Research Note

Xiaoyang Song

WN 2022

1 Generalized Latent Factor Model

1.1 Notations

Suppose that we have N people and J items, and each item has C possible categories. We define the following notations:

1. $Y_{ij} \in [0, C-1]$: the response of the ith person on item j, $i \in [0, N-1]$, $j \in [0, J-1]$.
2. p: number of latent factors
3. $\mathbf{x}_i = (\theta_{i1}, ..., \theta_{ip})^\top \in \mathbb{R}^{p \times 1}$: the factor scores for the ith person.
4. $\mathbf{X} \in \mathbb{R}^{N \times p}$: the factor score matrix.
5. $\boldsymbol{\beta}_j = (\beta_{j1}, ..., \beta_{jp})^\top \in \mathbb{R}^{K \times 1}$: the loading vector for item j.
6. $\boldsymbol{\beta} \in \mathbb{R}^{J \times p}$: the loading matrix.

1.2 Model

The generalized latent factor model assumes that Y_{ij} given $\mathbf{x}_i$ and $\boldsymbol{\beta}_j$ is a member of the exponential family with density function:

$$f(Y_{ij} = y \mid \boldsymbol{\beta}_j, \mathbf{x}_i) = \exp\left(\frac{y m_{ij} - b(m_{ij})}{\phi} + c(y, \phi)\right)$$

where $b(.)$ and $c(.)$ are two functions that decide which distribution in the exponential family the model distribution belongs to, ϕ is the scale parameter, and $m_{ij} = \boldsymbol{\beta}_j^\top \mathbf{x}_i$. Correspondingly, the joint likelihood is defined as the following:

$$L(\mathbf{x}_1, ..., \mathbf{x}_N, \boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_J) = \prod_{i=1}^{N} \prod_{j=1}^{J} f(Y_{ij} = y_{ij} \mid \mathbf{x}_j, \boldsymbol{\beta}_i)$$

2 Multidimensional Graded Model

2.1 Motivation

Motivation: we want to model the probability that the ith respondent chooses the category $k \in [0, C_j - 1]$ for item j using the information of the latent variables called *factors*, where C_j denotes the number of categories for item j and those categories are indexed by integer from 0 to $C_j - 1$ for simplicity.

2.2 Notations

Suppose that we have N people and J items, and each item has C possible categories. We define the following notations:

1. i: denotes the respondent i, $i \in [1, N]$.
2. j: denotes the item j, $j \in [1, J]$
3. C_j: the number of categories of item j.
4. p: number of latent factors
5. $\boldsymbol{\beta}_j$: a $p \times 1$ vector of factor coefficient for item j.
6. $\mathbf{X}$: a $N \times p$ matrix of factor scores of all respondents.
7. $\mathbf{x}_i$: a $p \times 1$ vector of factor scores for respondent i.
8. $\mathbf{Y}$: a $N \times J$ matrix of responses of all respondents.
9. $\mathbf{y}_i$: the response of respondent i, which is a $J \times 1$ vector.
10. y_{ij}: the response of respondent i on item j.
11. $\mathbf{d}_j$: a $(C_j - 1) \times 1$ vector of intercept term for item j.

2.3 MIRT Model

The MIRT model is defined by the following probabilities:

$$\mathbf{P}(y_{ij} \geq 0 \mid \boldsymbol{\beta}_j, \mathbf{d}_j, \mathbf{x}_i) = 1 \tag{1}$$

$$\mathbf{P}(y_{ij} \geq C_j \mid \boldsymbol{\beta}_j, \mathbf{d}_j, \mathbf{x}_i) = 0 \tag{2}$$

$$\mathbf{P}(y_{ij} \geq k \mid \boldsymbol{\beta}_j, \mathbf{d}_j, \mathbf{x}_i) = \frac{1}{1 + \exp\left(-\boldsymbol{\beta}_j^\top \mathbf{x}_i - d_{jk}\right)}, \quad \forall k \in [1, C_j - 1] \tag{3}$$

3 ReBoot Algorithm

Given the MIRT model, we can apply it to the distributed settings and formulate the `ReBoot` algorithm. Similar as the notation that is defined in the preceding section, suppose that we have N respondents and J items. We then distribute the data into m local machines and the estimated loadings and intercepts terms are denoted by $\widehat{\boldsymbol{\beta}}^{(k)}$ and $\widehat{\mathbf{d}}^{(k)}$ for $k \in [m]$, respectively. Let $\widetilde{n}$ denote the bootstrap sample size for each local estimator. The `ReBoot` algorithm can be formulated as the following.

