Multilevel Low Rank Matrices and Applications

Tetiana Parshakova

Ph.D. Dissertation Defense

Stanford

5/15/24

Contributions

1. W. Athas, Z. Nadeem, and **T. Parshakova**. (2022). Interpolation method and apparatus for arithmetic functions. US Patent Application No. 17/085,971.
2. **T. Parshakova**, F. Zhang, and S. Boyd. (2023). Implementation of an oracle-structured bundle method for distributed optimization. *Optimization and Engineering*, 1–34. Springer.
3. K. Choromanski, A. Sehanobish, H. Lin, Y. Zhao, E. Berger, **T. Parshakova**, et al. (2023). Efficient graph field integrators meet point clouds. In *Proceedings of the ICML*, 5978–6004. PMLR.
4. **T. Parshakova**, T. Hastie, E. Darve, and S. Boyd. (2024). Factor fitting, rank allocation, and partition in multilevel low rank matrices. To appear in *Optimization, Discrete Mathematics, and Applications to Data Sciences*. Springer.
5. S. Boyd, **T. Parshakova**, E. Ryu, and J. Suh. (2024). Optimization algorithm design via electric circuits. *Submitted*.
6. **T. Parshakova**, T. Hastie, and S. Boyd. (2024). Fitting multilevel factor models. *In preparation*.
7. **T. Parshakova**, T. Marcucci, and S. Boyd. (2024). Approximate distributed routing via low dimensional embedding. *In preparation*.

Contributions

1. W. Athas, Z. Nadeem, and **T. Parshakova**. (2022). Interpolation method and apparatus for arithmetic functions. US Patent Application No. 17/085,971.
2. **T. Parshakova**, F. Zhang, and S. Boyd. (2023). Implementation of an oracle-structured bundle method for distributed optimization. *Optimization and Engineering*, 1–34. Springer.
3. K. Choromanski, A. Sehanobish, H. Lin, Y. Zhao, E. Berger, **T. Parshakova**, et al. (2023). Efficient graph field integrators meet point clouds. In *Proceedings of the ICML*, 5978–6004. PMLR.
4. **T. Parshakova**, T. Hastie, E. Darve, and S. Boyd. (2024). Factor fitting, rank allocation, and partition in multilevel low rank matrices. To appear in *Optimization, Discrete Mathematics, and Applications to Data Sciences*. Springer.
5. S. Boyd, **T. Parshakova**, E. Ryu, and J. Suh. (2024). Optimization algorithm design via electric circuits. *Submitted*.
6. **T. Parshakova**, T. Hastie, and S. Boyd. (2024). Fitting multilevel factor models. *In preparation*.
7. **T. Parshakova**, T. Marcucci, and S. Boyd. (2024). Approximate distributed routing via low dimensional embedding. *In preparation*.

Outline

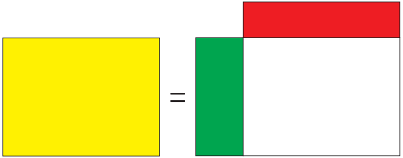
Low rank data

- in many applications data is organized in a matrix, $A \in \mathbf{R}^{m \times n}$
  - user ratings over movies
  - gene expressions in cells
- in practice the data is often approximately low rank [Eckart+Young36, Jolliffe02, Candès+Recht09, Udell+16]

$$A_{ij} \approx b_i^T c_j, \qquad b_i, c_j \in \mathbf{R}^r, \qquad r \ll \min\{m, n\}$$

  - per-user coefficients and per-movie factors
  - per-cell coefficients and per-gene factors

Low rank matrix approximation



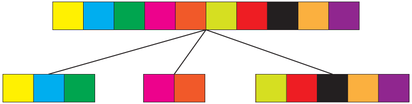
- find $B \in \mathbf{R}^{m \times r}$ and $C \in \mathbf{R}^{n \times r}$ such that $A \approx BC^T$

$$\text{minimize} \quad \|A - BC^T\|_F^2 = \sum_{i,j=1}^{m,n} (A_{ij} - b_i^T c_j)^2$$

