Generalized Matrix Factorization

Łukasz Kidziński   LUKASZ.KIDZINSKI@STANFORD.EDU
Department of Bioengineering
Stanford University
Stanford, CA 94305, USA

Francis K.C. Hui   FRANCIS.HUI@ANU.EDU.AU
Research School of Finance, Actuarial Studies and Statistics
Australian National University
Acton, ACT 2601, Australia

David I. Warton   DAVID.WARTON@UNSW.EDU.AU
School of Mathematics and Statistics
and Evolution & Ecology Research Centre
The University of New South Wales
Sydney, NSW 2052, Australia

Trevor J. Hastie   HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Abstract

Unmeasured or latent variables are often the cause of correlations between multivariate measurements and are studied in a variety of fields such as psychology, ecology, and medicine. For Gaussian measurements, there are classical tools such as factor analysis or principal component analysis with a well-established theory and fast algorithms. Generalized Linear Latent Variable models (GLLVM) generalize such factor models to non-Gaussian responses. However, current algorithms for estimating model parameters in GLLVMs require intensive computation and do not scale to large datasets with thousands of observational units or responses. In this article, we propose a new approach for fitting GLLVMs to such high-volume, high-dimensional datasets. We approximate the likelihood using penalized quasi-likelihood and use a Newton method and Fisher scoring to learn the model parameters. Our method greatly reduces the computation time and can be easily parallelized, enabling factorization at unprecedented scale using commodity hardware. We illustrate application of our method on a dataset of 48,000 observational units with over 2,000 observed species in each unit, finding that most of the variability can be explained with a handful of factors.

Keywords: Matrix Factorization, Generalized Linear Latent Variable Models, Generalized Linear Models, Generalized Linear Mixed-effect Models

1. Introduction

Latent factors underlying multivariate observations are of great interest in many applied disciplines. For example, in psychology or sociology researchers measure multiple correlated

test items to quantify certain constructs. They assume that responses can be described in terms of a small set of latent variables and that these latent variables can be interpreted as psychological or sociological traits (Bartholomew et al., 2011; Skrondal and Rabe-Hesketh, 2004; Spearman, 1904). In genetics, researchers measure gene expression in patients and expect that they correlate with intrinsic patient's features, often not measurable directly (Stegle et al., 2012). In ecology, researchers observe sets of species in independent observational units (e.g., sites) and assume the existence of latent features associated with the abundance of species (e.g., representing a low-dimensional community composition space Warton et al., 2015, 2016; Ovaskainen et al., 2017).

Data from such experiments or observational studies can typically be expressed as a matrix of responses $Y = [y_{ij}]$, where rows $i \in \{1, 2, ..., n\}$ correspond to observational units (locations), subjects, etc. and columns $j \in \{1, 2, ..., m\}$ correspond to different responses such as species, genes, etc. Linear latent factors are then incorporated as means of obtaining a low-rank approximation to the covariance structure of the rows in Y. Specifically, we assume that conditional on a set of latent variable u_i, responses y_{ij} are independent observations and that $u_i \in \mathbb{R}^p$ with $p \ll \min(m, n)$ (See Section 2 for more detailed notation and the model).

If all the responses are Gaussian in distribution, then we can find linear latent patterns with classical tools such as principal component analysis, singular value decomposition, or factor analytic models and solutions can be computed very quickly, even at scale. Many algorithms for fitting such large scale models have been suggested in the last two decades (Zou et al., 2006; Witten et al., 2009; Halko et al., 2011; Hirose and Yamamoto, 2015; Hirose and Imada, 2018). These algorithms are fast and scale well, since in this case there are known closed-form solutions leveraging the Gaussian structure. However, the theory and computation for Gaussian responses does not generalize easily to non-Gaussian cases.

Generalized Linear Latent Variable Models (GLLVMs, Moustaki and Knott, 2000; Niku et al., 2017) are a class of models which generalizes factor analysis to non-Gaussian responses. Specifically, they assume that responses follow distributions in the exponential family and that the mean for each response varies as a function of observed covariates and the aforementioned set of latent features via a known link function.

In GLLVMs, model parameters are usually estimated using Bayesian or maximum likelihood methods. Among a wide variety of Bayesian tools, practitioners use general purpose modeling software such as `Stan` (Carpenter et al., 2017), integrated nested Laplace approximations (INLA, Rue et al., 2009), as well as Bayesian Markov Chain Monte Carlo estimation with more specific software designed for GLLVMs (Blanchet et al., 2018; Hui, 2020).

In this work we focus on likelihood-based methods. First, we observe that, given the latent variable u_i for an observational unit i, all the responses y_{ij} are conditionally independent. Thus, the likelihood can be expressed as a product of m individual conditional likelihoods, after which we marginalize out the latent variables u_i. The key problem in this approach is that the integral over u_i does not have a closed form and must be estimated or approximated by some means. To solve this problem, a number of methods have been proposed, including the Laplace method or some variation thereof (Huber et al., 2004; Bianconcini and Cagnone, 2012; Niku et al., 2017; Robin, 2019), numerical integration methods using adaptive quadrature (Rabe-Hesketh et al., 2002), and variational approximations (Hui et al., 2017b; Niku et al., 2019a).

While these approaches lead to very accurate solutions, they are computationally expensive for high-volume or high-dimensional problems and are difficult to parallelize, making them infeasible for solving large-scale problems. An alternative is to look for approximate solutions. Pichler and Hartig (2020) proposed to model the multivariate joint likelihood using a deep neural network, substantially improving computational performance thanks to the use of Graphical Processing Units (GPUs). However, their method has currently only been developed for binary responses. Huber et al. (2004) observed that if the latent scores are treated as fixed parameters, then estimates of them obtained through the Laplace method coincide with variables derived using Penalized Quasi-Likelihood estimation, known from Generalized Linear Mixed Models (GLMMs, Breslow and Clayton, 1993; Green, 1987; Nie, 2007).

In this article, we propose an approximation to the GLLVM problem, along with two fast algorithms for solving it. We present its application on a large ecological dataset on coexistence of species across 48,000 observational units with over 2000 responses in each unit. We build on ideas from GLMMs for estimating parameters of GLLVMs. In particular, we develop an alternating algorithm which leverages the idea that a solution to a GLLVM can be approximated using penalized quasi-likelihood estimation. We propose two algorithms: a direct Newton method with a simplified Hessian and an Alternating Iteratively Reweighted Least Squares (AIRWLS) algorithm. As the name suggests, the AIRWLS algorithm uses classical Iteratively Reweighted Least Squares iterations, by applying them alternately to rows and columns of Y. Computationally, we demonstrate that our algorithms are orders of magnitude faster than existing methods. Moreover, the AIRWLS algorithm can be easily distributed across many machines enabling large-scale analysis. Finally, our approach also allows for some of the y_{ij} to be missing at random and produces predictions for these as part of the estimation procedure.

2. Model

We now provide a more precise mathematical formulation of the GLLVM. For observational unit $1 \leq i \leq n$ and response $1 \leq j \leq m$, we assume

$$y_{ij}|\mu_{ij} \sim \mathcal{F}(\mu_{ij}, \phi_j)$$
$$g(\mu_{ij}) = \eta_{ij} = \beta_{0j} + x_i'\beta_j + u_i'\lambda_j, \tag{1}$$

where β_{0j} is an intercept for each response, $x_i \in \mathbb{R}^d$ are observed covariates for the i-th observational unit with $\beta_j \in \mathbb{R}^d$ the corresponding response-specific regression coefficients, $u_i \in \mathbb{R}^p$ are the latent variables (also referred to as factor scores) for observational unit i with $\lambda_j \in \mathbb{R}^p$ the corresponding response-specific factor loadings, $\mathcal{F}(\mu_{ij}, \phi_j)$ denotes a distribution from the exponential family with mean μ_{ij} and response-specific dispersion parameter ϕ_j, and g is a known link function, e.g., the logit link for binary responses.

We assume that: (A1) $u_i \sim \mathcal{N}(0, I_p)$ where I_p is $p \times p$ identity matrix, (A2) $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_m]$ is a $p \times m$ upper triangular with positive elements on the diagonal, and (A3) all observational units are independent and, conditioned on u_i, all m in site i are also independent. In other words, conditional on u_i the model as defined by Model (1) specifies a GLM (McCullagh and Nelder, 1983) for each response j with mean μ_{ij} and dispersion parameter ϕ_j. Let Ψ denote all the model parameters, i.e. $\beta_{0,i}$, ϕ_j, β_j, λ_j for

3

all i and j. Assumption (A1) is typically made in the context of GLLVMs to ensure that the model is not location and scale invariant. Assumption (A2) is also set for parameter identifiability reasons, since without this assumption we could rotate the vectors u_i and λ_j without changing the value of η_{ij} in (1) (Huber et al., 2004). Moreover, the independence of observational units as in Assumption (A3) is common in many formulations of GLLVMs (e.g., Hirose and Yamamoto, 2015; Hui et al., 2017b; Niku et al., 2017).

