Negative binomial factor model for analyzing microbiome count data

September 6, 2020

1 Introducttion

xxx; thetta def;

2 Method

We consider the microbial abundance data of count type obtained after high-throughput sequencing of n samples as $\mathbf{Y} = [y_{ik}]_{n\times q} = [\mathbf{y}_1, \ldots, \mathbf{w}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times q}$. The problem of interest is to understand the association of the abundance $\mathbf{Y}$ to predictors/features $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times p}$, and control variables $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times c}$. $\mathbf{Z}$ consists of a set of variables that should always be included in the model and are thus not regularized. Depending on the application, we consider experimental

input such as age, gender (factor variable) as control variables. A typical characteristic of the abundance data is overdispersion (variance larger than mean). In the multivariate regression framework with the outcome $\mathbf{Y}$ and covariates $\{\mathbf{X}, \mathbf{Z}\}$, we assume that the abundance of each of the microbial species follows negative binomial distribution to model the underlying association between the two entity.

Using the alternative form of the negative binomial distribution [Zeileis et al., 2008], a generative model for the microbial abundance of the jth species in ith sample is given by

$$p(y_{ij}; \mu_{ij}, \phi_j) = \mathrm{NB}(y_{ij}; \mu^*_{ij}, \phi^*_j) = \binom{y_{ij} + \phi^*_j - 1}{y_{ij}} \frac{{\mu^*_{ij}}^{y_{ij}} {\phi^*_j}^{\phi^*_j}}{(\mu^*_{ij} + \phi^*_j)^{y_{ij}+\phi^*_j}}, \tag{1}$$

where μ^*_{ij} is the entry specific mean and $\phi_j \in \mathbb{R}^+$ is the species specific dispersion parameter. Let us jointly represent the dispersion parameters for q species by $\mathbf{\Phi}^* = [\phi^*_1, \ldots, \phi^*_q]$ and the entry specific parameter by $\boldsymbol{\mu}^* = [\mu^*_{ij}]_{n \times q}$. For the generative model (1), $\mathbb{E}(y_{ij}) = \mu^*_{ij}$, $\mathrm{var}(y_{ij}) = \mu^*_{ij} + \frac{{\mu^*_{ij}}^2}{\phi^*_j}$ and $\mathrm{var}(y_{ij}) \geqslant \mathbb{E}(y_{ij})$, making the model suitable for modeling the overdispersed count data of microbial abundance. Then the joint negative log-likelihood function is given by

$$\mathcal{L}(\boldsymbol{\mu}^*, \mathbf{\Phi}^*) = -\sum_{i=1}^{n} \sum_{k=1}^{q} \ell_k(\mu^*_{ik}, \phi^*_k), \tag{2}$$

where $\ell_k(\mu^*_{ik}, \phi^*_k) = \log p(y_{ik}; \mu^*_{ik}, \phi^*_k)$.

We associate covariates to the multivariate outcome by linking the entry specific

parameter $\boldsymbol{\mu}^*$ to the linear predictors $\boldsymbol{\eta}^* = [\eta_{ij}^*]_{n\times q}$ given by

$$g(\boldsymbol{\mu}^*) = \boldsymbol{\eta}^*(\mathbf{C}^*, \boldsymbol{\beta}^*, \mathbf{O}) = \mathbf{O} + \mathbf{Z}\boldsymbol{\beta}^* + \mathbf{X}\mathbf{C}^*, \tag{3}$$

where $g(\cdot)$ is any link function, $\mathbf{O} = [o_{ik}]_{n\times q} \in \mathbb{R}^{n\times q}$ is a fixed offset term, $\mathbf{C}^* = [\mathbf{c}_1^*, \ldots, \mathbf{c}_q^*] \in \mathbb{R}^{p\times q}$ is the coefficient matrix corresponding to the predictors, and $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^*, \ldots, \boldsymbol{\beta}_q^*] \in \mathbb{R}^{p_z\times q}$ is the coefficient matrix corresponding to the control variables. The intercept is included by taking the first column of $\mathbf{Z}$ to be $\mathbf{1}_n$, the $n \times 1$ vector of ones. We choose $g(x) = \log x$ as our link function. Depending on the problem, one may choose another link function satisfying $\boldsymbol{\mu}^* \geqslant 0$. For simplicity, we may write $\boldsymbol{\eta}^*(\mathbf{C}^*, \boldsymbol{\beta}^*, \mathbf{O})$ as $\boldsymbol{\eta}^*$ if no confusion arises.

We assume the outcomes are conditionally independent given $\mathbf{X}$ and $\mathbf{Z}$. For fixed dispersion parameter ϕ_j^*, the negative binomial distribution (1) belongs to the exponential family. Hence, we reparameterize and conveniently express the negative log-likelihood function (2) as

$$\mathcal{L}(\boldsymbol{\Theta}^*, \boldsymbol{\Phi}^*) = -\operatorname{tr}(\mathbf{Y}^{\mathrm{T}}\boldsymbol{\Theta}^*) + \operatorname{tr}(\mathbf{J}^{\mathrm{T}}\mathbf{B}(\boldsymbol{\Theta}^*)) + \sum_{i,j} \log \binom{y_{ij} + \phi_j^* - 1}{y_{ij}}, \tag{4}$$

where $\mathbf{J} = \mathbf{1}_{n\times q}$, $\operatorname{tr}(\mathbf{A})$ is the *trace* of a square matrix $\mathbf{A}$, $\mathbf{B}(\boldsymbol{\Theta}^*) = [b(\theta_{ij}^*)]_{n\times q}$ and $\boldsymbol{\Theta}^* = [\theta_{ij}^*]_{n\times q} \in \mathbb{R}^{n\times q}$ is the natural parameter of the exponential family when $\boldsymbol{\Phi}^*$ is fixed such that $\theta_{ij}^* = \log \frac{\mu_{ij}^*}{\mu_{ij}^* + \phi_j^*}$ and $b(\theta_{ij}^*) = -\phi_j^* \log(1 - e^{\theta_{ij}^*})$. It is trivial to show that $\mu_{ij}^* = b'(\theta_{ij}^*)$. Then, one can link θ_{ij}^* to the linear predictor as $g(b'(\theta_{ij}^*) = \eta_{ij}^*$.

