Notes: MPLN

immediate

September 21, 2023

1 Data Structure

Let Y_i denote the binary phenotype of interest, and X_{ij} represents the counts of jth feature (e.g. ICD code, NLP mentions, etc) of ith patient, $\boldsymbol{X} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\intercal}$ and $\boldsymbol{X}_i \in \mathbb{R}^p$, U_i denotes the healthcare utilization of patient i. Our data consist of n independent and identically distributed (iid) labelled data $\mathcal{L} = \{(\boldsymbol{X}_i, Y_i, U_i) : i \in \mathcal{I}_L\}$ where $n = |\mathcal{I}_L|$, and N iid unlabelled data $\mathcal{U} = \{(\boldsymbol{X}_i, U_i) : i \in \mathcal{I}_U\}$ where $N = |\mathcal{I}_U|$. We assume that the labelling mechanism is completely at random.

2 Model

We propose to model the EHR data using a mixture of multivariate Poisson-LogNormal (MPLN)[Aitchison and Ho, 1989, Subedi and Browne, 2020], with

$$\begin{aligned}
&\boldsymbol{X}_i \mid \boldsymbol{Z}_i \sim Poisson(e^{\boldsymbol{Z}_i}) \quad , \quad X_{ij} \mid Z_{ij} \ indep. \\
&\boldsymbol{Z}_i \mid (Y_i = y) \sim \mathcal{N}(\boldsymbol{D}_i^{(\mathrm{y})\intercal} \boldsymbol{B}_{d\times p}, \boldsymbol{\Sigma}_{p\times p}) \quad , \\
&Y_i \sim Ber(\pi) \quad iid
\end{aligned}$$

where $\boldsymbol{D}_i^{(\mathrm{y})} = (1, U_i, y)^{\intercal}$. In the model, $\boldsymbol{Z}_i \in \mathbb{R}^p$ represents the patient-level embedding for ith patient, and B_{2j} can be interpreted as the importance of the jth feature to the phenotype of interest, adjusting for healthcare utilization. $\boldsymbol{\Sigma}_{ij}$ describes the underlying relationships between the ith and jth concept. Due to the similarity and relatedness among p concepts, we assume $\boldsymbol{\Sigma}$ has a low rank. We decompose $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\intercal}$, where $\boldsymbol{V} \in \mathbb{R}^{p\times q}$ is orthonormal and the rank $q \leq p$, $\boldsymbol{\Lambda} \in \mathbb{R}^{q\times q}$ is diagonal. Assuming that across different institutions, the orientation/rotation ($\boldsymbol{V}$) is the same but the shape/scaling $\boldsymbol{\Lambda}$ can be different, we may obtain $\boldsymbol{V}$ by applying singular value decomposition (SVD) to the feature-level embedding matrix $\boldsymbol{E}$ and obtain $\boldsymbol{V} \in \mathbb{R}^{p\times q}$ as the first q left singular vectors. In that case, instead of estimating the full covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$, we only need to estimate the diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{q \times q}$, reducing the number of parameters to be estimated from pq to q. Alternatively, we can write the model as:

$$\begin{aligned}
&\boldsymbol{X}_i \mid \boldsymbol{Z}_i \sim Poisson(e^{\boldsymbol{Z}_i}) \quad , \quad X_{ij} \mid Z_{ij}\ indep. \\
&\boldsymbol{Z}_i \mid (Y_i = y) = \boldsymbol{D}_i^{(\mathrm{y})\top} \boldsymbol{B} + \boldsymbol{V} \mathbf{\Lambda}^{1/2} \boldsymbol{W}_i \\
&\boldsymbol{W}_i \sim \mathcal{N}(0_q, \boldsymbol{I}_{q \times q}) \quad iid, \quad Y_i \sim Ber(\pi) \quad iid
\end{aligned} \tag{1}$$

For model estimation, we first consider the supervised scenario, i.e. estimation of the model with $\mathcal{L}$. With Y_i known for $i \in \mathcal{I}_L$, the complete-data log-likelihood can be written as