- storage compression from mn to $2(m+n)r$
- interpretable factors
- solved via the singular value decomposition (SVD), proposed in 1907 [Schmidt07]

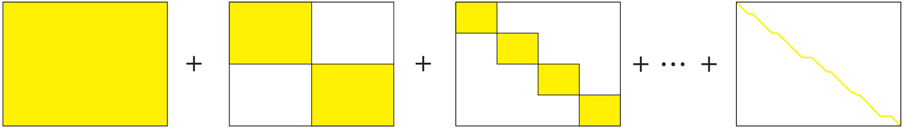
Hierarchically structured data

- biology: cells, tissues, organs
- geography: cities, states, countries
- finance: industries, groups, sectors
- healthcare: patients, clinics, regions
- education: students, classrooms, schools



Contiguous multilevel low rank matrices

- an $m \times n$ contiguous multilevel low rank (MLR) matrix A with L levels

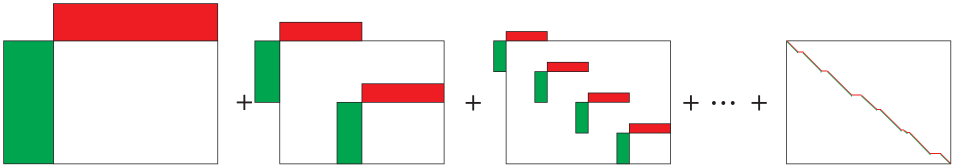


$$A = A^1 + \cdots + A^L, \qquad A^l = \mathbf{diag}(A_{l,1}, \ldots, A_{l,p_l})$$

- groups in partitions are contiguous ranges of row/column indices

Contiguous multilevel low rank matrices

- an $m \times n$ contiguous multilevel low rank (MLR) matrix A with L levels

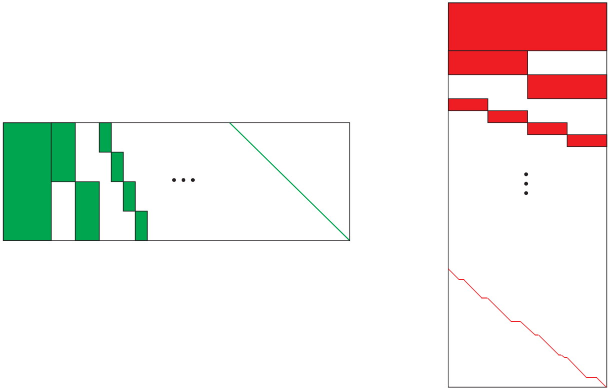


$$A_{l,k} = B_{l,k}C_{l,k}^T, \qquad B_{l,k} \in \mathbf{R}^{m_{l,k} \times r_l}, \qquad C_{l,k} \in \mathbf{R}^{n_{l,k} \times r_l}$$

- groups in partitions are contiguous ranges of row/column indices

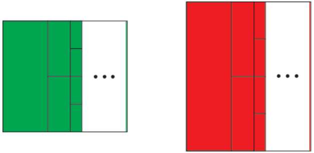
Factor form

- arrange factors such that $A = \tilde{B}\tilde{C}^T$



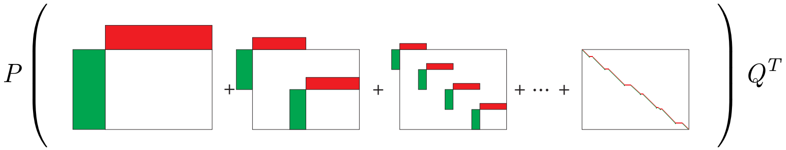
Compressed form

- $$B^l = \begin{bmatrix} B_{l,1} \\ \vdots \\ B_{l,p_l} \end{bmatrix} \in \mathbf{R}^{m \times r_l}, \qquad C^l = \begin{bmatrix} C_{l,1} \\ \vdots \\ C_{l,p_l} \end{bmatrix} \in \mathbf{R}^{n \times r_l}$$
- $$B = \begin{bmatrix} B^1 & \cdots & B^L \end{bmatrix} \in \mathbf{R}^{m \times r}, \qquad C = \begin{bmatrix} C^1 & \cdots & C^L \end{bmatrix} \in \mathbf{R}^{n \times r}$$
- $r = r_1 + \cdots + r_L$ is the MLR-rank of A



