

# Divide-and-Conquer Matrix Factorization

Lester Mackey

Collaborators: Ameet Talwalkar and Michael I. Jordan

January 23, 2012

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix}$$

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 5 & 4 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots & 1 \end{bmatrix}$$

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix}$$

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 1 & ? & 3 & 1 & \dots & 3 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix}$$

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix}$$

# Motivation: Low-rank Matrix Factorization

$$\begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} ?$$

# Motivation: Low-rank Matrix Factorization

**Goal:** Given a matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  formed by deleting and corrupting the entries of  $\mathbf{L}_0$ , recover the underlying low-rank matrix  $\mathbf{L}_0$ .

$$\mathbf{M} = \begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} = \mathbf{L}_0$$

- ① **Matrix completion (MC):** Small fraction of entries revealed
- ② **Robust matrix factorization (RMF):** Fraction of entries grossly corrupted

# Motivation: Low-rank Matrix Factorization

**Goal:** Given a matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  formed by deleting and corrupting the entries of  $\mathbf{L}_0$ , recover the underlying low-rank matrix  $\mathbf{L}_0$ .

$$\mathbf{M} = \begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} = \mathbf{L}_0$$

## Examples: Matrix completion

- Collaborative filtering: How will user  $i$  rate movie  $j$ ?
  - Netflix: 10 million users, 100K DVD titles
- Ranking on the web: Is URL  $j$  relevant to user  $i$ ?
  - Google News: millions of articles, millions of users

# Motivation: Low-rank Matrix Factorization

**Goal:** Given a matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  formed by deleting and corrupting the entries of  $\mathbf{L}_0$ , recover the low-rank matrix  $\mathbf{L}_0$ .

$$\mathbf{M} = \begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} = \mathbf{L}_0$$

## Examples: Robust matrix factorization

- Background modeling/foreground activity detection



(Candès, Li, Ma, and Wright, 2011)

# Motivation: Low-rank Matrix Factorization

**Goal:** Given a matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  formed by deleting and corrupting the entries of  $\mathbf{L}_0$ , recover the low-rank matrix  $\mathbf{L}_0$ .

$$\mathbf{M} = \begin{bmatrix} 5 & ? & 3 & 1 & \dots & 4 \\ 5 & ? & 4 & ? & \dots & ? \\ ? & 5 & ? & 3 & \dots & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 3 & 1 & \dots & 3 \\ 5 & 4 & 4 & 5 & \dots & 4 \\ 3 & 5 & 5 & 3 & \dots & 5 \end{bmatrix} = \mathbf{L}_0$$

## State of the art MF algorithms

- Strong recovery guarantees
- Plagued by expensive subroutines (e.g., truncated SVD)

## This talk

- Present divide and conquer approaches for scaling up any MF algorithm while maintaining strong recovery guarantees

# Roadmap

- 1 Introduction
- 2 Matrix Completion
  - Background
  - Divide-Factor-Combine
  - Simulations
  - Collaborative filtering
- 3 Robust Matrix Factorization
  - Background
  - Simulations
  - Video background modeling
- 4 Future Directions

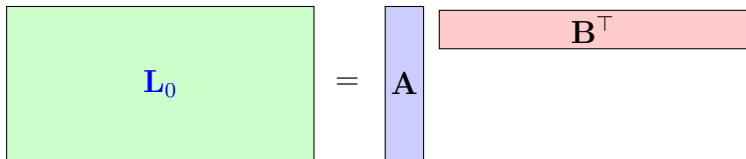
# Exact Matrix Completion

**Goal:** Given entries from a matrix  $\mathbf{L}_0 \in \mathbb{R}^{m \times n}$  with rank  $r \ll m, n$ , recover  $\mathbf{L}_0$ .

# Noisy Matrix Completion

**Goal:** Given entries from a matrix  $\mathbf{M} = \mathbf{L}_0 + \mathbf{Z} \in \mathbb{R}^{m \times n}$  where  $\mathbf{L}_0$  has rank  $r \ll m, n$  and  $\mathbf{Z}$  is entrywise noise, recover  $\mathbf{L}_0$ .

