Matrix Decomposition: Applications and Algorithms

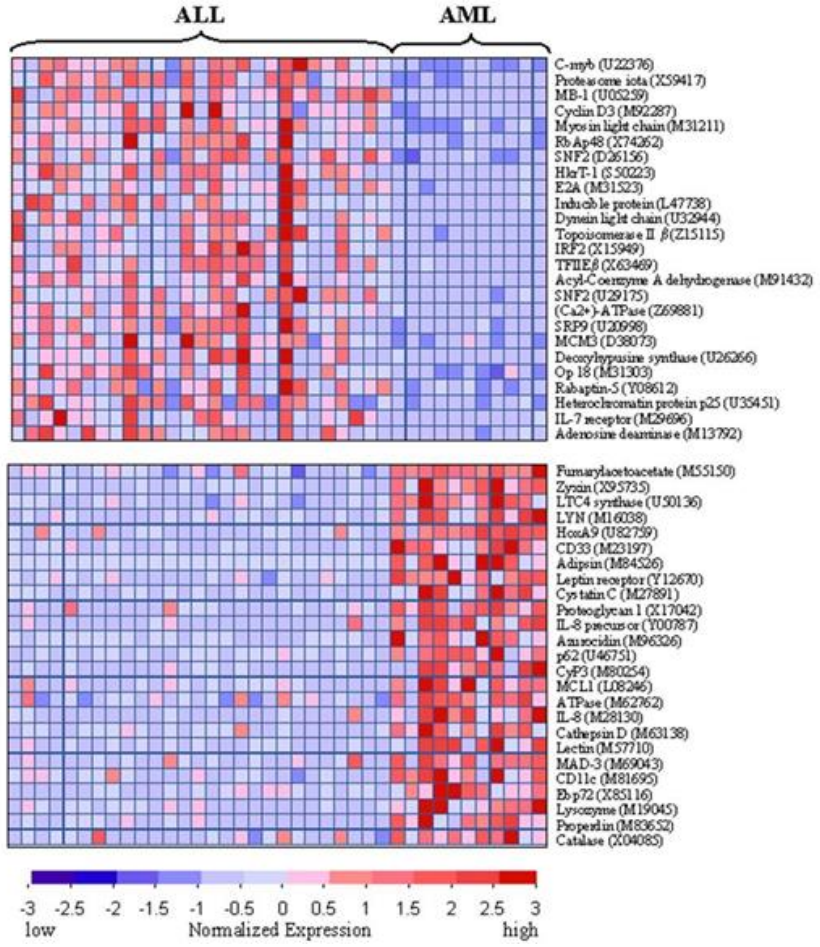
Praneeth Netrapalli

Microsoft Research India

Extracting Information from Data Matrices

- Huge amounts of data available in a lot of domains: online search, social media, health records etc.
- Information ***hidden*** in the data; Very useful and has many applications
- Specific context: Data as matrices

For example



Gene expression levels

Image credit: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, T. R. Golub, et al., Science 286 (531) 1999

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

User item ratings

Image credit: https://katbailey.github.io/post/matrix-factorization-with-tensorflow/

Frames

Frames in a video

Matrix Decomposition

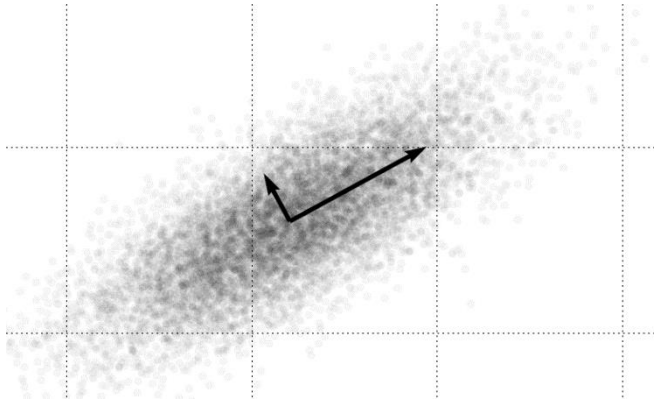
$$M \sim UV^T$$
$$s.t. U \in S_1, V \in S_2$$

