The Integrated Nested Laplace Approximation for Fitting Dirichlet Regression Models

Joaquín Martínez-Minaya, Finn Lindgren, Antonio López-Quílez, Daniel Simpson, and David Conesa

ABSTRACT

This article introduces a Laplace approximation to Bayesian inference in Dirichlet regression models, which can be used to analyze a set of variables on a simplex exhibiting skewness and heteroscedasticity, without having to transform the data. These data, which mainly consist of proportions or percentages of disjoint categories, are widely known as compositional data and are common in areas such as ecology, geology, and psychology. We provide both the theoretical foundations and a description of how Laplace approximation can be implemented in the case of Dirichlet regression. The article also introduces the package dirinla in the R-language that extends the R-INLA package, which can not deal directly with Dirichlet likelihoods. Simulation studies are presented to validate the good behavior of the proposed method, while a real data case-study is used to show how this approach can be applied. Supplementary materials for this article are available online.

1. Introduction

The use of regression models with multivariate or correlated responses has enormously increased in the last few years. Different R-packages have been developed to deal with them, MCMCglmm (Hadfield 2010), SabreR (Nowosad and Stepiński 2018) or mcglm (Bonat 2018), also packages which uses copulas (Masarotto and Varin 2017). The use of Multinomial likelihood in the case of multivariate discrete response is one of the most popular (Monyai et al. 2016; Odeyemi et al. 2019; Piccini et al. 2019).

Responses can also be continuous, such as Gaussian (Anderson 1958) or compositional data (Aitchison and Egozcue 2005; Hijazi and Jernigan 2009). Of particular interest are compositional data which consist of proportions or percentages of disjoint categories summing up to one, and play an important role in many fields such as ecology (Kobal, Kastelec, and Eler 2017; Douma and Weedon 2019), geology (Buccianti and Grunsky 2014; Engle and Rowan 2014), genomics (Tsilimigras and Fodor 2016; Shi, Zhang, and Li 2016; Washburne et al. 2017) or medicine (Dumuid et al. 2018; Fairclough et al. 2018).

One of the biggest problems one has to face when dealing with models with multivariate or correlated responses is that of performing inference. Their own complexity makes statistical analysis complicated. In the case of compositional data, there are different approaches to deal with these additional complications. One method, due to Aitchison (1986), is based in the idea that “information in compositional vectors is concerned with relative, not absolute magnitudes,” and uses log-ratio analysis to deal with the unit-sum constraint (Aitchison 1981, 1982, 1983, 1984). Dirichlet regression models (Hijazi and Jernigan 2009) are another good way of analyzing compositional data. By using appropriate link functions, Dirichlet regression provides a GLM-like framework that relates compositional data with other relevant variables of interest. Beta regression can be considered a special, and effectively univariate, case of the former with only two categories.

Different packages have been implemented in R (R Core Team 2021) that analyze compositional data using Beta regression and Dirichlet regression, both under the frequentist (Cribari-Neto and Zeileis 2010; Maier 2014) and the Bayesian paradigm. In the case of the latter, the largest challenge is implementing the posterior approximation. In particular, it has been implemented in BayesX (Klein et al. 2015), Stan (Sennhenn-Reußen 2018), BUGS (Van der Merwe 2019), and R-JAGS (Plummer 2016). These packages are mainly based on Markov chain Monte Carlo (MCMC) methods, which construct a Markov chain whose stationary distribution converges to the posterior distribution. However, the computational cost of MCMC can be high. On the other hand, the integrated nested Laplace approximation (INLA) methodology (Rue, Martino, and Chopin 2009), whose main idea is to approximate the posterior distribution using the Laplace integration method, has become an alternative to MCMC, guaranteeing a higher computational speed for Latent Gaussian models (LGMs).

The INLA methodology is now a well-established tool for Bayesian inference in several research fields, including ecology,
epidemiology, econometrics and environmental science, and is implemented in the \texttt{R-INLA} package (Rue et al. 2017). Nevertheless, and spite of its availability for a large number of models, \texttt{R-INLA} does not allow to deal with compositional data when the number of categories is bigger than 2, the reason being that it is constructed for models with univariate responses.

Our objective in this work is 2-fold. We present an expansion of the INLA method for the particular case of the Dirichlet regression, providing both its theoretical foundations and a description of how it can be implemented for its application, and we introduce the package \texttt{dirinla} in the R-language that allows its practical use. To do so, the remainder of the article is structured as follows. Section 2 provides the basics of the Dirichlet regression, while Section 3 gives the necessary hints about LGMs and the INLA approach to follow the remainder of the article. Section 4 depicts the new approach, and Section 5 introduces the \texttt{dirinla} package and how to use it. Simulation studies about the performance of the method introduced is presented in Section 6, followed by an illustration of its use on real data in Section 7. Finally, Section 8 concludes.

2. Dirichlet Likelihood

In what follows we present both the Dirichlet distribution and the Dirichlet regression, while introducing some assumptions and the notation that will be used in the rest of the article.

2.1. Dirichlet Distribution

The Dirichlet distribution is the generalization of the widely known Beta distribution, and it is defined by the following probability density,

\[ p(y | \alpha) = \frac{1}{B(\alpha)} \prod_{c=1}^{C} y_c^{\alpha_c - 1}, \]

where \( \alpha = (\alpha_1, \ldots, \alpha_C) \) is known as the vector of shape parameters for each category, \( \alpha_c > 0 \ \forall c, \ y_c \in (0, 1), \sum_{c=1}^{C} y_c = 1, \) and \( B(\alpha) \) is the Multinomial Beta function, which serves as the normalizing constant. The Multinomial Beta function is defined as \( \prod_{c=1}^{C} \Gamma(\alpha_c) / \Gamma(\sum_{c=1}^{C} \alpha_c) \). The sum of all \( \alpha \)'s, \( \alpha_0 = \sum_{c=1}^{C} \alpha_c \), is usually interpreted as a precision parameter (\( \tau \)). The Beta distribution is the particular case when \( C = 2 \). In addition, each variable is marginally Beta distributed with \( \alpha = \alpha_c \) and \( \beta = \alpha_0 - \alpha_c \).

Let \( y \sim \mathcal{D}(\alpha) \) denote a variable that is Dirichlet distributed. The expected values are \( E(y_c) = \alpha_c / \alpha_0, \) the variances are \( \text{var}(y_c) = [\alpha_c (\alpha_0 - \alpha_c)] / [\alpha_0^2 (\alpha_0 + 1)] \) and the covariances are \( \text{cov}(y_c, y_{c'}) = (-\alpha_c \alpha_{c'}) / [\alpha_0^2 (\alpha_0 + 1)] \).

2.1.1. Dealing with Zeros and Ones

Dirichlet distributions, as it happens with Beta distributions, are defined in the open interval \((0, 1)\). Nevertheless, in practice, data may come from the closed interval \([0, 1]\). Although converting compositional data into Multinomial data could be an option, there are many situations in real life (see references in the Introduction) where we have to use the continuous nature of the compositional data. In such cases, one option to deal with zeros and ones in Dirichlet distributions is to transform them (Maier 2014), in a similar way as it happens in Beta distributions (Smithson and Verkuilen 2006). In particular:

\[ y^* = \frac{y(N-1)+1/C}{N}, \]

with \( N \) being the number of observations of the \( C \) dimensional Dirichlet response. This transformation compresses the data symmetrically around 0.5, so extreme values are affected more than values lying close to 1/2. Additionally, as it is pointed out in Maier (2014), if \( N \to \infty \) the compression vanishes, that is, larger datasets are less affected by this transformation. However, for small \( N \), the transformation can introduce a noticeable bias. In the INLA software, a new method has been implemented for the Beta distribution that instead treats extreme observations as censored variables, but it is not clear how to extend this to the Dirichlet case.

From now on, we assume that the variable takes values in the open interval \((0, 1)\), so that the transformation (2) is not required.

2.2. Dirichlet Regression Models

Let \( Y \) be a matrix with \( C \) rows and \( N \) columns denoting \( C \) observations for the different categories \( C \) of the \( C \) dimensional response variable \( Y_{cn} \). Let \( \eta_{cn} \) be the linear predictor for the \( n \)th observation in the \( c \)th category, so \( \eta_{cn} \) is a matrix with \( C \) rows and \( N \) columns. Let \( V_{(c)}^{(e)} \), \( e = 1, \ldots, C \), represents a matrix with dimension \( N \times I_c \) that contains the covariate values for each individual and each category, so \( V_{(c)}^{(e)} \) shows the covariate values for the \( n \)th observation and the \( c \)th category. Let \( \beta \) be a matrix with \( I_c \) rows and \( C \) columns representing the regression coefficients in each dimension. Finally, let also \( \omega_{cn} \) represents a realization of a random effect for the \( c \)th category and the \( n \)th observation. Then, the relationship between the parameters of the Dirichlet distribution and the elements of the linear predictor (including random effects) is set up as

\[ g(\alpha_{cn}) = \eta_{cn}^* = V_{(c)}^{(e)} \beta_{c} + \omega_{cn}, \]

where \( g(\cdot) \) is the link-function. As \( \alpha_c > 0 \) for \( c = 1, \ldots, C \), log-link \( g(\cdot) = \log(\cdot) \) is used. The regression coefficients \( \beta_{c} \) are a column vector with \( I_c \) elements.