Algorithm 1: `ReBoot` on IRT model

Input: $\big\{\widehat{\boldsymbol{\beta}}^{(k)}, \widehat{\mathbf{d}}^{(k)}\big\}_{k=1}^{m}$, $\widetilde{n}$

for $k = 1, \ldots, m$ **do**
  for $i = 1, \ldots, \widetilde{n}$ **do**
    Draw a Bootstrap feature vector $\widetilde{\mathbf{x}}_i^{(k)}$ from the distribution $f_{\mathbf{x}}(\cdot)$;
    Draw a Bootstrap response $\widetilde{Y}_i^{(k)}$ according to $f_{Y|\mathbf{x}}\big(\cdot\,|\widetilde{\mathbf{x}}_i^{(k)}; \widehat{\boldsymbol{\beta}}^{(k)}, \widehat{\mathbf{d}}^{(k)}\big)$;
  end
  $\widetilde{\mathcal{D}}^{(k)} \leftarrow \big\{\widetilde{Y}_i^{(k)}\big\}_{i\in[\widetilde{n}]}$;
end
$\widetilde{\mathcal{D}} \leftarrow \cup_{k=1}^{m} \widetilde{\mathcal{D}}^{(k)}$;
$\widehat{\boldsymbol{\beta}}^{\mathrm{rb}}, \widehat{\mathbf{d}}^{\mathrm{rb}} \leftarrow \underset{\boldsymbol{\beta}, \mathbf{d}\in\mathcal{B}}{\operatorname{argmin}}\ \ell_{\widetilde{\mathcal{D}}}(\boldsymbol{\beta}, \mathbf{d})$.

Output: $\widehat{\boldsymbol{\beta}}^{\mathrm{rb}}, \widehat{\mathbf{d}}^{\mathrm{rb}}$

4 Rasch Model

Rasch model is a latent factor model for dichotomous data. Suppose that we have N observations and each observation has J item responses. Given the capability of ith person X_i and the difficulty parameter of jth item d_j, the probability of a correct response is

$$\mathbb{P}(Y_{ij} = 1 | X_i, d_j) = \frac{1}{1 + e^{-(X_i - d_j)}}.$$

We are interested in estimating the difficulty parameter for each item, i.e., $\mathbf{d} := (b_j)_{j\in[J]}$. One typically uses the maximum likelihood estimation(MLE) to estimate $\mathbf{d}$. Let $\mathbf{Y} := (Y_{ij})_{i\in[N], j\in[J]}$ denote the response matrix and $\mathbf{x} := (X_i)_{i\in[N]}$ denote the capability of N

individuals. The likelihood function is defined as

$$\begin{aligned} L(\mathbf{d}|\mathbf{Y},\mathbf{x}) &= \prod_{i=1}^{N}\int\prod_{j=1}^{J}\{\mathbb{P}(Y_{ij}=1|X_i,d_j)\}^{Y_{ij}}\{\mathbb{P}(Y_{ij}=0|X_i,d_j)\}^{1-Y_{ij}}\phi(X_i)dX_i \\ &= \prod_{i=1}^{N}\int\prod_{j=1}^{J}\frac{e^{Y_{ij}(X_i-d_j)}}{1+e^{X_i-d_j}}\phi(X_i)dX_i \\ &= \prod_{i=1}^{N}\int\prod_{j=1}^{J}\frac{e^{X_i-d_j}}{1+e^{X_i-d_j}}e^{(Y_{ij}-1)(X_i-d_j)}\phi(X_i)dX_i \\ &= \prod_{i=1}^{N}\left\{\left(\prod_{j=1}^{J}e^{-(Y_{ij}-1)d_j}\right)\int\prod_{j=1}^{J}\frac{e^{X_i-d_j}}{1+e^{X_i-d_j}}e^{(Y_{ij}-1)X_i}\phi(X_i)dX_i\right\}. \end{aligned} \tag{4}$$