$$\begin{aligned}
\log p(\boldsymbol{X}_L, \boldsymbol{W}_L, \boldsymbol{Y}_L; \boldsymbol{B}, \mathbf{\Lambda}, \pi) &= \sum_{y=0}^{1} \sum_{i \in \mathcal{I}_L} I(Y_i = y) \left\{ \log p(\boldsymbol{X}_i \mid \boldsymbol{W}_i, U_i, y; \boldsymbol{B}, \mathbf{\Lambda}, \pi) + \log p(\boldsymbol{W}_i) + \log p(y; \pi) \right\} \\
&= \sum_{y=0}^{1} I(Y_i = y) \Big[\mathbf{1}_n^\top \left\{ \boldsymbol{X}_L \odot (\boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{W}_L \mathbf{\Lambda}^{1/2} \boldsymbol{V}^\top) - \exp(\boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{W}_L \mathbf{\Lambda}^{1/2} \boldsymbol{V}^\top) \right\} \mathbf{1}_p \\
&\quad - \frac{||\boldsymbol{W}_L||_F^2}{2} - \frac{nq}{2} \log(2\pi) - \mathbf{1}_n^\top \log(\boldsymbol{X}_L!) \mathbf{1}_p + ny \log \pi + n(1-y) \log(1-\pi) \Big].
\end{aligned}$$

Since $\boldsymbol{W}_L$ is also latent, the evaluations of the log-likelihood of the observed data

$$\ell_L(\boldsymbol{B}, \mathbf{\Lambda}, \pi) = \log p(\boldsymbol{X}_L, \boldsymbol{Y}_L; \boldsymbol{B}, \mathbf{\Lambda}, \pi) = \log \int p(\boldsymbol{X}_L, \boldsymbol{W}_L, \boldsymbol{Y}_L; \boldsymbol{B}, \mathbf{\Lambda}, \pi) d\boldsymbol{W}_L$$

is intractable, as well as its maximization with respect to $(\boldsymbol{B}, \mathbf{\Lambda})$. In this setting, the most popular strategy to perform maximum likelihood is to use the EM algorithm [Moon, 1996], which requires the evaluation of $E_{W|X}\{\log p(\boldsymbol{X}_L, \boldsymbol{W}_L; \boldsymbol{B}, \mathbf{\Lambda}, \pi)\}$. This is challenging as it requires at least the first and second order of $p(\boldsymbol{W}_i | \boldsymbol{X}_i)$ which are unknown in general and hard to compute. Karlis [2005] and Silva et al. [2019] suggest achieving this task via numerical or Monte-Carlo integration, but this approach is too computationally demanding when dealing with even a moderate number of p. To circumvent this issue, we resort instead to a variational strategy and integrate out W under a tractable approximation of $p(\boldsymbol{W} | \boldsymbol{X})$.

2.1 Variational Approximation

Variational approximation [Wainwright et al., 2008] is an approximate inference technique which has been very popular in machine learning. It presents an alternative parameter estimation framework by using a computationally convenient approximating density in place of a more complex posterior density. Using computationally convenient Gaussian densities, complex posterior distributions are approximated by minimizing the Kullback-Leibler (KL) divergence between the true and the approximating densities and therefore, reducing the computational overhead.

Instead of directly maximizing the log likelihood, we maximize the evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{J}(\boldsymbol{B}, \boldsymbol{\Lambda}, q) &= \log p(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{B}, \boldsymbol{\Lambda}) - KL[q(\boldsymbol{W}), p(\boldsymbol{W} \mid \boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{B}, \boldsymbol{\Lambda}))] \\ &= E_q[\log p(\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}; \boldsymbol{B}, \boldsymbol{\Lambda})] - E_q[\log q(\boldsymbol{W})]\end{aligned}$$

In our variational approximation, we choose the set $\mathcal{Q}$ of product distribution of q-dimensional multivariate Gaussian with diagonal covariance matrices:

$$\mathcal{Q} = \left\{ q : q(\boldsymbol{W}) = \prod_i q_i(\boldsymbol{W}_i), q_i(\boldsymbol{W}_i) = \mathcal{N}(\boldsymbol{W}_i; \boldsymbol{m}_i, diag(\boldsymbol{s}_i \circ \boldsymbol{s}_i))), (\boldsymbol{m}_i, \boldsymbol{s}_i) \in \mathbb{R}^q \times \mathbb{R}_+^q \right\}$$