We use matrix notation whenever it is convenient and suitable. In particular, apart from Λ already defined, we use $B = [\beta_1, \beta_2, ..., \beta_m]$, $X' = [x_1, u_2, ..., x_n]$ and $U' = [u_1, u_2, ..., u_n]$ to define matrices corresponding to the regression coefficients, observed covariates, and factor loadings, respectively. We denote the matrix of responses as $Y = [y_{ij}]$ and the matrix of means $M = [\mu_{ij}]$. Finally, $f(y_{ij}|u_i, \Psi)$ is the density function of y_{ij} given u_i and Ψ, corresponding to the distribution $\mathcal{F}(\mu_{ij}, \Psi)$ as defined (1). We define $f(y_i|u_i\Psi)$ as the multivariate density function of the vector $y_i = (y_{i1}, ..., y_{im})'$.

The usual factor analytic model can be expressed in terms of Model (1) if we take $\mathcal{F}$ to be a Gaussian distribution and set x_i to only involve an intercept term. The methods introduced in this paper for solving Model (1) in the general form rely on matrix factorization concepts and, in turn, we refer to them as *generalized matrix factorization*.

3. Likelihood estimation

By the independence of components of y_i conditional on u_i we have $f(y_i|u_i, \Psi) = \prod_j f(y_{ij}|u_i, \Psi)$. We integrate out the random effects and write the marginal log-likelihood in (1) as:

$$
\begin{aligned}
\ell(\Psi) &= \sum_{i=1}^{n} \ell_i(\Psi) \\
&= \sum_{i=1}^{n} \log \left(\int \prod_{j=1}^{m} f(y_{ij}|u_i, \Psi) f(u_i) du_i \right),
\end{aligned}
\tag{2}
$$

where $f(u_i) \sim \mathcal{N}(0, I_p)$ and ℓ_i is the log-likelihood of the i-th row.

Except for the special case where all the responses are Gaussian and the identity link function is used, the integration in (2) can not be expressed in closed form, and has thus led to an extensive amount of research into overcoming this computational burden. For example, it can be directly computed using numerical integration methods, including Gauss-Hermite quadrature, adaptive quadrature, or Monte-Carlo integration. In brief, quadrature approaches aim at approximating the integral as a $(2R - 1)$ polynomial by evaluating the function at R quadrature points. Adaptive quadrature shifts and scales locations of sampling points at each step to minimize the error (Rabe-Hesketh et al., 2002). Monte-Carlo methods sample the function within its domain and average the values. Importance sampling, a more sample-efficient method, was introduced for linear mixed models by Kuk (1999) and Skaug (2002) and can also be used here. While these methods can yield solutions close to exact, they are very expensive computationally and scale poorly with the number of latent variables.

Alternatively, we can approximate the value of the integral (2) using a variational approximation or Taylor expansion (better known as the Laplace method). In the variational

approximation, recently introduced in the context of GLLVMs by Hui et al. (2017b); Niku et al. (2019a), Jensen's inequality is applied to obtain a lower bound for the log-likelihood that is computationally manageable. Maximizing this lower bound gives an approximate solution for (2). While the method is more efficient, it is still computationally expensive and difficult to apply on large datasets, especially because the number of parameters to estimate is increased considerably due to the introduction of the parameters characterizing the variational distributions.

The Laplace method is derived by taking a Taylor expansion of (2) around its mode. While a standard Laplace approximation is based on a second order expansion, and indeed we can view the Laplace method as a special case of adaptive quadrature with $R = 1$ quadrature point, higher-order approximations have also been analyzed (Bianconcini and Cagnone, 2012; Bianconcini et al., 2014). Although estimates from Laplace approximations may exhibit a non-negligible finite sample bias relative to aforementioned methods such as Monte-Carlo methods, they are consistent to order $O(m^{-1})$ (Kass and Raftery, 1995; Vonesh, 1996). Despite solving a simplified problem, the Laplace approximation still tends to be slow for large matrices. To further optimize computational efficiency, one can drop slowly varying terms, leading to class of the penalized quasi-likelihood methods (Breslow and Clayton, 1993; McGilchrist, 1994; Vonesh et al., 2002) which motivate the approach presented in this paper and which we describe in detail in Section 4.

All of the aforementioned methods are computationally prohibitive for problems with thousands of rows and/or columns and may require days, weeks, or more time to converge (see Pichler and Hartig 2020 and an example empirical study in Section 6.2 and Figure 5). In this article, we leverage the structure of (1) in two ways: (i) we show that by approximating the problem with a penalized log-likelihood we can efficiently estimate all the required gradients and Hessians necessary for estimation of model parameters; (ii) we show that thanks to this approximation, the problem can be decomposed to a set of smaller and relatively simple estimation problems based on individual rows and individual columns of the response matrix, and thus treated in an alternating fashion. Therefore, the algorithm can be parallelized enabling further performance optimization.

4. Penalized log-likelihood

In the pursuit of an efficient algorithm for estimating the model parameters in (1), we borrow ideas from Huber et al. (2004). They showed that, in the setting of GLLVMs, the maximum likelihood estimators of the latent variables derived from applying the Laplace method to the marginal likelihood in (2) are equivalent to those based on maximizing the Penalized Quasi-Likelihood (PQL) approach of Green (1987). On the other hand, the estimates of β and u_i are not equivalent and are indeed more biased for the PQL approach relative to the Laplace method. However, it has nevertheless been proven that they remain asymptotically consistent (Nie, 2007; Hui et al., 2017a) provided the size of each observational unit, which in this case corresponds to the number of responses, grows with the number of units. This insight is particularly promising in the context of our work since we focus on large-scale problems. In this section, we build on the derivation from PQL and Laplace methods proposed by Breslow and Clayton (1993) and Huber et al. (2004).

4.1 Derivation

We start by computing the log-likelihood $\ell_i(\Psi)$ for the i-th row. For ease of notation, we assume distributions in the exponential family with some known cumulant function $b(\cdot)$ and that the canonical link function is used. The developments can be extended to the case of a non-canonical link function, at the expense of added tedious algebra. We have

$$\exp(\ell_i(\Psi)) = \int \prod_j^m h(y_{ij}, \phi) \exp\left(\frac{y_{ij}\eta_{ij} - b(\eta_{ij})}{\phi_j}\right) \exp\left(-\frac{u_i'u_i}{2}\right) du_i$$

$$\propto \int \exp\left(\sum_{j=1}^m \frac{y_{ij}\eta_{ij} - b(\eta_{ij})}{\phi_j} - \frac{u_i'u_i}{2}\right) du_i, \tag{3}$$

where $g(\mu_{ij}) = \eta_{ij}$ as defined in Model 1.

We express the integral (3) in the form $\int \exp(-\kappa(u_i)) du_i$ in order to apply the Laplace method. Let κ' and κ'' denote first- and second-order partial derivatives of κ with respect to u_i, respectively. Then the Laplace approximation yields

$$\ell_i(\Psi) \approx -\frac{1}{2}\log|\kappa''(\tilde{u}_i)| - \kappa(\tilde{u}_i),$$

where $|\kappa|$ is the determinant of κ and $\tilde{u}_i$ is the solution to $\kappa'(u) = 0$, i.e. the minimum of $\kappa(u)$. Note that

$$\kappa'(u_i) = -\sum_{j=1}^m \frac{\lambda_j(y_{ij} - \mu_{ij})}{\phi_j} + u_i \tag{4}$$

and

$$\kappa''(u_i) = \sum_{j=1}^m \frac{\lambda_j\lambda_j'v(\mu_{ij})}{\phi_j} + I_p, \tag{5}$$

where $v(\mu_{ij}) = 1/g'(\mu_{ij})$ is the variance function associated with the exponential family of distribution when the canonical link function is used e.g., for the Bernoulli distribution with the canonical logit link, we have $v(\mu) = \mu(1-\mu)$.

From equation (5), κ'' can be rewritten as $\Lambda W\Lambda' + I_p$ where $\Lambda = [\lambda_1, ..., \lambda_m]$ is the $p \times m$ matrix of factor loadings and W is a $m \times m$ diagonal matrix with elements $w_j = v(\mu_{ij})/\phi_j$ for $j \in \{1, 2, ..., m\}$. Note that in the case of GLLVMs, the elements $\{w_j\}_{j=1}^m$ correspond precisely to iterative weights coming from a GLM (McCullagh and Nelder, 1983, Chapter 2.5).

For the i-th observational unit, we can now write

$$\ell_i(\Psi) \approx -\frac{1}{2}\log|\Lambda W\Lambda' + I_p| + \frac{1}{\phi_j}\sum_{j=1}^m (y_{ij}\hat{\eta}_{ij} - b(\hat{\eta}_{ij})) - \frac{1}{2}\tilde{u}_i'\tilde{u}_i, \tag{6}$$

where $\tilde{\eta}_{ij} = \beta_{0j} + x_i'\beta_j + \tilde{u}_i'\lambda_j$.

Breslow and Clayton (1993) suggested that since W varies slowly as a function of the model parameters and for fixed Λ, the first term in (6) could be ignored. In the setting of GLLVMs, Λ is not fixed, but if we consider the normalized log-likelihood $\frac{1}{mn}\sum_i \ell_i$ then we observe that for fixed m, this first term is asymptotically negligible as n gets large. Moreover if both m and n are growing, then the first term in (6) is asymptotically dominated by the second term; see also (Demidenko, 2013; Hui et al., 2017a). Hence for large sample sizes, we chose also to (conveniently) ignore this term in our approximation so as to facilitate computation.