Therefore, the log-likelihood function is

$$l(\mathbf{d}|\mathbf{Y},\mathbf{x}) = \sum_{i=1}^{N}\sum_{j=1}^{J}(1-Y_{ij})d_j + \sum_{i=1}^{N}\log\left(\int\prod_{j=1}^{J}\frac{e^{X_i-d_j}}{1+e^{X_i-d_j}}e^{(Y_{ij}-1)X_i}\phi(X_i)dX_i\right). \tag{5}$$

Taking derivative with respect to d_j, we have

$$\begin{aligned} \frac{\partial l(\mathbf{d}|\mathbf{Y},\mathbf{x})}{\partial d_j} &= \sum_{i=1}^{N}(1-Y_{ij}) - \sum_{i=1}^{N}\log\left(\int\frac{e^{-X_i}}{\{1+e^{-(X_i-d_j)}\}^2}e^{(Y_{ij}-1)X_i}\prod_{l\neq j}^{J}\frac{e^{X_i-d_j}}{1+e^{X_i-d_j}}e^{(Y_{ij}-1)X_i}\phi(X_i)dX_i\right) \\ &= \sum_{i=1}^{N}(1-Y_{ij}) - \sum_{i=1}^{N}\log\left(\int\frac{e^{-d_j}}{1+e^{X_i-d_j}}\prod_{l=1}^{J}\frac{e^{X_i-d_j}}{1+e^{X_i-d_j}}e^{(Y_{ij}-1)X_i}\phi(X_i)dX_i\right). \end{aligned} \tag{6}$$

5 Numerical Study

5.1 Experiments Setup

On the Rasch model, the only parameters of interests are the intercepts (i.e. difficulty parameters). Suppose that we have $N = 1000$ people and $J = 10$ items. For the Rasch model, the ground truth parameters are designed to be:

$$\mathbf{d} = [0, 0.5, 1, 1.5, 2, 0, -0.5, -1, -1.5, -2] \tag{7}$$

The slopes are fixed to be 1 and will not be freely estimated. In addition, we only conduct the small-scale simulation studies under the first regime that is introduced above. Similarly, the local estimator, average estimator, and the `ReBoot` estimator are computed in the experiment.

Due to the concerns about the identifiability problem (i.e. if X_i and d_j shift by the same constant simultaneously, the resulting model will be the same), we fix the intercept of the first item to be zero during the estimation procedure.

