td>
<td>1.00</td>
<td># beta.Tripra</td>
<td>1</td>
</tr>
<tr>
<td>theta.13</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Trire</td>
<td>1</td>
</tr>
<tr>
<td>theta.14</td>
<td>1</td>
<td>1.00</td>
<td># beta.Trire</td>
<td>1</td>
</tr>
<tr>
<td>theta.15</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Antodo</td>
<td>1</td>
</tr>
<tr>
<td>theta.16</td>
<td>1</td>
<td>1.01</td>
<td># beta.Antodo</td>
<td>1</td>
</tr>
<tr>
<td>theta.17</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Salrep</td>
<td>1</td>
</tr>
<tr>
<td>theta.18</td>
<td>1</td>
<td>1.00</td>
<td># beta.Salrep</td>
<td>1</td>
</tr>
<tr>
<td>theta.19</td>
<td>1</td>
<td>1.01</td>
<td># beta.Achmil</td>
<td>1</td>
</tr>
<tr>
<td>theta.20</td>
<td>1</td>
<td>1.00</td>
<td># beta.Achmil</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Belper</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Poatri</td>
<td>1</td>
</tr>
<tr>
<td>beta.Belper</td>
<td>1</td>
<td>1.00</td>
<td># beta.Poatri</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Junbuf</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Elyrep</td>
<td>1</td>
</tr>
<tr>
<td>beta.Junbuf</td>
<td>1</td>
<td>1.00</td>
<td># beta.Elyrep</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Junart</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Sagpro</td>
<td>1</td>
</tr>
<tr>
<td>beta.Junart</td>
<td>1</td>
<td>1.00</td>
<td># beta.Sagpro</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Airpra</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Plalan</td>
<td>1</td>
</tr>
<tr>
<td>beta.Airpra</td>
<td>1</td>
<td>1.00</td>
<td># beta.Plalan</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Elepal</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Agrsto</td>
<td>1</td>
</tr>
<tr>
<td>beta.Elepal</td>
<td>1</td>
<td>1.00</td>
<td># beta.Agrsto</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Rumace</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Lolper</td>
<td>1</td>
</tr>
<tr>
<td>beta.Rumace</td>
<td>1</td>
<td>1.00</td>
<td># beta.Lolper</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Viclat</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Alogen</td>
<td>1</td>
</tr>
<tr>
<td>beta.Viclat</td>
<td>1</td>
<td>1.00</td>
<td># beta.Alogen</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Brarut</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Brohor</td>
<td>1</td>
</tr>
<tr>
<td>beta.Brarut</td>
<td>1</td>
<td>1.00</td>
<td># beta.Brohor</td>
<td>1</td>
</tr>
<tr>
<td>alpha.Ranfla</td>
<td>1</td>
<td>1.00</td>
<td># alpha.Ranfla</td>
<td>1</td>
</tr>
</tbody>
</table>

## Multivariate psrf

## 1.01
The multivariate potential scale reduction factor is estimated to be close to one, which suggests convergence. I therefore conclude that the MCMC samples in \( Y_{\text{mcmc1d}} \) are representative of the target posterior distribution.

## 2.4 Analyze posterior

There are many ways to analyze posterior simulations. Here we consider (1) posterior summaries (e.g. means, and 95\% credible intervals, Section 2.4.1), (2) distributions of the correlations between indirect latent gradients and direct gradients (Section 2.4.2), (3) distributions of the site and species scores (i.e. Bayesian ordination, Section 2.4.3), and (4) posterior predictive checks (i.e. comparing observations with predicted distribution of observations, Section 2.4.4).

### 2.4.1 Posterior summaries

I begin by computing the posterior mean parameters and their 95\% credible intervals.

```r
Yabx <- list(a = a(YmcmcAll), b = b(YmcmcAll), x = x(YmcmcAll))
postMean <- lapply(Yabx, apply, 2, mean)
postCI <- lapply(Yabx, apply, 2, quantile, probs = c(0.025, 0.975))
```

For example, the posterior means and credible intervals of the species scores for *Achillea millefolium* and *Agrostis stolonifera* are,

```r
sp <- c("Achmil", "Agrsto"); rbind(mean = postMean$b[,sp], postCI$b[,sp])
##       Achmil  Agrsto
## mean  -0.6807541  0.7479060
## 2.5%  -1.2860593  0.1997832
## 97.5%  -0.1538944  1.3796509
```

The fact that these intervals do not overlap suggests that *Achillea millefolium* and *Agrostis stolonifera* are negatively associated, which is consistent with the fact that the two species never co-occur. I illustrate this using the `pairComp` function in the `ecoBayesIRT` package, with wrapping from the `reshape2` package.

```r
acast(melt(pairComp(Y[,sp])[, , 1, 2, drop = FALSE]), Var1 + Var3 ~ Var2 + Var4)
##       present_Agrsto absent_Agrsto
## present_Achmil           0          7
## absent_Achmil            10          3
```

Note that the credible intervals get wider for more extreme axes.

```r
plot(postMean$x^2, apply(postCI$x, 2, diff), xlab = "mean sq.", ylab = "CI width", las = 1)
```
2.4.2 Correlations with direct gradients

