Decentralized Low-Rank Matrix Completion

Qing Ling[1], Yangyang Xu[2], Wotao Yin[2], Zaiwen Wen[3]

1. Department of Automation, University of Science and Technology of China
2. Department of Computational and Applied Mathematics, Rice University
3. Department of Mathematics, Shanghai Jiaotong University

Outline

- Background: matrix completion
  centralized → decentralized

- Problem formulation
  nonconvex matrix factorization model + decentralized computing

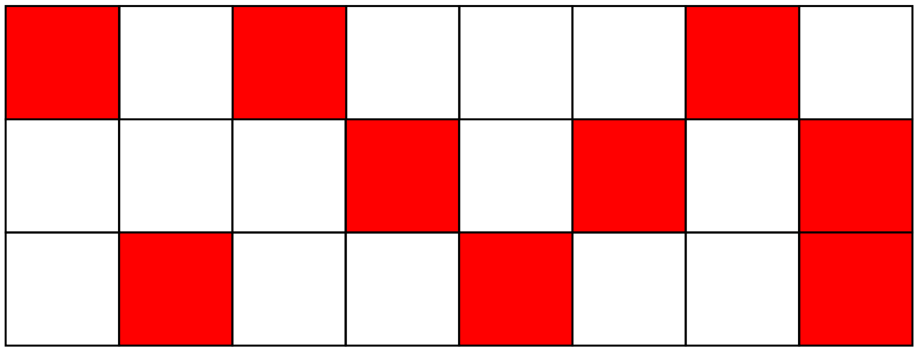
- Algorithm design
  Gauss-Seidel + decentralized implementation (with the ADM)

- Simulation and conclusion

Keywords: matrix completion matrix factorization ADM/ADMM
decentralized computing