Dirichlet distributions can also be parametrized in terms of the mean \( \mu_{cn} = \sum_{c=1}^{n} \frac{\exp(\eta_{cn})}{\sum_{c=1}^{n} \exp(\eta_{cn})} \alpha_c \) and the precision \( \tau \). In this case, the relationship between the mean and the linear predictor is stated as

\[ \mu_{cn} = \frac{\exp(\eta_{cn}^*)}{\sum_{c=1}^{C} \exp(\eta_{cn}^*)}, \]

\[ = \frac{\exp(\eta_{cn}^* V_{(c)}^{(e)} \beta_{c} + \omega_{cn})}{\sum_{c=1}^{C} \exp(\eta_{cn}^* V_{(c)}^{(e)} \beta_{c})}. \]

The regression coefficients \( \beta_{c} \) are now a column vector with \( I_c \) elements, while \( \omega_{cn} \) represents now a realization of a random effect in this parameterization. Moreover, as one of
the categories can be obtained as one less the sum of the rest, we
we can employ the same Multinomial logit strategy used in Multinomial regression: linear predictor of one category (base category) is set to zero, whereby this category is virtually omitted and becomes the reference.

The equivalence between the two parameterizations comes from the equalities

\[ \tilde{y}_{*c} = \tilde{\beta}_{*c} - \bar{\beta}_{*r} , \]

and

\[ \omega^c_{cn} = \omega_{cn} - \omega_{rn} , \]

where \( \tilde{y}_{*c} \) is a column vector with \( J \geq J_c \) elements that contains \( J_c \) nonzero elements and \( J - J_c \) zero-elements. \( J \) dimension comes from a general matrix \( V \) with dimension \( N \times J \) which contains all the covariates values (\( J \)) for all the observations used in all the categories. \( \tilde{\beta}_{*c} \) is also a column vector with \( J \geq J_c \) elements, and \( \bar{\beta}_{*r} \) and \( \omega_{rn} \) are the regression coefficients and the realization of a random effect for the base category, respectively.

Note that we can include in the formula as many random effects as we consider and from a different nature: temporal, spatial, etc. But here, without loss of generality, we have added just one to show the equivalence between both parameterizations (proof in Appendix A). As the proposal presented in the following sections relies on the fact that there are no other parameters to estimate in the observation likelihood, from now on we focus on the first parameterization presented.

Equation (3) can be rewritten in a vectorized form. In particular, if

\[ \tilde{\eta} = \begin{bmatrix} \eta_{*1} \\ \vdots \\ \eta_{*N} \end{bmatrix} \]

\[ (CN \times 1) \]

denotes a restructured linear predictor, being \( \eta_{*n} \) a column vector representing the linear predictor for the \( n \)th observation and all the categories, the model in matrix notation is

\[ \tilde{\eta} = A x , \]

where \( A \) is the matrix properly constructed with the covariates values and vectors of 1s for the random effects, and \( x \) a vector formed by the regression coefficients plus the realizations of the random effects. Posteriorly, we will refer to this vector as the Latent Gaussian random field.

3. INLA for Latent Gaussian Models (LGMs)

In this section, we start with a brief explanation about LGMs, the framework where we are going to fit the Bayesian Dirichlet regression (Section 3.1), followed by the main idea of the Laplace approximation (Section 3.2) and finishing with the INLA methodology (Section 3.3).
The main idea is to approximate the target with a scaled Gaussian density that matches the value and the curvature of the target distribution at the mode and evaluate the integral using this Gaussian instead. If \( x_0 \) is the point where \( f(x) \) has its maximum, then

\[
I_n \approx \int \exp(nf(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0))) \, dx
\]

\[
= \exp(nf(x_0)) \left( \frac{2\pi}{-nf''(x_0)} \right)^{\frac{1}{2}} = \bar{I}_n .
\]

If \( nf(x) \) is interpreted as the sum of log-likelihoods and \( x \) as the unknown parameter, the Gaussian approximation will be very accurate as \( n \to \infty \) under appropriate regularity conditions.

If we are interested in computing a marginal distribution \( p(\gamma) \) from a joint distribution \( p(\gamma, \theta) \), the Laplace approximation of the integral \( \int p(\gamma) \, d\gamma \) can be expressed as follows:

\[
p(\gamma) = \frac{p(\gamma)}{p(\gamma_{-1} | \gamma_1)} \bigg|_{\gamma_{-1} = \gamma_{-1}^*} \approx \frac{p(\gamma)}{p(\gamma_{-1} | \gamma_{1-1})} \bigg|_{\gamma_{-1} = \gamma_{-1}^* = \mu(\gamma)} ,
\]

where the first equality holds for any valid \( \gamma_{-1}^* \), and the mean \( \mu(\gamma) \) and precision \( Q(\gamma) \) are the parameters of the multivariate Gaussian density derived from the derivatives of \( \log p(\gamma) \) with respect to \( \gamma_{-1} \), for fixed \( \gamma_1 \). If the posterior is close to a Gaussian density, the results will be more accurate than if the posterior is very non-Gaussian. In this context, unimodality is necessary since the integrand is being approximated with a Gaussian at the mode \( \gamma_{-1}^* = \mu(\gamma) \).

### 3.3. INLA

The main idea of INLA approach is to approximate the posteriors of interest: the marginal posteriors for the latent field, \( p(\theta_k | y) \), and the marginal posteriors for the hyperparameters, \( p(\theta_k | y) \). These posteriors can be written as

\[
p(x_m | y) = \int p(x_m | \theta, y)p(\theta | y) \, d\theta ,
\]

\[
p(\theta_k | y) = \int p(\theta | y) \, d\theta_k .
\]

The nested formulation is used to compute \( p(x_m | y) \) by approximating \( p(x_m | \theta, y) \) and \( p(\theta | y) \), and then using numerical integration to integrate out \( \theta \). Similarly, \( p(\theta_k | y) \) can be computed by approximating \( p(\theta | y) \) and integrating out \( \theta_k \).

The marginal posterior distributions in (12) and (13) are computed using the Laplace approximation presented in Section 3.2. In Rue, Martino, and Chopin (2009) it is shown that the nested approach yields a very accurate approximation if applied to LGMs.

All this methodology can be used through R with the R-INLA package. For more details about R-INLA we refer the reader to Blangiardo and Cameletti (2015), Zuur, Ieno, and Saveliev (2017), Wang, Ryan, and Faraway (2018), Krainski et al. (2018), Moraga (2019), and Gómez-Rubio (2020), where practical examples and code guidelines are provided.

However, and despite the advantages of the R-INLA implementation, there are some limitations when we deal with multivariate responses. The Multinomial case is solved, as the Multinomial likelihood can be approximated using the Poisson trick. It consists of transforming the Multinomial likelihood into a Poisson likelihood with additional parameters (Baker 1994). However, there is no method available when we deal with compositional data. In what follows, we propose an expansion of the INLA method for a Dirichlet response variable.

### 4. Inference in Dirichlet Likelihoods

The INLA methodology is a tool that allows us to deal with a wide range of LGMs. However, when a multivariate response is required and several linear predictors are needed to explain it, the implemented R-INLA methodology has some limitations. In the particular case of Dirichlet likelihoods, the main idea to incorporate them in the R-INLA is first to approximate the effect of the log-likelihood on the posterior using the Laplace approach and then convert the multivariate response data into observations that R-INLA can deal with. The remainder of the section presents both the theoretical fundamentals to approximate the effect of the log-likelihood function \( \log p(Y | x, \theta) \) in the posterior using the Laplace approximation that provides the conditioned independent Gaussian pseudo-observations, and then an algorithmic representation of the method.

#### 4.1. Fundamentals of the Approximation

Let \( \eta_n := \eta_n^0 \) denote the linear predictor corresponding to the \( n \)th observation \( y_n := Y_n \). If \( l(y | x) \) represents \( -\log p(y | x) \) for any \( y \) and \( x \), then \( l(y_n | \eta_n) = -\log p(y_n | \eta_n) \) is the log-likelihood function expressed for the \( n \)th observation, being \( y_n \) and \( \eta_n \) vectors with \( C \) components.

Moreover, if \( \eta_n^0 \) is a vector with dimension \( C \), we express the gradient of \( l(y_n | \eta_n) \) as \( \nabla \eta_n = \nabla \eta_n(l(\eta_n^0, y_n)) \), and the Hessian of \( l(y_n | \eta_n) \) in \( \eta_n^0 \) as \( H_{\eta_n}^0 \). Depending on which is more computationally convenient, \( H_{\eta_n}^0 \) can be either the true Hessian \((\nabla^2 \eta_n(l(\eta_n^0, y_n))) \) or the expected Hessian \((E_{\eta_n}((\nabla^2 \eta_n(l(\eta_n^0, y_n)))) \) in \( \eta_n^0 \). Let \( L_n^0 \) be the result of applying the Cholesky factorization to \( H_{\eta_n}^0 \): \( L_n^0 = L_n^0(L_n^0)^T \).

**Theorem 4.1.** If the Laplace approximation method is applied to \( l(y_n | \eta_n) \) with respect to \( \eta_n^0 \), then the vector

\[
z_n^0 := (L_n^0)^T [y_n - (L_n^0)^T \eta_n^0] = (L_n^0)^T \eta_n^0 - (L_n^0)^{-1} \eta_n^0 ,
\]

is conditionally independent Gaussian distributed

\[
l(y_n | \eta_n) \approx l(z_n^0 | \eta_n) \approx l(z_n^0 | \eta_n) = \text{Constant} + \frac{1}{2} z_n^0 - (L_n^0)^T \eta_n^0 - (L_n^0)^{-1} \eta_n^0 ,
\]

that is, \( z_n^0 | \eta_n \sim N((L_n^0)^T \eta_n, I_d) \) and \( z_n^0 | \eta_n \sim N((L_n^0)^T \eta_n, I_d) \), and the constant value of the expression is \( l(Y_n | \eta_n) = -\frac{1}{2}(L_n^0)^T \eta_n^0 - (L_n^0)^{-1} \eta_n^0 \).