We compute the correlations between the site scores on the first axis and soil moisture and manure (code not shown for manure) for every MCMC sample, and then compute their posterior means and 95% credible intervals.

```r
dune.env <- as.numeric(dune.env$Manure)[siteOrder]
manureCors <- apply(x(YmcmcAll), 1, cor, manure)
sapply(list(moist = moistCors, manure = manureCors), mean)
```

<table>
<thead>
<tr>
<th>moist</th>
<th>manure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.78</td>
<td>-0.19</td>
</tr>
</tbody>
</table>

```r
sapply(list(moist = moistCors, manure = manureCors), quantile, probs = c(0.025, 0.975))
```

<table>
<thead>
<tr>
<th>moist</th>
<th>manure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.66</td>
<td>-0.42</td>
</tr>
<tr>
<td>0.88</td>
<td>0.05</td>
</tr>
</tbody>
</table>

The following figure makes clear that the ordination axis is an estimate of a gradient that is closely related to moisture.

![Correlation between moisture and the gradient](image)

2.4.3 Posterior ordinations

The ordinations of the sites and species are produced using the `ggplot2` package (see `suppmat.R` for code).
2.4.4 Posterior predictive checks

The simplest comparison between fitted values and the data is between the posterior mean probabilities of occurrence and raw presence-absence data (see suppmat.R for the code).

Another complementary way to compare the posterior mean fitted values with the observed presence-absence data is as a boxplot. Note that the fitted probabilities of occurrence tend to be larger for site-species combinations that actually occurred. However, note that the probabilities do not get too close to zero or one. This is because of the conservative prior distribution I chose to keep the model from making over-confident predictions (Section 2.1).
To generate a posterior predictive distribution, I simulate fake data, $Y_{sim}$, using the posterior distribution of the probabilities of occurrence, $p$.

$$Y_{sim} \leftarrow \text{array(rbinom(prod(dim(p)), 1, prob = p), dim(p))}$$

I compare this sample from the posterior predictive distribution with the observed species pairwise comparisons.

$$\text{obsPairs} \leftarrow \text{pairComp(Y)}$$

Here are two example comparisons.

$$\text{obsPairs[, , "Lolper", "Salrep", drop = FALSE]}$$

## , , Lolper, Salrep
##
## present absent
## present 1 11
## absent 2 6
The posterior predictive distribution of these pairwise comparisons is computed similarly.

```r
obsPairs[, , 1, 27, drop = FALSE]
## , , Belper, Brohor
## present absent
## present 4 2
## absent 1 13
```

As discussed in the main text, the number of pairs that fall outside of the prediction intervals is zero.

```r
lowPred <- apply(pairwiseYsim, 1:4, quantile, probs = 0.025)
highPred <- apply(pairwiseYsim, 1:4, quantile, probs = 0.975)
numComparisons <- prod(dim(lowPred)) - prod(dim(lowPred)[1:3])
sum((obsPairs > highPred) | (obsPairs < lowPred))
## [1] 0
```

One may now apply this object to explore a pairwise comparison between any two species. For example, as in the main text, I compare `Brohor` and `Potpal` (see `suppmat.R` for code).

![Graph showing pairwise comparison between Brohor and Potpal](image)

And here are the numbers behind the graph.
3 Direct gradient analysis (very short example)

Here is a short example of illustrating how MCMC can be used to fit item response models to ecological community and environmental data. These models account for both (1) the statistical dependence of community data on environmental variables and (2) the co-occurrence patterns among species that are not explained by the environmental variables alone.

3.1 Fitting the model

```r
iters <- 22000; burnin <- 2000; thin <- 1
Ymcmc1dwithEnv <- MCMCirtHier1d(Y, data.frame(moist = moist),
  mcmc = iters, burnin = burnin,
  thin = thin, ABO = priorPrecision,
  seed = 1)
moistCoefPosterior <- Ymcmc1dwithEnv[, "beta.moist"]
```

3.2 Analyze posterior

```r
quantile(moistCoefPosterior, probs = c(0.025, 0.975))
#
## 2.5% 97.5%
## -0.84926998 0.07312483
```

```r
moistCoefDensity <- logspline(moistCoefPosterior)
maxDen <- optimize(dlogspline, interval = c(-5, 5),
  fit = moistCoefDensity, maximum = TRUE)$objective
plot(moistCoefDensity, las = 1,
  xlab = "Moisture effect",
  ylab = "Posterior density",
  yaxs = "i", ylim = range(0, maxDen*1.05))
abline(v = 0, lwd = 0.5)
```
4 Two-dimensional model (non-exhaustive illustration)

In what follows, I attempt to fit a model with two gradients using the \texttt{MCMCirtKd} function. However, I am still uncertain precisely what model is being fitted. Nevertheless, what follows should be useful for anyone wishing to pursue fitting models with more than one indirect gradient. The code closely follows that for the one-dimensional model, and so many fewer comments are made.