$$M \sim U + V$$
$$s.t. U \in S_1, V \in S_2$$

Helps extract information from matrices in several applications

Classical decompositions: Eigen, Schur, Cholesky, QR, Jordan,…

Canonical Example: Principal Component Analysis



Motivation: Powerful tool for dimensionality reduction

Canonical Example: Principal Component Analysis

$$\min_{U,V} \left\| M - U V^T \right\|_F^2$$

Motivation: Powerful tool for dimensionality reduction

K-means Clustering

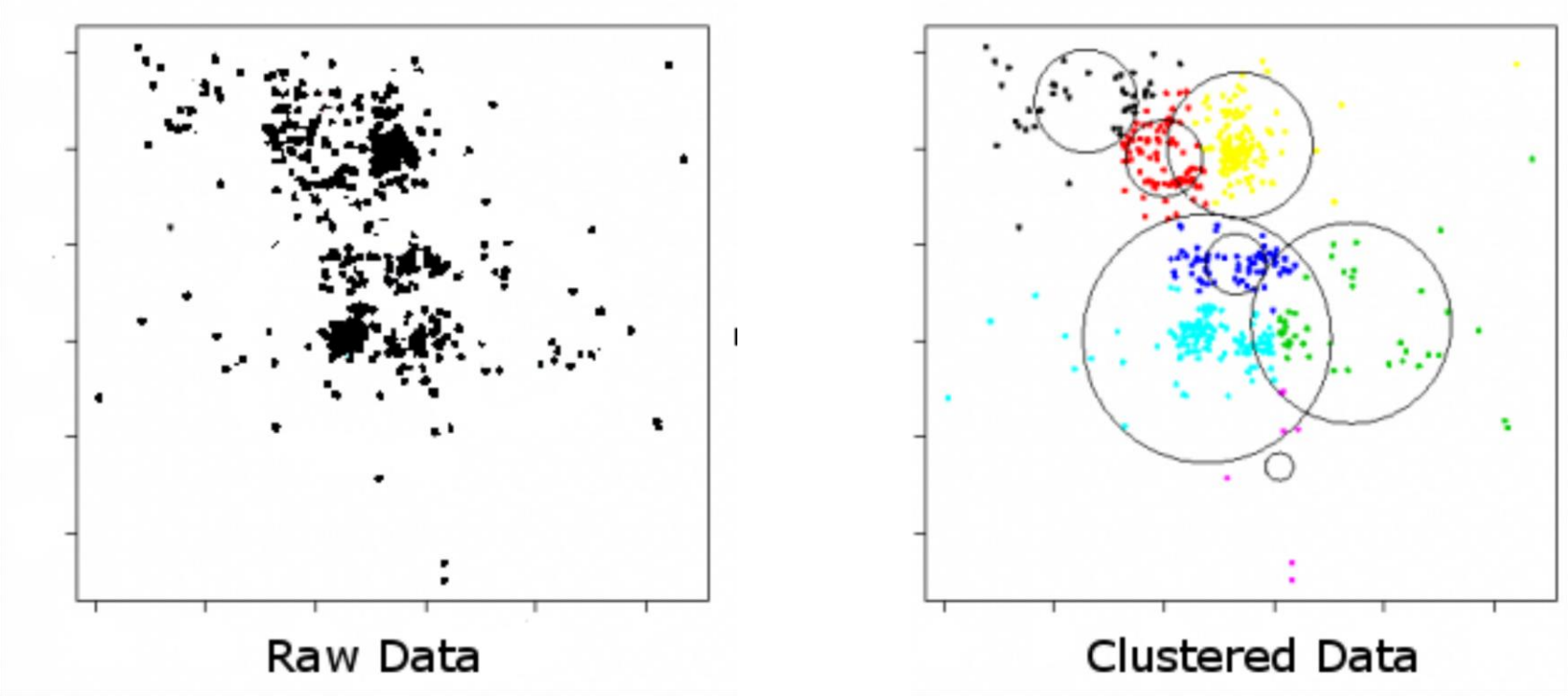


Image credit: http://stackoverflow.com/questions/1441319/whats-the-best-approach-to-recognize-patterns-in-data-and-whats-the-best-way

K-means Clustering

$$\min_{U,V} \left\| M - U V^T \right\|_F^2$$

s.t. each column of V^T is an indicator vector

In a Nutshell

- Many more applications: Matrix completion, Nonnegative PCA, Sparse coding, Subspace clustering …

- Several approaches
  - Classical methods are either algebraic or based on convex relaxations
  - Not designed for modern data applications
    - Many passes over data not feasible
    - Noisy data

Alternating Minimization (AltMin)