For missing entries in $\mathbf{Y}$, let us define an index set of the observed outcomes as

$$\boldsymbol{\Omega} = \{(i,k); w_{ik} \text{ is observed}, i = 1,\ldots,n, k = 1,\ldots,q\},$$

and denote the projection of $\mathbf{Y}$ onto $\boldsymbol{\Omega}$ by $\widetilde{\mathbf{Y}} = \mathcal{P}_{\boldsymbol{\Omega}}(\mathbf{Y})$, where $\tilde{y}_{ik} = y_{ik}$ for any $(i,k) \in \boldsymbol{\Omega}$ and $\tilde{y}_{ik} = 0$ otherwise. Accordingly, the negative log-likelihood function with incomplete data is given by

$$\mathcal{L}(\boldsymbol{\Theta}^*, \boldsymbol{\Phi}^*) = -\operatorname{tr}(\widetilde{\mathbf{Y}}^{\mathrm{T}}\boldsymbol{\Theta}^*) + \operatorname{tr}(\widetilde{\mathbf{J}}^{\mathrm{T}}\mathbf{B}(\boldsymbol{\Theta}^*)) + \sum_{i,j \in \boldsymbol{\Omega}} \log \binom{y_{ij} + \phi_j^* - 1}{y_{ij}},$$

where $\widetilde{\mathbf{J}} = \mathcal{P}_{\boldsymbol{\Omega}}(\mathbf{J})$ and $g(b'(\boldsymbol{\Theta}^*) = \boldsymbol{\eta}^*$. Henceforth, we mainly focus on the complete data case (4) when presenting our proposed model, as the extension to the missing data case by and large only requires replacing $\mathbf{Y}$ by $\widetilde{\mathbf{Y}}$ and $\mathbf{J}$ by $\widetilde{\mathbf{J}}$.

We minimize $\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ with respect to $\{\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}\}$ to estimate the model parameters such that $g(b'(\boldsymbol{\Theta})) = \boldsymbol{\eta} = \mathbf{O} + \mathbf{X}\mathbf{C} + \mathbf{Z}\boldsymbol{\beta}$ and $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_j]$. In the marginal modeling approach, one can separately estimate the model parameters for each outcomes using the framework of negative binomial regression model Zeileis et al. [2008]. However, in high-dimensional setting, the procedure ignores the dependency among the outcomes. Some of the recent development in the field of multivariate regression models with non-Gaussian outcomes models underlying dependency using low-rank sparse coefficient matrix.

We represent the underlying association in terms a few latent factors, each of which is constructed as a linear combination of covariates, resulting in the coefficient

matrix $\mathbf{C}^*$ to be of low-rank given by

$$\text{rank}(\mathbf{C}^*) \leqslant r^*. \tag{5}$$

We term the proposed model **N**egative **b**inomial **r**educed **r**ank **r**egression, denoted by NBRRR. Parameters estimate under the constraint have limited usage as it does not explore variable selection. On the other hand, in high dimensional setting, we assume that the regression association is driven by a few latent factors, each of which is constructed from a possibly different subset of the predictors, and, moreover, that each response may be associated with a possibly different subset of the latent factors. To be specific, this amounts to assuming a *co-sparse* SVD of $\mathbf{C}^*$ [Mishra et al., 2017], i.e., we decompose $\mathbf{C}^*$ as

$$\mathbf{C}^* = \mathbf{U}^*\mathbf{D}^*\mathbf{V}^{*\mathrm{T}}, \qquad \text{s.t.} \qquad \mathbf{U}^{*\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{U}^*/n = \mathbf{V}^{*\mathrm{T}}\mathbf{V}^* = \mathbf{I}_{r^*}, \tag{6}$$

where both the left singular vector matrix $\mathbf{U}^* = [\mathbf{u}_1^*, \ldots, \mathbf{u}^*_{r^*}] \in \mathbb{R}^{p\times r^*}$ and the right singular vector matrix $\mathbf{V}^* = [\mathbf{v}_1^*, \ldots, \mathbf{v}^*_{r^*}] \in \mathbb{R}^{q\times r}$ are assumed to be *sparse*, and $\mathbf{D} = \text{diag}\{d_1^*, \ldots, d_{r^*}\} \in \mathbb{R}^{r^*\times r^*}$ is the diagonal matrix with the nonzero singular values on its diagonal. The orthogonality constraints ensuring identifiability suggest that the sample latent factors, i.e., $(1/\sqrt{n})\mathbf{X}\mathbf{u}_k^*$ for $k = 1, \ldots, r^*$, are uncorrelated with each other, and the strength of the association between the latent factors and multivariate response $\mathbf{Y}$ is denoted by the singular values $\{d_1^*, \ldots, d_{r^*}\}$. We thus term the proposed model **s**parse **N**egative **b**inomial **fa**ctor **r**egression, denoted by NBFAR.

3 Estimation procedures

Joint estimation of the $\{\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}\}$ with suggested structures in Equation (5) or Equation (6) is a notoriously difficult estimation problem. In the marginal modeling approach, Zeileis et al. [2008] proposed an iterative procedure that update one parameters by keeping others fixed until convergence. In case of multivariate regression model with mixed outcomes, Luo et al. [2018], Mishra et al. [2017] proposed a similar alternating approach to estimate the structured model parameters. Following Luo et al. [2018], we propose an iterative procedure that cycles between $\mathbf{C}$-step, $\boldsymbol{\beta}$-step and $\boldsymbol{\Phi}$-step to estimate the model parameters in case of NBRRR. On the other hand, in case of NBFAR, the main idea is to extract the unit-rank components of $\mathbf{XC}$ one by one, i.e., sequentially; see Mishra et al. [2017]. The optimization problem to estimate the unit-rank components is solved by an iterative procedure that cycles between $\mathbf{u}$-step, $\mathbf{v}$-step, $\boldsymbol{\beta}$-step and $\boldsymbol{\Phi}$-step until convergence [Mishra et al., 2017].

3.1 NBRRR

The optimization problem to estimate the parameters of NBRRR is given by

$$(\widehat{\mathbf{C}}, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Phi}}) \equiv \underset{\mathbf{C},\, \boldsymbol{\beta},\, \boldsymbol{\Phi}}{\arg\min}\, \mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) \qquad \text{s.t.} \qquad \text{rank}(\mathbf{C}) \leqslant r, \tag{7}$$

where $g(b'(\boldsymbol{\Theta})) = \boldsymbol{\eta}(\mathbf{C}, \boldsymbol{\beta}, \mathbf{O}) = \mathbf{O} + \mathbf{X}\mathbf{C} + \mathbf{Z}\boldsymbol{\beta}$. The joint estimation of the unknown parameters $(\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi})$ is nontrivial. We solve the problem using a block upper bound successive minimization (BSUM) approach [Razaviyayn et al., 2013] that cycles between $\mathbf{C}$-step, $\boldsymbol{\beta}$-step and $\boldsymbol{\Phi}$-step to update the parameters $\mathbf{C}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Phi}$, respectively, until convergence.