Multilevel low rank matrices

- general $m \times n$ MLR matrix has the form



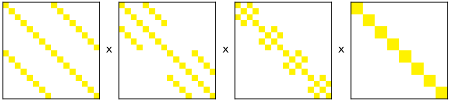
  - $P \in \mathbf{R}^{m \times m}$ is the row permutation matrix
  - $Q \in \mathbf{R}^{n \times n}$ is the column permutation matrix
- general hierarchical partition of the row and column index sets

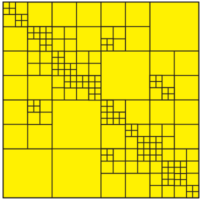
Multilevel low rank matrices

- permutations P and Q
- the number of levels L
- the block dimensions $m_{l,k}$ and $n_{l,k}$, $l = 1, \ldots, L$, $k = 1, \ldots, p_l$
- the two matrices B and C
- ranks r_i s.t. $r_1 + \cdots + r_L = r$

Related work

- Hierarchical matrices
  - $\mathcal{H}$-matrix [Greengard+Rokhlin87, Hackbusch99]
  - $\mathcal{H}^2$-matrix [Hackbusch+Borm02, Darve00]
  - hierarchically off-diagonal low-rank (HODLR) [Aminfar+16]
  - hierarchical semiseparable (HSS) matrix [Chandrasekaran+06]
- block low rank matrices [Amestoy+15]
- butterfly matrices [Parker95]
  - Monarch matrices [Dao+22]





Example: Distance matrix

- distance matrix for Venice roadmap
- $n = 5893$ nodes and 12098 edges
- $L = 14$ levels and MLR-rank $r = 98$
- compression ratio $30 : 1$

Method	Error (%)	Storage ($\times 10^5$)
LR	0.72	5.78
LR+D	0.71	5.78
HODLR	2.50	5.79
Monarch	0.87	5.88
MLR	**0.37**	5.78

Properties of MLR matrices

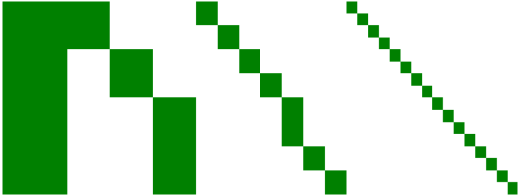
- matrix-vector multiply in $2(m+n)r$ flops vs mn in the dense case
- linear system solve
  - via recursive Sherman-Morrison-Woodbury in $O(nr^2)$ vs $O(n^3)$ in the dense case
  - via direct sparse solver

$$Ax = b \iff \begin{bmatrix} \tilde{C}^T & -I \\ 0 & \tilde{B} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

- k largest eigenvalues, total cost at iteration k
  - Arnoldi iteration with $O(nrk + nk^2)$ vs $O(n^2k + nk^2)$ dense case
  - Lanczos algorithm with $O(nrk + nk)$ vs $O(n^2k + nk)$ dense case

Example: Linear system solve

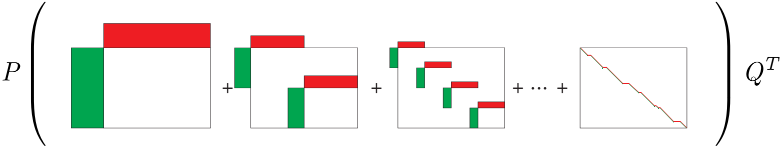
- solve $Ax = b$ with A positive definite MLR matrix
- $n = 10^5$
- dense matrix in single precision requires 37Gb
- hierarchical partition $p_1 = 1$, $p_2 = 3$, $p_3 = 7$, $p_4 = 16$, $p_5 = 10^5$
- ranks $r_1 = 30$, $r_2 = 20$, $r_3 = 10$, $r_4 = 5$, $r_5 = 1$
- compression ratio $750 : 1$