**Proof.** For proof of the theorem see Appendix B.  
\[\square\]
Theorem 4.1 allows us to convert the observation vector $y_n$ into Gaussian conditionally independent pseudo-observations $z^0_n$. More importantly, this theorem can be expanded to multiple observations. In particular, if we denote

$$\tilde{\eta}^0 = \begin{bmatrix} \eta^0_1 \\ \vdots \\ \eta^0_N \end{bmatrix}, \quad s^0 = \begin{bmatrix} s^0_1 \\ \vdots \\ s^0_N \end{bmatrix}, \quad L^0 = \begin{bmatrix} L^0_1 \\ \vdots \\ L^0_N \end{bmatrix},$$

then the following proposition stands.

**Proposition 4.2.** The matrix

$$z^0 := (L^0)^T \tilde{\eta}^0 - (L^0)^{-1} s^0 \tilde{\eta}^0$$

is conditionally independent Gaussian distributed by columns,

$$z^0 \mid \tilde{\eta} \sim \mathcal{N}((L^0)^T \tilde{\eta}, I_{CN}).$$

**Proof.** For proof of this proposition see Appendix B. \hfill \square

This approximation has been constructed for a generic $\tilde{\eta}^0$, but, as we are interested in building a Gaussian approximation of the effect of the likelihood on the posterior distribution, $\tilde{\eta}^0$ has been chosen as the posterior mode of $I(\tilde{\eta} \mid Y)$. Then:

$$p(Y \mid x, \theta) = p(Y \mid \tilde{\eta}) \approx p(z^0 \mid \tilde{\eta}) = p(z^0 \mid x, \theta).$$

The model posterior is factorized as

$$p(\theta, x \mid y) = p(\theta \mid y) \cdot p(x \mid y, \theta),$$

and the approximation is factorized as

$$\tilde{p}(\theta, x \mid z^0) = \tilde{p}(\theta \mid z^0) \cdot \tilde{p}(x \mid z^0, \theta).$$

Note that the approximation is constructed for a generic $\tilde{\eta}$, in other words, this approximation is conditionally dependent on the linear predictor. The linear predictor can be formed by fixed effects or random effects. Here, in the implementation of the algorithm, we focus on the case where fixed effects and iid random effects are added to the linear predictor.

### 4.2. The Algorithm

In what follows, we depict the different steps to compute the marginal posterior distributions of the latent field, $p(x_n \mid Y)$, and the marginal posterior distribution of the hyperparameters $p(\theta_k \mid Y)$. To obtain them, it is necessary to numerically find the mode of the posterior distribution of the linear predictor $\tilde{\eta}^0$. This can be done by means of an iterative method in a similar way to inlabru (Bachl et al. 2019). We define a functional $f(\tilde{p}_u)$ of the posterior distribution at $u$ which generates a latent field configuration. Our aim is to find an invariant point of the functional, so that $x^0 = f(\tilde{p}_u)$. The choice for $f(\cdot)$ is the joint conditional mode $f(\tilde{p}_u) = \arg\max \tilde{p}_u(x \mid z^0, \theta^0)$, being $\theta^0 = \arg\max \tilde{p}_u(\theta \mid z^0)$. The final algorithm is

1. Let $x^1$ and $\theta^1$ be initial candidate points for the latent variables and the hyperparameters.
2. Locate a good candidate in the latent variables for the computation of the new pseudo-observations. Compute the mode $(x^2)$ in $p(x \mid Y, \theta^1)$ by means of a quasi-Newton method (Dennis and Moré 1977) with line search strategy and Armijo conditions (Nocedal and Wright 2006) in $-\log(p(x \mid Y, \theta^1)) \propto l(Y \mid x, \theta^1) + l(x \mid \theta^1)$, being $p(x \mid \theta^1)$ multivariate Gaussian as we are in the LGMs context. $\tilde{\eta}^1$ can be easily calculated from the expression $\tilde{\eta} = Ax$.
3. Calculate the conditionally independent Gaussian pseudo-observations $\tilde{z}^1$. In this case, at the modal configuration established $x^1$, the Hessian matrix $H^1_\tilde{\eta}$ is computed. If the submatrix corresponding to the $n$th individual $H^1_{\tilde{\eta}^n}$ is not positive definite, the expected Hessian is used instead to guarantee a positive definite $H^1$. Following the approximation previously presented, the Cholesky factorization is computed in $H^1_\tilde{\eta} = L^1(L^1)^T$. The gradient $\tilde{g}^1_\tilde{\eta}$ is also calculated in $\tilde{\eta}^1$. According to the equation (16), the scale and rotation of the original observations are done to get the pseudo-observations $\tilde{z}^1$.
4. Call R-INLA. As pseudo-observations $\tilde{z}^1$ are conditionally independent Gaussian observations, we are able to call R-INLA. If in Step 2 algorithm has converged for a given tolerance, the posterior distributions obtained here are the one that we were looking for: $\tilde{p}(\theta \mid \tilde{z}^1)$ and $\tilde{p}(x \mid \tilde{z}^1)$. Else, repeat from Step 1 defining $x^3 = f(\tilde{p}_u)$ as the new starting point for the latent variables, and $\theta^3 = \arg\max \tilde{p}_u(\theta \mid \tilde{z}^1)$ the new starting point for the hyperparameters.

After depicting the complete method, we focus on an implementation in R (R Core Team 2021) of this approximation.

### 5. The R-package dirinla

In what follows we present dirinla, an R (R Core Team 2021) package developed to fit Dirichlet regression models. This package can be installed and upgraded via the repository https://github.com/inlabru-org/dirinla. To show how it works, we present an example of a Dirichlet regression without random effect, although, they can be included in the model, as done in Section 6. Then, this section is divided in three parts: the first one presents the necessary commands to perform a simulation from a Dirichlet regression model; the second one is devoted to show how to fit those models; and the last one depicts how to predict using the package (An extended version of this example is available in the vignette of the R-package dirinla). In particular, we first illustrate how to simulate 100 data points from a Dirichlet regression model with three different categories and one different covariate per category:

$$Y_{ni} \sim \text{Dirichlet}(\alpha_{1n}, \ldots, \alpha_{3n}), n = 1, \ldots, 100,$$

$$\log(\alpha_{1n}) = \beta_{01} + \beta_{11} v_{1n},$$

$$\log(\alpha_{2n}) = \beta_{02} + \beta_{21} v_{1n},$$

$$\log(\alpha_{3n}) = \beta_{03} + \beta_{31} v_{1n}.\]
\begin{align*}
\log(\alpha_{2n}) &= \beta_{20} + \beta_{12}s_{2n}, \\
\log(\alpha_{3n}) &= \beta_{23} + \beta_{13}s_{3n},
\end{align*}

(21)

being the parameters that compose the latent field $\beta_{01} = -1.5$, $\beta_{02} = -2$, $\beta_{03} = 0$ (the intercepts), and $\beta_{11} = 1$, $\beta_{12} = 2.3$, $\beta_{13} = -1.9$ (the slopes). Note that covariates are different for each category. This could be particularized for a situation where all of them are the same.

For simplicity, covariates are simulated from a Uniform distribution on (0,1). To posteriorly fit the model, and following the structure of LGMs, Gaussian prior distributions are assigned with precision $10^{-4}$ to all the elements of the Gaussian field.

### 5.1. Data Simulation

This section presents an example of how simulation can be conducted using the functions of `dirinla`.

First, we simulate the covariates from a Uniform(0,1):

```r
R> N <- 100
R> V <- as.data.frame(matrix(runif(N, 0, 1), ncol = 3))
R> names(V) <- paste0('v', 1:3)
```

We then define the formula that we want to fit to keep the values of the different categories in a list. This object will be used to construct the $A$ matrix. We use the function `formula_list()` from the package `dirinla`.

```r
R> formula <- y ~ 1 + v1 | 1 + v2 | 1 + v3
R> names_cat <- formula_list(formula)
```

The values for the parameters composing the latent field are assigned to conduct the simulation. As we have previously depicted, $\beta_{01} = -1.5$, $\beta_{02} = -2$, $\beta_{03} = 0$ are the intercepts, and $\beta_{11} = 1$, $\beta_{12} = 2.3$, $\beta_{13} = -1.9$ are the slopes:

```r
R> x <- c(-1.5, 1, -2, 2.3, 0, -1.9)
```

We call the function `data_stack_dirich()` of the package `dirinla` to construct the $A$ matrix presented in previous sections. This function uses the `inla.stack()` structure of the package R-INLA. As a consequence, the returning object is an `inla.stack` object. Observe that the arguments are the response variable $y$ (in this case it has not been generated yet), the names of the categories `covariates`, a matrix with the values of the covariates `data`, the number of categories `c` and the number of observations `N`. The sparse matrix $A$ is then computed.

```r
R> mus <- exp(x) / sum(exp(x))
R> C <- length(names_cat)
R> A_construct <- +  data_stack_dirich(y = +  as.vector(rep(NA, N * C)),
+  covariates = names_cat,
+  data = V,
+  d = C,
+  n = N)
```

The next step is to construct the linear predictor as $\tilde{\eta} = Ax$ using the parameters fixed in the latent field. Using the exponential transformation it is easy to get the parameters $\alpha$ of the Dirichlet distribution:

```r
R> eta <- A_construct
R> alpha <- exp(eta)
```

The last stage is to generate the response variable using the function `rdirichlet()` from `DirichletReg` (Maier 2014). The output is a matrix with the response variable summing their rows up to one.

```r
R> y <- rdirichlet(N, alpha)
```

### 5.2. Fitting the Model

We now present the functions of `dirinla` needed to fit Dirichlet regression models, the main one being `dirinlareg`. This function carries out all the steps presented in Section 4, and its use (for the simulated data from previous section) is as follows:

```r
R> model.inla <- dirinlareg( +  formula = y ~ 1 + v1 | 1 + v2 | 1 + v3,
+  y = y,
+  data.cov = V,
+  prec = 0.0001,
+  verbose = FALSE)
```

where we have to specify the model formula, the response variable $Y$ in a matrix format, the `data.frame` with the covariates `data.cov`, and the precision of the Gaussian prior (`prec`) for the latent field $x$. If we want to follow the process step by step, we can add the instruction `verbose = TRUE`.