$$\min_{U \in S_1, V \in S_2} f(U, V)$$

1. Choose a random V
2. Fix V and optimize over U
3. Fix U and optimize over V
4. Repeat Steps 2 & 3

In many cases, steps 2 and 3 efficiently solvable

Empirically widely used (e.g., k-means); has good performance

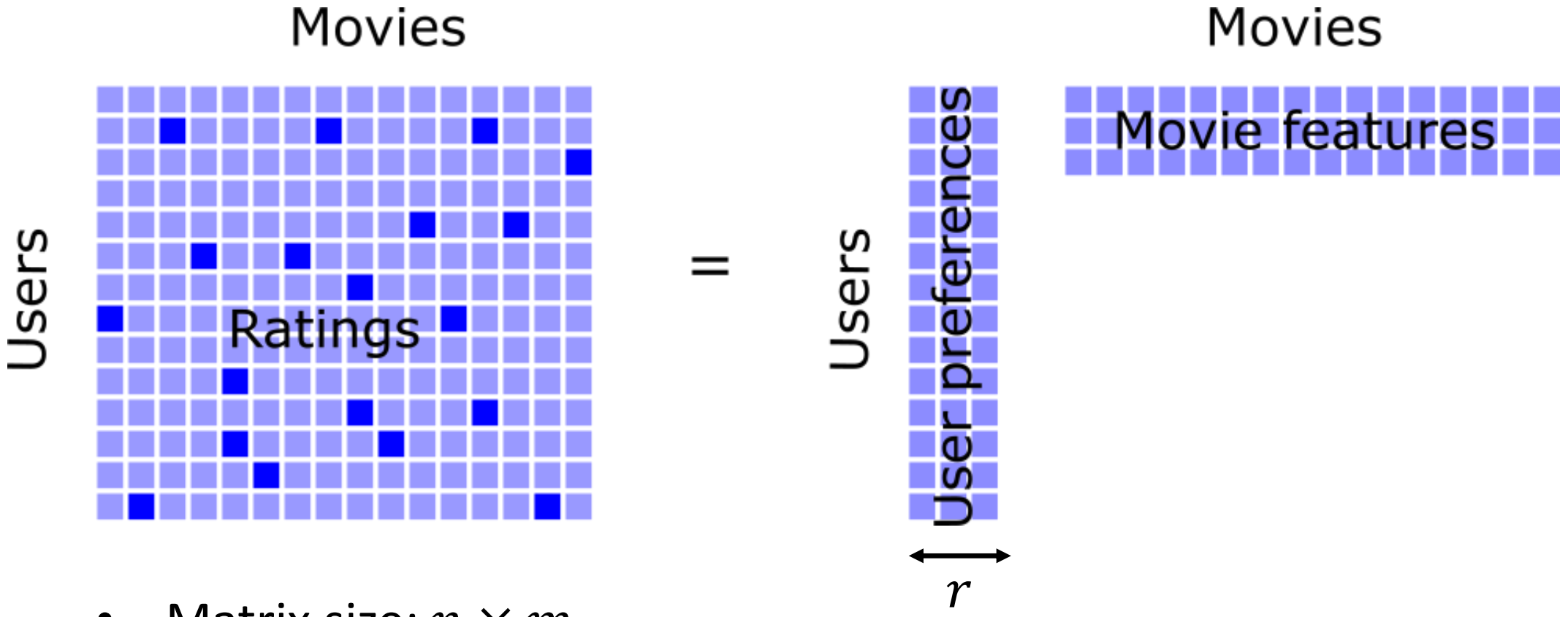
Theoretically significant progress in the last decade

This talk

- Matrix completion
- Robust PCA
- Sparse coding
- Conclusion

Part I: Matrix Completion

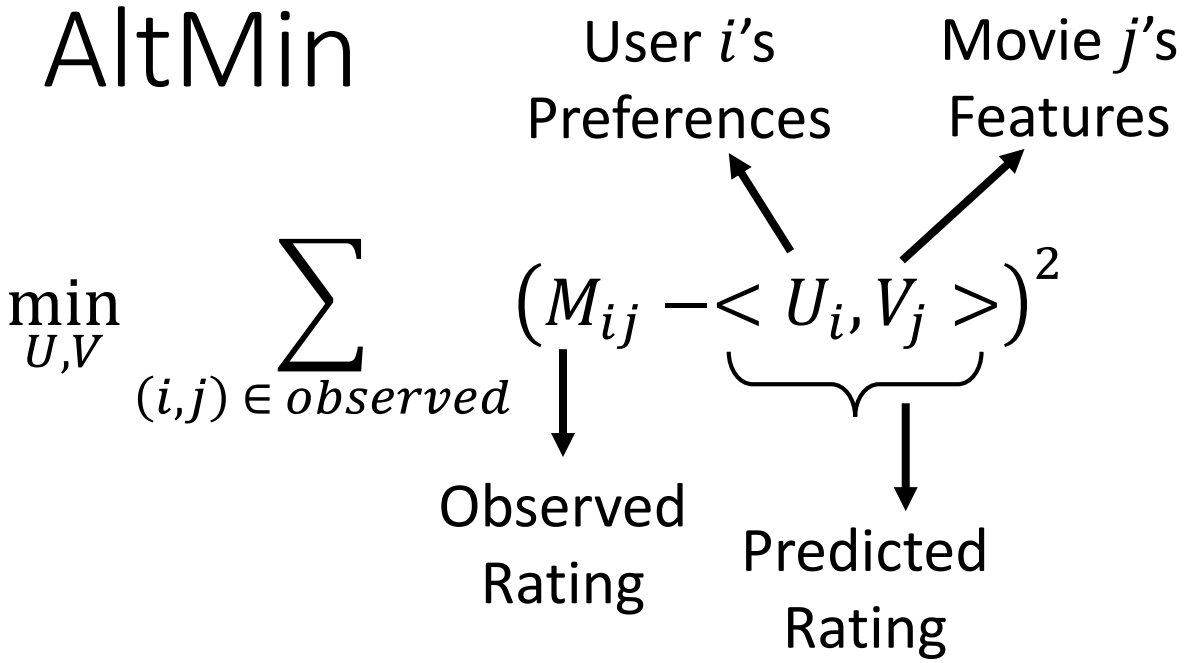
Matrix Completion



- Matrix size: $n \times m$
- Wlog $n \geq m$
- Rank = $r \ll n, m$

Goal: Infer the entire matrix with the minimum number of observations.