- **Good news:**  $\mathbf{L}_0$  has  $\sim (m + n)r \ll mn$  degrees of freedom



- Factored form:  $\mathbf{A}\mathbf{B}^T$  for  $\mathbf{A} \in \mathbb{R}^{m \times r}$  and  $\mathbf{B} \in \mathbb{R}^{n \times r}$
- **Bad news:** Not all low-rank matrices can be recovered

**Question:** What can go wrong?

# What can go wrong?

## Entire column missing

$$\begin{bmatrix} 1 & 2 & ? & 3 & \dots & 4 \\ 3 & 5 & ? & 4 & \dots & 1 \\ 2 & 5 & ? & 2 & \dots & 5 \end{bmatrix}$$

- No hope of recovery!

## Solution: Uniform observation model

Assume that the set of  $s$  observed entries  $\Omega$  is drawn uniformly at random:

$$\Omega \sim \text{Unif}(m, n, s)$$

# What can go wrong?

## Bad spread of information

$$\mathbf{L} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [1] [1 \ 0 \ 0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Can only recover  $\mathbf{L}$  if  $\mathbf{L}_{11}$  is observed

Solution: Incoherence with standard basis (Candès and Recht, 2009)

A matrix  $\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \in \mathbb{R}^{m \times n}$  with  $\text{rank}(\mathbf{L}) = r$  is  $(\mu, r)$ -coherent if

Singular vectors are **not too sparse**:  $\begin{cases} \max_i \|\mathbf{U}\mathbf{U}^\top \mathbf{e}_i\|^2 \leq \mu r / m \\ \max_i \|\mathbf{V}\mathbf{V}^\top \mathbf{e}_i\|^2 \leq \mu r / n \end{cases}$

and **not too cross-correlated**:  $\|\mathbf{U}\mathbf{V}^\top\|_\infty \leq \sqrt{\frac{\mu r}{mn}}$

# How do we recover $\mathbf{L}_0$ ?

First attempt:

$$\begin{aligned} & \text{minimize}_{\mathbf{A}} \quad \text{rank}(\mathbf{A}) \\ & \text{subject to} \quad \sum_{(i,j) \in \Omega} (\mathbf{A}_{ij} - \mathbf{M}_{ij})^2 \leq \Delta^2. \end{aligned}$$

**Problem:** Intractable to solve!

**Solution:** Solve **convex** relaxation (Fazel, Hindi, and Boyd, 2001; Candès and Plan, 2010)

$$\begin{aligned} & \text{minimize}_{\mathbf{A}} \quad \|\mathbf{A}\|_* \\ & \text{subject to} \quad \sum_{(i,j) \in \Omega} (\mathbf{A}_{ij} - \mathbf{M}_{ij})^2 \leq \Delta^2 \end{aligned}$$

where  $\|\mathbf{A}\|_* = \sum_k \sigma_k(\mathbf{A})$  is the trace/nuclear norm of  $\mathbf{A}$ .

**Questions:**

- Will the nuclear norm heuristic successfully recover  $\mathbf{L}_0$ ?
- Can nuclear norm minimization scale to large MC problems?

# Noisy Nuclear Norm Heuristic: Does it work?

Yes, with high probability.

## Typical Theorem

If  $\mathbf{L}_0$  is  $(\mu, r)$ -coherent,  $s = O(\mu rn \log^2(n))$  entries of  $\mathbf{M} \in \mathbb{R}^{m \times n}$  are observed uniformly at random, and  $\hat{\mathbf{L}}$  solves the noisy nuclear norm heuristic, then

$$\|\hat{\mathbf{L}} - \mathbf{L}_0\|_F \leq f(m, n)\Delta$$

with high probability when  $\|\mathbf{M} - \mathbf{L}_0\|_F \leq \Delta$ .