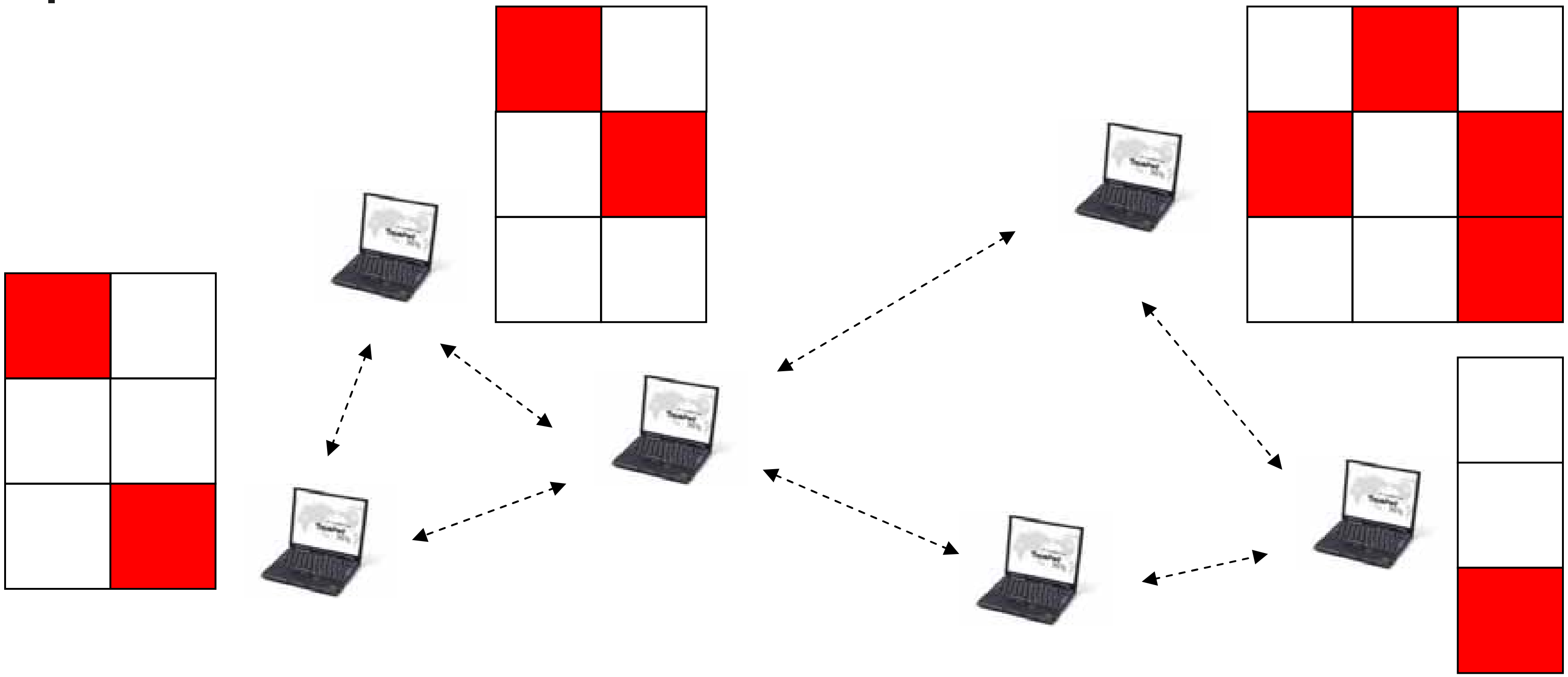
Background (I): matrix completion

- Matrix completion problem
  - Knowing some entries of a matrix, to recover the others
  - Important prior: the matrix is low-rank



- Related applications
  - Collaborative filtering
  - Internet traffic analysis
  - Sensor node localization

Background (II): decentralized matrix completion



- Distributed data in distributed agents & no fusion center
  - Privacy, cost of data collection, etc
  - Decentralized computing with limited information exchange

Problem formulation (I): two models

- A connected network with L distributed agents
  - Agent i observes some entries of a data matrix $\mathbf{W}_i \in \mathcal{R}^{N \times M_i}$
  - The whole data matrix is with rank r<<min(N, M)

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_L] \in \mathcal{R}^{N \times M}$$

  - Observation over a subset

$$\{W_{n,m}\},\ (n,m) \in \Omega \subset \{(n,m) : 1 \le n \le N, 1 \le m \le M\}$$

- Nonconvex matrix factorization model

$$\min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \quad \frac{1}{2}||\mathbf{XY} - \mathbf{Z}||_F^2, \qquad \mathbf{Z} \in \mathcal{R}^{N \times M},\ \mathbf{X} \in \mathcal{R}^{N \times r},\ \mathbf{Y} \in \mathcal{R}^{r \times M}$$

$$s.t. \quad Z_{n,m} = W_{n,m}, \quad \forall (n,m) \in \Omega.$$

- Convex nuclear norm minimization model

$$\min_{\mathbf{Z}} \quad ||\mathbf{Z}||_*, \qquad \mathbf{Z} \in \mathcal{R}^{N \times M}$$

$$s.t. \quad Z_{n,m} = W_{n,m}, \quad \forall (n,m) \in \Omega.$$

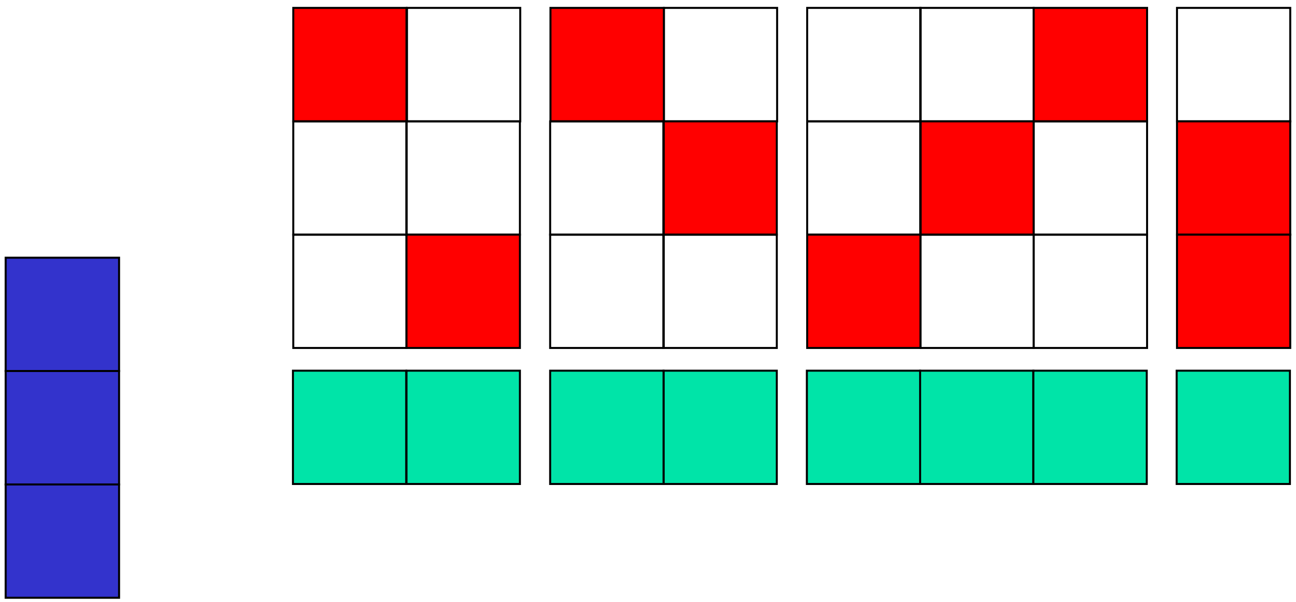
Problem formulation (II): decentralized computing