AltMin


User i's Preferences
Movie j's Features
$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$
Observed Rating
Predicted Rating

AltMin

User i's Preferences

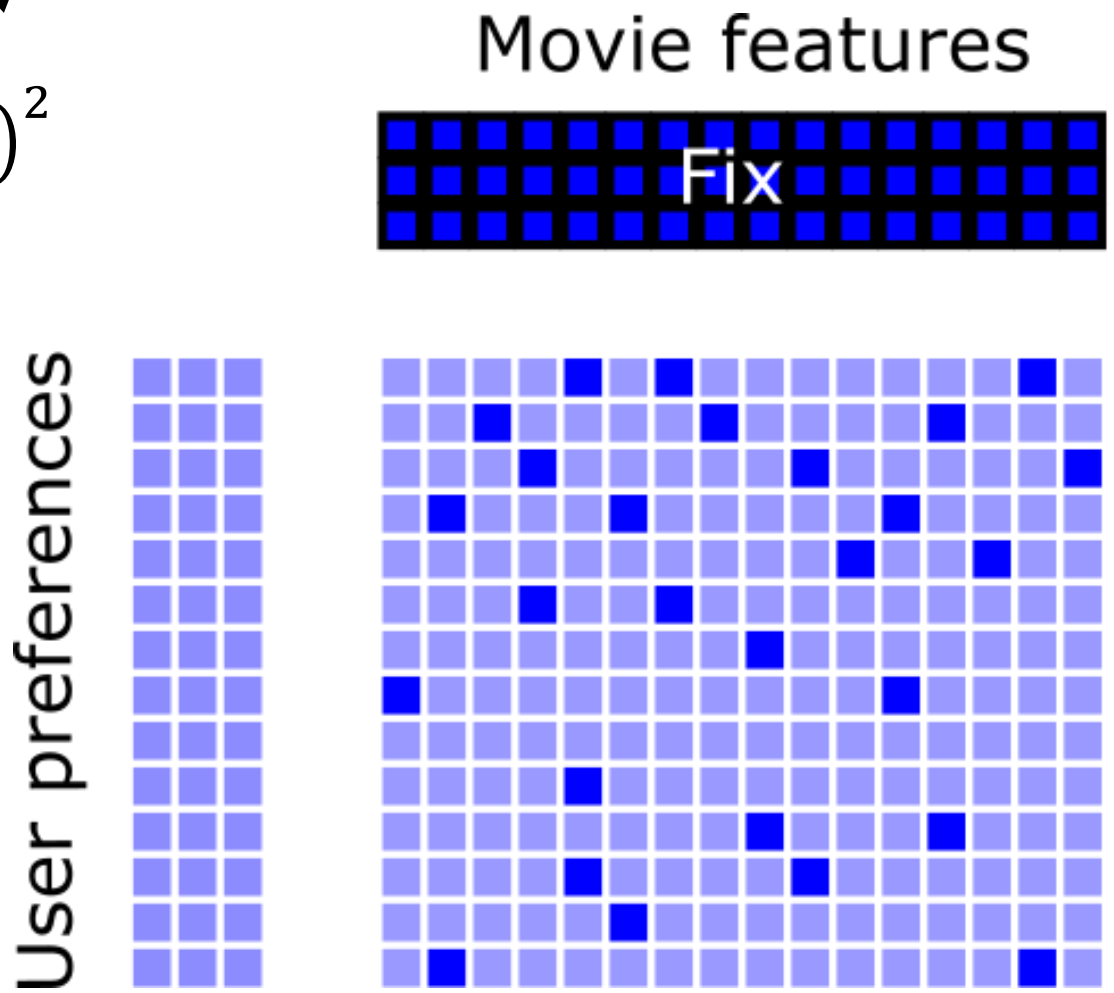
Movie j's Features

$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

Observed Rating

Predicted Rating

Fix V, optimize over U



AltMin

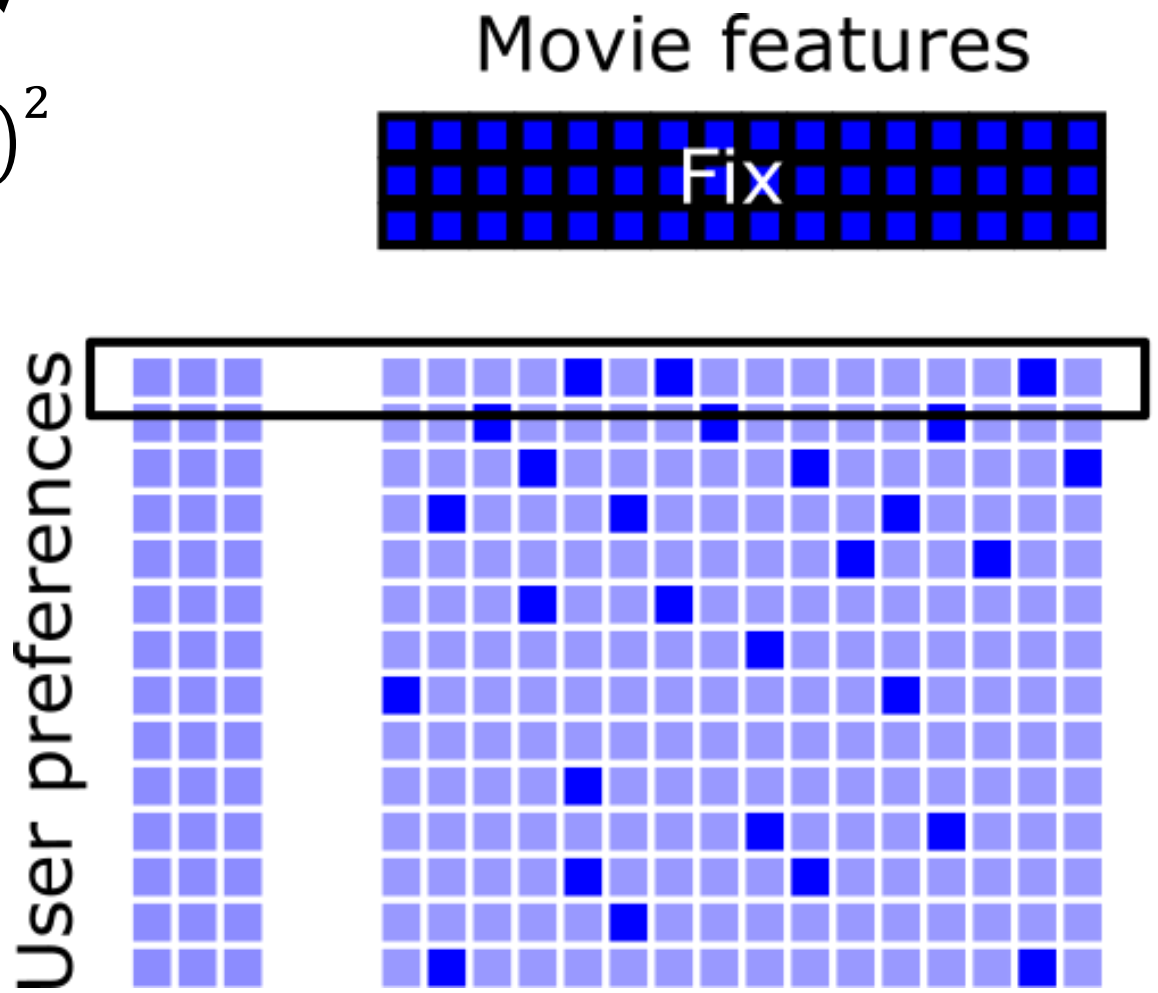
$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