Example: Linear system solve

- direct dense solve using Cholesky
  - extrapolated time (from 10s for $10^4 \times 10^4$ matrix) is **2.7h** on M2 chip
- recursive SMW
  - solve in **200ms** on M2 chip
- MLR solve is $\times 50000$ faster than the dense one

Fitting problems



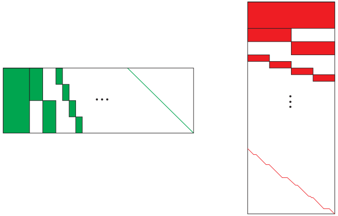
- how to fit the factors?
- how to allocate ranks across levels?
- how to choose hierarchical partition?

Outline

Factor fitting

- fix hierarchical partition and rank allocation
- optimize over the factors B and C

Alternating least squares

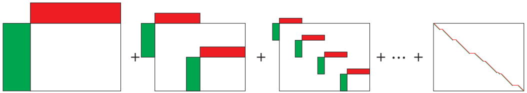


- recall $\hat{A} = \tilde{B}\tilde{C}^T = \hat{A}(B, C)$ is bi-linear
- an alternating least squares (ALS) algorithm to minimize

$$\|P^T A Q - \hat{A}(B, C)\|_F^2$$

  over B, then C, then B, etc
- $O(mnr)$ per iteration (conjugate gradient)

Block coordinate descent



- update the factors in one level in each iteration
- for level l we choose $B_{l,k}$ and $C_{l,k}$ to minimize

$$\left\| R - \mathbf{blkdiag}(B_{l,1}\, C_{l,1}^T, \ldots, B_{l,p_l}\, C_{l,p_l}^T) \right\|_F^2$$

  where R is the current residual

$$R = P^T A Q - \sum_{j \neq l} \mathbf{blkdiag}(B_{j,1}\, C_{j,1}^T, \ldots, B_{j,p_j}\, C_{j,p_j}^T)$$

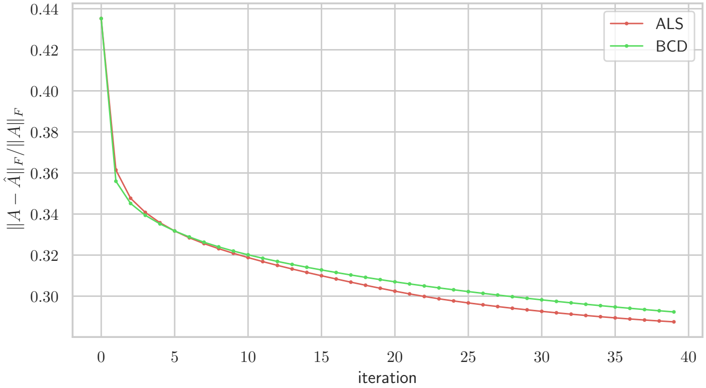
- $O(mnr)$ for single V-epoch (blockwise partial SVDs)

Comparison

- one iteration for ALS: approximately minimizing over B and then over C
- one iteration for BCD: one V-epoch

Comparison

- discrete Gauss transform matrix
- $m = 5000$ and $n = 7000$, $L = 14$, and $r_1 = \cdots = r_{14} = 5$



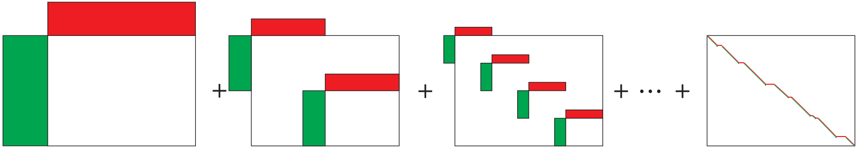
Outline

Rank allocation

- fix hierarchical partition
- optimize over the factors B and C and ranks $r_1, \ldots, r_L$ s.t. $r_1 + \cdots + r_L = r$