- Decentralized matrix completion with the nonconvex model

$$\mathbf{W} = \mathbf{X}\mathbf{Y} \longleftrightarrow \mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_L] = \mathbf{X}[\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_L]$$

$$\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_L]$$

  - Public matrix X: common to all agents
  - Private matrix Y_i: held by agent i



Problem formulation (III): nonconvex vs. convex

- Nonconvex vs. convex in decentralized computing
  - Nonconvex: efficient computation of X and Y_i (and Z_i)
  - Convex: decentralized SVD as a subroutine

Algorithm design (I): Gauss-Seidel method

- Centralized Gauss-Seidel method: LMaFit

$$\boxed{\begin{array}{lll} \min\limits_{\mathbf{X},\mathbf{Y},\mathbf{Z}} & \frac{1}{2}||\mathbf{X}\mathbf{Y}-\mathbf{Z}||_F^2, & \\ s.t. & Z_{n,m}=W_{n,m}, & \forall (n,m)\in\Omega. \end{array}}$$

$$\begin{aligned}
\mathbf{X}(t+1) &= \mathbf{Z}(t)\mathbf{Y}^T(t)(\mathbf{Y}(t)\mathbf{Y}^T(t))^{-1}, \\
\mathbf{Y}(t+1) &= (\mathbf{X}^T(t+1)\mathbf{X}(t+1))^{-1}\mathbf{X}^T(t+1)\mathbf{Z}(t), \\
\mathbf{Z}(t+1) &= \mathbf{X}(t+1)\mathbf{Y}(t+1)+\underbrace{P_\Omega(\mathbf{W}-\mathbf{X}(t+1)\mathbf{Y}(t+1))}_{\text{projection}}.
\end{aligned}$$

[WYZ10] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm. Mathematical Programming Computation, To Appear

Algorithm design (I): Gauss-Seidel method

- Centralized Gauss-Seidel method: LMaFit

$$\boxed{\begin{array}{ll} \min\limits_{\mathbf{X},\mathbf{Y},\mathbf{Z}} & \frac{1}{2}||\mathbf{X}\mathbf{Y}-\mathbf{Z}||_F^2, \\ s.t. & Z_{n,m} = W_{n,m}, \quad \forall (n,m) \in \Omega. \end{array}}$$

$$\begin{aligned} \mathbf{X}(t+1) &= \mathbf{Z}(t)\mathbf{Y}^T(t)(\mathbf{Y}(t)\mathbf{Y}^T(t))^{-1}, \\ \mathbf{Y}(t+1) &= (\mathbf{X}^T(t+1)\mathbf{X}(t+1))^{-1}\mathbf{X}^T(t+1)\mathbf{Z}(t), \\ \mathbf{Z}(t+1) &= \mathbf{X}(t+1)\mathbf{Y}(t+1) + P_\Omega(\mathbf{W} - \mathbf{X}(t+1)\mathbf{Y}(t+1)). \end{aligned}$$

- A simple but nontrivial revision: replace X(t+1) with

$$\mathbf{X}(t+1) = c\mathbf{Z}(t)\mathbf{Y}^T(t), \quad c > 0,$$

Since we care about Z other than X and Y

Algorithm design (II): decentralized implementation

- If agent i knows $X^{(i)}=X$, the updates of Y_i and Z_i are easy

$$\boxed{\mathbf{Y}(t+1) = (\mathbf{X}^T(t+1)\mathbf{X}(t+1))^{-1}\mathbf{X}^T(t+1)\mathbf{Z}(t),}$$