User i's Preferences

Movie j's Features

Observed Rating

Predicted Rating



Fix V, optimize over U

Decouples into multiple least squares problems

AltMin

User i's Preferences

Movie j's Features

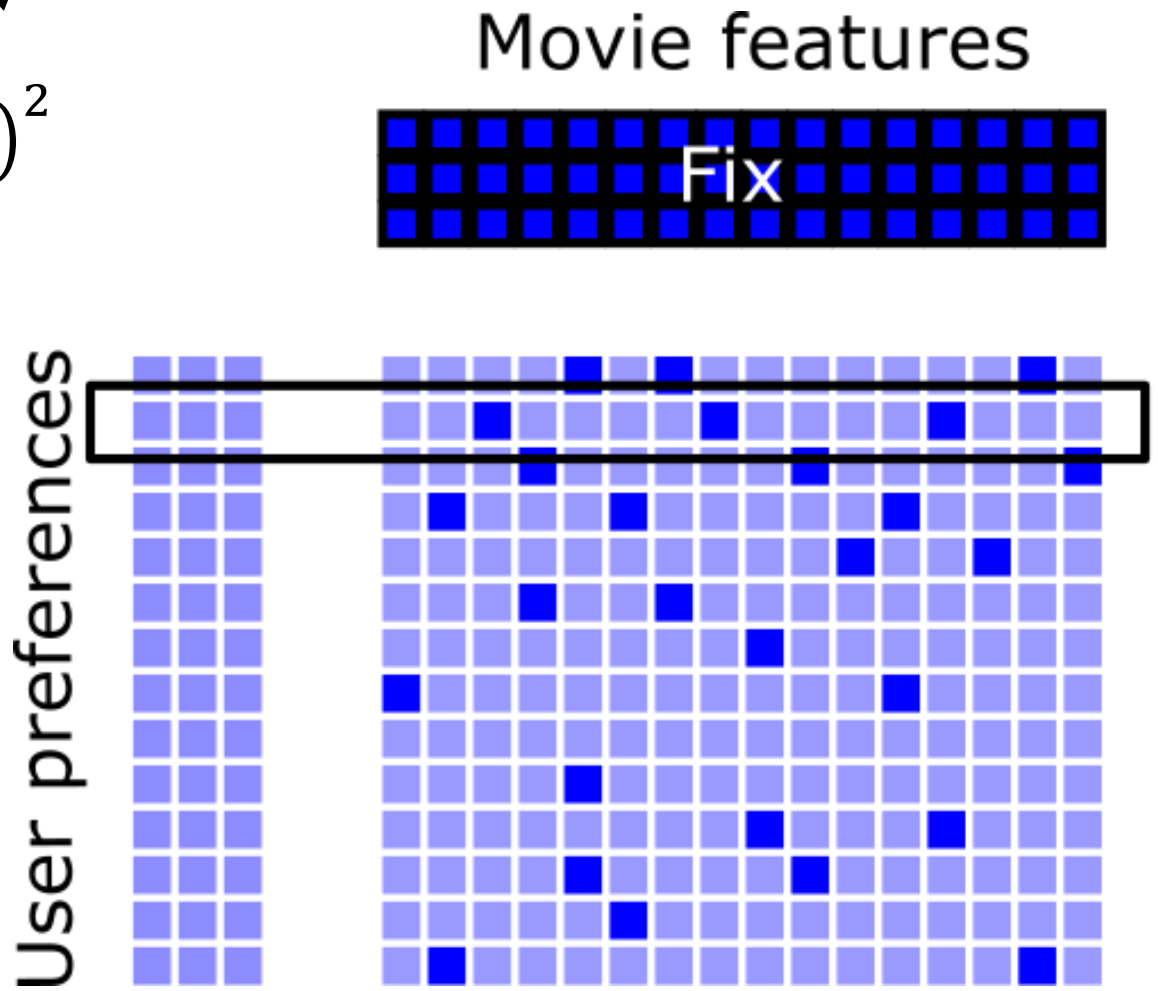
$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