Rank exchange algorithm

$$R = P^T A Q - \sum_{j \neq l} \mathbf{blkdiag}(B_{j,1} C_{j,1}^T, \ldots, B_{j,p_j} C_{j,p_j}^T)$$



Rank exchange algorithm

$$R = P^T A Q - \sum_{j \neq l} \mathbf{blkdiag}(B_{j,1} C_{j,1}^T, \ldots, B_{j,p_j} C_{j,p_j}^T)$$

- incrementing rank allocated to level l by 1, decreases the Frobenius norm squared error by

$$\delta_l^+ = \sum_{k=1}^{p_l} \sigma_{r_l+1}^2(R_{l,k})$$

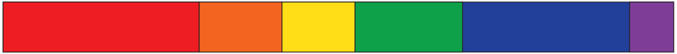
- decrementing rank allocated to level l by 1, increases Frobenius norm squared error by

$$\delta_l^- = \sum_{k=1}^{p_l} \sigma_{r_l}^2(R_{l,k})$$

Rank exchange algorithm

- find the levels $i \neq j$ for which the predicted net decrease is maximized

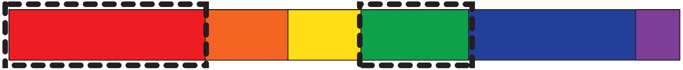
$$i, j = \underset{i \neq j}{\operatorname{argmax}} \left(\delta_i^+ - \delta_j^- \right)$$



Rank exchange algorithm

- find the levels $i \neq j$ for which the predicted net decrease is maximized

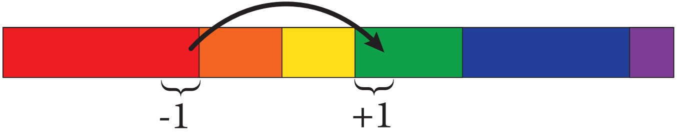
$$i, j = \underset{i \neq j}{\operatorname{argmax}} \left(\delta_i^+ - \delta_j^- \right)$$



Rank exchange algorithm

- find the levels $i \neq j$ for which the predicted net decrease is maximized

$$i, j = \underset{i \neq j}{\operatorname{argmax}} \left(\delta_i^+ - \delta_j^- \right)$$



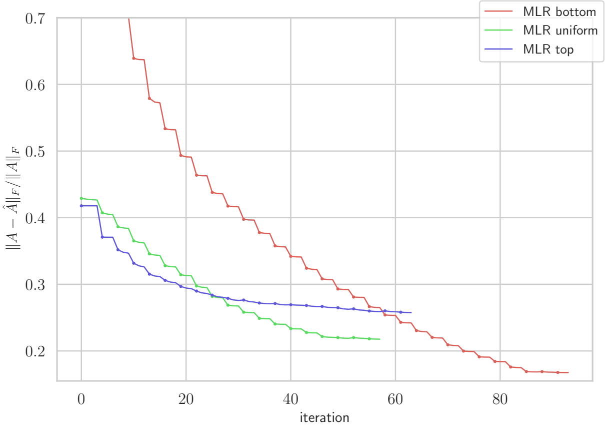
Rank exchange algorithm

- find the levels $i \neq j$ for which the predicted net decrease is maximized

$$i, j = \underset{i \neq j}{\operatorname{argmax}} \left(\delta_i^+ - \delta_j^- \right)$$

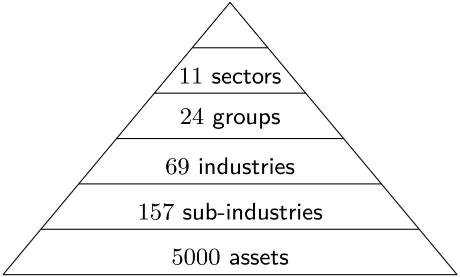
Rank exchange algorithm

- discrete Gauss transform matrix
- $m = 5000$, $n = 7000$, $L = 14$, and $r = 28$