C-step: For fixed $\boldsymbol{\beta}$ and $\boldsymbol{\Phi}$, we denote $\boldsymbol{\Theta}$ as function of $\mathbf{C}$ by $\boldsymbol{\Theta}(\mathbf{C})$. Suppose $\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}), \boldsymbol{\Phi})$ is L-Lipschitz continuous gradient for some s_c i.e., $\|\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}})) - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}))\| \leqslant s_c\|\check{\mathbf{C}} - \mathbf{C}\|$ for any conformable $\check{\mathbf{C}}$. Then

$$\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}), \boldsymbol{\Phi}) \leqslant \mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}})), \boldsymbol{\Phi}) + \mathrm{tr}(\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}}), \boldsymbol{\Phi}))^{\mathrm{T}}\{\mathbf{C} - \check{\mathbf{C}})\}) + s_c\|\check{\mathbf{C}} - \mathbf{C}\|^2/2,$$

upper bound the negative log-likelihood function $\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}), \boldsymbol{\Phi})$. The statement holds for any s_c that upper bound $\sup_{\mathbf{C}} \|\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}))\|$. Let us denote the current estimate of $\mathbf{C}$ by $\check{\mathbf{C}}$. Using the framework of BSUM algorithm, we update the parameter $\mathbf{C}$ as

$$\bar{\mathbf{C}} \equiv \arg\min_{\check{\mathbf{C}}} \mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}}), \boldsymbol{\Phi})) + \mathrm{tr}(\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}}), \boldsymbol{\Phi}))^{\mathrm{T}}\{\mathbf{C} - \check{\mathbf{C}})\}) + s_c\|\check{\mathbf{C}} - \mathbf{C}\|^2/2,$$
$$\text{s.t.} \quad \mathrm{rank}(\mathbf{C}) \leqslant r. \tag{8}$$

The unique optimal solution is given by $\bar{\mathbf{C}} = \mathbb{T}^{(r)}(\check{\mathbf{C}} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{C}}), \boldsymbol{\Phi})/s_c)$ where $\mathbb{T}^{(r)}(\mathbf{M})$ extracts r SVD components of matrix $\mathbf{M}$.

$\boldsymbol{\beta}$-step: For fixed $\mathbf{C}$ and $\boldsymbol{\Phi}$, we denote $\boldsymbol{\Theta}$ as function of $\boldsymbol{\beta}$ as $\boldsymbol{\Theta}(\boldsymbol{\beta})$). Let us assume that the $\mathcal{L}(\boldsymbol{\Theta}(\boldsymbol{\beta}), \boldsymbol{\Phi})$ is L-Lipschitz continuous gradient for some s_b. Following the $\mathbf{C}$-step, the optimization problem to update the current parameter $\check{\boldsymbol{\beta}}$ in the BSUM framework is given by

$$\bar{\boldsymbol{\beta}} \equiv \arg\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\Theta}(\check{\boldsymbol{\beta}}), \boldsymbol{\Phi}) + \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\boldsymbol{\beta}}), \boldsymbol{\Phi})^{\mathrm{T}}\{\boldsymbol{\beta} - \check{\boldsymbol{\beta}})\} + \frac{s_b}{2}\|\check{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2, \tag{9}$$

resulting in $\bar{\boldsymbol{\beta}} = \check{\boldsymbol{\beta}} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\boldsymbol{\beta}}), \boldsymbol{\Phi})/s_b$.

$\boldsymbol{\Phi}$-step: For fixed $\mathbf{C}$ and $\boldsymbol{\beta}$, we update $\boldsymbol{\Phi}$ by minimizing the negative log-likelihood function with respect to $\boldsymbol{\Phi}$, which can be obtained by a standard algorithm such as Newton-Raphson [R Core Team, 2019]; see Section 4.3 of the SM for more details.

We have relegated the details of the computation of $\{s_c, s_b\}$ to Section 4.2 in the Supplementary Materials. For convenience, let us denote the problem by NBRRR$(\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}; \mathbf{Y}, \mathbf{Z}, \mathbf{X}, \mathbf{O}, r)$. We have summarized the suggested procedure in Algorithm 1.

Algorithm 1 NBRRR($\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}; \mathbf{W}, \mathbf{Z}, \mathbf{X}, \mathbf{O}, r$)

Given: $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{O}$ and desirable rank $r \geqslant 1$.
Initialize: $\mathbf{C}^{(0)} = \mathbf{0}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\Phi}^{(0)}$, $t \leftarrow 0$.
Set $s_b = \max_{1 \leqslant j \leqslant q} \|\mathbf{Z}^{\mathrm{T}} \mathrm{diag}(\mathbf{Y}_{\cdot j} + 1)\mathbf{Z}\|/2$, $s_c = \max_{1 \leqslant j \leqslant q} \|\mathbf{X}^{\mathrm{T}} \mathrm{diag}(\mathbf{Y}_{\cdot j} + 1)\mathbf{X}\|/2$
repeat
(1) $\mathbf{C}$-step: $\mathbf{C}^{(t+1)} = \mathbb{T}^{(r)}(\mathbf{C}^{(t)} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}^{(t)}), \boldsymbol{\Phi}^{(t)})/s_c)$ where $g(b'(\boldsymbol{\Theta}(\mathbf{C}^{(t)}))) = \boldsymbol{\eta}(\mathbf{C}^{(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{O})$, and $\mathbb{T}^{(r)}(\mathbf{M})$ extracts r SVD components of matrix $\mathbf{M}$.
(2) $\boldsymbol{\beta}$-step: $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\boldsymbol{\beta}^{(t)}), \boldsymbol{\Phi}^{(t)})/s_b$ where $g(b'(\boldsymbol{\Theta}(\boldsymbol{\beta}^{(t)}))) = \boldsymbol{\eta}(\mathbf{C}^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{O})$.
(3) $\boldsymbol{\Phi}$-step: $\boldsymbol{\Phi}^{(t+1)} = \arg\min_{\boldsymbol{\Phi}} \mathcal{L}(\boldsymbol{\Theta}^{(t+1)}, \boldsymbol{\Phi})$ where $g(b'(\boldsymbol{\Theta}^{(t+1)})) = \boldsymbol{\eta}(\mathbf{O}, \mathbf{C}^{(t+1)}, \boldsymbol{\beta}^{(t+1)})$; see paragraph 4.3 of the SM.
$t \leftarrow t + 1$.
until convergence,
e.g., $\|[\mathbf{C}^{(t+1)}\ \boldsymbol{\beta}^{(t+1)}] - [\mathbf{C}^{(t)}\ \boldsymbol{\beta}^{(t)}]\|_F / \|[\mathbf{C}^{(t)}\ \boldsymbol{\beta}^{(t)}]\|_F \leqslant \epsilon$ with $\epsilon = 10^{-6}$.
return $\widehat{\mathbf{C}}$, $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\Phi}}$.

