Divide-and-Conquer Matrix Factorization

Lester Mackey[†]

Collaborators: Ameet Talwalkar[*] and Michael I. Jordan[*]

[†]Stanford University [*]UC Berkeley

May 15, 2013

Motivation: Large-scale Matrix Completion

Goal: Estimate a matrix $\mathbf{L}_0 \in \mathbb{R}^{m \times n}$ given a subset of its entries

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

Examples

- 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
- Link prediction: Is user i friends with user j?
  - Facebook: 500 million users

Motivation: Large-scale Matrix Completion

Goal: Estimate a matrix $\mathbf{L}_0 \in \mathbb{R}^{m \times n}$ given a subset of its entries

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

State of the art MC algorithms

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

This talk

- Present divide and conquer approaches for scaling up any MC algorithm while maintaining strong estimation guarantees

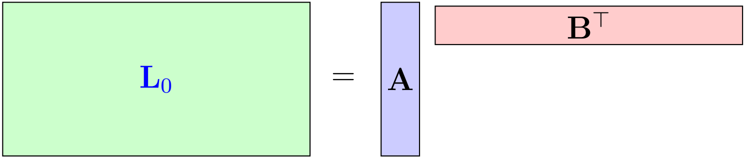
Exact Matrix Completion

Goal: Estimate a matrix $\mathbf{L}_0 \in \mathbb{R}^{m\times n}$ given a subset of its entries

Noisy Matrix Completion

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

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



  - Factored form: $\mathbf{A}\mathbf{B}^\top$ 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 Ω is drawn uniformly at random:

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

What can go wrong?

Bad spread of information

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

- 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 $\mathrm{rank}(\mathbf{L}) = r$ is *incoherent* if

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

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

How do we estimate $\mathbf{L}_0$?

First attempt:

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

Problem: Computationally intractable!

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

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

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 $\mathbf{L}_0$ with rank r is incoherent, $s \gtrsim rn\log^2(n)$ 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). See also 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): $\mathrm{O}(|\Omega|(m+n)^3 + |\Omega|^2(m+n)^2 + |\Omega|^3)$
- 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: Many provably accurate MC algorithms are too expensive for large-scale or real-time matrix completion

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

Divide-Factor-Combine (DFC)

Our Solution: Divide and conquer

1. Divide $\mathbf{M}$ into submatrices.
2. Complete each submatrix **in parallel**.
3. Combine submatrix estimates to estimate $\mathbf{L}_0$.

Advantages

- Submatrix completion is often much cheaper than completing $\mathbf{M}$
- Multiple submatrix completions can be carried out in parallel
- DFC works with **any** base MC algorithm
- With the right choice of division and recombination, yields estimation guarantees comparable to those of the base algorithm

DFC-PROJ: Partition and Project

1. Randomly partition $\mathbf{M}$ into t column submatrices $\mathbf{M} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_t \end{bmatrix}$ where each $\mathbf{C}_i \in \mathbb{R}^{m \times l}$
2. Complete the submatrices **in parallel** to obtain
$$\begin{bmatrix} \hat{\mathbf{C}}_1 & \hat{\mathbf{C}}_2 & \cdots & \hat{\mathbf{C}}_t \end{bmatrix}$$
   - Reduced cost: Expect t-fold speed-up per iteration
   - Parallel computation: Pay cost of one cheaper MC
3. Project submatrices onto a single low-dimensional column space
   - Estimate column space of $\mathbf{L}_0$ with column space of $\hat{\mathbf{C}}_1$
$$\hat{\mathbf{L}}^{proj} = \hat{\mathbf{C}}_1 \hat{\mathbf{C}}_1^{+} \begin{bmatrix} \hat{\mathbf{C}}_1 & \hat{\mathbf{C}}_2 & \cdots & \hat{\mathbf{C}}_t \end{bmatrix}$$
   - Common technique for randomized low-rank approximation

     (Frieze, Kannan, and Vempala, 1998)
   - Minimal cost: $\mathrm{O}(mk^2 + lk^2)$ where $k = \operatorname{rank}(\hat{\mathbf{L}}^{proj})$
4. **Ensemble:** Project onto column space of each $\hat{\mathbf{C}}_j$ and average

DFC: Does it work?

Yes, with high probability.

Theorem (Mackey, Talwalkar, and Jordan, 2011)

If $\mathbf{L}_0$ with rank r is incoherent and $s = \omega(r^2 n \log^2(n)/\epsilon^2)$ entries of $\mathbf{M} \in \mathbb{R}^{m \times n}$ are observed uniformly at random, then $l = o(n)$ 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 ($l/n \to 0$)
- Implies exact recovery for noiseless ($\Delta = 0$) setting

DFC: Does it work?

Yes, with high probability.

Proof Ideas:

1. If $\mathbf{L}_0$ is incoherent (has good spread of information), its partitioned submatrices are incoherent w.h.p.
2. Each submatrix has sufficiently many observed entries w.h.p.