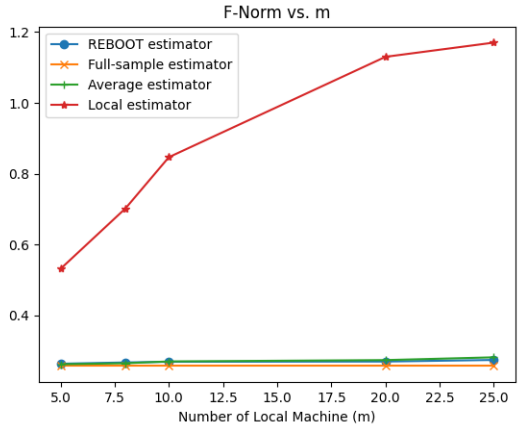

Figure 1: F-norm for all estimators

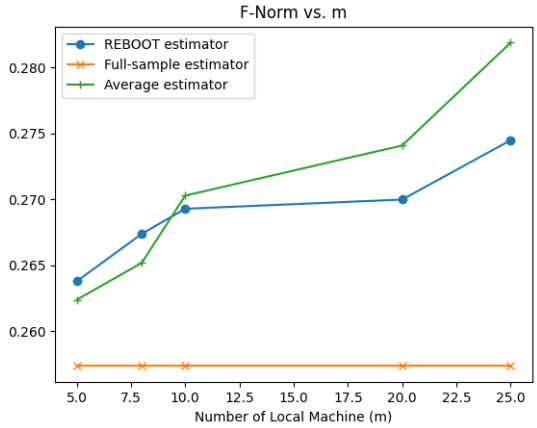

Figure 2: F-norm for all but local estimator

5.2 Experimental Results

From the simulation results we can observe that the `ReBoot` estimator gradually has its advantages exhibited as the number of machines increases (i.e. the local sample size decreases), which is consistent to the pattern that is observed in simulation study for MIRT model.

5.3 Discussions

This simulation study has several potential problems that need to be addressed.

- The proper way to choose ground truth for simulation.
- The proper way to impose identification constraint.
- `mirt()` function can not estimate item whose responses from all people are only in one category. A potential solution is to manually discard these invalid data.

6 DistributedPCA Algorithm

For MIRT model, to aggregate the loading estimators from m local machines, we can also adopt the `DistributedPCA` algorithm by slightly modifying it. The `DistributedPCA` algorithm for MIRT model is formulated below, where $\widehat{\boldsymbol{\beta}}^{(k)}$ is the local estimator for loadings of the kth machine.

Algorithm 2: DistributedPCA

Input: $\{\widehat{\boldsymbol{\beta}}^{(k)}\}_{k=1}^{m}$

for $k = 1, \ldots, m$ **do**

  Compute $\widehat{\boldsymbol{\Sigma}}^{(k)} \leftarrow \widehat{\boldsymbol{\beta}}^{(k)}\widehat{\boldsymbol{\beta}}^{(k)\top}$;

  Compute the eigenvectors $\widehat{\mathbf{V}}^{(k)}$ of $\widehat{\boldsymbol{\Sigma}}^{(k)}$ and send it to the central server.

end

On the central server, compute $\widetilde{\boldsymbol{\Sigma}} \leftarrow \frac{1}{m}\sum_{k=1}^{m}\widehat{\mathbf{V}}^{(k)}\widehat{\mathbf{V}}^{(k)\top}$ and its eigenvector $\widetilde{\mathbf{V}}$;

for $k = 1, \ldots, m$ **do**

  Compute the eigenvalues $\widehat{\boldsymbol{\Lambda}}^{(k)} \leftarrow \mathrm{diag}(\widetilde{\mathbf{V}}^{\top}\widehat{\boldsymbol{\Sigma}}^{(k)}\widetilde{\mathbf{V}})$ locally and send it to the central server;

end

On the central server, compute $\widetilde{\boldsymbol{\Lambda}} \leftarrow \frac{1}{m}\Sigma_{k=1}^{m}\widehat{\boldsymbol{\Lambda}}^{(k)}$;

$\widehat{\boldsymbol{\beta}}^{\mathrm{dpca}} \leftarrow \widetilde{\mathbf{V}}\widetilde{\boldsymbol{\Lambda}}^{\frac{1}{2}}$.

Output: $\widehat{\boldsymbol{\beta}}^{\mathrm{dpca}}$

A MIRT model simulation

A.1 Experiments Setup

The simulation study is conducted on the MIRT model that is introduced in the preceding section, where the ground truth parameters for slopes (i.e. loadings) and intercepts (i.e. difficulty parameters) are borrowed from Cai's (2010) paper. Specifically, two regimes of experiments are conducted:

- Fixed global sample size ($N = 1000$ and $N = 2500$) and gradually increase the number of machines.
- Fixed local sample size ($n = 125$) and gradually increase the number of machines.

Under both regimes, the local estimator, average estimator (i.e. the arithmetic mean of the local estimators), `ReBoot` Estimator and full-sample estimator are computed and evaluated.

Due to the concerns about the identifiability problem of the estimated loading matrix, the Positive-Diagonal-Lower-Triangular (PLT) constraint is enforced on the parameter matrix before the estimation.