Example: Asset covariance matrix

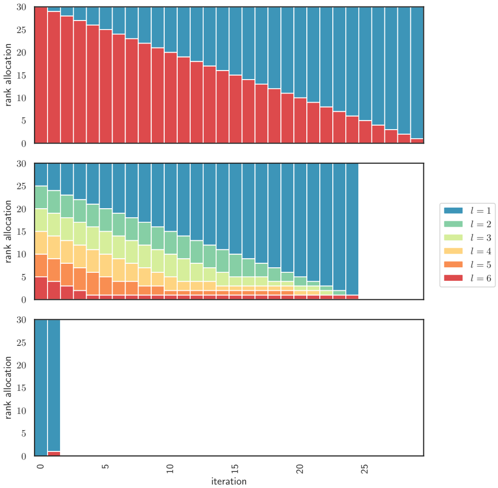
- 5000 asset returns over 300 days
- Global Industry Classification Standard (GICS)



Example: Asset covariance matrix

- $m = n = 5000$, $r = 30$, and $L = 6$
- compression ratio $80 : 1$

Method	Error (%)	Storage ($\times 10^5$)
LR	16.2	1.50
LR+D	**15.4**	1.50
HODLR	38.8	1.50
Monarch	18.0	1.56
MLR	**15.4**	1.50



Outline

Nested spectral dissection

1. $\tilde{R}_1 = (A - B_{1,1}\, C_{1,1}^T)$
2. $R_1 = P_1^T \tilde{R}_1 Q_1$
   - permutations P_1^T, Q_1^T maximize the sum of squares of residuals within the two diagonal blocks
3. $\tilde{R}_2 = R_1 - \begin{bmatrix} B_{2,1}\, C_{2,1}^T & 0 \\ 0 & B_{2,2}\, C_{2,2}^T \end{bmatrix}$
4. $R_2 = P_2^T \tilde{R}_1 Q_2$
   - permutations P_2^T, Q_2^T maximize the sum of squares of residuals within the four diagonal blocks, local for the two groups above
5. $\tilde{R}_3 = R_2 - \begin{bmatrix} B_{3,1}\, C_{3,1}^T & 0 & 0 & 0 \\ 0 & B_{3,2}\, C_{3,2}^T & 0 & 0 \\ 0 & 0 & B_{3,3}\, C_{3,3}^T & 0 \\ 0 & 0 & 0 & B_{3,4}\, C_{3,4}^T \end{bmatrix}$
6. $\ldots$

Permutation

- represent the partition as a vector $x \in \{-1, 1\}^n$
- maximize the sum of squares of residuals within the two groups

$$x^T S x = \sum_{i,j} x_i x_j R_{ij}^2 = \sum_{x_i = x_j} R_{ij}^2 - \sum_{x_i \neq x_j} R_{ij}^2 = 2 \sum_{x_i = x_j} R_{ij}^2 - \|R\|_F^2$$

- maximum bisection problem

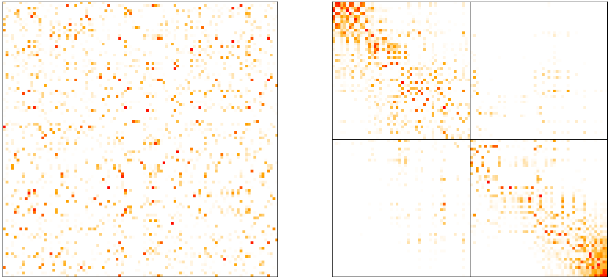
$$\begin{array}{ll} \text{maximize} & x^T S x \\ \text{subject to} & x \in \{-1, 1\}^n, \quad \mathbf{1}^T x = 0 \end{array}$$