In the supervised setting, let $\boldsymbol{M}_L = (\boldsymbol{m}_1, \cdots, \boldsymbol{m}_n)^\intercal \in \mathbb{R}^{n \times q}$ and $\boldsymbol{S}_L = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n)^\intercal \in \mathbb{R}^{n \times q}$. Let $\boldsymbol{\Theta} = \{\boldsymbol{B}, \boldsymbol{\Lambda}, \pi\}$. The ELBO on the labeled set can be derived as

$$\begin{aligned}\mathcal{J}_L(\boldsymbol{\Theta}, \boldsymbol{M}_L, \boldsymbol{S}_L; \boldsymbol{X}_L, \boldsymbol{D}_L) = \sum_{y=0}^{1} I(\boldsymbol{Y}_L = y) \Bigg\{ & \mathbf{1}_n^\intercal \left[\boldsymbol{X}_L \odot (\boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{M}_L \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal) - \mathbb{E}_q \left\{ \exp(\boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{W}_L \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal) \right\} \right] \mathbf{1} \\ & - \frac{1}{2} \mathbf{1}_n^\intercal \left\{ \boldsymbol{M}_L \odot \boldsymbol{M}_L + \boldsymbol{S}_L \odot \boldsymbol{S}_L - 2 \log(\boldsymbol{S}_L) - \mathbf{1}_{n \times q} \right\} \mathbf{1}_q - \mathbf{1}_n^\intercal \log(\boldsymbol{X}_L!) \mathbf{1}_p \\ & + ny \log \pi + n(1-y) \log(1-\pi) \Bigg\},\end{aligned} \tag{2}$$

where

$$\mathbb{E}_q \left\{ \exp(\boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{W}_L \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal) \right\} = \exp \left\{ \boldsymbol{D}_L^{(\mathrm{y})} \boldsymbol{B} + \boldsymbol{M}_L \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal + \frac{1}{2} (\boldsymbol{S}_L \odot \boldsymbol{S}_L)(\boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal \odot \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^\intercal) \right\} =: \boldsymbol{A}_L^{(\mathrm{y})}.$$

2.2 Supervised Setting

In the supervised setting, maximizing the ELBO leads to the variational EM (VEM) algorithm, which alternates between two steps until convergence: (I) VE step: update $(\boldsymbol{M}, \boldsymbol{S})$ keeping $(\boldsymbol{B}, \boldsymbol{\Lambda}^{1/2})$ fixed; (II) M step: update $(\boldsymbol{B}, \boldsymbol{\Lambda}^{1/2})$ keeping $(\boldsymbol{M}, \boldsymbol{S})$ fixed. Since the blockwise gradients can be easily derived, we may still use gradient-based local optimization algorithms for maximizing the ELBO, with block-wise gradients:

$$\begin{gathered}
\frac{\partial \mathcal{J}_L}{\partial \boldsymbol{M}_L} = (\boldsymbol{X}_L - \boldsymbol{A}_L)\boldsymbol{V}\boldsymbol{\Lambda}^{1/2} - \boldsymbol{M}_L \\
\frac{\partial \mathcal{J}_L}{\partial \boldsymbol{S}_L} = \boldsymbol{S}_L^{\oslash} - \boldsymbol{S}_L - \boldsymbol{A}_L(\boldsymbol{V}\boldsymbol{\Lambda}^{1/2} \odot \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}) \odot \boldsymbol{S}_L \\
\frac{\partial \mathcal{J}_L}{\partial \boldsymbol{B}} = (\boldsymbol{X}_L - \boldsymbol{A}_L)^{\intercal}\boldsymbol{D}_L \\
\frac{\partial \mathcal{J}_L}{\partial \boldsymbol{\Lambda}^{1/2}} = diag\left\{\boldsymbol{V}^{\intercal}\left[(\boldsymbol{X}_L - \boldsymbol{A}_L)^{\intercal}\boldsymbol{M}_L - \{\boldsymbol{A}_L^{\intercal}(\boldsymbol{S}\odot\boldsymbol{S})\} \odot \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\right]\right\}
\end{gathered} \tag{3}$$

3 Simulations

In the simulations, we fit the model in supervised setting as well as unsupervised and semi-supervised settings, but for the bias issue, we can first just focus on the supervised setting. That corresponds to *fixedV(sup)* in the simulations.

3.1 Data Generation

The datasets are generated as the following:

(i) Generate $\widetilde{\boldsymbol{\Sigma}} = AR_1(0.5)$. Find its eigen-decomposition $\widetilde{\boldsymbol{\Sigma}} = \boldsymbol{V}\widetilde{\boldsymbol{\Lambda}}\boldsymbol{V}^{\intercal}$. Let $\boldsymbol{\Lambda} = diag(seq(10, 5, length.out = q))$, and $\boldsymbol{\Sigma} = \boldsymbol{V}_{[,1:q]}\boldsymbol{\Lambda}_{[1:q,1:q]}\boldsymbol{V}_{[,1:q]}^{T}$.