3.2 NBFAR

3.2.1 Sequential approach

Motivated by Mishra et al. [2017], we propose to sequentially extract the unit-rank components of $\mathbf{C}$, i.e., $(d_k, \mathbf{u}_k, \mathbf{v}_k)$, for $k = 1, \ldots, r$. For any k, let us denote $\mathbf{C}_k = d_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}}$. Given an estimate of the unit-rank components $\widehat{\mathbf{C}}_i$ for $i = 1, \ldots, k-1$, the optimization problem to extract the kth component is given by

$$(\hat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Phi}}) \equiv \underset{\mathbf{u},\mathbf{d},\mathbf{v},\boldsymbol{\beta},\boldsymbol{\Phi}}{\arg\min} \quad \mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) + \rho(\mathbf{C}; \lambda), \tag{10}$$
$$\text{s.t.} \quad \mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}},\ \mathbf{u}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u}/n = \mathbf{v}^{\mathrm{T}}\mathbf{v} = 1, \boldsymbol{\Theta} = \boldsymbol{\Theta}(\mathbf{C}, \boldsymbol{\beta}, \mathbf{O}^{(k)}),$$

where $\rho(\mathbf{C};\lambda)$ is a sparsity-inducing penalty function with tuning parameter λ, and $\mathbf{O}^{(k)} = \mathbf{O} + \mathbf{X}\sum_{i=2}^{k}\widehat{\mathbf{C}}_{i-1}$ with $\mathbf{O}^{(1)} = \mathbf{O}$ (the original offset matrix). We refer the general problem (10) as negative binomial co-sparse unit-rank estimation, NB-CURE($\mathbf{C},\boldsymbol{\beta},\boldsymbol{\Phi};\mathbf{Y},\mathbf{X},\mathbf{Z},\mathbf{O}^{(k)},\rho$). Denote the estimate of the unit-rank component of $\mathbf{C}$ as $\widehat{\mathbf{C}}_k = \widehat{d}_k\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^{\mathrm{T}}$. We remark that the low-dimensional parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Phi}$ are re-estimated at the intermediate steps, and their final estimates are obtained from the last step.

We use the elastic net penalty and its adaptive version [Zou and Hastie, 2005, Zou and Zhang, 2009, Mishra et al., 2017], i.e., for the kth step,

$$\rho(\mathbf{C};\lambda) = \rho(\mathbf{C};\mathbf{W},\lambda,\alpha) = \alpha\lambda\|\mathbf{W}\circ\mathbf{C}\|_1 + (1-\alpha)\lambda\|\mathbf{C}\|_F^2. \tag{11}$$

Here $\|\cdot\|_1$ denotes the ℓ_1 norm, the operator "$\circ$" stands for the Hadamard product, $\mathbf{W} = [w_{ij}]_{p\times q}$ is a pre-specified weighting matrix, λ is a tuning parameter controlling the overall amount of regularization, and $\alpha \in (0,1)$ controls the relative weights between the two penalty terms. Several other penalties, such as the lasso ($\alpha = 1,\gamma = 0$), the adaptive lasso ($\alpha = 1,\gamma > 0$), and the elastic net ($0 < \alpha < 1$, $\gamma = 0$), are its special cases. In the kth step of NBFAR, we let $\mathbf{W}_k = |\widetilde{\mathbf{C}}_k|^{-\gamma}$, where $\gamma = 1$ and $\widetilde{\mathbf{C}}_k = \widetilde{d}_k\widetilde{\mathbf{u}}_k\widetilde{\mathbf{v}}_k^{\mathrm{T}}$ is an initial estimate of $\mathbf{C}_k$. As such, $w_{ijk} = w_k^{(d)}w_{ik}^{(u)}w_{jk}^{(v)}$, with

$$w_k^{(d)} = |\widetilde{d}_k|^{-\gamma}, \mathbf{w}_k^{(u)} = [w_{1k}^{(u)},...,w_{pk}^{(u)}]^{\mathrm{T}} = |\widetilde{\mathbf{u}}_k|^{-\gamma}, \mathbf{w}_k^{(v)} = [w_{1k}^{(v)},...,w_{qk}^{(v)}]^{\mathrm{T}} = |\widetilde{\mathbf{v}}_k|^{-\gamma}. \tag{12}$$

To streamline the presentation, for now let us assume that we are able to solve

NB-CURE and select the tuning parameter λ suitably. In each step, through the construction of the offset term, the regression effects from the previous steps are adjusted or "deflated" in order to enable NB-CURE to target a new unit-rank component. The procedure terminates after a pre-specified number of steps or when $\widehat{d}_k$ is estimated to be zero.

3.2.2 Computation

Compared to lasso, a small amount of ridge penalty in the elastic net allows correlated predictors to be in or out of the model together, thereby improving the convexity of the problem and enhancing the stability of optimization [Zou and Hastie, 2005, Mishra et al., 2017]; in our work, we fix $\alpha = 0.95$ and write $\rho(\mathbf{C}; \mathbf{W}, \lambda, \alpha) = \rho(\mathbf{C}; \mathbf{W}, \lambda)$ for simplicity. Now we express the general form of NB-CURE$(\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}; \mathbf{Y}, \mathbf{Z}, \mathbf{X}, \mathbf{O}, \rho)$ as