Permutation

- spectral partition

$$\begin{array}{ll} \text{minimize} & x^T(\mathbf{diag}(S\mathbf{1}) - S)x \\ \text{subject to} & \|x\|_2^2 = n, \quad \mathbf{1}^T x = 0 \end{array}$$

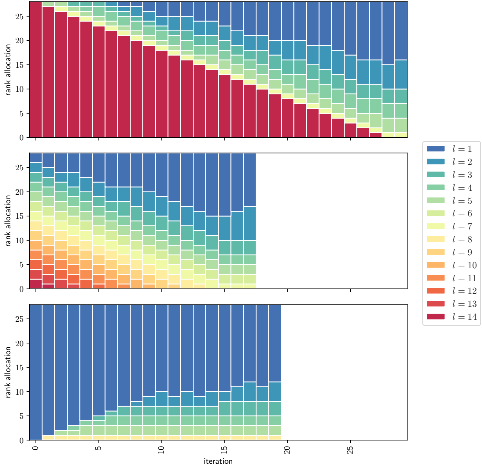
- *e.g.*, the sum of terms on the block diagonal increases by 80% after permutation



Example: Discrete Gauss transform matrix

- $A_{ij} = e^{-\|t_i - s_j\|_2^2/h^2}$ and $s_j, t_i \in \mathbf{R}^d$
- $m = 5000,\ n = 7000,\ r = 28,\ L = 14,\ d = 3,$ and $h = 0.2$
- compression ratio $100 : 1$

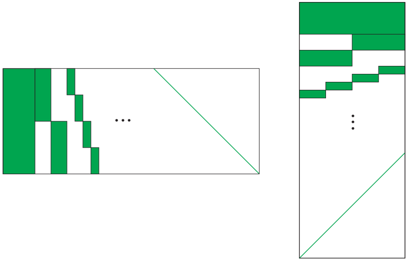
Method	Error (%)	Storage ($\times 10^5$)
LR	41.8	3.36
HODLR	72.5	3.39
Monarch	44.0	3.60
MLR bottom	**16.8**	3.36
MLR uniform	21.8	3.36
MLR top	25.8	3.36



Outline

PSD MLR

- symmetric positive semidefinite (PSD) MLR matrices
  - each block $A_{l,k} = B_{l,k} B_{l,k}^T$ is PSD



- PSD MLR is a covariance matrix in multilevel factor model (MFM) [Aitkin+81]

$$\Sigma = \begin{bmatrix} F & D^{1/2} \end{bmatrix} \begin{bmatrix} F & D^{1/2} \end{bmatrix}^T = FF^T + D$$

Multilevel factor model

$$y = Fz + e$$

- $F \in \mathbf{R}^{n \times s}$ is structured factor loading matrix
- $z \in \mathbf{R}^s$ are factor scores, with $z \sim \mathcal{N}(0, I_s)$
- $e \in \mathbf{R}^n$ are unique terms, with $e \sim \mathcal{N}(0, D)$

MLE-based fitting

- observe $Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} \in \mathbf{R}^{N \times n}$
- the log-likelihood based on N points

$$\ell(F, D; Y) = -\frac{nN}{2} \log(2\pi) - \frac{N}{2} \log\det(FF^T + D) - \frac{1}{2} \mathbf{Tr}((FF^T + D)^{-1} Y^T Y)$$

- if also observe latent data $z_1, \ldots, z_N \in \mathbf{R}^s$, the log-likelihood simplifies

$$\ell(F, D; Y, Z) = -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log\det D - \frac{1}{2} \| D^{-1/2}(Y - ZF^T) \|_F^2 - \frac{1}{2} \| Z \|_F^2$$

EM algorithm

- E step: compute

$$Q(F, D; F^0, D^0) = \mathbf{E}\left(\ell(F, D; Y, Z) \mid Y, F^0, D^0\right)$$

- M step: find F^1 and D^1 using

$$\begin{array}{ll} \text{maximize} & Q(F, D; F^0, D^0) \\ \text{subject to} & \left[\begin{array}{cc} F & D^{1/2} \end{array}\right] \text{ is the factor of PSD MLR} \end{array}$$