Y-update

$$\downarrow$$

$$\mathbf{Y}_i(t+1) = ((\mathbf{X}^{(i)}(t+1))^T\mathbf{X}^{(i)}(t+1))^{-1}(\mathbf{X}^{(i)}(t+1))^T\mathbf{Z}_i(t),$$

$$\boxed{\mathbf{Z}(t+1) = \mathbf{X}(t+1)\mathbf{Y}(t+1) + P_{\Omega}(\mathbf{W} - \mathbf{X}(t+1)\mathbf{Y}(t+1)).}$$

Z-update

$$\downarrow$$

$$\mathbf{Z}_i(t+1) = \mathbf{X}^{(i)}(t+1)\mathbf{Y}_i(t+1) + P_{\Omega_i}(\mathbf{W}_i - \mathbf{X}^{(i)}(t+1)\mathbf{Y}_i(t+1)).$$

- How to update $X^{(i)}$? Choose c=1/L:

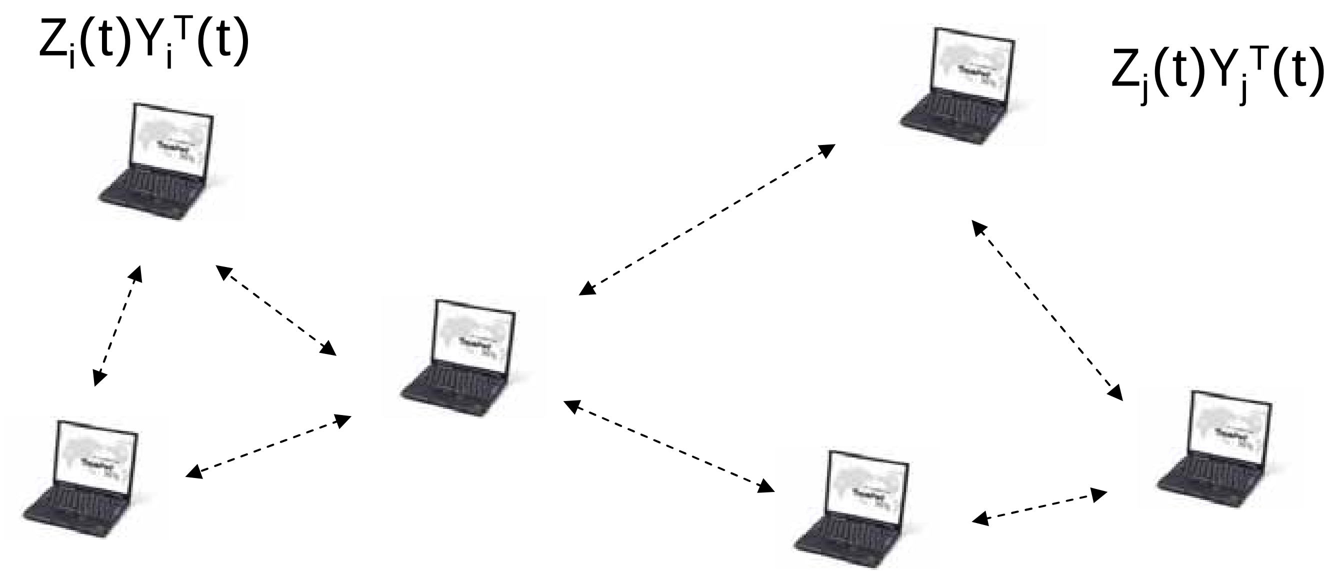
$$\boxed{\mathbf{X}(t+1) = c\mathbf{Z}(t)\mathbf{Y}^T(t), \quad c > 0,}$$

X-update is average consensus

$$\downarrow$$

$$\mathbf{X}^{(i)}(t+1) = \frac{1}{L}\sum_{i=1}^{L}\mathbf{Z}_i(t)\mathbf{Y}_i^T(t),$$

Algorithm design (III): average consensus



How to let all agents have the average?