Once the model is fitted, we can summarize the posterior distribution of the fixed effects by means of the function `summary_fixed` applied to the object generated. This object belongs to the `dirinlaregmodel` class. Three model selection criteria are also displayed: Deviance Information Criterion (Spiegelhalter et al. 2002, DIC), Watanabe-Akaike information criteria (Gelman, Hwang, and Vehtari 2014, WAIC), and the mean of the logarithm of the conditional predictive ordinate (Gneiting and Raftery 2007, LCPO). Lastly, the number of observations and the number of categories are also depicted (Table 1). Using the implemented `plot` method, we obtain the marginal posterior distributions of the latent field for the different categories (Figure 1). These marginals and a summary of each one are stored in `model.inla$marginals_fixed` and `model.inla$summary_fixed` (see example.R from the supplementary code to see the details of the code). The `plot` method also depicts the posterior predictive distribution in the simplex (Figure 2).

Finally, the posterior distribution for the scale parameters of the Dirichlet $\alpha$ can also be computed. In particular, `model.inla$marginals_fixed` and `model.inla$summary_fixed` provide the marginals and a summary for each category. Mean parameters can also be obtained via `model.inla$marginals_means` or `model.inla$summary_means`, and similarly, `model.inla$marginals_precision` or `model.inla$summary_precision` provides the precision parameters.
Table 1. Output obtained when summary command is employed in a dirinnlaregmodel object.

|        | mean   | sd      | 0.025quant | 0.5quant   | 0.975quant | mode   |
|--------|--------|---------|------------|------------|------------|--------|
| y1     | Intercept | -1.5542 | 0.2014     | -1.9497    | -1.5542    | -1.1591 | -1.5542 |
|        | v1     | 1.0181  | 0.3690     | 0.2936     | 1.0181     | 1.7420  | 1.0181  |
| y2     | Intercept | -1.7601 | 0.2329     | -2.2174    | -1.7601    | -1.3032 | -1.7601 |
|        | v2     | 1.9847  | 0.3971     | 1.2051     | 1.9847     | 2.7636  | 1.9847  |
| y1     | Intercept | -0.0593 | 0.2500     | -0.5501    | -0.0593    | 0.4312  | -0.0593 |
|        | v3     | -1.9470 | 0.4158     | -2.7633    | -1.9470    | -1.1314 | -1.9470 |

DIC = 1821.0247, WAIC = 1827.6268, LCPO = 913.8319
Number of observations: 100
Number of Categories: 3

Figure 1. Marginal posterior distributions of the latent field for the different categories. Real values are indicated with a red vertical line. The sample size is 50.

Figure 2. Posterior predictive distribution in the simplex. Points represent the original data.
5.3. Prediction

In most cases, practitioners want to be able to predict the composition of a new observation. The package also provides a function `predict` to compute posterior predictive distributions for new individuals. To show how it works, we now present how to predict for a value of \( v1 = 0.2, v2 = 0.5, \) and \( v3 = -0.1 \):

```r
R> model.prediction <- +  predict(model.inla, +  data.pred.cov = data.frame(v1 = 0.2, +  v2 = 0.5, +  v3 = -0.1))
```

The resulting object also belongs to the `dirinlaregmodel` class. In a similar way as above, the elements `summary_predictive_alphas` and `marginals_predictive_alphas` describe the posterior predictive distribution for the scale parameters of the Dirichlet \( \alpha \), obtained for the new values of the covariates. In addition, means \( (\mu) \) and precisions \( (r) \) are available via `summary_predictive_means`, `marginals_predictive_means`, `summary_predictive_precision` and `marginals_predictive_precision`.

6. Simulation Studies

This section provides a comparison of the performance of the INLA approach for Dirichlet regression models using the `dirinla` package with the widely used method for Bayesian inference using MCMC algorithms, R-JAGS (Plummer 2016). The comparison was performed in four different simulated scenarios, and consists of comparisons of computational times and method accuracy.

In all the cases to compare `dirinla` with R-JAGS, we employed three different methods to make inference: a standard application of the R-JAGS package with a number of iterations enough to achieve a given effective sample size; the INLA methodology through the `dirinla` package; and a "long" application of the R-JAGS package (for simplicity called long R-JAGS from now on), in this case with a large amount of iterations to get really good representation of the posterior distributions. All computations were performed on a computer with a processor Intel(R) Core(TM) i5-10500 CPU @ 3.10GHz and 32 Gb RAM memory.

To compare the different ways of inference, we implemented four different strategies. The first one was to plot the approximate marginal posterior distributions of each parameter jointly with the real value. The second one consisted on comparing computational times between the three different methods. The remaining two strategies involved the computation of two different ratios:

\[
\text{ratio}_1 = \frac{E(\phi_{\text{method}}) - E(\phi_{\text{long R-JAGS}}))}{SD(\phi_{\text{long R-JAGS}})} , \quad (22)
\]

\[
\text{ratio}_2 = \frac{SD(\phi_{\text{method}})}{SD(\phi_{\text{long R-JAGS}})} , \quad (23)
\]

where `method` refers to the method we want to compare with long R-JAGS. It can be either `dirinla` or R-JAGS. \( \phi \) is the parameter of interest. \( E(\cdot) \) represents the computed mean of the marginal posterior distribution, and \( SD(\cdot) \) is the computed standard deviation of the posterior distribution. In what follows, we make a brief description of the four different scenarios with the aim of extending the explanation at a later stage.

1. Simulation 1: the first scenario comprised the simulation of data coming from a Dirichlet regression with just intercepts in the model, with different data sizes.
2. Simulation 2: the second scenario is similar to the previous one, but adding some complexity to the model. In particular, by adding a covariate per category, and checking with different data sizes.
3. Simulation 3: in the third scenario, we show how the method behaves when random effects are added to the model. The simulations and model include random effects levels.
4. Simulation 4: the objective of this last scenario is to check how our INLA approach for Dirichlet regression behaves when the number of categories increases. We simulated data from a Dirichlet regression with no slopes but increasing the number of categories.

6.1. Simulation 1

Our first setting is based on a Dirichlet regression with four categories and one parameter per category, the intercept, that is

\[
Y_{ni} \sim \text{Dirichlet}(\alpha_{1n}, \ldots, \alpha_{4n}), \quad n = 1, \ldots, N,
\]

\[
\log(\alpha_{1n}) = \beta_1, \quad \log(\alpha_{2n}) = \beta_2, \quad \log(\alpha_{3n}) = \beta_3, \quad \log(\alpha_{4n}) = \beta_4. \quad (24)
\]

Five different datasets of sizes \( N = 50, 100, 500, 1000, 10,000 \) with this structure were simulated letting \( \beta_0, \tau \) to be \(-2, 4, 1.2, -3.1, 1.3\), respectively. We used vague prior distributions for the latent field \((x)\). In particular, \( p(x | \tau) \sim \mathcal{N}(0, \tau = 0.0001) \), \( m = 1, \ldots, 4. \) As the response values are not close to 0 and 1, no transformation was needed.

As above mentioned, for each simulated dataset, we employed three different methods to make inference: a standard application of the R-JAGS package with 2000 iterations, a burnin of 200, a thin number of 5 and 3 chains; the INLA methodology through the `dirinla` package; and long R-JAGS, using 1,000,000 iterations with a burnin of 100,000, a thin number of 5 and 3 chains. To compare methodologies, the strategies presented at the beginning of the section were used.

As seen in Figure 3 (the rest of the plots are shown in the Simulation 1 of the supplementary material in https://jmartinezminaya.github.io/supplementary/CODA/tests.html), the posterior densities have similar shape, with the new method tending to agree more with the long R-JAGS result than the more variable short run R-JAGS does, illustrating how our method reduces estimator variability, at the potential cost of a generally small bias. With respect the computational effort needed to get those results, Figure 4 displays that, `dirinla` methodology has a faster computational speed for given data sizes, for the given R-JAGS chain lengths. In Figure 5, we see that ratio_1-`dirinla`
is so close to 0 as happen with ratio1-R-JAGS, and ratio2-dirinla is always close to 1, similar to ratio2-R-JAGS, meaning that using dirinla we obtain as good approximations as with R-JAGS reducing considerably the computational cost.