We thus conclude that for approximate maximum likelihood estimation, we can use the following approximation

$$\log \int \prod_{j=1}^{m} f(y_{ij}|u_i, \Psi)f(u_i)du_i \approx C + \sum_{j=1}^{m}(y_{ij}\tilde{\eta}_{ij} - b(\tilde{\eta}_{ij})) - \frac{1}{2}\tilde{u}_i'\tilde{u}_i, \tag{7}$$

where $\tilde{u}_i$ maximizes (4) and C is some constant as a function of the model parameters. Note that (7) has the form of a penalized quasi-likelihood. This result implies that, provided m is sufficiently large, instead of integrating the left-hand side of (3), we can obtain a good approximation by minimizing

$$L(\Psi) = -\sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij}\tilde{\eta}_{ij} - b(\tilde{\eta}_{ij})) + \frac{1}{2}\sum_{i=1}^{n} u_i'u_i. \tag{8}$$

Importantly, we can solve (8) very efficiently using a Newton algorithm as we demonstrate in Section 4.2. First, we discuss an approach inspired by iteratively reweighted least squares, where we alternate between the estimation of U and Λ (Section 4.2.1). In Section 4.2.2 we introduce heuristics for estimating Hessians which allows for a direct Newton algorithm on all the parameters, and substantially reduces the computations in each iteration of the Newton algorithm.

4.2 Newton algorithms

One approach to optimizing (8) is via alternating minimization (Robin, 2019) with respect to U and Λ. For ease of notation assume $\phi_j = 1$ is known e.g., in the case of Poisson and Bernoulli distributed responses. In our proposed iterative algorithms, estimates of dispersion parameters can be updated after each iteration of the Newton algorithm, if required. Specifically for estimating ϕ_j we can use a method of moments or maximum likelihood after each iteration of our proposed algorithms (Nelder and Wedderburn, 1972).

The gradient of L in (8) with respect to the latent variables u_i is given by

$$\begin{aligned} \frac{\partial L}{\partial u_i} &= -\sum_{j=1}^{m} \frac{(y_{ij} - \mu_{ij})\mu'_{ij}(\eta_{ij})}{v(\mu_{ij})}\lambda_j + u_i \\ &= -\sum_{j=1}^{m}(y_{ij} - \mu_{ij})\lambda_j + u_i \end{aligned}$$

where $v(\mu_{ij})$ is the variance function of $\mathcal{F}$ and the second line follows from the assumption of a canonical link. In the case of a non-canonical link, a similar expression can be written

involving additional weights. Likewise

$$
\begin{aligned}
\frac{\partial^2 L}{\partial u_i \partial u_i'} &= \sum_{j=1}^{m} \frac{\mu_{ij}'(\eta_{ij})^2}{v(\mu_{ij})} \lambda_j \lambda_j' + I_p \\
&= \sum_{j=1}^{m} v(\mu_{ij}) \lambda_j \lambda_j' + I_p,
\end{aligned}
\tag{9}
$$

and noting that

$$
\frac{\partial^2 L}{\partial u_i \partial u_k'} = 0 \text{ for } k \neq i.
$$

Hence the Hessian has the block diagonal form $d^2 L = \operatorname{diag}(H_1, H_2, \ldots, H_n)$ with $H_i = \partial u_i \partial u_i' / \partial^2 L$.

Similarly, we can also straightforwardly calculate gradients and Hessians of L in (8) with respect to λ_j and β_j,

$$
\begin{aligned}
\frac{\partial L}{\partial \lambda_j} &= -\sum_{i=1}^{n} (y_{ij} - \mu_{ij}) u_i, \\
\frac{\partial L}{\partial \beta_j} &= -\sum_{i=1}^{n} (y_{ij} - \mu_{ij}) x_i
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial^2 L}{\partial \lambda_j \partial \lambda_j'} &= \sum_{i=1}^{n} v(\mu_{ij}) u_i u_i' \\
\frac{\partial^2 L}{\partial \beta_j \partial \beta_j'} &= \sum_{i=1}^{n} v(\mu_{ij}) x_i x_i' \\
\frac{\partial^2 L}{\partial \beta_j \partial \lambda_j'} &= \sum_{i=1}^{n} v(\mu_{ij}) x_i u_i'.
\end{aligned}
$$

Furthermore,

$$
\frac{\partial^2 L}{\partial \lambda_i \partial \lambda_k'} = \frac{\partial^2 L}{\partial \beta_i \partial \beta_k'} = \frac{\partial^2 L}{\partial \beta_i \partial \lambda_k'} = 0 \text{ for } k \neq i.
$$

Note that for fixed u_i, we can view $(x_i', u_i')'$ as an enlarged covariate vector, with response-specific parameters $(\beta_j', \lambda_j')'$. In turn, the update step for each parameter $\theta \in \{u_i, \lambda_j, \beta_j\}$ takes the form

$$
\theta_{t+1} = \theta_t + s[-d^2 L(\theta_t)]^{-1} \nabla L(\theta_t),
\tag{10}
$$

where θ_t is the estimator of θ in t-th iteration, ∇L is the gradient of L with respect to θ, $d^2 L(\theta)$ is the corresponding Hessian of L at θ, and $s > 0$ is an arbitrary step size.

We now proceed to discuss two iterative algorithms for computing the update step in (10) efficiently. The first approach uses Fisher scoring and gives an exact update, leveraging the fact that $d^2 L(\theta)$ can be approximated by the Fisher information matrix (plus an identity

matrix) and this substitution is exact in the case where a canonical link is used. The second approach uses only the diagonal of the Hessian $d^2 L(\theta)$, which can be computed very quickly. In both approaches, after each update, we rotate the matrices U and Λ so as to satisfy the identifiability assumptions in (A1) and (A2).

4.2.1 FISHER SCORING AND ALTERNATING ITERATIVELY REWEIGHTED LEAST SQUARES

For optimizing (8) we leverage the fact that when we use the canonical link, Hessians as defined in Section 4.2 are equal to the Fisher information (Nelder and Wedderburn, 1972) of the likelihood (8). We showed in Section 4.1 that the approximation to the negative log-likelihood takes the form

$$\ell(\Psi) = \sum_{i=1}^{n} \ell_i(\Psi) \approx -\sum_{i=1}^{n} \left(\frac{1}{2} u_i' u_i + \sum_{j=1}^{m} (y_{ij} \eta_{ij} - b(\eta_{ij})) \right)$$

$$= -\frac{1}{2} \sum_{i=1}^{n} u_i' u_i - \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{ij} \eta_{ij} - b(\eta_{ij})). \tag{11}$$

For $(\beta_j, \lambda_j)_{j=1}^{m}$ known, we can obtain each of the u_i, $i = 1, \ldots, n$ in (11) by solving n separate penalized GLMs (Green, 1987; Breslow and Clayton, 1993), where in each the m responses are treated as the "observations". Conversely, with $(u_i)_{i=1}^{n}$ known, we can obtain (β_j, λ_j), $j = 1, \ldots, m$ by solving m separate GLMs in (11) (not penalized, and with predictors (x_i', u_i')).

Maximum likelihood for a single GLM is typically performed using the Newton algorithm implemented via iteratively reweighted least squares (IRWLS), or penalized least squares when there is a penalty. Hence minimizing (11) can be achieved by alternating and parallel IRWLS algorithms applied to the rows and columns of the response matrix.

We illustrate the derivation of the t-th IRWLS update step for u_i. Let $u_i^{(t)}$ be the estimate of u_i in the t-th iteration, $I(u_i^{(t)})$ be the (penalized) Fisher information matrix, and W_t be the iterative weight m-vector in the t-th iteration as defined in (6). Let $M_i^{(t)} = [\mu_{i,\cdot}]'$ be the i-th row of the matrix of means estimated in the t-th iteration, and $Y_i = [y_{i,\cdot}]'$ be the ith row of the response matrix (both written in column-vector form). We first derive the update step (10) with step size $s = 1$.

$$u_i^{(t+1)} = u_i^{(t)} + [I(u_i^{(t)})]^{-1} \nabla L(u_i^{(t)}),$$
$$= u_i^{(t)} + [\Lambda W_t \Lambda' + I_p]^{-1} [\Lambda(Y_i - M_i^{(t)}) - u_i^{(t)}],$$
$$= [\Lambda W_t \Lambda' + I_p]^{-1} \Lambda W_t [\Lambda' u_i^{(t)} + W_t^{-1}(Y_i - M_i^{(t)})]$$
$$= [\Lambda W_t \Lambda' + I_p]^{-1} \Lambda W_t Z_t,$$

where

$$Z_t = \Lambda' u_i^{(t)} + W_t^{-1}(Y_i - M_i^{(t)})$$

is a working response. Hence we obtain $u_i^{(t+1)}$ by a ridge regression of Z_t on Λ' with weights W_t. When the step size $s \neq 1$, our update is instead $u_i^{(t)} + s(u_i^{(t+1)} - u_i^{(t)})$. Note that

although the (fixed) $(\beta_j)_{j=1}^m$ do not appear explicitly in these equations, the jth element of $M_i^{(t)}$ includes $x_i'\beta_j$ as part of its linear predictor—an *offset* in GLM parlance.