(ii) For each patient, generate $Y_i \sim Ber(0.4)$, $U_i \sim Pois(10)$, let $\boldsymbol{D}_i = (1, \log(U_i + 1), Y_i)^{\intercal}$, $\boldsymbol{B} = (0\boldsymbol{1}_p, 0.1\boldsymbol{1}_p, (0.8\boldsymbol{1}_2^{\intercal}, 0.2\boldsymbol{1}_{18}^{\intercal}, 0.1\boldsymbol{1}_{30}^{\intercal}, -0.1\boldsymbol{1}_{50}^{\intercal}, 0\boldsymbol{1}_{p-100}^{\intercal}))$.

(iii) Generate $\boldsymbol{W}_i \sim \mathcal{N}(0_q, \boldsymbol{I}_{q\times q})$ and let $\boldsymbol{Z}_i = \boldsymbol{D}_i^{\intercal}\boldsymbol{B} + \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{W}_i$.

(iv) Generate $\boldsymbol{X}_i \sim Poisson(e^{\boldsymbol{Z}_i})$.

We then randomly choose n samples for which we assume Y_i is observed. To study the behavior of the optimization algorithm and assess our method's robustness to various model specifications, we varied the following generative parameters:

(a) The number of observed phenotype labels $n = 100, 200, 400$;

(b) The total number of patients $N = 5000, 10000, 20000$;

(c) The number of selected features $p = 100, 200, 400$;

(d) The rank $q = 5, 10, 20$.

3.2 Benchmark Methods

For comparison, we considered (1) *lowrank(sup)* supervised MPLN without known $\boldsymbol{V}$ adpated from Chiquet et al. [2018].(2) *fixedV(sup)*: supervised MPLN with known $\boldsymbol{V}$, (3) *lowrank(unsup)*: unsupervised MPLN without known $\boldsymbol{V}$, (4) *fixedV(unsup)*: unsupervised MPLN model with known $\boldsymbol{V}$, (5) *lowrank(semisup)*: semi-supervised MPLN model without known $\boldsymbol{V}$, (6) *fixedV(semisup)*: semi-supervised MPLN model with known $\boldsymbol{V}$. The differences among the models are summarized in table 1.

Method	**known $\boldsymbol{V}$**	**labels**	**$\mathcal{U}$ used**	**# training**	**# model params.**	**# var. params.**
lowrank(sup)	no	yes	no	n	$dp + pq$	$2nq$
fixedV(sup)	yes	yes	no	n	$dp + q$	$2nq$
lowrank(unsup)	no	no	yes	N	$dp + pq$	$2Nq$
fixedV(unsup)	yes	no	yes	N	$dp + q$	$2Nq$
lowrank(semisup)	no	yes	yes	$N + n$	$dp + pq$	$2(N + n)q$
fixedV(semisup)	yes	yes	yes	$N + n$	$dp + q$	$2(N + n)q$

Table 1: Comparison between different models.

3.3 Results

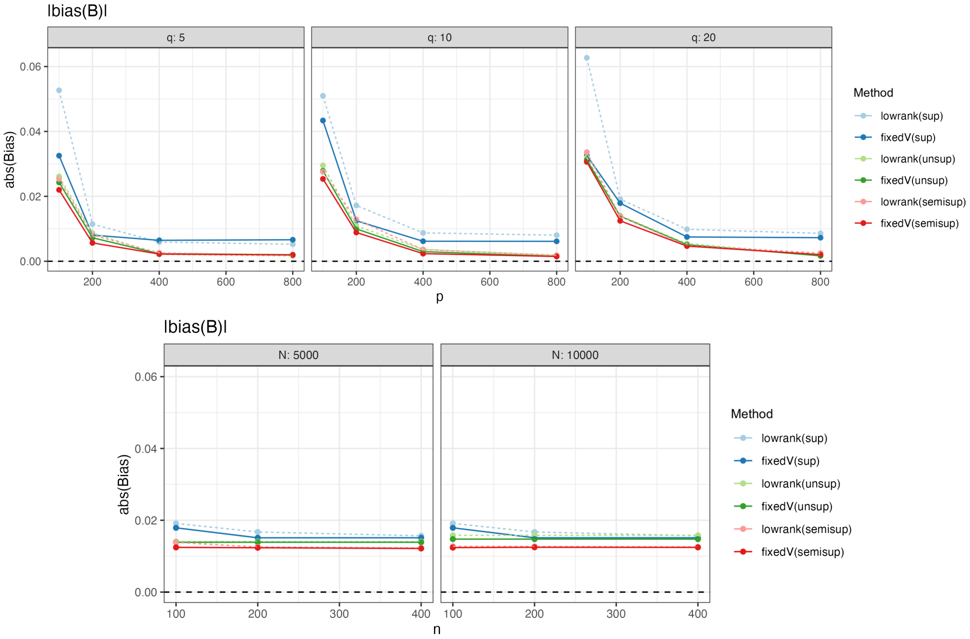

Figure 1: Top: varying p and q with $N = 5000$, $n = 100$. Bottom: varying N and n with $p = 200$, $q = 20$.