Observed Rating

Predicted Rating

Fix V, optimize over U

Decouples into multiple least squares problems



AltMin

User i's Preferences

Movie j's Features

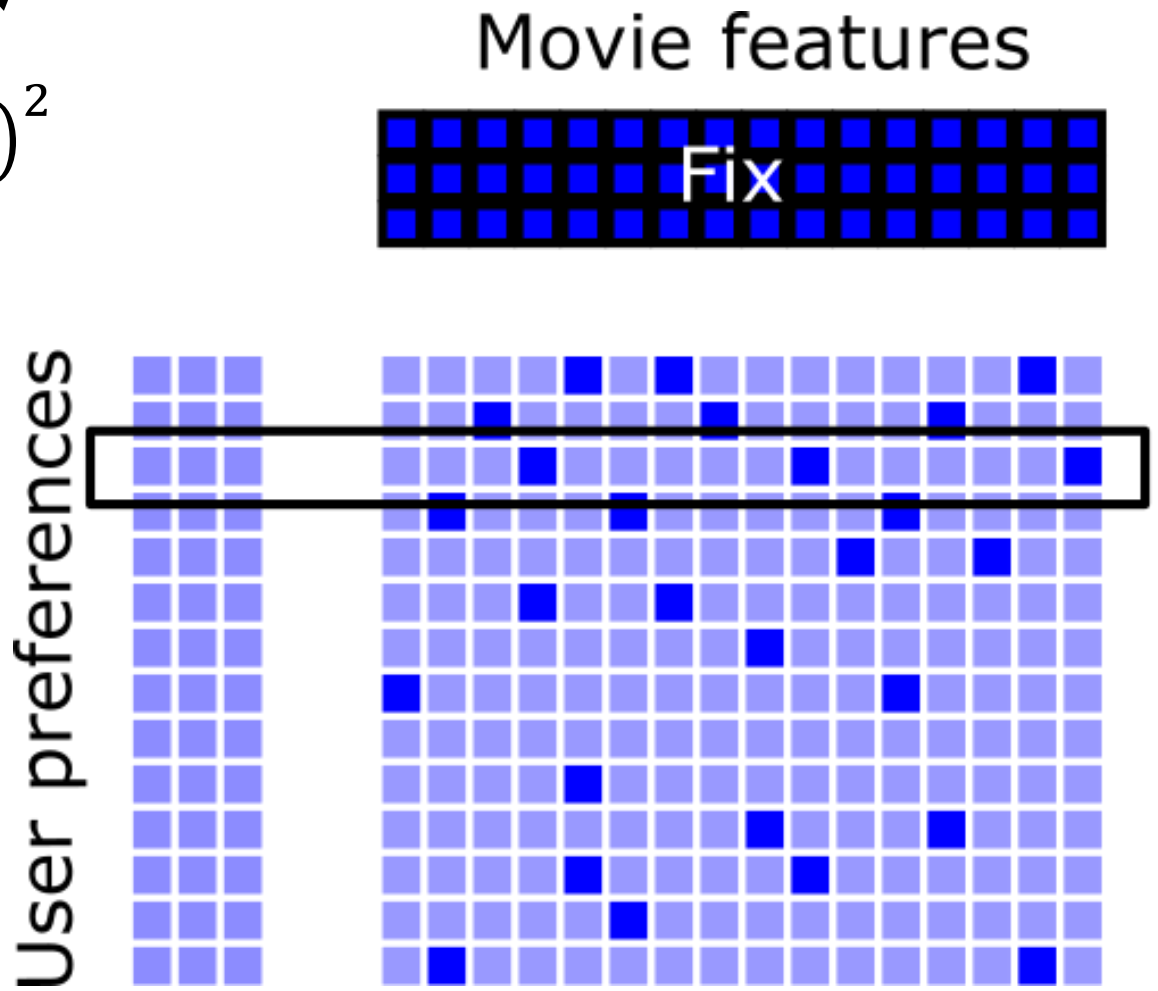
$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