$\Rightarrow$ Submatrix completion succeeds

3. Random submatrix captures the full column space of $\mathbf{L}_0$ w.h.p.
   - Analysis builds on randomized ℓ_2 regression work of Drineas, Mahoney, and Muthukrishnan (2008)

$\Rightarrow$ Column projection succeeds

DFC Estimation Error

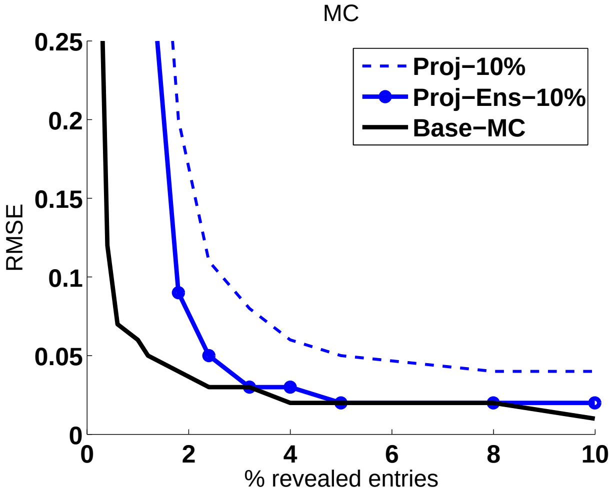


Figure : Estimation error of DFC and base algorithm (APG) with $m = 10K$ and $r = 10$.

DFC Speed-up

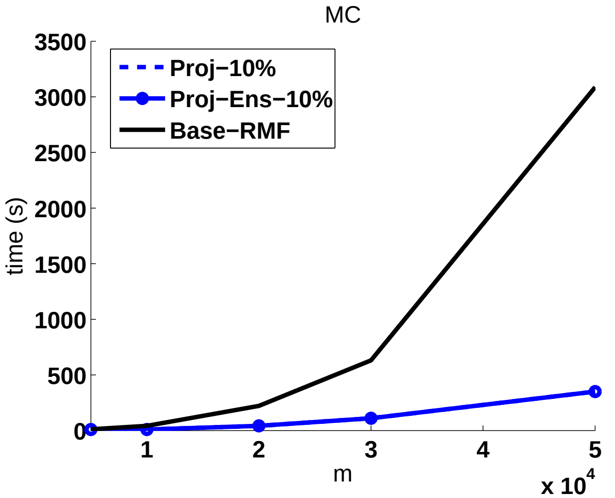


Figure : Speed-up over base algorithm (APG) for random matrices with $r = 0.001m$ and 4% of entries revealed.

Application: Collaborative filtering

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

Challenges

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

The Data

- **Netflix Prize Dataset**[1]
  - 100 million ratings in $\{1, \ldots, 5\}$
  - 17,770 movies, 480,189 users

[1] `http://www.netflixprize.com/`

Application: Collaborative filtering

Method	Netflix	
	RMSE	**Time**
Base algorithm (APG)	0.8433	2653.1s
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

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)

Examples:

- Background modeling/foreground activity detection

$\mathbf{M}$

$\mathbf{L}_0$

$\mathbf{S}$

(Candès, Li, Ma, and Wright, 2011)

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)

$\mathbf{M}$ $\mathbf{L}_0$ $\mathbf{S}_0$

- $\mathbf{S}_0$ 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:

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

Problem: Computationally intractable!

Solution: Convex relaxation

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

where $\|\mathbf{S}\|_1 = \sum_{ij} \mathbf{S}_{ij}$ is the ℓ_1 entrywise norm of $\mathbf{S}$.

Question: Does it work?

- Will noisy *Principal Component Pursuit* (PCP) recover $\mathbf{L}_0$?

Question: Is it efficient?

- Can noisy PCP scale to large RMF problems?

Noisy Principal Component Pursuit: Does it work?

Yes, with high probability.

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

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

$$r = O\left(m/\log^2 n\right) \quad \text{and} \quad s \le c \cdot mn,$$

the minimizer to the problem

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

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

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

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

- See also Agarwal, Negahban, and Wainwright (2011)

Noisy Principal Component Pursuit: Is it efficient?

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 $\text{SVD}(m, n, k) = O(mnk)$

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

DFC: Does it work?

Yes, with high probability.

Theorem (Mackey, Talwalkar, and Jordan, 2011)

If $\mathbf{L}_0$ with rank r is incoherent, and $\mathbf{S}_0 \in \mathbb{R}^{m\times n}$ contains $s \leq c \cdot mn$ non-zero entries with uniformly distributed locations, then

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

random columns suffice to have

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

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

- Can sample polylogarithmic number of columns
- Implies exact recovery for noiseless ($\Delta = 0$) setting

DFC Estimation Error

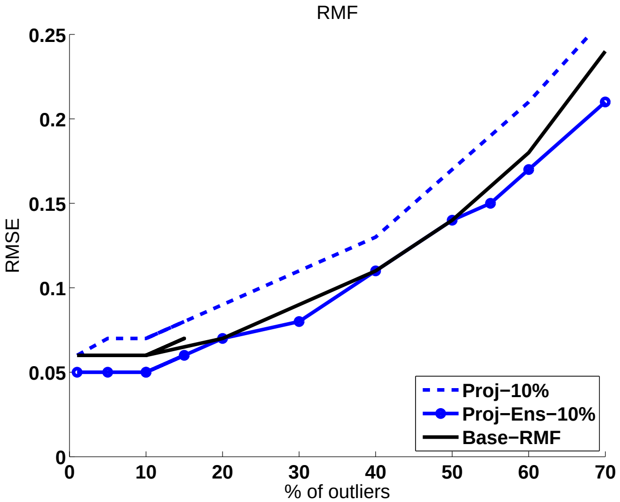


Figure : Estimation error of DFC and base algorithm (APG) with $m = 1K$ and $r = 10$.