Parameter Estimates We studied how the bias of parameter estimates change with varying N, n, p, q. Figure 1(top) shows the bias of $\widehat{\boldsymbol{B}}_2$ (the coefficients corresponding to the difference in mean between two groups) with varying p and q when $N = 5000$, $n = 100$. As p increases, we observe decrease in bias for all estimators. Intuitively speaking, we are using p measures from each patient i to estimate the variational parameters $\boldsymbol{m}_i \in \mathbb{R}^q$ and $\boldsymbol{s}_i \in \mathbb{R}^q$. Figure 1(bottom) shows that the bias stabilizes after $n = 200$ and does not decrease further with increasing n or N, suggesting that the bias term is not dominated by the sample size.

Figure 2 shows $||\widehat{\boldsymbol{B}}_2 - \boldsymbol{B}_2||_2/||\boldsymbol{B}_2||_2$ and $||\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}||_F/||\boldsymbol{\Sigma}||_F$ with varying p and q with $N = 5000$, $n = 100$. We observe increase in the L_2-norm error of $\widehat{\boldsymbol{B}}_2$ for the supervised estimators, which is likely due to small sample size to estimate the growing number of parameters. The L_2-norm error of $\widehat{\boldsymbol{B}}_2$ for unsupervised and semi-supervised estimators stayed relatively stable with growing p. For $\widehat{\boldsymbol{\Sigma}}$, the relative F-norm error of $\widehat{\boldsymbol{\Sigma}}$ for methods without known $\boldsymbol{V}$ (dotted lines) unsurprisingly increases with larger p. By fixing $\boldsymbol{V}$ and reducing the number of parameters from pq to q, we can observe that (i) the relative F-norm error of $\widehat{\boldsymbol{\Sigma}}$ of the supervised estimator does not change with p as the it is likely dominated by n, and (ii) the relative F-norm error pf $\widehat{\boldsymbol{\Sigma}}$ of the unsupervised and semi-supervised estimators decrease with increasing p, since with large N the bias is likely dominated by estimation of the variational parameters. Comparing across different q, we see that both $||\widehat{\boldsymbol{B}}_2 - \boldsymbol{B}||_2/||\boldsymbol{B}||_2$ and $||\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}||_F/||\boldsymbol{\Sigma}||_F$ increases with larger q for all methods. This is as expected as the number of variational parameters to be estimated grows with the training size and q. In EHR settings, we expect q to be small relative to p, due to similarity and relatedness among the EHR concepts.