A.2 Experimental Results

As for statistical error metrics, the SinTheta distance is used to measure the error for the loading matrix, while the Frobenius norm is used for intercepts. The experimental results are provided below:

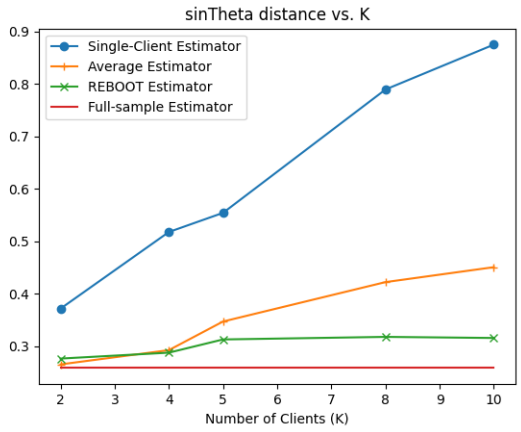

Figure 3: $N = 1000$

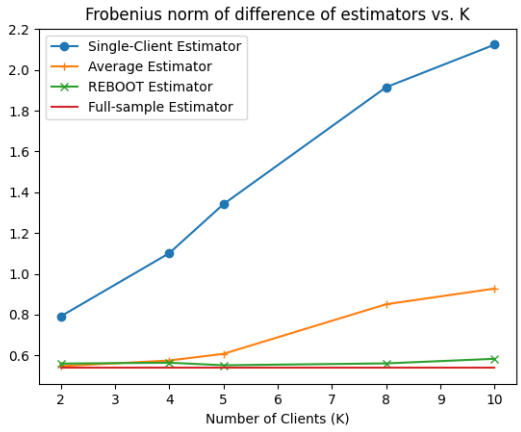

Figure 4: $N = 1000$

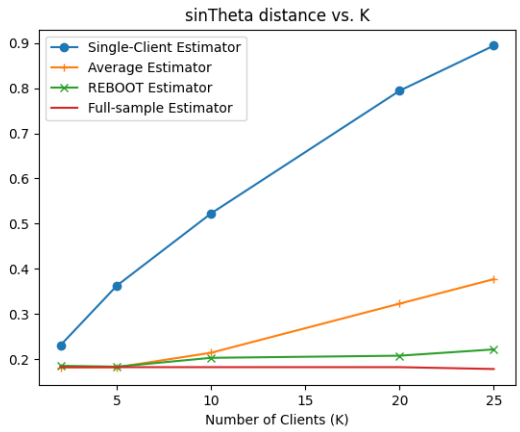

Figure 5: $N = 2500$

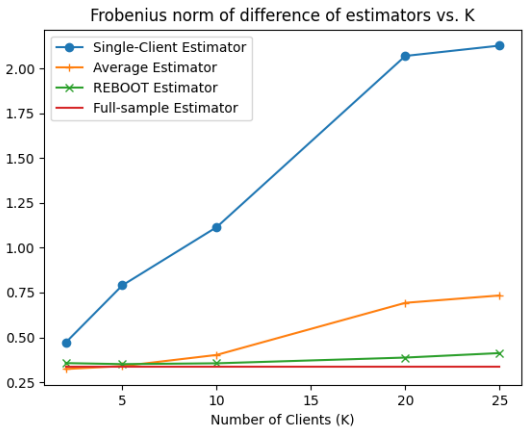

Figure 6: $N = 2500$

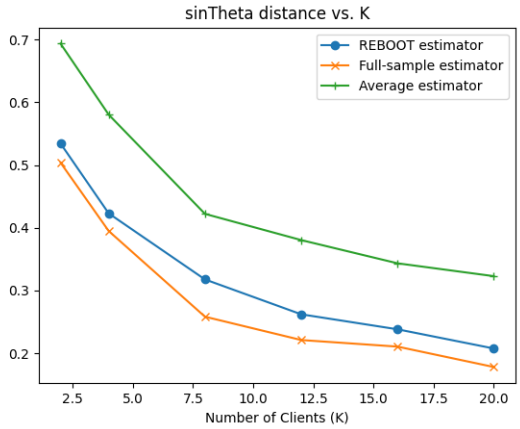

Figure 7: $n = 125$

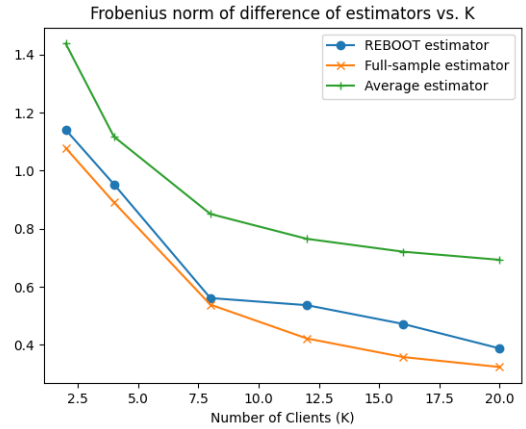

Figure 8: $n = 125$

A.3 Discussions

From the experimental results, the `ReBoot` estimator tends to give the best estimation for both loadings and intercepts for the most times. When the number of machines is small, implying that the local sample size is sufficient, the average estimator tends to have similar statistical error as the `ReBoot` estimator. For all cases, full-sample estimator gives the best estimator, which is consistent with the theoretical expectation, and the local estimator has the worst performance.

Experiment setup: fixed global sample size: $N = 1000$

K	2	4	5	8	10
Single client estimator	0.3715	0.5177	0.5542	0.7896	0.8752
Average estimator	0.2651	0.2923	0.3469	0.4221	0.4506
`REBOOT` estimator	0.2762	0.2875	0.3127	0.3176	0.3155
`MIRT` estimator	0.2582				

Table 1: sinTheta distance for estimators of slopes

Experiment setup: fixed global sample size: $N = 1000$

K	2	4	5	8	10