$$(\hat{d}, \widehat{\mathbf{u}}, \widehat{\mathbf{v}}, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Phi}}) \equiv \underset{\mathbf{u},\mathbf{d},\mathbf{v},\boldsymbol{\beta},\boldsymbol{\Phi}}{\arg\min} \Big\{ F_\lambda(d, \mathbf{u}, \mathbf{v}, \boldsymbol{\beta}, \boldsymbol{\Phi}) = \mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) + \rho(\mathbf{C}; \mathbf{W}, \lambda) \Big\}, \qquad (13)$$
$$\text{s.t.} \quad \mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}},\ \mathbf{u}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u}/n = \mathbf{v}^{\mathrm{T}}\mathbf{v} = 1, \boldsymbol{\Theta} = \boldsymbol{\Theta}(\mathbf{C}, \boldsymbol{\beta}, \mathbf{O}),$$

where $\mathbf{W} = w^{(d)}\mathbf{w}^{(u)}\mathbf{w}^{(v)\mathrm{T}}$. Following the estimation procedure of NBRRR model, we solve the problem using tthhe framework of BSUM approach [Razaviyayn et al., 2013] that cycles between $\mathbf{u}$-step, $\mathbf{v}$-step, $\boldsymbol{\beta}$-step and $\boldsymbol{\Phi}$-step to update the parameters in block of $(\mathbf{u}, d)$, $(\mathbf{v}, d)$, $\boldsymbol{\beta}$ and $\boldsymbol{\Phi}$, respectively, until convergence.

u-step For fixed $\{\mathbf{v}, \boldsymbol{\beta}, \boldsymbol{\Phi}\}$ with $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$, we rewrite the objective function (13) in terms of the product variable $\check{\mathbf{u}} = d\mathbf{u}$ to avoid the quadratic constraints. For simplicity, we write $\boldsymbol{\Theta}$ as function of $\check{\mathbf{u}}$ as $\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}})$. In terms of $\check{\mathbf{u}}$, let us assume that the $\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}}), \boldsymbol{\Phi})$ is L-Lipschitz continuous gradient for some s_u. Following the **C**-step in Section 3.1, the optimization problem to update $\check{\mathbf{u}}$ is given by

$$\widehat{\mathbf{u}} \equiv \arg\min_{\mathbf{a}} \mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}}), \boldsymbol{\Phi}) + \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}}), \boldsymbol{\Phi}))^{\mathrm{T}}(\mathbf{a} - \check{\mathbf{u}}) + \frac{s_u}{2}\|\check{\mathbf{u}} - \mathbf{a}\|^2 + \rho(\mathbf{a}\mathbf{v}^{\mathrm{T}}; \mathbf{W}, \lambda).$$

Following Zou and Hastie [2005], the unique optimal solution is given by

$$\widehat{\mathbf{u}} = \frac{\mathbf{S}(\check{\mathbf{u}} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}}), \boldsymbol{\Phi})/s_u; \alpha\lambda\mathbf{v}^{\mathrm{T}}\mathbf{w}^{(v)}w^{(d)}\mathbf{w}^{(u)}/s_u)}{1 + 2\lambda(1-\alpha)\|\mathbf{v}\|_2^2/s_u}, \tag{14}$$

where $\mathbf{S}(\mathbf{t}; \tilde{\lambda}) = \mathrm{sign}(\mathbf{t})(|\mathbf{t}| - \tilde{\lambda})_+$ is the elementwise soft-thresholding operator on any $\mathbf{t} \in \mathbb{R}^p$. Now, using the equality constraint, i.e., $\|\mathbf{X}\mathbf{u}\|_2 = \sqrt{n}$, we can retrieve the individual estimates of $(d, \mathbf{u})$ from $\widehat{\mathbf{u}}$.

v-step For fixed $\{\mathbf{u}, \boldsymbol{\beta}, \boldsymbol{\Phi}\}$, we write As in the **u**-step, we rewrite the objective function (13) in terms of the product $\check{\mathbf{v}} = d\mathbf{v}$. For simplicity, we write $\boldsymbol{\Theta}(\mathbf{C}, \boldsymbol{\beta}, \mathbf{O})$ as $\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}})$. Following the **u**-step, the optimization problem to update $\check{\mathbf{v}}$ is given by

$$\widehat{\mathbf{v}} \equiv \arg\min_{b} \mathcal{L}(\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}}), \boldsymbol{\Phi}) + \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}}), \boldsymbol{\Phi})^{\mathrm{T}}(\mathbf{b} - \check{\mathbf{v}}) + \frac{s_v}{2}\|\check{\mathbf{v}} - \mathbf{b}\|^2 + \rho(\mathbf{u}\mathbf{b}^{\mathrm{T}}; \mathbf{W}, \lambda), \tag{15}$$

where $\mathcal{L}(\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}}), \boldsymbol{\Phi})$ is L-Lipschitz continuous gradient for some s_v. Following the $\mathbf{u}$-step, the unique optimal solution is given by

$$\widehat{\mathbf{v}} = \frac{\mathbf{S}(\check{\mathbf{v}} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}}), \boldsymbol{\Phi})/s_v; \alpha\lambda\mathbf{u}^{\mathrm{T}}\mathbf{w}^{(u)}w^{(d)}\mathbf{w}^{(v)}/s_v}{1 + 2\lambda(1-\alpha)\|\mathbf{u}\|_2^2/s_v}. \tag{16}$$

Again, we retrieve the estimates of $(d, \mathbf{v})$ from the equality constraint $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$.

$\boldsymbol{\beta}$-step: For fixed $\mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}}$ and $\boldsymbol{\Phi}$, we denote $\boldsymbol{\Theta}$ as function of $\boldsymbol{\beta}$ as $\boldsymbol{\Theta}(\boldsymbol{\beta})$). Following $\boldsymbol{\beta}$-step of NBRRR in Section 3.1, we update the current estimate $\check{\boldsymbol{\beta}}$ by $\widehat{\boldsymbol{\beta}} = \check{\boldsymbol{\beta}} - \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\boldsymbol{\beta}}), \boldsymbol{\Phi})/s_b$.

$\boldsymbol{\Phi}$-step: For fixed $\mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}}$ and $\boldsymbol{\beta}$, we update $\boldsymbol{\Phi}$ by minimizing the negative log-likelihood function with respect to $\boldsymbol{\Phi}$, which can be obtained by a standard algorithm such as Newton-Raphson [R Core Team, 2019]; see Section 4.3 of the SM for more details.

We have relegated the details of the computation of $(s_u,\ s_v,\ s_b)$ and $(\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\mathbf{u}}\mathbf{v}^{\mathrm{T}}), \boldsymbol{\Phi}),\ \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\mathbf{u}\check{\mathbf{v}}^{\mathrm{T}}), \boldsymbol{\Phi}),\ \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\Theta}(\check{\boldsymbol{\beta}}), \boldsymbol{\Phi}))$ to the SM. We have summarized the computation procedure in Algorithm 2.