6.2. Simulation 2

The second setting is based on a Dirichlet regression with a different covariate per category:

\[ Y_{ni} \sim \text{Dirichlet}(\alpha_{1n}, \ldots, \alpha_{4n}), \]  
\[ n = 1, \ldots, N, \]
\[ \log(\alpha_{1n}) = \beta_{01} + \beta_{11}v_{1n}, \]
\[ \log(\alpha_{2n}) = \beta_{02} + \beta_{12}v_{2n}, \]
\[ \log(\alpha_{3n}) = \beta_{03} + \beta_{13}v_{3n}, \]
\[ \log(\alpha_{4n}) = \beta_{04} + \beta_{14}v_{4n}. \]

\[ (25) \]

Again, we simulated five different datasets of sizes \( N = 50, 100, 500, 1000, 10,000 \). We set values for \( \beta_{0c} \) and \( \beta_{1c}, c = 1, \ldots, 4 \) to \(-1.5, 1, -3, 1.5, 2, -3, -1.5, 5\), respectively, and we simulated covariates from a Uniform distribution with mean in the interval \((0, 1)\). We assigned vague prior distributions for the latent field \( (x = [\beta_{0c}, \beta_{1c}, c = 1, \ldots, 4]). \) Particularly, \( p(x_m) \sim N(0, \tau = 0.0001), m = 1, \ldots, 8. \) As the data generated did not present zeros and ones, we did not use any transformation.

We employed the same three different inference methods with the same configurations and the four strategies used in the previous simulation. In Figure 6 (see Simulation 2 in https://jmartinez-minaya.github.io/supplementary/CODA/tests.html for further plots), the posterior distributions are similar in both \( \beta_{0c} \) and \( \beta_{1c}, c = 1, \ldots, 4. \) Figure 7 depicts that dirinla has a faster computational speed for given data sizes and for given R-JAGS chain lengths. In Figure 8, we show that ratio1-dirinla and ratio1-R-JAGS are so close to 0 in all cases, and ratio2-dirinla and ratio2-R-JAGS are always close to 1. Again, we conclude that using dirinla we get similar approximations to R-JAGS, while substantially reducing the computational cost.
Figure 5. Simulation 1: comparing accuracy between dirinla and R-JAGS computing two different measures: ratio 1 (a) and ratio 2 (b) for $\beta_0, \mu_c, c = 1, \ldots, 4$.

Figure 6. Simulation 2: marginal posterior distributions of the latent field for the different categories, and using different methodologies R-JAGS, dirinla, and long R-JAGS, when the sample size is 100. Black vertical lines represent real values.

6.3. Simulation 3

The third setting is based on a Dirichlet regression with a different covariate per category without intercept and adding two shared independent random effects.

\begin{align*}
Y_{nn} & \sim \text{Dirichlet}(\alpha_{1n}, \ldots, \alpha_{4n}), n = 1, \ldots, N, \\
\log(\alpha_{1n}) & = \beta_{11} v_{n1} + \omega_{1n}, \\
\log(\alpha_{2n}) & = \beta_{12} v_{n2} + \omega_{1n}, \\
\log(\alpha_{3n}) & = \beta_{13} v_{n3} + \omega_{2n}, \\
\log(\alpha_{4n}) & = \beta_{14} v_{n4} + \omega_{2n}.
\end{align*}

We simulated four different datasets of sizes $N = 50, 100, 500, 1000$. We set values for $\beta_{1c}$ for $c = 1, \ldots, 4$ to $-1.5, 2, 1, -3$, respectively, and we simulated covariates from a Uniform distribution on the interval $(-1, 1)$. Random effects $\omega_1$ and $\omega_2$
Figure 7. Simulation 2: computational time in seconds for the different simulated data with $N = 50, 100, 500, 1000, \text{ and } 10,000$, and with the different methodologies: R-JAGS, dirinla, and long R-JAGS.

$\omega_2$ were simulated from Gaussian distributions with mean 0 and standard deviations $\sigma_1 = 1/2$ (precision $\tau_1 = 4$) and $\sigma_2 = 1/3$ (precision $\tau_2 = 9$) varying the levels of the factor ($I$), in particular, they were set to $I = 2, 5, 10, 25$. The
Figure 9. Simulation 3: marginal posterior distributions of the parameters corresponding to the fixed effects using different methodologies R-JAGS, dirinla with different priors and long R-JAGS, when the sample size is 100 and the levels of the factor (I) are 2. Black vertical lines represent real values.

Figure 10. Simulation 3: marginal posterior distributions of the hyperparameters with their logarithmic transformations using different methodologies R-JAGS, dirinla, and long R-JAGS, when the sample size is 100 and the levels of the factor (I) are 2. Priors are also depicted in the plot. Black vertical lines represent real values.

Figure 11. Simulation 3: computational time in seconds for the different simulated data with N = 50, 100, 500, and 1000, and with the different methodologies: R-JAGS, dirinla, and long R-JAGS when the levels of the factor (I) are 2.

$i_n$ sub-index assigns each individual $n$ to a level of the factor.

As we are in the context of Bayesian LGMs, we established Gaussian prior distributions for the latent field, in this case, formed by the parameters corresponding to the fixed effects and the random effects. In particular, we assigned Gaussian prior distributions with mean 0 and precision 0.0001 to $\{\beta_1, c = 1, \ldots, 4\}$, and Gaussian priors distribution with mean 0 and precisions $\tau_1$ and $\tau_2$ for the two shared random effects $\omega_1$ and $\omega_2$. Two types of priors for the $\tau_1$ and $\tau_2$ parameters were employed. Half-Gaussian priors with location 0 and precision parameter 1 were used for R-JAGS, long R-JAGS, and dirinla. For dirinla, and additional model with a PC-prior(1, 0.01) (Simpson et al. 2017) was also used. The generated data did not contain zeros and ones, so we did not use any transformation. As these models have two hyperparameters, we increased the number of iterations of the R-JAGS method to 20,000 and burnin to 2000 to achieve a given effective sample size of the MCMC method.

In Figures 9 and 10 (see Simulation 3 in the supplementary material, https://jmartinez-minaya.github.io/supplementary/CODA/tests.html for the complete simulation summaries), we display the posterior distribution of the parameters and the hyperparameters with two levels in the factors obtained with the different methods and the different priors. In the case of the precision parameters, we also depict the priors used for
Figure 12. Simulation 3: comparing accuracy between dirinla and R-JAGS when the prior used is Half Gaussian computing two different measures: ratio1 (a) and ratio2 (b) for β1, c = 1, . . . , 4, and σ1 and σ2, when the levels of the factor (f) are 2. Figure 11 shows that dirinla provides higher computational efficiency. And, in view of Figure 12, ratio1–dirinla and ratio1–R-JAGS are both closed to 0. Similar is the behavior of ratio2, in both cases, it is closed to 1, proving that when we include random effects in the model, we can also get similar approximations by decreasing the computational cost. If we look at the rest of the simulations conducted (See Simulation 3 in the supplementary material, https://jmartinez-minaya.github.io/supplementary/CODA/tests.html), we observe that large N and small J shows the influence of the different priors. And large J display the difficulties of dirinla in approaching the posterior of the hyperparameters.

6.4. Simulation 4
This last simulation is based on a Dirichlet regression with just one parameter per category and without covariates as in Simulation 1. The idea is that for a fixed data size, we increase the number of categories and then simulate from all those models and compare inferential methods. In particular, we used N = 100 and C = 5, 10, 15, 20, and 30. We also employed R-JAGS, dirinla with the same configuration as in simulations 1 and 2. However, in the case of long
R-JAGS, iterations were reduced to 100,000 and burnin to 10,000.

After comparing computational times needed (Figure 13), we also plot (see Figure 14) the measures ratio₁ and ratio₂ computed for the Dirichlet means $\mu_c$, $c = 1, \ldots, C$ to see how accurate is our method in comparison with R-JAGS. Once more, dirinla shows a higher computational speed than R-JAGS and a similar accuracy.

7. Real Example: Arabidopsis thaliana

After validating the use of the package and the approximation in simulated examples, this section shows an application of the INLA approach for Dirichlet regression models in a real setting. In particular, we worked with a collection of 301 accessions of the annual plant Arabidopsis thaliana in the Iberian Peninsula using the four genetic clusters (gc) inferred in Martínez-Minaya et al. (2019): gc1, gc2, gc3, and gc4; categories whose values are summing up to one. Here, we depict how these four gc’s can be modeled using Dirichlet regression in terms of two covariates: the annual mean temperature (BIO1) and the annual precipitation (BIO12) (Martinez-Minaya et al. 2019). The aim is explain the distribution of each gc using those two climatic covariates. The complete dataset was downloaded from the repository https://zenodo.org/record/2552025#.YtbLgHZByUl.