We use a similar procedure for finding λ_j and β_j when the u_i are known. The problem of finding λ_j and β_j in

$$g(\mu_{\cdot j}) = X\beta_j + U\lambda_j,$$

where $U' = [u_1, ..., u_n]$, can be rewritten as

$$g(\mu_{\cdot j}) = (X, U)\gamma,$$

where $(\cdot, \cdot)$ stands for horizontal concatenation of matrices and $\gamma' = (\beta_j', \lambda_j')$. Again we can solve it using IRWLS, this time without the penalty term, and with (X, U) in place of Λ. At the end of each iteration we rotate U and Λ to fulfill the identifiability assumptions (A1)-(A2). For the full summary of our alternating two-step procedure, we refer to Algorithm 1.

Algorithm 1: ALTERNATING ITERATIVELY REWEIGHTED LEAST SQUARES

1. Initialize U, B, Λ randomly, where $B = [\beta_1, \beta_2, ..., \beta_m]$.

2. Repeat until the convergence condition:

    (a) Perform one step of IRWLS to regress rows of Y on Λ. Store regression coefficients as U.

    (b) Perform one step of IRWLS to regress columns of Y on (X, U). Store regression parameters as $(B', \Lambda')'$.

    (c) Transform data to comply with assumptions (A1)-(A2):

        i. Find a rotation Θ such that $\text{Cov}(U\Theta) = I_p$, using, for example, principal component analysis,

        ii. Compute $U_0 = U\Theta$ and $\Lambda_0 = \Theta^{-1}\Lambda$,

        iii. Find a QR decomposition of $\Lambda_0 = QR$,

        iv. Return R and $U_0 Q$ as new estimates of Λ and U respectively.

For the convergence condition in Algorithm 1, in our implementation we take the change in log-likelihood relative to the new log-likelihood, i.e. we stop when $|L_{k-1} - L_k|/|L_k| < \varepsilon$, where L_k is the log-likelihood in k-th iteration and ε is a sufficiently small value.

Note that, for each $i \in \{1, ..., n\}$ in (11) the optimization problem can be decoupled and solved independently, allowing for parallelization of computation. Similarly for each $j \in \{1, ..., m\}$ in (11) optimization problems can be decoupled.

4.2.2 QUASI-NEWTON WITH DIAGONAL HESSIANS

Another approach is to derive an updated step directly from (10) by computing and inverting the Hessian explicitly. For each of the parameters u, λ, and β, Hessians are block diagonal and computing a Newton step requires inverting all blocks, which would be computationally expensive for large n, since the Hessians are $np \times np$ or $mp \times mp$ matrices. To speed up

computations, we propose a quasi-Newton method where we only use the diagonals of blocks in Hessians. These diagonals and their inverses in (10) can be computed quickly at the expense of slowing down convergence in terms of the number of steps. Our empirical study shows that this approximation reduces computation time, despite increasing the number of steps.

Note that the diagonal elements of (9) can be computed by taking

$$
\begin{aligned}
\operatorname{diag}\left(\frac{\partial^2 L}{\partial u_i \partial u_i'}\right) &= \operatorname{diag}\left(\sum_{j=1}^m v(\mu_{ij})\lambda_j\lambda_j' + I_p\right) \\
&= (\Lambda \circ \Lambda)v(\mu_{i,\cdot})' + \mathbb{1}_p,
\end{aligned}
\tag{12}
$$

where $\circ$ denotes the element-wise multiplication (Hadamard product), $\operatorname{diag}(\cdot)$ denotes the diagonal of the given matrix, $v(\mu_{i,\cdot})$ is the i-th row of variances, and $\mathbb{1}_p = [1, 1, ..., 1]'$. The diagonal elements of Hessians of λ_j and β_j are derived analogously. Therefore we obtain

$$
\begin{aligned}
\operatorname{diag}\left(\frac{\partial^2 L}{\partial \beta_j \partial \beta_j'}\right) &= (X' \circ X')v(\mu_{\cdot,j}), \\
\operatorname{diag}\left(\frac{\partial^2 L}{\partial \lambda_j \partial \lambda_i'}\right) &= (U' \circ U')v(\mu_{\cdot,j}).
\end{aligned}
\tag{13}
$$

The full algorithm follows the steps Algorithm 1 except for steps 2(a) and 2(b) where we replace the AIRWLS update with an explicit implementation of (10) with gradients given by (12) and Hessians given by (13)

4.3 Regularized Generalized Matrix Factorization

In practice, we do not know the dimension of the latent space and we may need to estimate it from the data. Methods for selecting the dimension range from cross-validation, information criteria testing (Bai and Ng, 2002; Hirose and Yamamoto, 2015), a somewhat arbitrary choice for the threshold of the variance explained (Smith et al., 2015), or L_1 penalty added to the log-likelihood (Hui et al., 2018). We propose another smooth shrinkage parameter, motivated by regularized matrix factorization (Zou et al., 2006).

Instead of controlling the rank by explicitly choosing the number of latent variables, we can set a large upper bound on the number of latent variables (e.g. $\sqrt{m}$) and then regularize the latent variables with an extra term $\frac{1}{2}\|\Lambda\|_2^2$ added to the PQL criterion. If we control penalties with a scaling parameter γ, this leads to the regularized objective function

$$
L_2(\Psi) = \sum_{i=1}^n \sum_{j=1}^m (y_{ij}\hat{\eta}_{ij} - b(\hat{\eta}_{ij})) + \frac{\gamma}{2}\|U\|_2^2 + \frac{\gamma}{2}\|\Lambda\|_2^2,
\tag{14}
$$

where $\|\cdot\|_2$ is the Frobenius norm. Srebro et al. (2005) show that solving (14) with U and Λ of sufficiently high rank is equivalent to solving

$$
L_*(\Psi) = \sum_{i=1}^n \sum_{j=1}^m (y_{ij}\hat{\eta}_{ij} - b(\hat{\eta}_{ij})) + \gamma\|M\|_*,
\tag{15}
$$

where $\| \cdot \|_*$ denotes the nuclear norm and $M = U\Lambda'$.

Equation (15) can be interpreted as a relaxed version of a rank constraint on M. To tune dimensionality we can control the penalty parameter γ in (14). In particular, for sufficiently large γ some singular values of M vanish, effectively reducing dimensionality of the latent space (See Figure 6 for an illustrative example).

We illustrate empirical properties of this method of dimensionality selection in a simulation study in Section 7.3.

5. Evaluation

The baseline method for comparing the proposed algorithms is the R package `gllvm` (Niku et al., 2019b), applied with default settings. This is a state-of-the-art method using a variational approximation (Hui et al., 2017b) and automatic differentiation (Niku et al., 2019a) to improve computationally efficiency of maximum likelihood approaches to GLLVM estimation. Since other approaches to estimation of parameters in GLLVM are comparable or worse in terms of speed, we only use the `gllvm` package for comparison.

We use a series of metrics and techniques to evaluate the quality of the fit of the models proposed in this article. For example, although the above algorithms minimize a version of penalized deviance, in certain applications practitioners might instead be interested in properties of the latent space, estimates of the fixed effect parameters, or predictive performance of the model. In this section, we review key metrics and methods used for comparing the performance of estimation approaches for GLLVMs.

Deviance for evaluating fit to responses. Following Nelder and Wedderburn (1972) we define deviance as

$$D(Y, \hat{M}) = 2 \sum_{i=1}^{n} \sum_{j=1}^{m} \Big(\log \big(p(y_{ij} \mid \hat{\eta}_{ij}) \big) - \log \big(p(y_{ij} \mid \hat{\eta}_{0,ij}) \big) \Big),$$

where $\hat{M} = [\hat{\eta}_{ij}]_{ij}$ are the parameters predicted from the model, $\hat{\eta}_{0,ij}$ are the linear predictors from the saturated model, and p is the density of the distribution assumed in (1).

The absolute value of the deviance is usually hard to interpret. Therefore, we choose to calculate the ratio of deviances between either two fitted models or between the fitted model and the null model. The latter ratio is interpreted as unexplained deviance.

Procrustes error for evaluating fit of the latent space. In certain applications we are interested in how much variance the latent space explains. Since vectors spanning latent spaces are not identifiable, we use a metric that rotates them before comparison. We follow Niku et al. (2019a) and for the purpose of this work we define Procrustes as

$$P(\Lambda_0, \hat{\Lambda}) = \min_{\Omega} \|\Lambda_0 - \Omega\hat{\Lambda}\|_F, \text{ subject to } \Omega'\Omega = I,$$

where $\| \cdot \|_F$ is the Frobenius norm, Λ_0 stands for ground truth latent loadings, $\hat{\Lambda}$ are predicted loadings, and Ω is a rotation matrix.

This metric is only available in simulations, where we have access to the ground-truth factor loadings. See Section 7 for examples.