Single client estimator	0.7915	1.1012	1.3416	1.9153	2.1240
Average estimator	0.5475	0.5751	0.6078	0.8510	0.9279
`REBOOT` estimator	0.5599	0.5644	0.5518	0.5610	0.5835
`MIRT` estimator	0.5383				

Table 2: Frobenius norm of difference of intercept estimators

Experiment setup: fixed global sample size: $N = 2500$

K	2	5	10	20	25
Single client estimator	0.2307	0.3630	0.5228	0.7943	0.8942
Average estimator	0.1816	0.1818	0.2141	0.3228	0.3770
`REBOOT` estimator	0.1851	0.1832	0.2028	0.2075	0.2214
`MIRT` estimator	0.1780				

Table 3: sinTheta distance for estimators of slopes

Experiment setup: fixed global sample size: $N = 2500$

K	2	5	10	20	25
Single client estimator	0.4710	0.7902	1.1151	2.0704	2.1280
Average estimator	0.3235	0.3398	0.4021	0.6928	0.7339
`REBOOT` estimator	0.3565	0.3508	0.3554	0.3874	0.4124
`MIRT` estimator	0.3233				

Table 4: Frobenius norm of difference of intercept estimators

Experiment setup: fixed local sample size: $n = 125$

K	2	4	8	12	16	20
Average estimator	0.6935	0.5803	0.4221	0.3803	0.3433	0.3228
`REBOOT` estimator	0.5343	0.4229	0.3176	0.2620	0.2381	0.2075
`MIRT` estimator	0.5038	0.3948	0.2582	0.2210	0.2105	0.1780

Table 5: sinTheta distance for estimators of slopes

Experiment setup: fixed local sample size: $n = 125$

K	2	4	8	12	16	20
Average estimator	1.4382	1.1175	0.8510	0.7652	0.7208	0.6928
`REBOOT` estimator	1.1412	0.9534	0.5610	0.5365	0.4717	0.3874
`MIRT` estimator	1.0784	0.8910	0.5383	0.4218	0.3575	0.3233

Table 6: Frobenius norm of difference of intercept estimators