The Dirichlet regression model that relates the proportion of each gc, that is, the multivariate response $Y_{\bullet n}$, to the
\[ Y_{*m} \sim \text{Dirichlet}(\alpha_{1n}, \ldots, \alpha_{4n}), \quad n = 1, \ldots, 301, \]
\[ \log(\alpha_{1n}) = \beta_{01} + \beta_{11}BI0_{1n} + \beta_{21}BI0_{12n}, \]
\[ \log(\alpha_{2n}) = \beta_{02} + \beta_{12}BI0_{1n} + \beta_{22}BI0_{12n}, \]
\[ \log(\alpha_{3n}) = \beta_{03} + \beta_{13}BI0_{1n} + \beta_{23}BI0_{12n}, \]
\[ \log(\alpha_{4n}) = \beta_{04} + \beta_{14}BI0_{1n} + \beta_{24}BI0_{12n}. \]  

Remark that due to the different scale of the covariates, we have scaled both to have mean 0 and standard deviation 1. We assigned vague prior distributions for the latent field, in particular
\[ p(x_m) \sim N(0, \tau = 0.01), \quad m = 1, \ldots, 12. \] As the data did not include zeros and ones, we did not do any transformation. In a similar approach as in the previous section, the number of iterations used in R-JAGS were 20,000 with a burnin of 2000, a thinning of 5 and 3 chains, while in the case of the long R-JAGS we used 1,000,000 of iterations with a burnin of 100,000, a thinning of 5 and 3 chains.

Figure 15 displays the marginal posterior distributions for \( \beta_{0c} \), \( \beta_{1c} \), and \( \beta_{2c}, c = 1, \ldots, 4 \). In most cases distributions obtained with R-JAGS perfectly match with those obtained using dirinla. R-JAGS took 405.152 sec, dirinla 8.517 sec and long R-JAGS 9886.677.

As in the previous examples, we also computed the measures \( \text{ratio}_1 \) and \( \text{ratio}_2 \) for dirinla and R-JAGS. Results are depicted in Table 2 and, we can observe that approaches done for dirinla seems to have a good behavior in comparison with R-JAGS not only in accuracy but also in computational cost.

**8. Concluding Remarks and Future Work**

In this article the INLA methodology is extended to fit models with Dirichlet response. In particular, we present both the calculations and a package to make inference and prediction for Dirichlet regression. The main idea underneath the proposed method is to approximate the multivariate likelihoods with univariate ones that can be fitted by R-INLA, in particular, Gaussian likelihoods. This idea is similar to the one proposed for modeling Multinomial likelihood in R-INLA, where using the Poisson trick (Baker 1994) to reparametrize the model we just need to fit independent Poisson observations. Simpson et al. (2016) use a similar strategy, constructing a Poisson approximation to the true log-Gaussian Cox process likelihood and enabling to make...
inference on a regular lattice over the observation window by counting the number of points in each cell. This technique has been already implemented in the R package \texttt{inlabru} (Bachl and Lindgren 2018; Bachl et al. 2019).

It is widely known that there exist a close connection between the Bayesian integrated likelihood and an adjusted profile likelihood (Lee, Nelder, and Pawitan 2018). In particular, the integrated likelihood is approximately the adjusted profile likelihood in the case of orthogonal parameters (Cox and Reid 1987), we just need to use the Laplace integral approximation to proof it. For the case of Dirichlet response that we deal with in this Appendix, we have not found a similar approximation for the Dirichlet regression. Maybe, the method we have developed can help to do approximations using the adjusted profile likelihood of Dirichlet response.

Finally, all examples presented here are focused on models that include fixed and iid random effects. As we are converting the multivariate observations to conditionally independent Gaussian observations that only depend on the linear predictor, and not directly on the individual latent variables, it should be expected that more complex random effects can be incorporated in the model structures. In particular, the \textit{R-INLA} method itself uses the same hierarchical model building technique to separate general complex structured random effect components (including spatial, temporal, and spatio-temporal models) from the observation models, that are only linked to the combined linear predictor values. For such more general models, the computational scaling cost benefits in comparison with \textit{R-JAGS} should be even more pronounced, due to the sparse matrix algebra taken advantage of by \textit{R-INLA}.

**Appendices**

**Appendix A. Equivalence between Parameterizations**

In this Appendix, we present the proof for the equivalence between the two parameterizations on Dirichlet regression models presented in the Equations (3) and (4).

As mentioned in the text, we just need to construct a matrix \( V \) with dimension \( N \times J \) which contains all the observed values for all the observations of all the covariates using in the different categories. Note that if the used covariates are the same in all the categories \( V \) is equal to \( V^{(c)} \). We rewrite equation (3) as

\[
g(\alpha_{cn}) = n^*_{cn} = V_{cn} \tilde{\beta}_{nc} + \omega_{cn},
\]

being \( \tilde{\beta}_{nc} \) a \( J \) column vector having \( l_{cn} \) non zero values, and being zero the rest of the elements; and Equation (4)

\[
\mu_{cn} = \frac{\exp(n^*_{cn} \omega_{cn})}{\sum_{c=1}^{C} \exp(n^*_{cn} \omega_{cn})} = \frac{\exp(V_{cn} \tilde{y}_{nc} + \omega_{cn})}{\sum_{c=1}^{C} \exp(V_{cn} \tilde{y}_{nc} + \omega_{cn})},
\]

being \( \tilde{y}_{nc} \) a \( J \) column vector having \( l_{cn} \) nonzero values and being zero the rest of the elements. \( \omega_{cn}^{(c)} \) represents a realization of a random effect for this parameterization.

In the alternative parameterization (28), as we have \( C \) categories for which the means must always sum up to 1, we employ a Multinomial logit strategy as in Multinomial regression, where the linear predictor of one category is set to zero, whereby it is virtually omitted and becomes the reference. Let \( \eta_{1n}^{\mu} \) be the reference, then:

\[
\eta_{j,n}^{\mu} = 0.
\]

Rewriting the Equation (29) reveals that

\[
\mu_{cn} = \frac{\exp(V_{cn} \tilde{y}_{nc} + \omega_{cn})}{\sum_{c=1}^{C} \exp(V_{cn} \tilde{y}_{nc} + \omega_{cn})} = \frac{\exp(V_{cn} \tilde{\beta}_{nc} - V_{cn} \tilde{\beta}_{n1} + \omega_{cn} - \omega_{1n})}{\sum_{c=1}^{C} \exp(V_{cn} \tilde{\beta}_{nc} - V_{cn} \tilde{\beta}_{n1} + \omega_{cn} - \omega_{1n})},
\]

so that

\[
\tilde{y}_{nc} = \tilde{\beta}_{nc} - \tilde{\beta}_{n1},
\]

and

\[
\omega_{cn}^{\mu} = \omega_{cn} - \omega_{1n},
\]

as we wanted to proof.

**Appendix B. Likelihood Approximation Effect**

This Appendix presents both the proof for the Theorem 4.1 and the Proposition 4.2, which is the expansion of the theorem to multiple observations. Let \( \eta_{n} := \eta_{an} \) denote the linear predictor corresponding to the uth observation \( y_{n} := Y_{an} \); let also define \( l(y | x) = -\log p(y | x) \) for any \( y \) and \( x \), and so let \( l(y_{n} | \eta_{n}) = -\log p(y_{n} | \eta_{n}) \) denote the log-likelihood function expressed for the uth observation, being \( y_{n} \in \mathbb{R}^{C} \) and \( \eta_{n} \in \mathbb{R}^{C} \). Using the Taylor series expansion in vector \( \eta_{n}^{0} \) we obtain the approximation:

\[
l(y_{n} | \eta_{n}) \approx l(y_{n} | \eta_{n}^{0}) = l(y_{n} | \eta_{n}^{0}) + \left[ \nabla_{\eta_{n}} l(y_{n} | \eta_{n}) \right]^{T} (\eta_{n} - \eta_{n}^{0}) + \frac{1}{2} (\eta_{n} - \eta_{n}^{0})^{T} H_{\eta_{n}} (\eta_{n} - \eta_{n}^{0}) = l(y_{n} | \eta_{n}^{0}) + (\eta_{n}^{0} - \eta_{n}^{0})^{T} H_{\eta_{n}} (\eta_{n} - \eta_{n}^{0}) + \frac{1}{2} (\eta_{n} - \eta_{n}^{0})^{T} H_{\eta_{n}} (\eta_{n} - \eta_{n}^{0})\]

\[
= C_{1} + \frac{1}{2} (\eta_{n} - \eta_{n}^{0})^{T} ((H_{\eta_{n}})^{-1} S_{\eta_{n}}^{0} - S_{\eta_{n}}^{0} (H_{\eta_{n}})^{-1} S_{\eta_{n}}^{0})),
\]

where \( S_{\eta_{n}}^{0} = \nabla_{\eta_{n}} l(y_{n} | \eta_{n}^{0}) \), and \( H_{\eta_{n}}^{0} \) is either the true Hessian \( (\nabla_{\eta_{n}}^{2} l(y_{n} | \eta_{n}^{0}, y_{n})) \) or the expected Hessian \( E_{\eta_{n}}[\eta_{n} | (\nabla_{\eta_{n}}^{2} l(y_{n} | \eta_{an}, y_{n}))] \). \( C_{1} \) is a constant whose value is \( l(y_{n} | \eta_{n}^{0}) - \frac{1}{2} (\eta_{n}^{0} - \eta_{n}^{0})^{T} H_{\eta_{n}}^{0} (\eta_{n}^{0} - \eta_{n}^{0}) \). Considering now the Cholesky factorization of \( H_{\eta_{n}}^{0} \), \( H_{\eta_{n}}^{0} = L_{n}^{0} (L_{n}^{0})^{T} \), expression (33) can be rewritten as follows:

\[
l(y_{n} | \eta_{n}) \approx l(y_{n} | \eta_{n}^{0}) = C_{1} + \frac{1}{2} (\eta_{n}^{0} - L_{n}^{0})^{T} (\eta_{n}^{0} - (H_{\eta_{n}}^{0})^{-1} S_{\eta_{n}}^{0})),
\]

Defining

\[
z_{n}^{0} := (L_{n}^{0})^{T} (\eta_{n}^{0} - (H_{\eta_{n}}^{0})^{-1} S_{\eta_{n}}^{0}) = (L_{n}^{0})^{T} (\eta_{n}^{0} - (L_{n}^{0})^{T} S_{\eta_{n}}^{0}),
\]

a conditionally Gaussian approximation is constructed.

\[
l(y_{n} | \eta_{n}) \approx l(z_{n}^{0} | \eta_{n}) = C_{1} + \frac{1}{2} (z_{n}^{0} - (L_{n}^{0})^{T} \eta_{n})^{T} [z_{n}^{0} - (L_{n}^{0})^{T} \eta_{n}].
\]
Thus, \( z_n^0 \mid \eta_n \sim \mathcal{N}((L_n^0)^T \eta_n, I_d) \), that is, \( z_{nc}^0 \mid \eta_n \sim \mathcal{N}((L_n^0)^T \eta_n, 1) \). The observation vector \( y_n \) has been converted into conditionally independent Gaussian pseudo-observations \( z_{nc}^0 \). This approximation can be expanded to the \( N \) observations.