DFC Speed-up

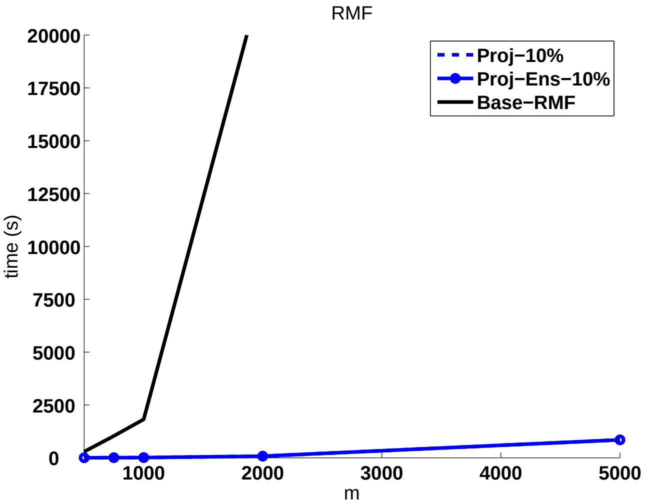


Figure : Speed-up over base algorithm (APG) for random matrices with $r = 0.01m$ and 10% of entries corrupted.

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 $\mathbf{S}_0$

$\mathbf{M}$

$\mathbf{L}_0$

$\mathbf{S}_0$

Challenges

- 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
- Base algorithm: 1.5 hours
- DFC: 8 minutes

Application: Video background modeling

Example: Significant foreground variation

Specs

- 1 minute of airport surveillance (Li, Huang, Gu, and Tian, 2004)
- 1000 frames, 25344 pixels
- Base algorithm: half an hour
- DFC: 7 minutes

Future Directions

New Theory

- Analyze statistical implications of divide and conquer algorithms
  - Trade-off between statistical and computational efficiency
  - Impact of ensembling

New Divide-and-Conquer Strategies

- Other ways to reduce computation while preserving accuracy

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 $\mathbf{W}$.

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

2. Recover the low rank components of $\mathbf{C}$ and $\mathbf{R}$ **in parallel** to obtain $\hat{\mathbf{C}}$ and $\hat{\mathbf{R}}$
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, Tyrtyshnikov, and Zamarashkin, 1997)
   - Minimal cost: $O(mk^2 + lk^2 + dk^2)$ where $k = \mathrm{rank}(\hat{\mathbf{L}}^{nys})$
4. **Ensemble:** Run p times in parallel and average estimates

Future Directions

New Theory

- Analyze statistical implications of divide and conquer algorithms
    - Trade-off between statistical and computational efficiency
    - Impact of ensembling

New Divide-and-Conquer Strategies

- Other ways to reduce computation while preserving accuracy

New Datasets / Applications

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

Future Directions

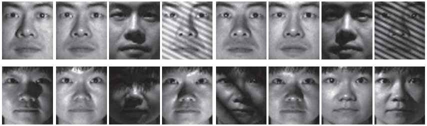
New Datasets / Applications

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

Subspace Segmentation (with Yadong Mu and Shih-Fu Chang)

- Given (corrupted) data points drawn from a union of subspaces, identify the subspaces
- Low-rank representation (Liu, Lin, and Yu, 2010)

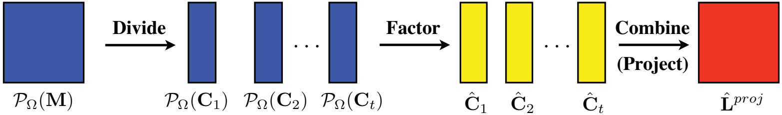
$$\begin{aligned} \text{minimize}_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_{2,1} \\ \text{subject to} \quad & \mathbf{M} = \mathbf{M}\mathbf{L} + \mathbf{S} \end{aligned}$$



- Applications to face clustering, video content detection, multimedia event detection, and image tagging (20K Flickr images: LRR 1.5 days $\rightarrow$ DFC-LRR 1 hour)

The End

Thanks!



References I

Agarwal, A., Negahban, S., and Wainwright, M. J. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. In *International Conference on Machine Learning*, 2011.

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.

Frieze, A., Kannan, R., and Vempala, S. Fast Monte Carlo algorithms for finding low-rank approximations. In *Foundations of Computer Science*, 1998.

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.

References II

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.

Liu, G., Lin, Z., and Yu, Y. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

Mackey, L., Talwalkar, A., and Jordan, M. I. Divide-and-conquer matrix factorization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 1134–1142. 2011.

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]`, 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. In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 1518 –1522, 2010.