Mean squared errors for evaluating fixed effect coefficients. Researchers may also be interested in the accuracy of estimates of the fixed effects, i.e. β_j in (1). A natural way to evaluate the fit is to compute the Euclidean distance between true and predicted parameters

$$F(\beta, \hat{\beta}) = \|B - \hat{B}\|_2^2,$$

where $B = [\beta_1, \beta_2, ..., \beta_m]$.

As with Procrustes error, evaluation of F requires access to the true β and as such the metric is also only available in simulations.

Predictive performance In certain situations some responses may not be observed and need to be predicted. The AIRWLS algorithm introduced in Section 4.2.1 can be used in such missing data settings since in each regression step we can use only a subset of rows or columns as long as there are enough observations. That is, our method can be used for sparsely observed data.

For the Newton method (Section 4.2), in order to compute gradients we need the fully observed matrix. However, following ideas from the Soft-Impute method (Mazumder et al., 2010), in each iteration, we can use predictions from the previous step to *impute* missing values and then compute gradients.

The above feature of our proposed algorithms also enables us to straightforwardly employ cross-validation for assessing the fit and choosing model tuning parameters. To assess overall out-of-sample goodness-of-fit we can randomly sample elements of the observed response matrix Y, hold them out, and compute the out-of sample deviance of the predictions. Depending on the objective, multiple techniques for sampling can be used, including uniform sampling of matrix entries, sampling based on response values, or sampling based on values of predictors. We use this method for evaluating models on real datasets (Section 6.2) and in simulations for choosing the optimal shrinkage parameter (Section 7.3).

6. Data studies

We report two applications from Ecology, with two distributions of responses: Poisson (Section 6.1) and Bernoulli (Section 6.2). In Section 6.2 we describe a study on a 48,737 × 4841 matrix (plant species at many locations). We compare three methods: AIRWLS, Newton, and variational approximation (Niku et al., 2019a). We implement the AIRWLS method as described in Section 4.2.1. We implement the Newton method using formulas for gradients and Hessians as obtained in Section 4.2.2 and we refer to it as *Newton*. We compare these methods with the implementation of variational approximation estimation in GLLVM provided by Niku et al. (2019b) in the `gllvm` package and we refer to that method as `gllvm`.

6.1 Study 1: Abundance of ants

In order to validate our algorithms, we start by analyzing a small dataset of 41 ant species measured at 30 study sites in March–April 2008 (Gibb and Cunningham, 2011). Together with the abundance of ants at these sites, researchers measured environmental variables on each site (% cover of shrubs, bare ground, coarse woody debris, etc.). For a full report on the data acquisition methodology and the study design, see Gibb and Cunningham (2011).
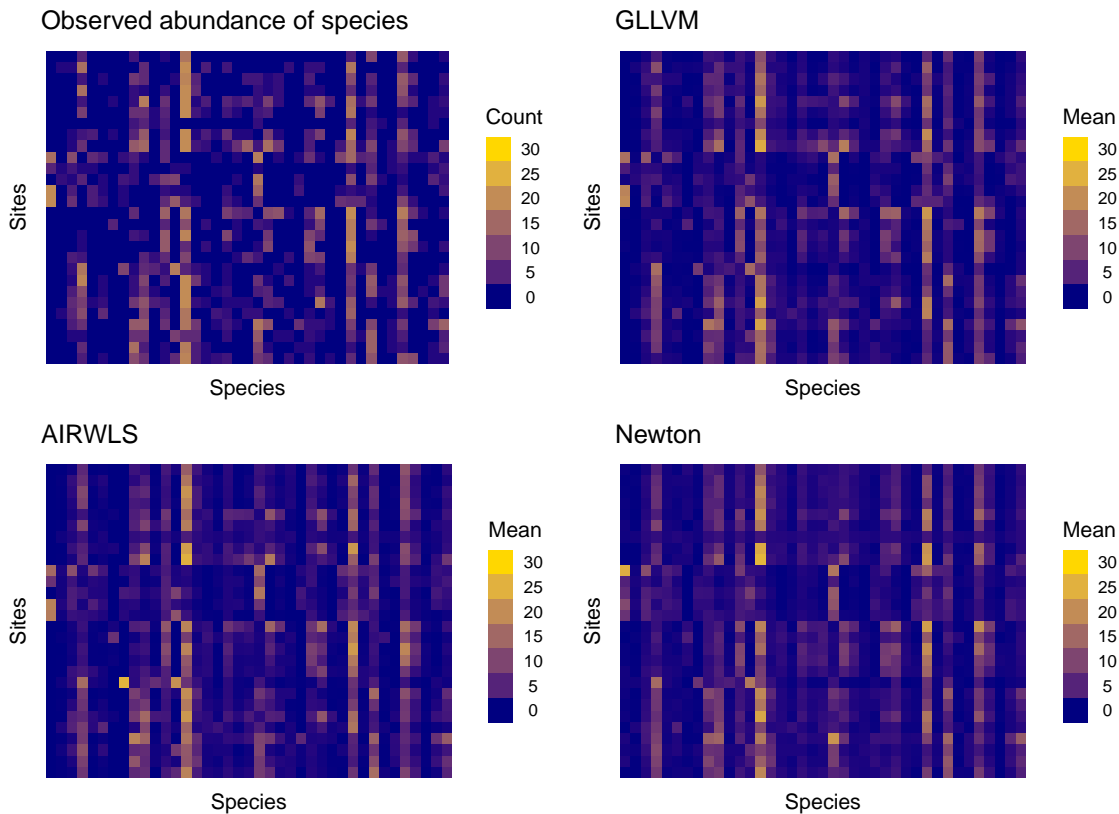
Figure 1: Qualitative validation of our algorithms on a dataset of abundance of 41 ant species measured at 30 observational sites. In our experiment, we assumed responses are Poisson-distributed and we fitted models using a baseline `gllvm` (top right) and our proposed methods AIRWLS (bottom left) and Newton (bottom right). We found that all three methods capture qualitatively similar features of the observed matrix. This observation is confirmed in our quantitative analysis.

Here, we show how GLLVMs can be used for identifying intrinsic environmental factors that are not expressed in measured habitat structure variables. We denote abundance data by Y and measured environmental variables by X. We set the number of factors to $p = 2$ for ordination purposes. In Figure 1 we present predicted means of all species for all environments.

To further validate the method, we removed one of the environmental variables and examined if some of its variability was captured by a latent variable (which could be interpreted as a missing covariate). That is, we hypothesized that some of variability explained by the variable can be explained by a latent factor. We arbitrarily chose to remove `Shrub.cover`. In Figure 2 we present variability of sites in the factor scores space and the relation between the second component and `Shrub.cover`. The Pearson correlation coefficient between the two was -0.49 and it was statistically significant with p-value equal 0.005 in a two-sided t-test.
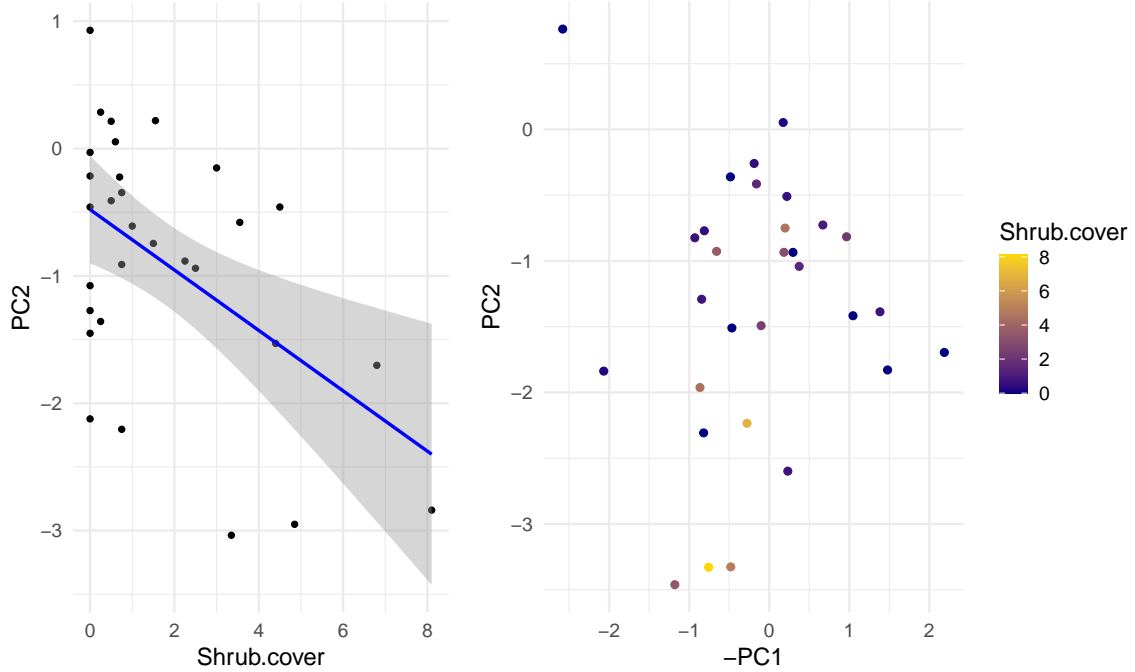
Figure 2: Validation of the latent decomposition. In the dataset of abundance of ants we held out a known feature of observational sites (`Shrub.cover`) and compared it to estimated latent scores. We used the AIRWLS algorithm for model fitting and found that the second latent score correlates with the held-out feature (left). We present the 2-dimensional space spanned by latent scores of observational units and the corresponding value of `Shrub.cover` for each unit (right).