- See Candès and Plan (2010); Mackey, Talwalkar, and Jordan (2011); Keshavan, Montanari, and Oh (2010); Negahban and Wainwright (2010)
- Implies **exact** recovery in the noiseless setting ( $\Delta = 0$ )

# Noisy Nuclear Norm Heuristic: Does it scale?

## Not quite...

- Standard interior point methods (Candès and Recht, 2009):  
 $O(|\Omega|(m+n)^3 + |\Omega|^2(m+n)^2 + |\Omega|^3)$
- More efficient, tailored algorithms:
  - Singular Value Thresholding (SVT) (Cai, Candès, and Shen, 2010)
  - Augmented Lagrange Multiplier (ALM) (Lin, Chen, Wu, and Ma, 2009a)
  - Accelerated Proximal Gradient (APG) (Toh and Yun, 2010)
  - All require rank- $k$  truncated SVD on **every** iteration

**Take away:** Provably accurate MC algorithms are still **too expensive** for large-scale or real-time matrix completion

**Question:** How can we **scale up** a given matrix completion algorithm and still **retain recovery guarantees**?

# Divide-Factor-Combine (DFC)

## Idea: Divide and conquer

- 1 Divide  $M$  into submatrices.
- 2 Factor each submatrix **in parallel**.
- 3 Combine submatrix estimates to recover  $L_0$ .

## Advantages

- Factoring a submatrix is often much cheaper than factoring  $M$
- Multiple submatrix factorizations can be carried out in parallel
- DFC works with **any** base MC algorithm
- With the right choice of division and recombination, yields recovery guarantees comparable to those of the base algorithm

# DFC-NYS: Generalized Nyström Decomposition

- 1 Choose a random column submatrix  $\mathbf{C} \in \mathbb{R}^{m \times l}$  and a random row submatrix  $\mathbf{R} \in \mathbb{R}^{d \times n}$  from  $\mathbf{M}$ . Call their intersection  $\mathbf{W}$ .

$$\mathbf{M} = \begin{bmatrix} \mathbf{W} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{M}_{21} \end{bmatrix} \quad \mathbf{R} = [\mathbf{W} \quad \mathbf{M}_{12}]$$

- 2 Recover the low rank components of  $\mathbf{C}$  and  $\mathbf{R}$  **in parallel** to obtain  $\hat{\mathbf{C}}$  and  $\hat{\mathbf{R}}$ 
  - **Reduced cost:** Expect  $\min(n/l, m/d)$  speed-up per iteration
- 3 Recover  $\mathbf{L}_0$  from  $\hat{\mathbf{C}}$ ,  $\hat{\mathbf{R}}$ , and their intersection  $\hat{\mathbf{W}}$

$$\hat{\mathbf{L}}^{nys} = \hat{\mathbf{C}}\hat{\mathbf{W}} + \hat{\mathbf{R}}$$

- Generalized Nyström method (Goreinov, Tyrtshnikov, and Zamarashkin, 1997)
  - **Minimal cost:**  $O(mk^2 + lk^2 + dk^2)$  where  $k = \text{rank}(\hat{\mathbf{L}}^{nys})$
- 4 **Ensemble:** Run  $p$  times in parallel and average estimates

# DFC-PROJ: Partition and Project

- 1 Randomly partition  $\mathbf{M}$  into  $n/l$  column submatrices  $\mathbf{M} = [\mathbf{C}_1 \ \mathbf{C}_2 \ \cdots \ \mathbf{C}_{n/l}]$  where each  $\mathbf{C}_i \in \mathbb{R}^{m \times l}$

- 2 Complete the submatrices **in parallel** to obtain

$$[\hat{\mathbf{C}}_1 \ \hat{\mathbf{C}}_2 \ \cdots \ \hat{\mathbf{C}}_{n/l}]$$

- 3 Recover a single factorization for  $\mathbf{M}$  by projecting each submatrix onto the column space of  $\hat{\mathbf{C}}_1$

$$\hat{\mathbf{L}}^{proj} = \hat{\mathbf{C}}_1 \hat{\mathbf{C}}_1^+ [\hat{\mathbf{C}}_1 \ \hat{\mathbf{C}}_2 \ \cdots \ \hat{\mathbf{C}}_{n/l}]$$

- **Minimal cost:**  $O(mk^2 + lk^2)$  where  $k = \text{rank}(\hat{\mathbf{L}}^{proj})$

- 4 **Ensemble:** Project onto column space of each  $\hat{\mathbf{C}}_j$  and average

## Advantages over DFC-NYS

- Utilizes the entire matrix  $\mathbf{M}$
- Column matrices are easier to factor when  $n > m$

# DFC: Does it work?

Yes, with high probability.