Recursive Sherman-Morrison-Woodbury

- PSD MLR

$$\Sigma = F_1 F_1^T + \cdots + F_{L-1} F_{L-1}^T + D$$

- define

$$\begin{aligned} F_{(l+1)+} &= \begin{bmatrix} F_{l+1} & \cdots & F_{L-1} \end{bmatrix} \\ M_0 &= (F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} F_l \\ H_l &= M_0 (I_{p_l r_l} + F_l^T M_0)^{-1/2} \end{aligned}$$

- SMW

$$(F_{l+} F_{l+}^T + D)^{-1} = (F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} - H_l H_l^T$$

- inverse is MLR matrix

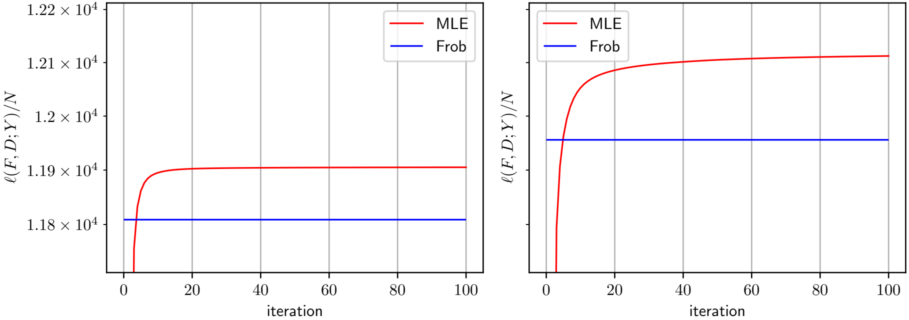
$$\Sigma^{-1} = -H_1 H_1^T - \cdots - H_{L-1} H_{L-1}^T + D^{-1}$$

Efficient computation

- computation of MLR Σ^{-1}
  - time complexity $O(nr^2 + p_{L-1} r_{\max} r^2)$
  - extra memory used is $3nr + 2p_{L-1} r_{\max} r$
- EM iteration
  - time complexity $O(p_{L-1} nr^2 + nr^3 + p_{L-1} nrN + p_{L-1} r_{\max} r^2)$

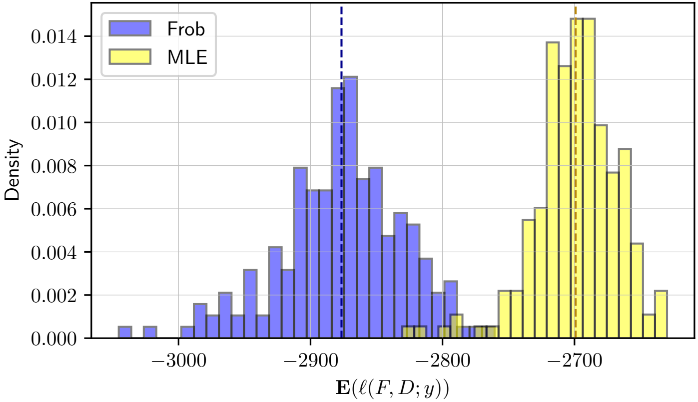
Example: Asset covariance matrix

- $n = 5000$, $L = 6$, $N = 300$, and $r = 30$
- compression ratio $80 : 1$
- log-likelihood for factor model (left) and multilevel factor model (right)



Example: Synthetic multilevel factor model

- $n = 1000$, $L = 5$, $r = 15$, $s = 77$, SNR of 4
- compression ratio $30 : 1$
- histograms over 100 runs each with sample size 200



Outline

Summary

- MLR matrices are natural extensions for low rank matrices
- fast linear algebra and storage compression
- Frobenius norm and MLE-based fitting methods
- model general hierarchical structures
- identify factors explaining data at global and local scales

Thanks!