Turning to computation time, the `gllvm` fit converged in 2.3 seconds, Newton in 0.2 seconds, and AIRWLS in 0.5 seconds on 1 CPU and in 0.04 seconds when run in parallel across 8 CPUs. Results from these methods were comparable in terms of deviance explained: `gllvm` 79%, AIRWLS 79%, and Newton 75%.

6.2 Study 2: Large scale coexistence of species

In this study, we analyzed data from systematic flora surveys along the east coast of New South Wales, using data obtained from the New South Wales Government (NSW Department of Planning Industry and Environment). Transects of fixed area were exhaustively searched and all plant species found in them identified to species, where possible. We were interested in understanding co-occurrence patterns of different plants.

A total of $48,737$ transects were surveyed over the last two decades, and $4,841$ species have been recorded as present or absent from each. Each observation $y_{i,j}$ is binary (presence/absence of species j at location i). For each observational unit, we were provided 9 covariates describing it.

We first filtered out columns and rows with less than 0.1% positive responses and we were left with 48,331 observational units and 2,211 species. Next we fitted a model with $p = 20$ on the full dataset using the proposed Newton algorithm in Section 4.2.2. We used that model for sampling missing data. Specifically, we held out 500 elements (i, j) for which the model predicted 1 with at least 0.5 probability and 500 for which the model predicted 0 with at least 0.5 probability. We then proceeded to fit the model with the resulting matrix with missing data, as described in Section 5. Based on the scree plot (Figure 3, left panel) we set $p = 3$. We evaluated the fit by calculating the out-sample deviance and area under the receiver operator curve (AUC), and compared predictions from only the fixed effect and the full model (Figure 3, right panel).

Due to the large scale of the problem, we were unable to use `gllvm`. We tested only the Newton algorithm and it converges in 3 hours on commodity hardware. Our model with only fixed effects had AUC $= 0.72$ and explained 39% of the out-of-sample deviance in the held-out dataset. By contrast, the full model containing three latent variables models had AUC $= 0.87$ and explained 58% of the out-of-sample deviance (Figure 3, right panel).
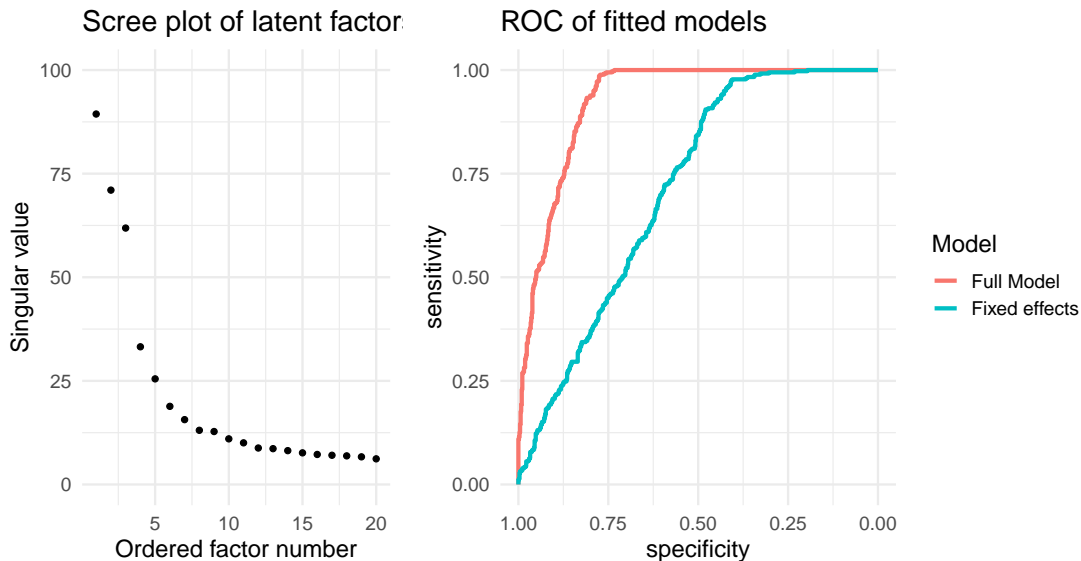

Figure 3: Model selection in the large scale coexistence of species dataset. We used the Newton algorithm to fit the model. To chose the number of factors, we used a scree plot, i.e. we plotted singular values of the latent decomposition (left), defined as the diagonal of Λ. The drop between the third and fourth value motivated the choice of the dimension of the latent space $p = 3$. To validate if the latent space contains meaningful information, we compared ROC curves computed on a held-out dataset of 500 matrix entries and found a substantial increase of the predictive power of the full model compared to the model only using fixed effects (right).

Next, we looked at how the size of the dataset influences performance of algorithms. We sampled rows and columns and built models for the subset of the data. We chose

$\rho \in \{0.005, 0.01, 0.015, ..., 0.065\}$ and sampled $\lfloor \rho \cdot n \rfloor$ rows and $\lfloor \rho \cdot m \rfloor$ columns, i.e. from 0.5% to 6.5% of the total number of rows and columns. Based on the scree plot of singular values of the full model we chose the number of latent factors $p = 3$ (Figure 3). In order to use a metric comparable across different samplings, we used mean deviance.

We compare our Newton implementation and AIRWLS model with the baseline `gllvm`. The proposed Newton method performed best both in terms of deviance and computation time (Figure 4). Elementary linear extrapolation suggests that computing a solution for the full dataset via `gllvm` would take at least 2 weeks. However, we failed to apply `gllvm` to the full dataset due to anticipated memory constraints.
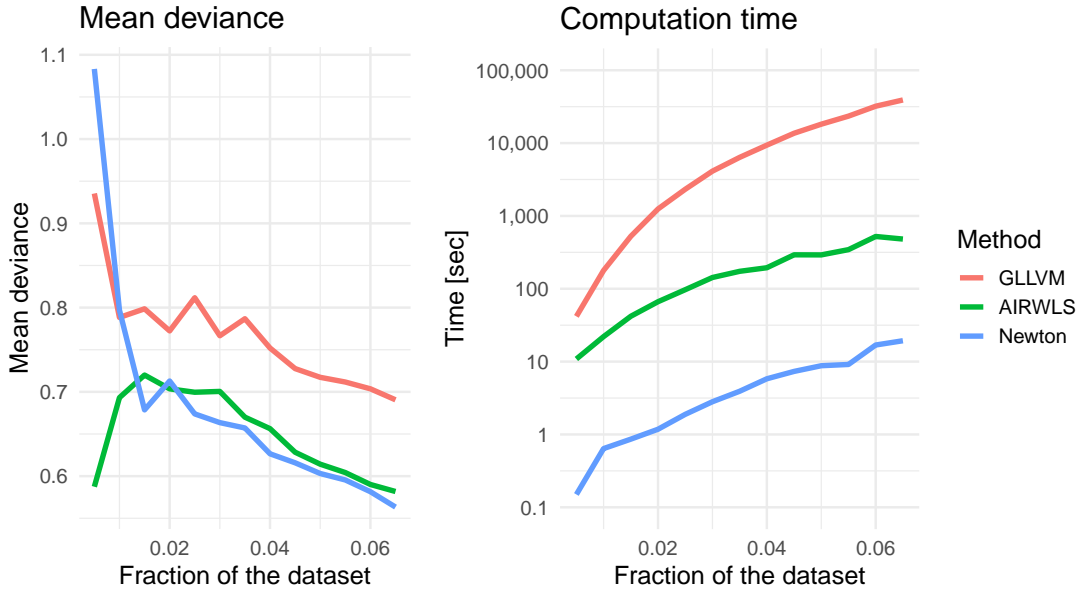

Figure 4: Accuracy and computation time as a function of the dataset size. We sampled fractions of columns and rows of the large dataset of coexistence of species and compared proposed methods on sampled subsets. A fraction $\rho \in \{0.01, ..., 0.065\}$ corresponded to $\lfloor \rho \cdot 48737 \rfloor$ rows and $\lfloor \rho \cdot 2211 \rfloor$ columns. We fitted models with $p = 3$ latent variables, and found that both AIRWLS and Newton methods outperform the baseline `gllvm` implementation both in terms of explained deviance (left) and computation time (right).

7. Simulations

We conducted a numerical study to empirically investigate how the number of responses, the number of latent variables, and the distribution of responses influence the performance of the proposed algorithms for estimating GLLVMs.

7.1 Setting

We designed simulations to mimic the setting from the data study described in Section 6.1. To that end, we started with a GLLVM model fitted to the ant abundance data. We used sample estimates of the covariance matrix of X and Λ as the basis for sampling new multivariate Gaussian variables for X and Λ. For U and B, we constructed matrices by randomly simulating each element independently from a standard Gaussian distribution. From these quantities, we were able to construct a matrix of linear predictors for each row $[\eta_{i1}, ..., \eta_{ip}]$ in (1).