**Theorem** (Mackey, Talwalkar, and Jordan, 2011)

If  $\mathbf{L}_0$  is  $(\mu, r)$ -coherent and  $s$  entries of  $\mathbf{M} \in \mathbb{R}^{m \times n}$  are observed uniformly at random, then

$$l = O\left(\frac{\mu^2 r^2 n^2 \log^2(n)}{s \epsilon^2}\right)$$

random columns suffice to have

$$\|\hat{\mathbf{L}}^{proj} - \mathbf{L}_0\|_F \leq (2 + \epsilon) f(m, n) \Delta$$

with high probability when  $\|\mathbf{M} - \mathbf{L}_0\|_F \leq \Delta$  and the noisy nuclear norm heuristic is used as a base algorithm.

- Can sample vanishingly small fraction of columns ( $l/n \rightarrow 0$ ) whenever  $s = \omega(n \log^2(n))$
- Implies exact recovery for noiseless ( $\Delta = 0$ ) setting

# DFC: Does it work?

Yes, with high probability.

## Proof Ideas:

- 1 Uniform column/row sampling yields **submatrices with low coherence** (high spread of information) w.h.p.
  - 2 Each submatrix has **sufficiently many observed entries** w.h.p.  
⇒ Submatrix completion succeeds
  - 3 Uniform sampling of columns/rows **captures the full column/row space** of  $\mathbf{L}_0$  w.h.p.
    - Noisy analysis builds on randomized  $\ell_2$  regression work of Drineas, Mahoney, and Muthukrishnan (2008)
- ⇒ Generalized Nyström method and column projection succeed

## DFC Noisy Recovery Error

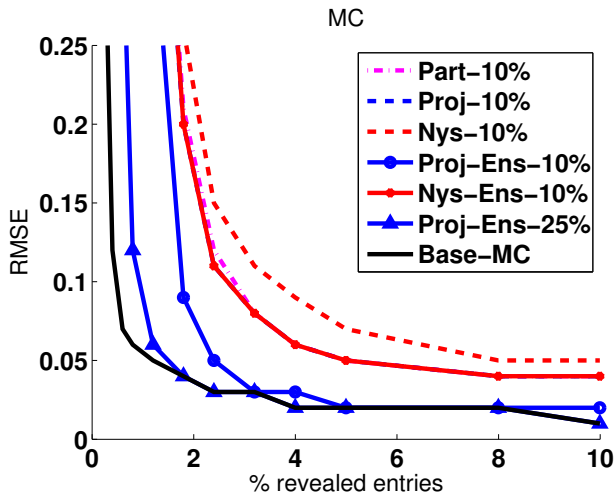


Figure: Recovery error of DFC relative to base algorithms with ( $m = 10K, r = 10$ ).

# DFC Speed-up

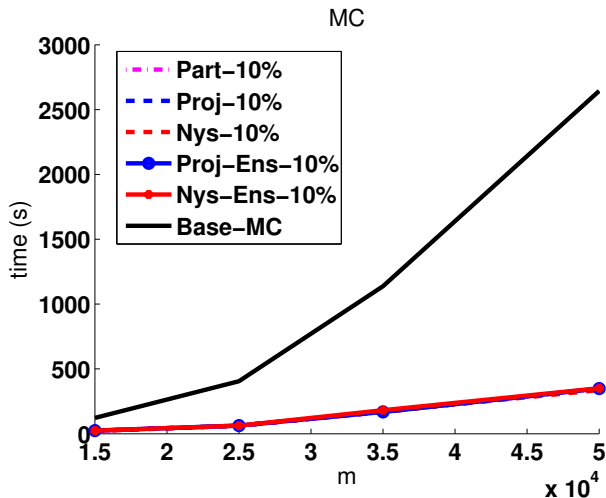


Figure: Speed-up over APG for random matrices with  $r = 0.001m$  and 4% of entries revealed.