Algorithm 2 is a block coordinate descent optimization procedure for minimizing the nonsmooth and nonconvex objective functions $F_\lambda(d, \mathbf{u}, \mathbf{v}, \boldsymbol{\beta}, \boldsymbol{\Phi})$. This type of problem has been studied in, e.g., Gorski et al. [2007], Razaviyayn et al. [2013] and Mishra et al. [2017]. In each of the sub-problems, the algorithm minimizes a *convex* surrogate that majorizes the objective function, which results in a unique and bounded solution when the elastic net penalty is used [Mishra et al., 2017].

Algorithm 2 Negative Binomial Co-Sparse Factor Regression

Initialize: $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\Phi}^{(0)}$, and set the maximum number of steps $r \geqslant 1$, e.g., an upper bound of rank($\mathbf{C}$).
Set $s_b = \max_{1 \leqslant j \leqslant q} \|\mathbf{Z}^{\mathrm{T}} \mathrm{diag}(\mathbf{Y}_{\cdot j} + 1)\mathbf{Z}\|/2$.
for $k \leftarrow 1$ to r **do**
(1) Update offset: $\mathbf{O}^{(k)} = \mathbf{O} + \mathbf{X} \sum_{i=2}^{k} \widehat{\mathbf{C}}_{i-1}$
(2) Initialize: $\widetilde{\mathbf{C}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Phi}} = \mathrm{NBRRR}(\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}; \mathbf{Y}, \mathbf{Z}, \mathbf{X}, \mathbf{O}^{(k)}, 1)$ with $\widetilde{\mathbf{C}} = \tilde{d}\tilde{\mathbf{u}}\tilde{\mathbf{v}}^{\mathrm{T}}$.
(3) Set $\mathbf{u}^{(0)} = \tilde{\mathbf{u}}$, $\mathbf{v}^{(0)} = \tilde{\mathbf{v}}$, $d^{(0)} = \tilde{d}$, $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$, $\boldsymbol{\Phi}^{(0)} = \widetilde{\boldsymbol{\Phi}}$ and $\mathbf{W}$ using (12).
(4) Solve NB-CURE($\mathbf{C}, \boldsymbol{\beta}, \boldsymbol{\Phi}; \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{O}^{(k)}, \rho$) such that $\mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}}$.
repeat

$$s_u = \|\mathbf{X}^{\mathrm{T}}\mathbf{X} + \sum_{i=1}^{n} \mathbf{x}_i \Big(\sum_{k=1}^{q} y_{ik} v_k^{(t)2} \Big) \mathbf{x}_i^{\mathrm{T}} \|/2, \;\; s_v = \frac{\max_{1 \leqslant j \leqslant q} \mathbf{u}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathrm{diag}(\mathbf{Y}_{\cdot j} + 1)\mathbf{X}\mathbf{u}}{2}$$

(I) $\mathbf{u}$-step: Set $\check{\mathbf{u}} = d^{(t)}\mathbf{u}^{(t)}$ and $\mathbf{v} = \mathbf{v}^{(t)}$. Update $\check{\mathbf{u}}^{(t+1)}$ using (14). Recover block variable $(\tilde{d}^{(t+1)}, \mathbf{u}^{(t+1)})$ using equality constraint in (13).
(II) $\mathbf{v}$-step: Set $\check{\mathbf{v}} = \tilde{d}^{(t+1)}\mathbf{v}^{(t)}$ and $\mathbf{u} = \mathbf{u}^{(t+1)}$. Update $\check{\mathbf{v}}^{(t+1)}$ using (16). Recover block variable $(d^{(t+1)}, \mathbf{v}^{(t+1)})$ using equality constraint in (13).
(III) $\boldsymbol{\beta}$-step: $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \frac{1}{s_b} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}^{(t+1)}, \boldsymbol{\beta}^{(t)}), \boldsymbol{\Phi})$ where $\mathbf{C}^{(t+1)} = d^{(t+1)}\mathbf{u}^{(t+1)}\mathbf{v}^{(t+1)\mathrm{T}}$.
(IV) $\boldsymbol{\Phi}$-step: $\boldsymbol{\Phi}^{(t+1)} = \arg\min_{\boldsymbol{\Phi}} \mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}), \boldsymbol{\Phi})$.
$t \leftarrow t + 1$.
until convergence, e.g., the relative ℓ_2 change in parameters is less than $\epsilon = 10^{-6}$.
(6) $\widehat{\mathbf{u}}_k = \widehat{\mathbf{u}}$, $\widehat{d}_k = \widehat{d}$, $\widehat{\mathbf{v}}_k = \widehat{\mathbf{v}}$, $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\Phi}} = \widehat{\boldsymbol{\Phi}}$ and $\widehat{\mathbf{C}}_k = \widehat{d}_k \widehat{\mathbf{u}}_k \widehat{\mathbf{v}}_k^{\mathrm{T}}$.
if $\widehat{d}_k = 0$ **then**
Set $\widehat{r} = k$; $k \leftarrow r$;
end if
end for
return $\widehat{\mathbf{C}} = \sum_{k=1}^{\widehat{r}} \widehat{\mathbf{C}}_k$, $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\Phi}}$.

Thus, using Theorem/Corollary 2(a) of Razaviyayn et al. [2013], we can conclude that any limit point of the sequence of solutions generated by the algorithm is a coordinate-wise minimum of the objective function. Algorithm **??** always converges in our extensive numerical studies. Both of the estimation procedures for GOFAR are implemented, tested, validated, and made publicly available in a user-friendly R package, gofar.