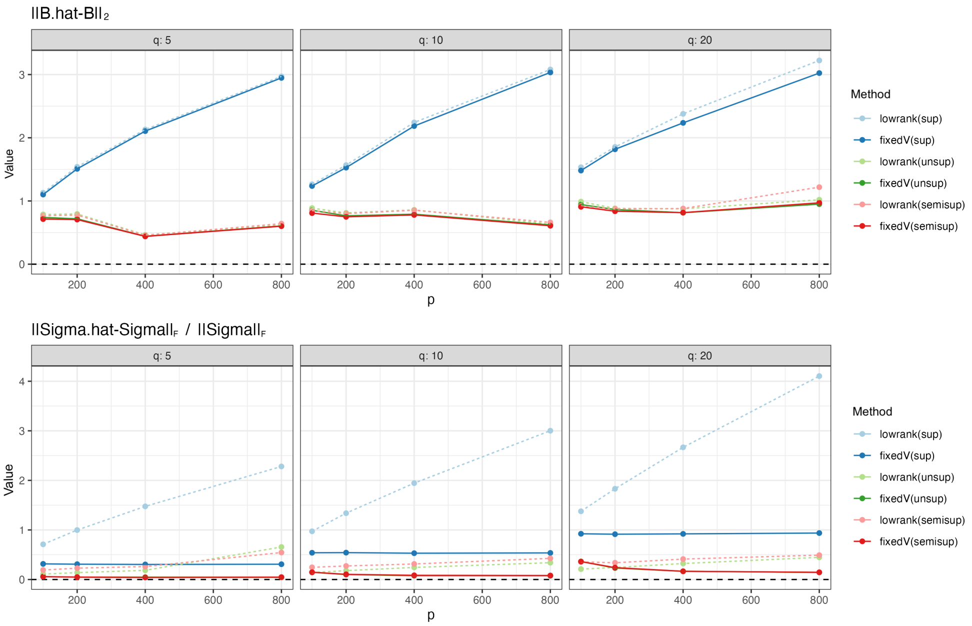

Figure 2: $||\widehat{\boldsymbol{B}}_2 - \boldsymbol{B}_2||_2/||\boldsymbol{B}_2||_2$ and $||\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}||_F/||\boldsymbol{\Sigma}||_F$ with varying p and q, with $N = 5000$, $n = 100$.

Figure 3 shows $||\widehat{\boldsymbol{B}}_2 - \boldsymbol{B}_2||_2$ and $||\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}||_F/||\boldsymbol{\Sigma}||_F$ with varying N and n with $p = 200$, $q = 20$.

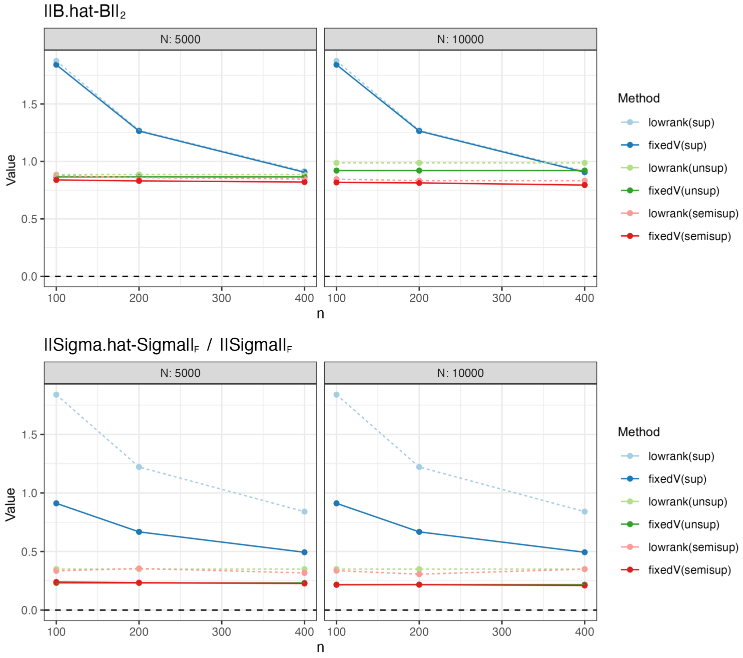

Figure 3: $||\widehat{\boldsymbol{B}}_2 - \boldsymbol{B}_2||_2/||\boldsymbol{B}_2||_2$ and $||\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}||_F/||\boldsymbol{\Sigma}||_F$ with varying N and n, with $p = 200$, $q = 20$.

As expected, as n increases, we see reduction in both errors for supervised methods. The errors of semi-supervised methods remains mostly unchanged with growing n or N, suggesting that the errors come from the variational approximation procedure, thus cannot be further reduced with increasing training sample size.

4 Suggested simulation settings

It would be very helpful if we can get a sense of how the bias and error of parameter estimates change with changing n, p, q by using the debiasing method. We can use the data generation setting in section 3.1, and $q = 20$, $p = 100, 200, 400$, $n = 100, 400$ or $n = 5000, 10000$.

References

John Aitchison and CH Ho. The multivariate poisson-log normal distribution. *Biometrika*, 76(4): 643–653, 1989.

Julien Chiquet, Mahendra Mariadassou, and Stéphane Robin. Variational inference for probabilistic poisson pca. *The Annals of Applied Statistics*, 12(4):2674–2698, 2018.

Dimitris Karlis. Em algorithm for mixed poisson and other discrete distributions. *ASTIN Bulletin: The Journal of the IAA*, 35(1):3–24, 2005.

Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6): 47–60, 1996.

Anjali Silva, Steven J Rothstein, Paul D McNicholas, and Sanjeena Subedi. A multivariate poisson-log normal mixture model for clustering transcriptome sequencing data. *BMC bioinformatics*, 20 (1):1–11, 2019.

Sanjeena Subedi and Ryan Browne. A parsimonious family of multivariate poisson-lognormal distributions for clustering multivariate count data. *arXiv preprint arXiv:2004.06857*, 2020.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.