Communicating with neighbors and updating the value

Algorithm design (IV): average consensus

- Exactly solving the average consensus problem
  - Randomized gossip, alternating direction method (ADM)
  - Iterative algorithm: dividing each iteration into S slots
- The ADM is a powerful tool for decentralized optimization

$$\boxed{\mathbf{X}^{(i)}(t+1) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{Z}_i(t)\mathbf{Y}_i^T(t), \quad \forall i.}$$

$$\boxed{\begin{array}{ll} \min & \frac{1}{2}\sum\limits_{i=1}^{L} \|\mathbf{U}^{(i)} - \mathbf{Z}_i(t)\mathbf{Y}_i^T(t)\|_F^2, \\ s.t. & \mathbf{U}^{(i)} = \mathbf{U}^{(j)}, \quad \forall j \in \mathcal{N}_i, \ \forall i. \end{array}}$$

[B1999] D. Bertsekas. Numerical Optimization, Second Edition. Athena Scientific, 1999

[SRG2008] I. Schizas, A. Ribeiro, and G. Giannakis. Consensus in ad hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals. IEEE Transactions on Signal Processing, 2008

[LTYY2012] Q. Ling, M. Tao, W. Yin, and X. Yuan. A multi-block alternating direction method with parallel splitting for decentralized consensus optimization. Journal of Wireless Communications and Networking, Submitted

Algorithm design (IV): average consensus

- Exactly solving the average consensus problem
  - Randomized gossip, alternating direction method (ADM)
  - Iterative algorithm: dividing each iteration into S slots

$$\mathbf{X}^{(i)}(t+1) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{Z}_i(t)\mathbf{Y}_i^T(t), \quad \forall i.$$

- Exact average consensus iterates with the ADM

multiplier   positive constant   neighbors

$$\mathbf{X}^{(i)}(t+\frac{s+1}{S}) = \frac{\mathbf{Z}_i(t)\mathbf{Y}_i^T(t) - \boldsymbol{\alpha}^{(i)}(t+\frac{s}{S}) + \beta|\mathcal{N}_i|\mathbf{X}^{(i)}(t+\frac{s}{S}) + \beta\sum_{j\in\mathcal{N}_i}\mathbf{X}^{(j)}(t+\frac{s}{S})}{1+2\beta|\mathcal{N}_i|}.$$

$$\boldsymbol{\alpha}^{(i)}(t+\frac{s+1}{S}) = \boldsymbol{\alpha}^{(i)}(t+\frac{s}{S}) + \beta\left(|\mathcal{N}_i|\mathbf{X}^{(i)}(t+\frac{s+1}{S}) - \sum_{j\in\mathcal{N}_i}\mathbf{X}^{(j)}(t+\frac{s+1}{S})\right).$$

Algorithm design (V): inexact average consensus

- Exact average consensus
  - Optimal when the network is connected
  - The decentralized algorithm = the centralized algorithm
  - Extra communication & coordination costs
- Inexact average consensus
  - Simply let S=1 in the ADM; no more iterates
  - Different from the centralized algorithm

Algorithm design (V): optimization framework

- Updating own private from own public

$$\mathbf{Y}_i(t+1) = ((\mathbf{X}^{(i)}(t+1))^T \mathbf{X}^{(i)}(t+1))^{-1} (\mathbf{X}^{(i)}(t+1))^T \mathbf{Z}_i(t),$$

$$\mathbf{Z}_i(t+1) = \mathbf{X}^{(i)}(t+1)\mathbf{Y}_i(t+1) + P_\Omega(\mathbf{W}_i - \mathbf{X}^{(i)}(t+1)\mathbf{Y}_i(t+1)).$$

- Updating own public from own private & neighboring public

$$\mathbf{X}^{(i)}(t+1) = \frac{\mathbf{Z}_i(t)\mathbf{Y}_i^T(t) - \alpha^{(i)}(t) + \beta|\mathcal{N}_i|\mathbf{X}^{(i)}(t) + \beta\sum_{j\in\mathcal{N}_i}\mathbf{X}^{(j)}(t)}{1 + 2\beta|\mathcal{N}_i|},$$

$$\alpha^{(i)}(t+1) = \alpha^{(i)}(t) + \beta\left(|\mathcal{N}_i|\mathbf{X}^{(i)}(t+1) - \sum_{j\in\mathcal{N}_i}\mathbf{X}^{(j)}(t+1)\right).$$

Question: can we protect the private information Z_i and Y_i?