Our procedure was parametrized with the number of observational units n, the number of responses in each unit m, the number of latent variables p, and the distribution of responses. We assumed a canonical link for each distribution. For each set of parameters, in each experimental trial we generated a response matrix following (1).

As in Section 6, we estimated model coefficients using the `gllvm` package, and our proposed PQL approximation using both Newton and AIRWLS methods. We evaluated performance using mean deviance for goodness-of-fit, Procrustes error for the fit of the latent space, and mean squared error for fixed-effect coefficients estimates as described in Section 5.

We ran simulations with $n, m \in \{200, 400, 600\}$ and $p \in \{2, 3\}$. For each set of simulation parameters, we repeated the experiment with 100 generated datasets. For all three methods, we used the same stopping criterion with a relevant error tolerance equal to 10^{-3}.

7.2 Results

We observed that `gllvm` and Newton methods achieved similar performance in terms of deviance explained, Procrustes error, and the MSE of fixed effects, while AIRWLS performs slightly better (Figure 5). The main gain however comes in computation time. With the same stopping criteria used in all three methods, computation times differ by orders of magnitude. On average, the Newton algorithm took 21 seconds to compute, AIRWLS method took 117 seconds, while `gllvm` 66 minutes (Figure 5, bottom right).

7.3 Model selection: Unknown p and regularization

Our cross-validation framework introduced in Section 5 allows us to fine-tune the model parameters, particularly for predictive applications. For illustration, we simulate Poisson responses using the process described in Section 7.1 with $n = m = 100$ and $p = 2$. We hold out randomly selected 5% of responses and used the AIRWLS algorithm to estimate model parameters. We set an upper bound for the size of the latent space to 10.

We illustrate two approaches to model selection: regularization introduced in Section 4.3 with a smooth shrinkage parameter γ and regularization by choosing a fixed number of factors p. For the smooth regularization, in order to choose the best γ we held out 5% of matrix entries and fitted the model using the Newton method with $\gamma \in \{0, 1, ..., 60\}$ in (14). For each γ we computed deviance on the held-out set (Figure 6 left). When we chose a fixed number of factors, we considered models with $p \in \{1, 2, ..., 50\}$, and similarly calculated the out-of-sample deviance on the 5% of observations in the held-out set. As presented in

Figure 6, we achieved slightly better predictions with a smaller shrunken model compared to the fixed-dimension model.

We conclude that cross-validation and regularization are promising concepts for applications of GLLVMs, however a thorough analysis of the theoretical properties of cross-validation and regularization is beyond the scope of this work.

8. Discussion

Our PQL-based methods are orders of magnitude faster than current state-of-the-art algorithms for estimating model parameters for GLLVMs. While the estimates are biased (Nie, 2007; Hui et al., 2017a), it has been proven that they generally perform well when the number of units and/or responses becomes large (Huber et al., 2004; Rabe-Hesketh and Skrondal, 2004), which was also validated in our simulation and data studies. Our algorithms are elementary to implement using existing GLM routines. Specifically, we provide an R implementation via the open source package gmf[1] enabling integration with existing workflows and further extension of our algorithms.

Throughout this work we have illustrated an applicability of our methodology in the context of ecology. However, similar problems can be found in other disciplines, whenever we are interested in extracting latent factors underlying certain responses. Our methods are particularly useful when matrices are large, for example in studies of behavior of subjects online with thousands of individuals and items or web pages they view. Not only are our methods are fast, but they can also be easily decomposed and parallelized across multiple machine, leveraging modern open parallel computing platforms such as Apache Spark (Zaharia et al., 2010).

References

Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

David J Bartholomew, Martin Knott, and Irini Moustaki. *Latent variable models and factor analysis: A unified approach*, volume 904. John Wiley & Sons, 2011.

Silvia Bianconcini and Silvia Cagnone. Estimation of generalized linear latent variable models via fully exponential laplace approximation. *Journal of Multivariate Analysis*, 112:183–193, 2012.

Silvia Bianconcini et al. Asymptotic properties of adaptive maximum likelihood estimators in latent variable models. *Bernoulli*, 20(3):1507–1531, 2014.

FG Blanchet, Gleb Tikhonov, and Anna Norberg. Hmsc: Hierarchical modelling of species community. *R package version*, pages 2–2, 2018.

Norman E Breslow and David G Clayton. Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, 88(421):9–25, 1993.

---

1. http://github.com/kidzik/gmf/

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Eugene Demidenko. *Mixed models: theory and applications with R*. John Wiley & Sons, 2013.

Heloise Gibb and Saul A Cunningham. Habitat contrasts reveal a shift in the trophic position of ant assemblages. *Journal of Animal Ecology*, 80(1):119–127, 2011.

Peter J Green. Penalized likelihood for general semi-parametric regression models. *International Statistical Review/Revue Internationale de Statistique*, pages 245–259, 1987.

Nathan Halko, Per-Gunnar Martinsson, Yoel Shkolnisky, and Mark Tygert. An algorithm for the principal component analysis of large data sets. *SIAM Journal on Scientific computing*, 33(5):2580–2594, 2011.

Kei Hirose and Miyuki Imada. Sparse factor regression via penalized maximum likelihood estimation. *Statistical Papers*, 59:633–662, 2018.

Kei Hirose and Michio Yamamoto. Sparse estimation via nonconcave penalized likelihood in factor analysis model. *Statistics and Computing*, 25(5):863–875, 2015.

Philippe Huber, Elvezio Ronchetti, and Maria-Pia Victoria-Feser. Estimation of generalized linear latent variable models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):893–908, 2004.

Francis K. C. Hui. *boral: Bayesian Ordination and Regression AnaLysis*, 2020. R package version 1.9.

Francis K C Hui, Samuel Mueller, and A H Welsh. Joint selection in mixed models using regularized pql. *Journal of the American Statistical Association*, 112(519):1323–1333, 2017a.

Francis K C Hui, David I Warton, John T Ormerod, Viivi Haapaniemi, and Sara Taskinen. Variational approximations for generalized linear latent variable models. *Journal of Computational and Graphical Statistics*, 26(1):35–43, 2017b.

Francis K C Hui, Emi Tanaka, and David I Warton. Order selection and sparsity in latent variable models via the ordered factor lasso. *Biometrics*, 74(4):1311–1319, 2018.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90:773–795, 1995.

Anthony YC Kuk. Laplace importance sampling for generalized linear mixed models. 1999.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug): 2287–2322, 2010.

P McCullagh and JA Nelder. *Generalized linear models*, volume 37. Chapman and Hall, 1983.

CA McGilchrist. Estimation in generalized mixed models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):61–69, 1994.

Irini Moustaki and Martin Knott. Generalized latent trait models. *Psychometrika*, 65(3): 391–411, 2000.

John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

Lei Nie. Convergence rate of mle in generalized linear and nonlinear mixed-effects models: Theory and applications. *Journal of Statistical Planning and Inference*, 137(6):1787–1804, 2007.

Jenni Niku, David I Warton, Francis KC Hui, and Sara Taskinen. Generalized linear latent variable models for multivariate count and biomass data in ecology. *Journal of Agricultural, Biological and Environmental Statistics*, 22(4):498–522, 2017.

Jenni Niku, Wesley Brooks, Riki Herliansyah, Francis KC Hui, Sara Taskinen, and David I Warton. Efficient estimation of generalized linear latent variable models. *PloS one*, 14 (5):e0216129, 2019a.

Jenni Niku, Francis KC Hui, Sara Taskinen, and David I Warton. gllvm: Fast analysis of multivariate abundance data with generalized linear latent variable models in r. *Methods in Ecology and Evolution*, 10(12):2173–2182, 2019b.

Otso Ovaskainen, Gleb Tikhonov, Anna Norberg, F Guillaume Blanchet, Leo Duan, David Dunson, Tomas Roslin, and Nerea Abrego. How to make more out of community data? a conceptual framework and its implementation as models and software. *Ecology Letters*, 20(5):561–576, 2017.

Maximilian Pichler and Florian Hartig. A new method for faster and more accurate inference of species associations from novel community data. *arXiv preprint arXiv:2003.05331*, 2020.

Sophia Rabe-Hesketh and Anders Skrondal. *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Chapman and Hall/CRC, 2004.

Sophia Rabe-Hesketh, Anders Skrondal, and Andrew Pickles. Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal*, 2(1):1–21, 2002.

Geneviève Robin. *Low-rank methods for heterogeneous and multi-source data*. PhD thesis, École Polytechnique, Paris, 2019.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

Hans J Skaug. Automatic differentiation to facilitate maximum likelihood estimation in nonlinear random effects models. *Journal of Computational and Graphical Statistics*, 11 (2):458–470, 2002.

Anders Skrondal and Sophia Rabe-Hesketh. *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Crc Press, 2004.

Alison B Smith, Aanandini Ganesalingam, Haydn Kuchel, and Brian R Cullis. Factor analytic mixed models for the provision of grower information from national crop variety testing programs. *Theoretical and Applied Genetics*, 128(1):55–72, 2015.

Charles Spearman. "general intelligence" objectively determined and measured. 1904.

Nathan Srebro, Noga Alon, and Tommi S Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances In Neural Information Processing Systems*, pages 1321–1328, 2005.