To present a proof for Proposition (4.2), we rewrite \( l(Y \mid \eta) \) for all the observations \( N \) as in Equation (34).

\[
\begin{align*}
  l(Y \mid \eta) & \approx \\
  & \approx NC_1 + \frac{1}{2} \sum_{n=1}^{N} ((L_n^0)^T \eta_n - (L_n^0)^T (\eta_n^0 - (H_n^0)^{-1} g_n^0))^T \\
  & \quad \times [(L_n^0)^T \eta_n - (L_n^0)^T (\eta_n^0 - (H_n^0)^{-1} g_n^0)].
\end{align*}
\]

Using the notation

\[
\begin{align*}
  \eta^0 &= \begin{bmatrix} \eta_{n}^0 \end{bmatrix}_{CN \times 1}, \\
  g_n^0 &= \begin{bmatrix} g_{nc}^0 \end{bmatrix}_{CN \times 1}, \\
  L_n^0 &= \begin{bmatrix} L_{nc}^0 \end{bmatrix}_{CN \times CN}, \\
  H_n^0 &= \begin{bmatrix} H_{nc}^0 \end{bmatrix}_{CN \times CN}, \\
  \eta^0 &= \begin{bmatrix} \eta_{n}^0 \end{bmatrix}_{CN \times 1}.
\end{align*}
\]

Equation (37) can be rewritten as follows:

\[
\begin{align*}
l(Y \mid \tilde{\eta}) & \approx \\
  & \approx NC_1 + \frac{1}{2} \sum_{n=1}^{N} ((L_n^0)^T \tilde{\eta} - (L_n^0)^T (\tilde{\eta}^0 - (H_n^0)^{-1} g_n^0))^T \\
  & \quad \times [(L_n^0)^T \tilde{\eta} - (L_n^0)^T (\tilde{\eta}^0 - (H_n^0)^{-1} g_n^0)].
\end{align*}
\]

Defining

\[
z^0 := (L^0)^T (\tilde{\eta}^0 - (H_0^0)^{-1} g_0^0) = (L^0)^T \tilde{\eta}^0 - (L^0)^T (H_0^0)^{-1} g_0^0,
\]

we obtain \( p(z^0 \mid \tilde{\eta}) \),

\[
z^0 \mid \tilde{\eta} \sim \mathcal{N}((L^0)^T \tilde{\eta}, I_{CN}),
\]

and the observation matrix \( Y \) has been turned into Gaussian conditionally independent pseudo-observations \( z^0 \), a likelihood which R-INLA can deal with.

\section*{Appendix C. Calculus for the Dirichlet Likelihood}

In this Appendix, we present all the calculus required for the particular case of the Dirichlet likelihood. For the sake of simplicity, we present those required just for one observation. We start with presenting the likelihood in terms of the linear predictor, we continue with the gradient, followed by the Hessian and finishing with the expected Hessian.

\subsection*{C.1. Likelihood}

The density function corresponding to Dirichlet distribution has been depicted in expression (1). Using the Dirichlet regression displayed in Equation (3), we know that \( \alpha_c = \exp(\eta_c) \), \( c = 1, \ldots, C \). Then the density function or the likelihood for just one observation can be expressed as

\[
p(y \mid \eta_1, \ldots, \eta_C) = \frac{1}{B(\exp(\eta_1), \ldots, \exp(\eta_C))} \prod_{c=1}^{C} \exp(\exp(\eta_c) - 1). \tag{41}
\]

Taking logarithms and using the definition of the B function, the next expression is obtained:

\[
\log p(y \mid \eta) = \log \left( \frac{\Gamma(\exp(\eta_1)) \cdots \Gamma(\exp(\eta_C))}{\Gamma(\exp(\eta_1) + \cdots + \exp(\eta_C))} \right) + \sum_{c=1}^{C} (\exp(\eta_c) - 1) \log(\gamma_c). \tag{42}
\]

\subsection*{C.2. Gradient}

Here, the gradient of the log-likelihood is calculated.

\[
\frac{\partial l}{\partial \eta_c} = \exp(\eta_c) \left[ \frac{\phi(\exp(\eta_c)) - \phi \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right] - \exp(\eta_c) \log(\gamma_c),
\]

where \( c = 1, \ldots, C \) and \( \phi \) is the digamma function.

\subsection*{C.3. Hessian}

The second derivatives are calculated for the log-likelihood. Let \( c \) and \( d \) two naturals such as \( 1 \leq c \leq C \) and \( 1 \leq d \leq C \), then

\[
\frac{\partial^2 l}{\partial^{2} \eta_c} = \exp(\eta_c) \left[ \phi(\exp(\eta_c)) - \phi \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right] + \exp(2\eta_c) \left[ \phi^1(\exp(\eta_c)) - \phi^1 \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right] - \exp(\eta_c) \log(\gamma_c),
\]

and

\[
\frac{\partial^2 l}{\partial \eta_c \partial \eta_d} = - \exp(\eta_c) \exp(\eta_d) \left[ \phi^1 \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right],
\]

where \( \phi \) is the digamma function and \( \phi^1 \) is the trigamma function.

\subsection*{C.4. Expected Hessian}

The expected second derivatives are calculated for the log-likelihood. Thus,

\[
E \left( \frac{\partial^2 l}{\partial \eta_c} \right) = \exp(2\eta_c) \left[ \phi^1(\exp(\eta_c)) - \phi^1 \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right]
\]

and

\[
E \left( \frac{\partial^2 l}{\partial \eta_c \partial \eta_d} \right) = - \exp(\eta_c) \exp(\eta_d) \left[ \phi^1 \left( \sum_{c=1}^{C} \exp(\eta_c) \right) \right],
\]

where \( \phi \) is the digamma function and \( \phi^1 \) is the trigamma function.
Supplementary Materials

R-package dirinla: R-package dirinla containing code to perform the Dirichlet regression using R-INLA described in the article. The package also contains the necessary code to reproduce the real and simulation examples presented in the article. It can be installed from https://github.com/inlabru-org/dirinla.

Simulations: the document test.html (also in https://jmartinez-minaya.github.io/supplementary/CODA/tests.html) contains a complete scenario of simulations by checking the performance of the dirinla R-package.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

The authors gratefully acknowledge the Ministerio de Educación y Ciencia (Spain) for financial support (jointly financed by the European Regional Development Fund) via Research Grant PID2019-106341 GB-100 (DC and AL-Q), the Basque Government through the BERC 2018-2021 program (MM), the Canadian Natural Sciences and Engineering Research Council and the Canadian Research Chairs program (DS).

ORCID

Joaquin Martinez-Minaya http://orcid.org/0000-0002-1016-8734
Finn Lindgren http://orcid.org/0000-0002-5833-2011
Antonio López-Quílez http://orcid.org/0000-0002-7022-9260
David Conesa https://orcid.org/0000-0002-5442-5691