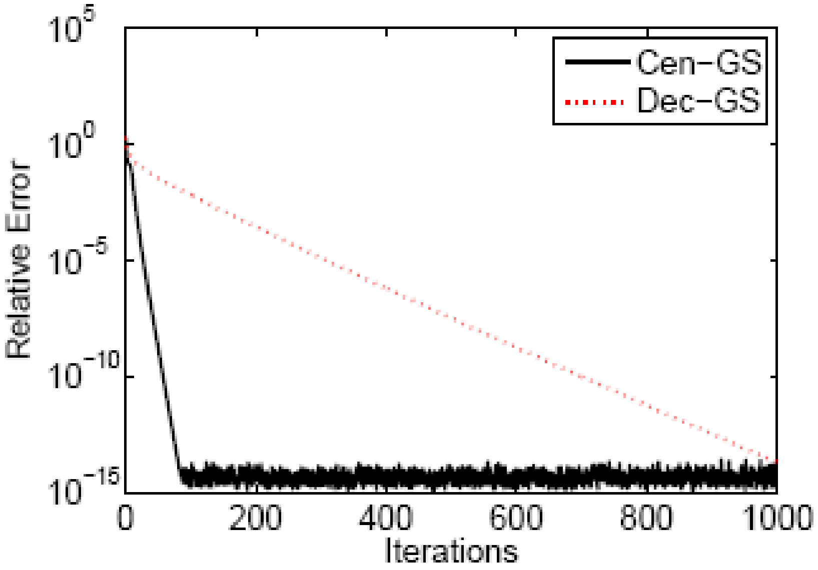
Simulation (I): simulation settings

- Network settings
  - L=25 agents are randomly distributed in a 100×100 area
  - Two agents are neighbors if their distance < 30
- Data settings
  - N=300 rows, M=500 columns, 20 columns per agent
  - 100×p percent randomly chosen entries can be observed
  - True rank K=4
- Performance evaluation: relative error

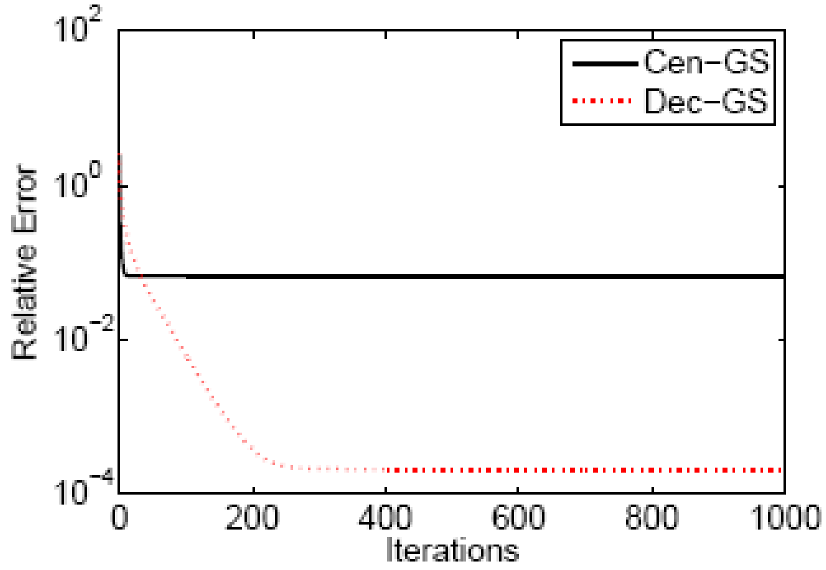
$$||\mathbf{W} - \mathbf{X}\mathbf{Y}||_F / ||\mathbf{W}||_F$$

Simulation (II): convergence

Knowing 80% data, p=0.8



Estimated rank r=4=K



Estimated rank r=6>K

- Decentralized = centralized
- Linear convergence rate

- Decentralized > centralized
- Centralized with adaptive rank

Conclusion

- Concluding remarks
  - Discuss matrix completion in a decentralized manner
  - Use a nonconvex matrix factorization model
  - Main feature: private <-> public
  - Gauss-Seidel for private, average consensus for public
- Open questions
  - When can we protect data privacy?
  - Convergence? Nonconvex model + inexact average consensus

Thank you!