Oliver Stegle, Leopold Parts, Matias Piipari, John Winn, and Richard Durbin. Using probabilistic estimation of expression residuals (peer) to obtain increased power and interpretability of gene expression analyses. *Nature protocols*, 7(3):500, 2012.

Edward F Vonesh. A note on the use of laplace's approximation for nonlinear mixed-effects models. *Biometrika*, 83(2):447–452, 1996.

Edward F Vonesh, Hao Wang, Lei Nie, and Dibyen Majumdar. Conditional second-order generalized estimating equations for generalized linear and nonlinear mixed-effects models. *Journal of the American Statistical Association*, 97(457):271–283, 2002.

David I Warton, F Guillaume Blanchet, Robert B O'Hara, Otso Ovaskainen, Sara Taskinen, Steven C Walker, and Francis KC Hui. So many variables: joint modeling in community ecology. *Trends in Ecology & Evolution*, 30(12):766–779, 2015.

David I Warton, F Guillaume Blanchet, Robert O'Hara, Otso Ovaskainen, Sara Taskinen, Steven C Walker, and Francis K C Hui. Extending joint models in community ecology: A response to beissinger et al. *Trends in ecology & evolution*, 31(10):737–738, 2016.

Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, Ion Stoica, et al. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.
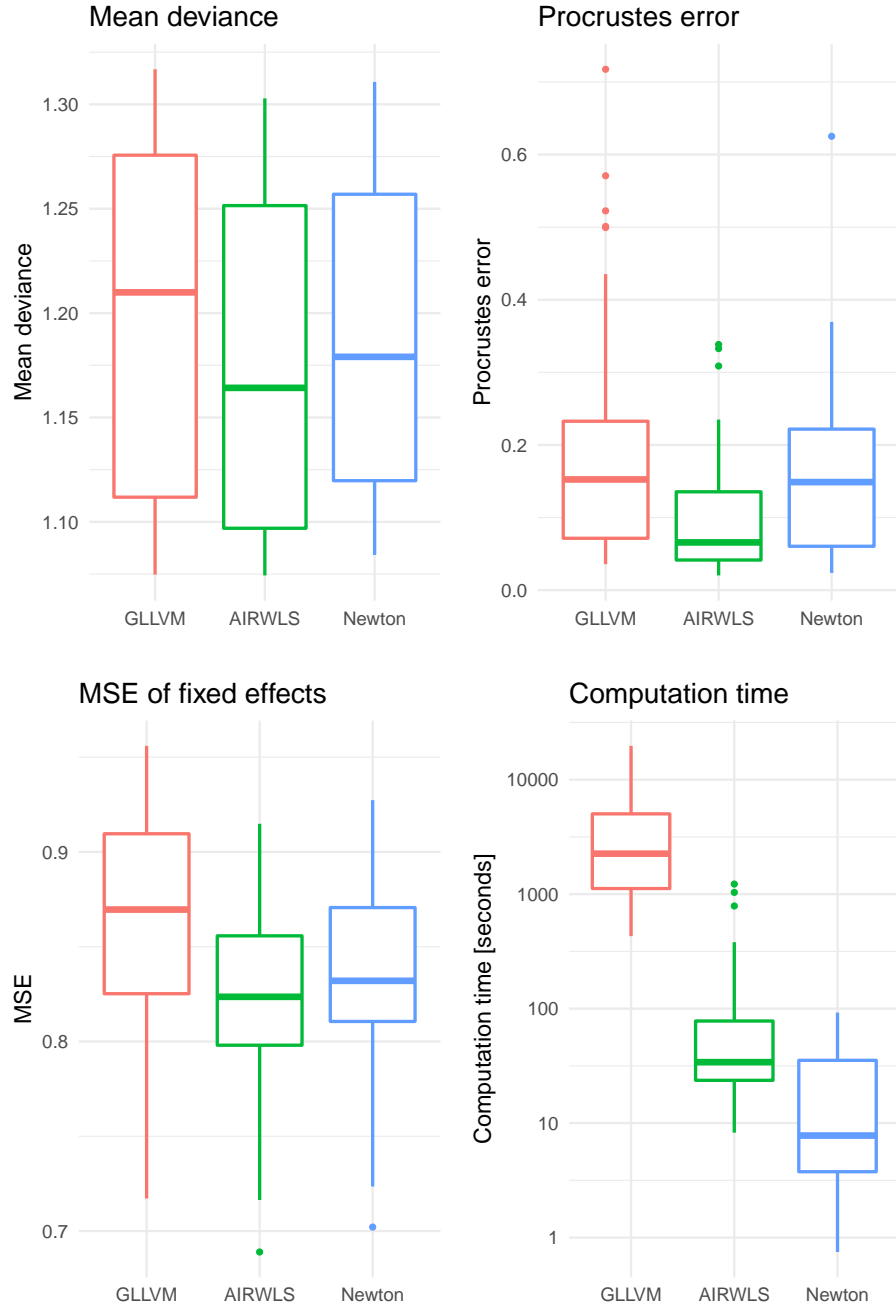
Figure 5: Comparison of proposed methods with the `gllvm` package. In this simulation study, we sampled data such that covariances of X and V matched those of the sample estimates from the ant abundance data. We varied $n, m \in \{200, 400, 600\}$, $p \in \{2, 3\}$ and the distribution of responses (Poisson or Binomial). Based on generating 100 datasets for each combination of simulation parameters, we found that the AIRWLS slightly outperforms other methods on all metrics (mean deviance, Procrustes error, and MSE of fixed effects), while the Newton algorithm is superior in terms of the computation time.
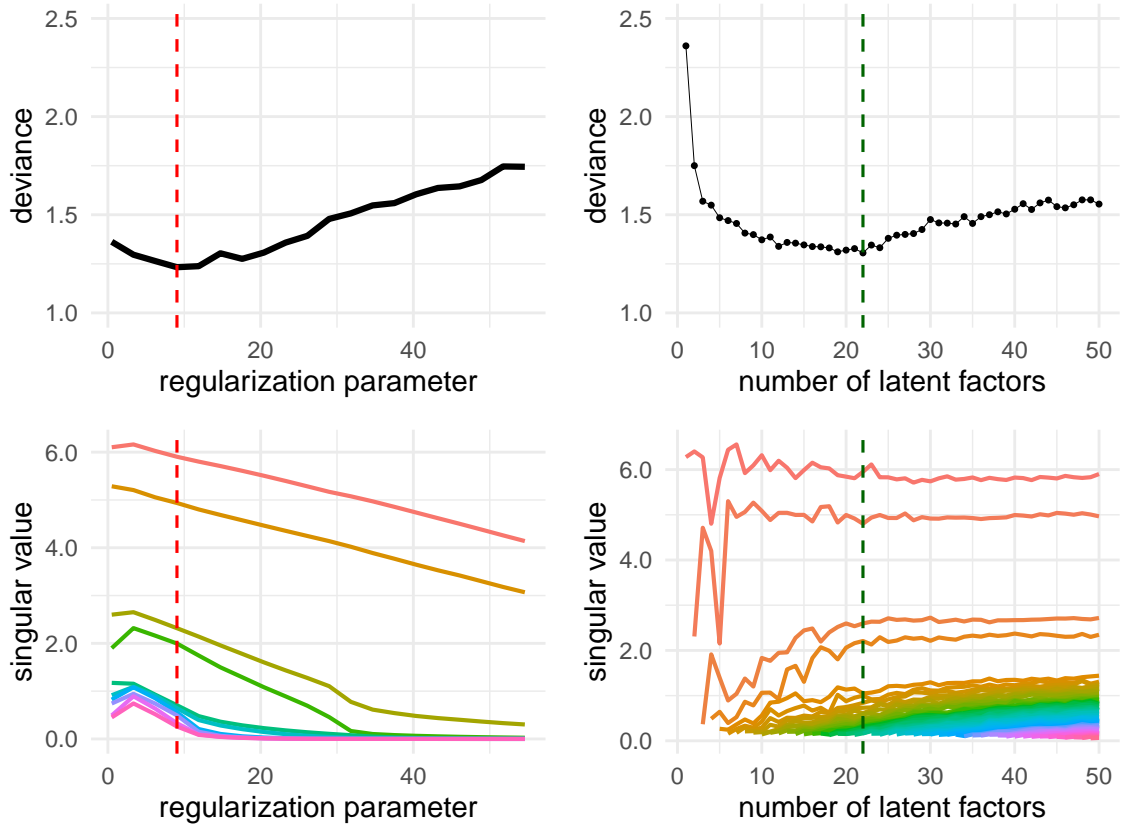
Figure 6: Model selection using the cross-validation framework. We present two methods for choosing the complexity of the model: regularization (left panels) and fixed-dimension (right panels). Cross-validation allowed us to select optimal parameters for prediction on a hold-out dataset of 5% of observations (dashed lines). The number of nonzero singular values for a given γ is the selected rank. As expected the singular values all shrink when the regularizing parameter increased (bottom left), while they remain stable when we add more factors, without extra regularization (bottom right). In this example, the best regularized model performs better than the best fixed-dimension model, despite having a smaller number of factors fixed to 10.