# Application: Collaborative filtering

**Task:** Given a sparsely observed matrix of user-item ratings, predict the unobserved ratings

## Issues

- Full-rank rating matrix
- Noisy, non-uniform observations

## The Data

- **Netflix Prize Dataset**<sup>1</sup>
  - 100 million ratings in  $\{1, \dots, 5\}$
  - 17,770 movies, 480,189 users

---

<sup>1</sup><http://www.netflixprize.com/>

# Application: Collaborative filtering

Method	Netflix	
	RMSE	Time
APG	0.8433	2653.1s
DFC-NYS-25%	0.8832	890.9s
DFC-NYS-10%	0.9224	487.6s
DFC-NYS-ENS-25%	0.8486	964.3s
DFC-NYS-ENS-10%	0.8613	546.2s
DFC-PROJ-25%	0.8436	689.5s
DFC-PROJ-10%	0.8484	289.7s
DFC-PROJ-ENS-25%	0.8411	689.5s
DFC-PROJ-ENS-10%	0.8433	289.7s

# Noisy Robust Matrix Factorization

**Goal:** Given a matrix  $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 + \mathbf{Z}$  where  $\mathbf{L}_0$  is low-rank,  $\mathbf{S}_0$  is sparse, and  $\mathbf{Z}$  is entrywise noise, recover  $\mathbf{L}_0$

(Chandrasekaran, Sanghavi, Parrilo, and Willsky, 2009; Candès, Li, Ma, and Wright, 2011; Zhou, Li, Wright, Candès, and Ma, 2010)

- $\mathbf{S}_0$  can be viewed as an outlier/gross corruption matrix
  - Ordinary PCA breaks down in this setting
- **Harder than MC:** outlier locations are unknown
- **More expensive than MC:** dense, fully observed matrices

# How do we recover $\mathbf{L}_0$ ?

First attempt:

$$\begin{aligned} & \text{minimize}_{\mathbf{L}, \mathbf{S}} \quad \text{rank}(\mathbf{L}) + \lambda \text{card}(\mathbf{S}) \\ & \text{subject to} \quad \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F \leq \Delta. \end{aligned}$$

**Problem:** Intractable to solve!

**Solution:** Convex relaxation

$$\begin{aligned} & \text{minimize}_{\mathbf{L}, \mathbf{S}} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ & \text{subject to} \quad \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F \leq \Delta. \end{aligned}$$

where  $\|\mathbf{S}\|_1 = \sum_{ij} \mathbf{S}_{ij}$  is the  $\ell_1$  entrywise norm of  $\mathbf{S}$ .

**Question:** Does it work? **Yes!**

**Question:** Does it scale? **Not quite.**

**Idea:** Divide and conquer

# DFC Noisy Recovery Error

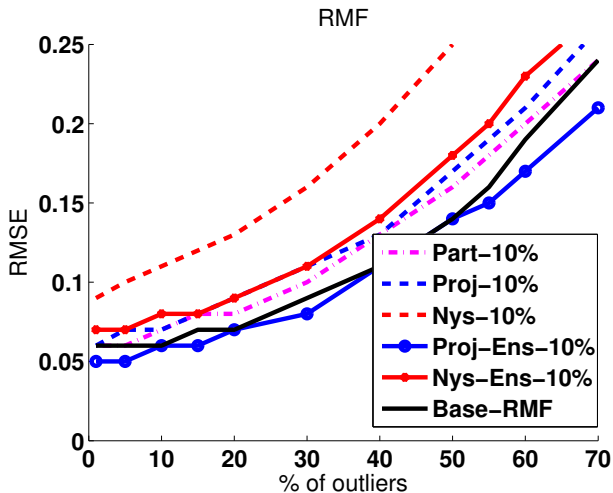


Figure: Recovery error of DFC relative to base algorithms with ( $m = 1K, r = 10$ ).

# DFC Speed-up

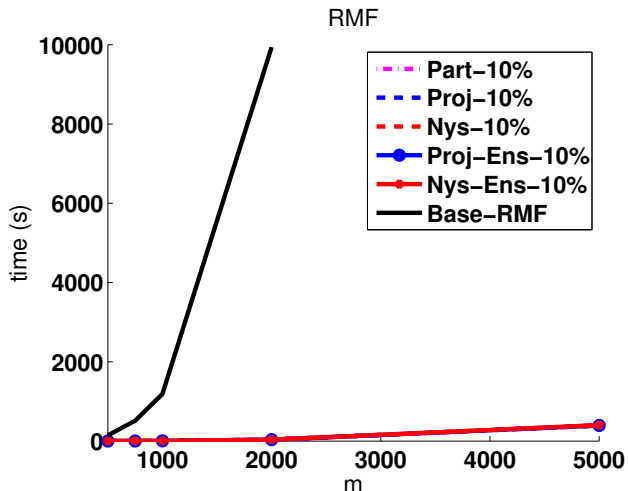


Figure: Speed-up over APG for random matrices with  $r = 0.01m$  and 10% of entries corrupted.