4.1 Fitting the model

\begin{verbatim}
iters <- 20000; burnin <- 2000; set.seed(1); thin <- 1; priorPrecision <- 5
Ymcmc2d1 <- MCMCirtKd(Y, dimensions = 2, mcmc = iters, burnin = burnin, thin = thin,
                      store.item = TRUE, store.ability = TRUE, seed = 10,
                      B0 = priorPrecision)
Ymcmc2d2 <- MCMCirtKd(Y, dimensions = 2, mcmc = iters, burnin = burnin, thin = thin,
                      store.item = TRUE, store.ability = TRUE, seed = 100,
                      B0 = priorPrecision)
Ymcmc2d3 <- MCMCirtKd(Y, dimensions = 2, mcmc = iters, burnin = burnin, thin = thin,
                      store.item = TRUE, store.ability = TRUE, seed = 1000,
                      B0 = priorPrecision)
Ymcmc2d <- setIndex(as.mcmc.list(list(Ymcmc2d1, Ymcmc2d2, Ymcmc2d3)))
Ymcmc2dAll <- setIndex(as.mcmc(as.matrix(Ymcmc2d)))
\end{verbatim}
This output indicates that the posterior sample has 6000 MCMC iterations, 20 sites, 27 species, and 2 axes.

4.2 Assessing and correcting the Markov chains

4.2.1 Computing fitted values

```r
xb2d <- apply2arrays(x(Ymcmc2dAll),
                     aperm(b(Ymcmc2dAll), c(1, 3, 2)),
                     1, 1, "%*%"
)  
dimnames(xb2d)[2:3] <- dimnames(Y)
```

```r
for(i in 1:3) a(Ymcmc2d[[i]]) <- -a(Ymcmc2d[[i]])
a(Ymcmc2dAll) <- -a(Ymcmc2dAll)
```

```r
eta2d <- sweep(xb2d, c(1, 3), a(Ymcmc2dAll), "+")
p2d <- pnorm(eta2d)
```

The range of posterior mean probability of occurrence is,

```r
range(pMean2d <- apply(p2d, c(2, 3), mean))
## [1] 0.09515673 0.88930883
```

4.2.2 Fixing sign ambiguities with orthogonal Procrustes rotation

```r
Exb2d <- aapply(xb2d, 2:3, mean)
svdExb2d <- svd(Exb2d)$u[, 1:axesIRT(Ymcmc2dAll), drop = FALSE]
```

```r
xList <- lapply(Ymcmc2d, x)
Qlist2d <- lapply(xList, aapply, 1, orthProcrustesRotMat, Z = svdExb2d, .drop = FALSE)
for(i in 1:3) {
  x(Ymcmc2d[[i]]) <- apply2arrays(x(Ymcmc2d[[i]]), Qlist2d[[i]], 1, 1, "%*%")
  b(Ymcmc2d[[i]]) <- apply2arrays(b(Ymcmc2d[[i]]), Qlist2d[[i]], 1, 1, "%*%")
}
```

```r
class(Ymcmc2d) <- "mcmc.list"
Ymcmc2dAll <- setIndex(as.mcmc(as.matrix(Ymcmc2d)))
```
4.2.3 Convergence diagnostics

Because procrustes rotation confines the gradient values to a subspace of $\mathbb{R}^n$, where $n$ is the number of sites, a singular matrix problem arises when computing the convergence diagnostics.

```r
gelman.diag(Ymcmc2d)
```

```
## Error in chol.default(W): the leading minor of order 121 is not positive definite
```

However, this issue is easily solved by removing the gradient values for one site and one axis.

```r
par2remove <- with(attr(Ymcmc2dAll, "index"),
                   which((type == "theta") & (object == 1) & (axis == 2)))
Ymcmc2dDiag <- as.mcmc.list(lapply(Ymcmc2d, compose(as.mcmc, take),
                                    2, -par2remove))
```
<table>
<thead>
<tr>
<th></th>
<th>Point est. Upper C.I.</th>
<th># theta.6.1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha.Elyrep</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Elyrep.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Elyrep.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Sagpro</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Sagpro.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Sagpro.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Plalan</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Plalan.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Plalan.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Agrsto</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Agrsto.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Agrsto.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Lolper</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Lolper.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Lolper.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Alogen</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Alogen.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Alogen.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>alpha.Brohor</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Brohor.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>beta.Brohor.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.1.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.2.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.2.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.3.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.3.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.4.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.4.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.5.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theta.5.2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

## Multivariate psrf

## 1.01

### 4.3 Analyze posterior

#### 4.3.1 Correlations with direct gradients

Exploring correlations with direct gradients is more complicated with two indirect gradients. This is because the direct gradient may correlate with both of the indirect gradients. I have written a `plot2dCors` function, included in the `ecoBayesIRT` package, for plotting a sample of the joint posterior distribution of the correlations with both indirect gradients (blue point cloud). I have also used the `kde` function in the `ks` package for plotting kernel density estimates of the 95% credible region (blue ellipses). The `plot2dCors` function also plots a null distribution of correlations with random permutations of the direct gradient (red) for reference.
Correlations between moisture and indirect gradient I

Correlations between manure and indirect gradient II

Correlations between manure and indirect gradient I