Observed Rating

Predicted Rating

Fix V, optimize over U

Decouples into multiple least squares problems



AltMin

User i's Preferences

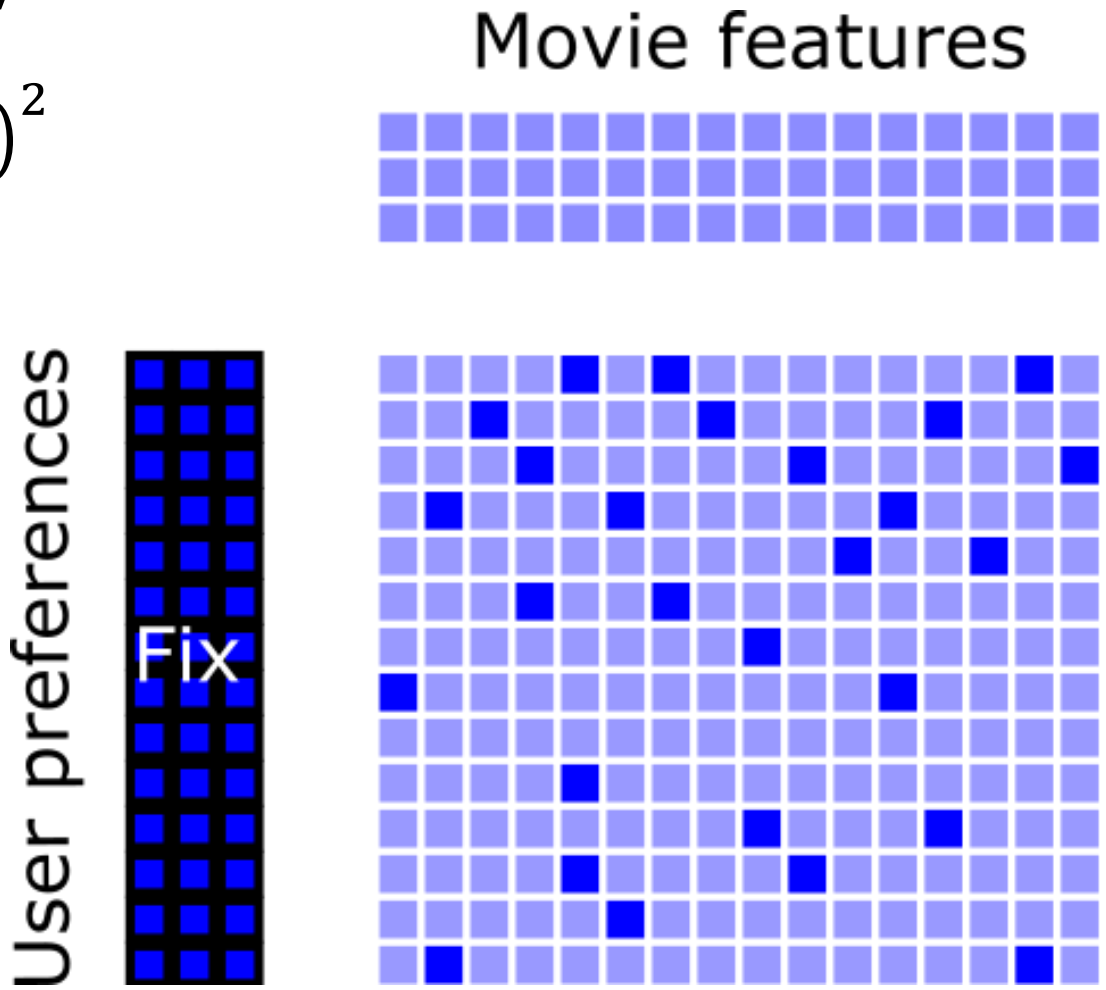
Movie j's Features

$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

Observed Rating

Predicted Rating