# Application: Video background modeling

## Task

- Each video frame forms one column of matrix  $\mathbf{M}$
- Decompose  $\mathbf{M}$  into stationary background  $\mathbf{L}_0$  and moving foreground objects  $\mathbf{S}_0$

## Issues

- Video is noisy
- Foreground corruption is often clustered, not uniform

# Application: Video background modeling

**Example:** Changes in illumination

## Specs

- 1.5 minutes of lobby surveillance (Li, Huang, Gu, and Tian, 2004)
- 1546 frames, 20480 pixels
- ALM running time: 5135.3s
- DFC-PROJ running time: 476.9s

# Application: Video background modeling

**Example:** Significant foreground variation

## Specs

- 1 minute of airport surveillance (Li, Huang, Gu, and Tian, 2004)
- 1000 frames, 25344 pixels
- ALM running time: 1871.0s
- DFC-PROJ running time: 392.8s

# Future Directions

## New Theory

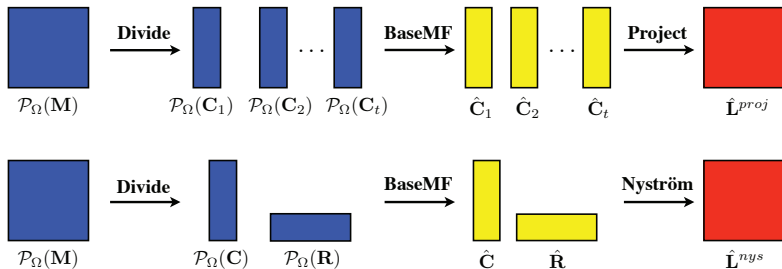
- Analyze statistical implications of divide and conquer algorithms
  - Trade-off between statistical and computational efficiency
  - Impact of ensembling
- Analysis of alternative optimization problems
  - Weighted trace norm for non-uniform sampling (Salakhutdinov and Srebro, 2010; Negahban and Wainwright, 2010)
  - Non-convex matrix completion with recovery guarantees (Keshavan, Montanari, and Oh, 2010)

## New Applications

- Practical problems with large-scale or real-time MF requirements

## The End

Thanks!



# References I

- Cai, J. F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4), 2010.
- Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
- Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.
- Candès, E.J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Sparse and low-rank matrix decompositions. In *Allerton Conference on Communication, Control, and Computing*, 2009.
- Chandrasekaran, V., Parrilo, P. A., and Willsky, A. S. Latent variable graphical model selection via convex optimization. preprint, 2010.
- Drineas, P., Mahoney, M. W., and Muthukrishnan, S. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30:844–881, 2008.
- Fazel, M., Hindi, H., and Boyd, S. P. A rank minimization heuristic with application to minimum order system approximation. In *In Proceedings of the 2001 American Control Conference*, pp. 4734–4739, 2001.
- Goreinov, S. A., Tyrtyshnikov, E. E., and Zamarashkin, N. L. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1-3):1–21, 1997.
- Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 99: 2057–2078, 2010.
- Li, L., Huang, W., Gu, I. Y. H., and Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.
- Lin, Z., Chen, M., Wu, L., and Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report UILU-ENG-09-2215, 2009a.
- Lin, Z., Ganesh, A., Wright, J., Wu, L., Chen, M., and Ma, Y. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. UIUC Technical Report UILU-ENG-09-2214, 2009b.
- Mackey, L., Talwalkar, A., and Jordan, M. I. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems 24*. 2011.