3.3 Tuning

4 Appendix

4.1 Partial derivative of $\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi})$ for fixed $\mathbf{\Phi}$:

For the ease of notation, we write $\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi})$ as $\mathcal{L}(\mathbf{\Theta})$. Here $\mathbf{\Theta}$ is linked to linear predictor $\boldsymbol{\eta}$ as $g(b'(\mathbf{\Theta})) = \boldsymbol{\eta}$ and

$$\begin{aligned}
&\mathcal{L}(\mathbf{\Theta}) = -\operatorname{tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{\Theta}) + \operatorname{tr}(\mathbf{J}^{\mathrm{T}}\mathbf{B}(\mathbf{\Theta})) + \text{const},\\
&\frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \theta_{ij}} = -y_{ij} + b'(\theta_{ij}) \implies \frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \eta_{ij}} = (-y_{ij} + b'(\theta_{ij}))\frac{\partial \eta_{ij}}{\partial \theta_{ij}}.
\end{aligned}$$

For $g(x) = \log x$, we have $\frac{\partial \eta_{ij}}{\partial \theta_{ij}} = \frac{\phi_j}{\phi_j + \exp \eta_{ij}}$ and implies $\frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \eta_{ij}} = \phi_j\left(\frac{-y_{ij} + \exp \eta_{ij}}{\phi_j + \exp \eta_{ij}}\right)$. Jointly we represent

$$\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\eta}) = \frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \boldsymbol{\eta}} = -\mathbf{Y} \circ \mathbf{B}^1(\boldsymbol{\eta}) + \mathbf{B}^2(\boldsymbol{\eta}),$$

where $\mathbf{B}^1(\boldsymbol{\eta}) = [b_1(\eta_{ij})]_{n\times q}$ with $b_1(\eta_{ij}) = \frac{\phi_j}{\phi_j+\exp\eta_{ij}}$ and $\mathbf{B}^2(\boldsymbol{\eta}) = [b_2(\eta_{ij})]_{n\times q}$ with $b_2(\eta_{ij}) = \frac{\phi_j \exp\eta_{ij}}{\phi_j+\exp\eta_{ij}}$. Then, we compute $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}} = \mathbf{Z}^{\mathrm{T}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}}$ and $\frac{\partial\mathcal{L}}{\partial\mathbf{C}} = \mathbf{X}^{\mathrm{T}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}}$. For $\mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}}$, $\frac{\partial\mathcal{L}}{\partial\mathbf{u}} = \mathbf{X}^{\mathrm{T}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}}\mathbf{v}$ and $\frac{\partial\mathcal{L}}{\partial\mathbf{v}} = \left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}}\right)^{\mathrm{T}}\mathbf{X}\mathbf{u}$.

4.2 Computation of s_c, s_b, s_u and s_v

For the ease of notation, we write $\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ as $\mathcal{L}(\boldsymbol{\Theta})$. We define s_c to be an upper bound on

$$\sup_{\mathbf{C}} \|\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}))\| = \max_{1\leqslant j\leqslant q}\sup_{\mathbf{C}_{\cdot j}} \|\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}_{\cdot j}))\|,$$

where $\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{C}_{\cdot j})) = \mathbf{X}^{\mathrm{T}}\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\eta}_{\cdot j})\mathbf{X}$. To proceed with the analysis, we simplify and bound $\boldsymbol{\nabla}^2\mathcal{L}(\eta_{ij})$ as

$$\boldsymbol{\nabla}^2\mathcal{L}(\eta_{ij}) = (y_{ij}+1)\frac{\phi_j\exp\eta_{ij}}{(\phi_j+\exp\eta_{ij})^2} \leqslant \frac{y_{ij}+1}{4}.$$

This implies $s_c = \max_{1\leqslant j\leqslant q}\|\mathbf{X}^{\mathrm{T}}\mathrm{diag}(\mathbf{Y}_{\cdot j}+1)\mathbf{X}\|/2$. Similarly, we compute $s_b = \max_{1\leqslant j\leqslant q}\|\mathbf{Z}^{\mathrm{T}}(\mathbf{Y}_{\cdot j}+1)\mathbf{Z}\|/2$.

Now, we analyze $\sup_{\mathbf{a}}\|\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{a}))\|$ to compute s_u. To begin with, we have $\frac{\partial\mathcal{L}}{\partial\mathbf{u}} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\eta})\mathbf{v} = \sum_{i=1}^n \mathbf{x}_i\sum_{k=1}^q \boldsymbol{\nabla}\mathcal{L}(\eta_{ik})v_k$. This implies $\frac{\partial^2\mathcal{L}}{\partial\mathbf{u}\partial\mathbf{u}^{\mathrm{T}}} = \sum_{i=1}^n \mathbf{x}_i\Big(\sum_{k=1}^q \boldsymbol{\nabla}^2\mathcal{L}(\eta_{ik})v_k^2\Big)\mathbf{x}_i^{\mathrm{T}}$. Hence,

$$s_u = \|\sum_{i=1}^n \mathbf{x}_i\Big(\sum_{k=1}^q (y_{ik}+1)v_k^2\Big)\mathbf{x}_i^{\mathrm{T}}\|/2 = \|\mathbf{X}^{\mathrm{T}}\mathbf{X} + \sum_{i=1}^n \mathbf{x}_i\Big(\sum_{k=1}^q y_{ik}v_k^2\Big)\mathbf{x}_i^{\mathrm{T}}\|/2.$$

Now, we analyze $\sup_{\mathbf{b}}\|\boldsymbol{\nabla}^2\mathcal{L}(\boldsymbol{\Theta}(\mathbf{b}))\|$ to compute s_v. To begin with, we have $\frac{\partial\mathcal{L}}{\partial\mathbf{v}} =$

$\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\eta}}\right)^{\mathrm{T}} \mathbf{X}\mathbf{u} = \sum_{i=1}^{n} \mathbf{x}_i^{\mathrm{T}} \mathbf{u} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\eta}_{i.})$. This implies

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} = \mathrm{diag}[\mathbf{u}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathrm{diag}(\boldsymbol{\nabla}^2 \mathcal{L}(\boldsymbol{\eta}_{.1})) \mathbf{X}\mathbf{u}, \ldots, \mathbf{u}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathrm{diag}(\boldsymbol{\nabla}^2 \mathcal{L}(\boldsymbol{\eta}_{.q})) \mathbf{X}\mathbf{u}].$$

. Then

$$s_v = \max_{1 \leqslant j \leqslant q} \mathbf{u}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathrm{diag}(\boldsymbol{\nabla}^2 \mathcal{L}(\boldsymbol{\eta}_{.j})) \mathbf{X}\mathbf{u}/2 = \max_{1 \leqslant j \leqslant q} \mathbf{u}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathrm{diag}[\mathbf{Y}_{.j} + 1] \mathbf{X}\mathbf{u}/2.$$