References

Aitchison, J. (1981), “A New Approach to Null Correlations of Proportions,” Mathematical Geology, 13, 175–189. [805]
Aitchison, J. (1982), “The Statistical Analysis of Compositional Data,” Journal of the Royal Statistical Society, Series B, 44, 139–177. [805]
Aitchison, J. (1983), “Principal Component Analysis of Compositional Data,” Biometrics, 70, 57–65. [805]
Aitchison, J. (1984), “The Statistical Analysis of Geochemical Compositions,” Journal of the International Association for Mathematical Geology, 16, 531–564. [805]
Aitchison, J. (1986), The Statistical Analysis of Compositional Data, London: Chapman and Hall. [805]
Aitchison, J., and Egozcue, J. J. (2005), “Compositional Data Analysis: Where Are We and Where Should We Be Heading?” Mathematical Geology, 37, 829–850. [805]
Anderson, T. W. (1958), An Introduction to Multivariate Statistical Analysis (Vol. 2), New York: Wiley. [805]
Bachl, F. E., and Lindgren, F. (2018), inlabru: Spatial Inference using Integrated Nested Laplace Approximation, R package version 2.1.9. [820]
Bachl, F. E., Lindgren, F., Borchers, D. L., and Illian, J. B. (2019), “inlabru: an R Package for Bayesian Spatial Modelling from Ecological Survey Data,” Methods in Ecology and Evolution, 10, 760–766. [809,820]
Baker, S. G. (1994), “The Multinomial-Poisson Transformation,” Journal of the Royal Statistical Society, Series D, 43, 495–504. [808,819]
Barndorff-Nielsen, O., and Cox, D. (1989), Asymptotic Techniques for Use in Statistics, Boca Raton, FL: Chapman and Hall/CRC. [807]
Blangiardo, M., and Cameletti, M. (2015), Spatial and Spatio-Temporal Bayesian Models with R-INLA, Chichester: Wiley. [808]
Bonat, W. H. (2018), “Multiple Response Variables Regression Models in R: The mcglm Package,” Journal of Statistical Software, 84, 1–30. [805]
Buccianti, A., and Grunsky, E. (2014), “Compositional Data Analysis in Geochemistry: Are we Sure to See what Really Occurs during Natural Processes?” Journal of Geochemical Exploration, 141, 1–5. [805]
Cox, D. R., and Reid, N. (1987), “Parameter Orthogonality and Approximate Conditional Inference,” Journal of the Royal Statistical Society, Series B, 49, 1–18. [820]
Cribari-Neto, F., and Zeileis, A. (2010), “Beta Regression in R,” Journal of Statistical Software, 34, 1–24. [805]
Dennis, J. E., and Moré, J. J. (1977), “Quasi-Newton Methods, Motivation and Theory,” SIAM Review, 19, 46–89. [809]
Douma, J. C., and Weedon, J. T. (2019), “Analysing Continuous Proportions in Ecology and Evolution: A Practical Introduction to Beta and Dirichlet Regression,” Methods in Ecology and Evolution, 10, 1412–1430. [805]
Dumuid, D., Stanford, T. E., Martín-Fernández, J.-A., Pedišić, Ž., Maher, C. A., Lewis, L. K., Hron, K., Katzmarzyk, P. T., Chaput, J.-P., Fogelholm, M., Hu, G., Lambert, E. V., Maia, J., Sarmiento, O. L., Standage, M., Barreira, T. V., Broyles, S. T., Tudor-Locke, C., TrembLAY, M. S., and Olds, T. (2018), “Compositional Data Analysis for Physical Activity, Sedentary Time and Sleep Research,” Statistical Methods in Medical Research, 27, 3726–3738. [805]
Engle, M. A., and Rowan, E. L. (2014), “Geochemical Evolution of Produced Waters from Hydraulic Fracturing of the Marcellus Shale, Northern Appalachian Basin: A Multivariate Compositional Data Analysis Approach,” International Journal of Coal Geology, 126, 45–56. [805]
Fairclough, S. J., Dumuid, D., Mackintosh, K. A., Stone, G., Dagger, R., Stratton, G., Davies, I., and Boddy, L. M. (2018), “Adiposity, Fitness, Health-Related Quality of Life and the Realization of Time between Children’s School Day Activity Behaviours: A Compositional Data Analysis,” Preventive Medicine Reports, 11, 254–261. [805]
Gelman, A., Hwang, J., and Vehtari, A. (2014), “Understanding Predictive Information Criteria for Bayesian Models,” Statistics and Computing, 24, 997–1016. [810]
Gneiting, T., and Raftery, A. E. (2007), “Strictly Proper Scoring Rules, Prediction, and Estimation,” Journal of the American Statistical Association, 102, 359–378. [810]
Gómez-Rubio, V. (2020), Bayesian Inference with INLA, Boca Raton, FL: CRC Press. [808]
Hadfield, J. D. (2010), “MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package,” Journal of Statistical Software, 33, 1–22. [805]
Hijazi, R. H., and Jernigan, R. W. (2009), “Modelling Compositional Data using Dirichlet Regression Models,” Journal of Applied Probability & Statistics, 4, 77–91. [805]
Klein, N., Kneib, T., Klasen, S., and Lang, S. (2015), “Bayesian Structured Additive Distributional Regression for Multivariate Responses,” Journal of the Royal Statistical Society, Series C, 64, 569–591. [805]
Kobal, M., Kastelec, D., and Eler, K. (2017), “Temporal Changes of Forest Species Composition Studied by Compositional Data Approach,” iForest-Biogeosciences and Forestry, 10, 729–738. [805]
Krainski, E. T., Gómez-Rubio, V., Bakka, H., Lenzi, A., Castro-Camilo, D., Simpson, D., Lindgren, F., and Rue, H. (2018), “Advanced Spatial Modeling with Stochastic Partial Differential Equations using R and INLA, Boca Raton, FL: CRC Press. [808]
Lee, Y., Nelder, J. A., and Pawitan, Y. (2018), Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood (Vol. 153), Boca Raton, FL: CRC Press. [820]
Maier, M. J. (2014), DirichletReg: Dirichlet Regression for Compositional Data in R. [805,806,810]
Martínez-Minaya, J., Conesa, D., Fortin, M.-J., Alonso-Blanco, C., Picó, F. X., and Marcé, A. (2019), “A Hierarchical Bayesian Beta Regression Approach to Study the Effects of Geographical Genetic Structure and Spatial Autocorrelation on Species Distribution Range Shifts,” Molecular Ecology Resources, 19, 929–943. [818]
Masarotto, G., and Varin, C. (2017), “Gaussian Copula Regression in R,” Journal of Statistical Software, 77, 1–26. [805]
Monyai, S., Lesaoana, M., Darikwa, T., and Nyamugure, P. (2016), “Application of Multinomial Logistic Regression to Educational Factors of the 2009 General Household Survey in South Africa,” Journal of Applied Statistics, 43, 128–139. [805]
Moraga, P. (2019), Geospatial Health Data: Modeling and Visualization with R-INLA and Shiny, Boca Raton, FL: CRC Press. [808]
Nocedal, J., and Wright, S. (2006), Numerical Optimization, New York: Springer. [809]
Nowosad, J., and Stepinski, T. (2018), “Spatial Association between Region-alizations Using the Information-Theoretical V-Measure,” International Journal of Geographical Information Science, 32, 2386–2401. [805]

Odeyemi, Y., Pollind, M., Peeler, R., Nozawa, K., Vesely, D., Page, A., Rakoski, C., and El-Askary, H. (2019), “Review of Climate Research and Funding 1993–2017: A Multinomial Logistic Regression Approach,” Journal of Environmental Informatics Letters, 1, 94–101. [805]

Piccini, C., Marchetti, A., Rivieccio, R., and Napoli, R. (2019), “Multinomial Logistic Regression with Soil Diagnostic Features and Land Surface Parameters for Soil Mapping of Latium (Central Italy),” Geoderma, 352, 385–394. [805]

Plummer, M. (2016), rjags: Bayesian Graphical Models using MCMC, R package version 4-6. [805,812]

R Core Team (2021), R: A Language and Environment for Statistical Computing, Vienna, Austria: R Foundation for Statistical Computing. [805,809]

Rue, H., and Held, L. (2005), Gaussian Markov Random Fields: Theory and Applications, London: Chapman & Hall. [807]

Rue, H., Martino, S., and Chopin, N. (2009), “Approximate Bayesian Inference for Latent Gaussian Models by using Integrated Nested Laplace Approximations,” Journal of the Royal Statistical Society, Series B, 71, 319–392. [808,808]

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. (2017), “Bayesian Computing with INLA: A Review,” Annual Review of Statistics and Its Application, 4, 395–421. [806,807]

Sennhenn-Reulen, H. (2018), Bayesian Regression for a Dirichlet Distributed Response using Stan, arXiv preprint arXiv:1808.06399. [805]

Shi, P., Zhang, A., and Li, H. (2016), “Regression Analysis for Microbiome Compositional Data,” The Annals of Applied Statistics, 10, 1019–1040. [805]

Simpson, D., Illian, J. B., Lindgren, E., Sørbye, S. H., and Rue, H. (2016), “Going off Grid: Computationally Efficient Inference for log-Gaussian Cox Processes,” Biometrika, 103, 49–70. [819]

Simpson, D., Rue, H., Riebler, A., Martins, T. G., and Sørbye, S. H. (2017), “Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors,” Statistical Science, 32, 1–28. [816]

Smithson, M., and Verkuilen, J. (2006), "A Better Lemon Squeezer: Maximum-likelihood Regression with Beta-Distributed Dependent Variables," Psychological Methods, 11, 54–71. [806]

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," Journal of the Royal Statistical Society, Series B, 64, 583–639. [810]

Tsilimigras, M. C., and Fodor, A. A. (2016), "Compositional Data Analysis of the Microbiome: Fundamentals, Tools, and Challenges," Annals of Epidemiology, 26, 330–335. [805]

Van der Merwe, S. (2019), "A Method for Bayesian Regression Modelling of Composition Data," South African Statistical Journal, 53, 55–64. [805]

Wang, X., Ryan, Y. Y., and Faraway, J. J. (2018), Bayesian Regression Modeling with INLA, Boca Raton, FL: Chapman and Hall/CRC. [808]

Washburne, A. D., Silverman, J. D., Leff, J. W., Bennett, D. J., Darcy, J. L., Mukherjee, S., Fierer, N., and David, L. A. (2017), "Phylogenetic Factorization of Compositional Data Yields Lineage-level Associations in Microbiome Datasets," PeerJ, 5, e2969. [805]

Zuur, A. F., Ieno, E. N., and Saveliev, A. A. (2017), Beginner’s Guide to Spatial, Temporal, and Spatial-Temporal Ecological Data Analysis with R-INLA, Newburgh: Highland Statistics Ltd. [808]