# References II

- Min, K., Zhang, Z., Wright, J., and Ma, Y. Decomposing background topics from keywords by principal component pursuit. In *Conference on Information and Knowledge Management*, 2010.
- Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. [arXiv:1009.2118v2\[cs.IT\]](https://arxiv.org/abs/1009.2118v2), 2010.
- Peng, Y., Ganesh, A., Wright, J., Xu, W., and Ma, Y. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Conference on Computer Vision and Pattern Recognition*, 2010.
- Recht, B. A simpler approach to matrix completion. [arXiv:0910.0651v2\[cs.IT\]](https://arxiv.org/abs/0910.0651v2), 2009.
- Salakhutdinov, R. and Srebro, N. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *NIPS*, 2010.
- Toh, K. and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615–640, 2010.
- Zhou, Z., Li, X., Wright, J., Candès, E. J., and Ma, Y. Stable principal component pursuit. [arXiv:1001.2363v1\[cs.IT\]](https://arxiv.org/abs/1001.2363v1), 2010.

# Noisy Principal Component Pursuit: Does it work?

Yes, with high probability.

**Theorem** (Zhou, Li, Wright, Candès, and Ma, 2010)

If  $\mathbf{L}_0$  is  $(\mu, r)$ -coherent and  $\mathbf{S}_0 \in \mathbb{R}^{m \times n}$  contains  $s$  non-zero entries with uniformly distributed locations, then if

$$r \leq \frac{\rho_r m}{\mu \log^2 n} \quad \text{and} \quad s \leq \rho_s mn$$

the minimizer to the problem

$$\begin{aligned} & \text{minimize}_{\mathbf{L}, \mathbf{S}} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ & \text{subject to} \quad \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F \leq \Delta. \end{aligned}$$

with  $\lambda = 1/\sqrt{n}$  satisfies

$$\|\hat{\mathbf{L}} - \mathbf{L}_0\|_F \leq f(m, n)\Delta$$

with high probability when  $\|\mathbf{M} - \mathbf{L}_0 - \mathbf{S}_0\|_F \leq \Delta$ .

# Noisy Principal Component Pursuit: Does it scale?

## Not quite...

- Standard interior point methods:  $O(n^6)$  (Chandrasekaran, Sanghavi, Parrilo, and Willsky, 2009)
- More efficient, tailored algorithms:
  - Accelerated Proximal Gradient (APG) (Lin, Ganesh, Wright, Wu, Chen, and Ma, 2009b)
  - Augmented Lagrange Multiplier (ALM) (Lin, Chen, Wu, and Ma, 2009a)
  - Require rank- $k$  truncated SVD on **every** iteration
  - Best case  $SVD(m, n, k) = O(mnk)$

**Take away:** Provably accurate RMF algorithms are still **too expensive** for large-scale or real-time matrix factorization

**Idea:** Leverage the divide-and-conquer techniques developed for MC in the RMF setting

# Divide-Factor-Combine (DFC)

**Idea:** Same divide-and-conquer strategies apply

- 1 Divide  $M$  into submatrices.
- 2 Factor each submatrix **in parallel**.
  - Using any base RMF algorithm
- 3 Combine submatrix estimates to recover  $L_0$ .
  - Using DFC-NYS or DFC-PROJ

**Question:** Is it efficient?

- Yes, as in the MC case.

**Question:** Does it work?

# DFC: Does it work?

Yes, with high probability.

**Theorem** (Mackey, Talwalkar, and Jordan, 2011)

If  $\mathbf{L}_0$  is  $(\mu, r)$ -coherent and  $\mathbf{S}_0 \in \mathbb{R}^{m \times n}$  contains  $s \leq \rho_s mn$  non-zero entries with uniformly distributed locations, then

$$l = O\left(\frac{\mu^2 r^2 \log^2(n)}{\epsilon^2}\right)$$

columns suffice to have

$$\|\hat{\mathbf{L}}^{proj} - \mathbf{L}_0\|_F \leq (2 + \epsilon)f(m, n)\Delta$$

with high probability when  $\|\mathbf{M} - \mathbf{L}_0 - \mathbf{S}_0\|_F \leq \Delta$  and noisy principal component pursuit is used as the base algorithm.

- Implies exact recovery for noiseless ( $\Delta = 0$ ) setting
- Similar result holds for DFC-NYS