4.3 Newton-Raphson to update $\boldsymbol{\Phi}$:

Given $\mathbf{C}$ and $\boldsymbol{\beta}$, $\boldsymbol{\mu} = g^{-1}(\mathbf{O} + \mathbf{X}\mathbf{C} + \mathbf{Z}\boldsymbol{\beta})$ is fixed. The first and the second derivative $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Phi})$ (2) for any jth dispersion parameter ϕ_j is given by

$$\begin{aligned}
-\frac{\partial \mathcal{L}}{\partial \phi_j} &= \sum_{i=1}^{n} \left[\left(\sum_{k=0}^{y_{ij}-1} \frac{1}{k + \phi_j} \right) + \frac{\mu_{ij} - y_{ij}}{\mu_{ij} + \phi_j} - \log\left(1 + \frac{\mu_{ij}}{\phi_j}\right) \right] \\
-\frac{\partial^2 \mathcal{L}}{\partial \phi_j^2} &= \sum_{i=1}^{n} \left[\left(\sum_{k=0}^{y_{ij}-1} \frac{-1}{(k + \phi_j)^2} \right) + \frac{y_{ij} - \mu_{ij}}{(\mu_{ij} + \phi_j)^2} + \frac{\mu_{ij}}{\phi_j(\phi_j + \mu_{ij})} \right].
\end{aligned}$$

Then we update the current estimate ϕ_j by $\hat{\phi}_j = \phi_j - \left(\frac{\partial^2 \mathcal{L}}{\partial \phi_j^2}\right)^{-1} \frac{\partial \mathcal{L}}{\partial \phi_j}$.

Reparameterized model in terms of α; n [Ismail and Jemain, 2007]

$$p(y_{ij}; \mu_{ij}, \phi_j) = \mathrm{NB}(y_{ij}; \mu_{ij}^*, \phi_j^*) = \binom{y_{ij} + \phi_j^* - 1}{y_{ij}} \frac{{\mu_{ij}^*}^{y_{ij}} {\phi_j^*}^{\phi_j^*}}{(\mu_{ij}^* + \phi_j^*)^{y_{ij} + \phi_j^*}},$$

$$\mathcal{L} = \sum_{i=1}^{n}\left[\Big(\sum_{k=0}^{y_{ij}-1}\log\frac{k\alpha_j+1}{\alpha_j}\Big) + y_{ij}\log\alpha_j - \Big(y_{ij}+\frac{1}{\alpha_j}\Big)\log(1+\mu_{ij}\alpha_j)\right]$$

$$\frac{\partial\mathcal{L}}{\partial\alpha_j} = \sum_{i=1}^{n}\left[\Big(\sum_{k=0}^{y_{ij}-1}\frac{k}{k\alpha_j+1}\Big) - \frac{\mu_{ij}}{\alpha_j}\frac{y_{ij}\alpha_j+1}{\mu_{ij}\alpha_j+1} + \frac{1}{\alpha_j^2}\log\Big(1+\mu_{ij}\alpha_j\Big)\right]$$

$$\frac{\partial^2\mathcal{L}}{\partial\alpha_j^2} = \sum_{i=1}^{n}\left[\Big(\sum_{k=0}^{y_{ij}-1}\frac{-k^2}{(k\alpha_j+1)^2}\Big) + \frac{\mu_{ij}}{\alpha_j^2(1+\alpha_j\mu_{ij})} - \frac{2}{\alpha_j^3}\log\Big(1+\mu_{ij}\alpha_j\Big)\right.$$
$$\left. - \frac{\mu_{ij}y_{ij}}{\alpha_j(\mu_{ij}\alpha_j+1)} + \frac{\mu_{ij}(y_{ij}\alpha_j+1)(2\mu_{ij}\alpha_j+1)}{\alpha_j^2(\mu_{ij}\alpha_j+1)^2}\right]$$

$$\frac{\partial^2\mathcal{L}}{\partial\alpha_j^2} = \sum_{i=1}^{n}\left[\Big(\sum_{k=0}^{y_{ij}-1}\frac{-k^2}{(k\alpha_j+1)^2}\Big) - \frac{2}{\alpha_j^3}\log\Big(1+\mu_{ij}\alpha_j\Big)\right.$$
$$\left. + \frac{\mu_{ij}(1-y_{ij}\alpha_j)(\mu_{ij}\alpha_j+1) + \mu_{ij}(y_{ij}\alpha_j+1)(2\mu_{ij}\alpha_j+1)}{\alpha_j^2(\mu_{ij}\alpha_j+1)^2}\right].$$

$$\frac{\partial^2\mathcal{L}}{\partial\alpha_j^2} = \sum_{i=1}^{n}\left[\Big(\sum_{k=0}^{y_{ij}-1}\frac{-k^2}{(k\alpha_j+1)^2}\Big) - \frac{2}{\alpha_j^3}\log\Big(1+\mu_{ij}\alpha_j\Big)\right.$$
$$\left. + \frac{\mu_{ij}[2+\mu_{ij}\alpha_j(3+y_{ij}\alpha_j)]}{\alpha_j^2(\mu_{ij}\alpha_j+1)^2}\right].$$

$$\begin{aligned}\text{Residual deviance} &= 2(\text{loglik(saturated)} - \text{loglik(proposed)})\\ &= 2(\mathcal{L}(\mathbf{Y};\mathbf{Y},\boldsymbol{\Phi}) - \mathcal{L}(\mathbf{Y};\boldsymbol{\mu},\boldsymbol{\Phi}))\\ &= 2\left[\sum_i\sum_j y_{ij}\log\frac{y_{ij}}{\mu_{ij}} - (y_{ij}+\phi_j)\log\left(\frac{\phi_j+y_{ij}}{\phi_j+\mu_{ij}}\right)\right]\end{aligned}$$

References

Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

Noriszura Ismail and Abdul Aziz Jemain. Handling overdispersion with negative binomial and generalized poisson regression models. In *Casualty actuarial society forum*, volume 2007, pages 103–58. Citeseer, 2007.

Chongliang Luo, Jian Liang, Gen Li, Fei Wang, Changshui Zhang, Dipak K Dey, and Kun Chen. Leveraging mixed and incomplete outcomes via reduced-rank modeling. *Journal of Multivariate Analysis*, 167:378–394, 2018.

Aditya Mishra, Dipak K Dey, and Kun Chen. Sequential co-sparse factor regression. *Journal of Computational and Graphical Statistics*, 26(4):814–825, 2017.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL `https://www.R-project.org/`.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

Achim Zeileis, Christian Kleiber, and Simon Jackman. Regression models for count data in r. *Journal of statistical software*, 27(8):1–25, 2008.

Hui Zou and Trevor J. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00503.x.

Hui Zou and Hao Helen Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37(4):1733, 2009.