Fix U, optimize over V



AltMin

User i's Preferences

Movie j's Features

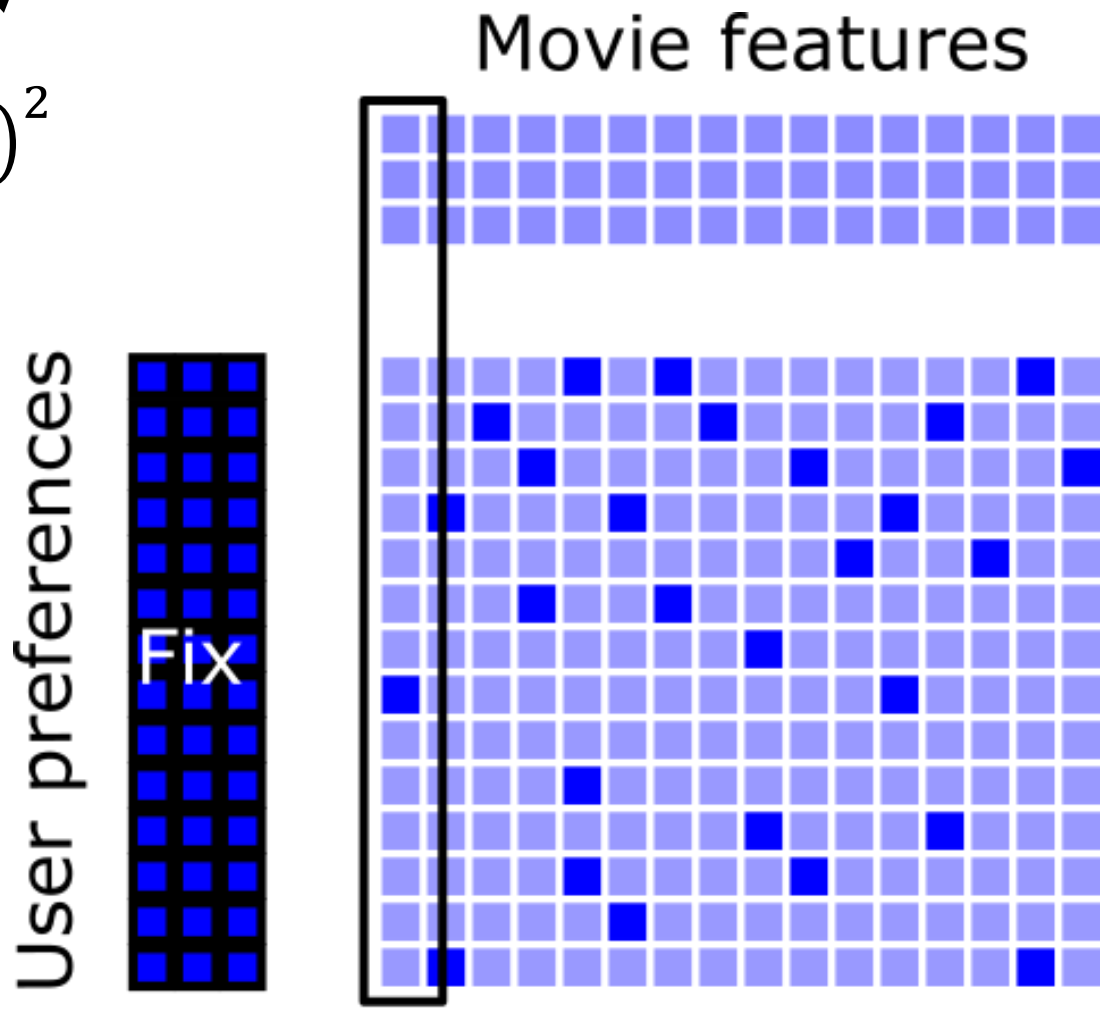
$$\min_{U,V} \sum_{(i,j) \in observed} \left(M_{ij} - < U_i, V_j >\right)^2$$

Observed Rating

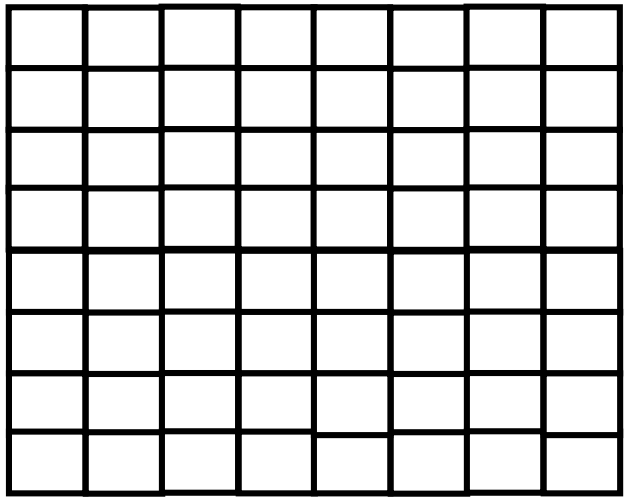
Predicted Rating

Fix U, optimize over V

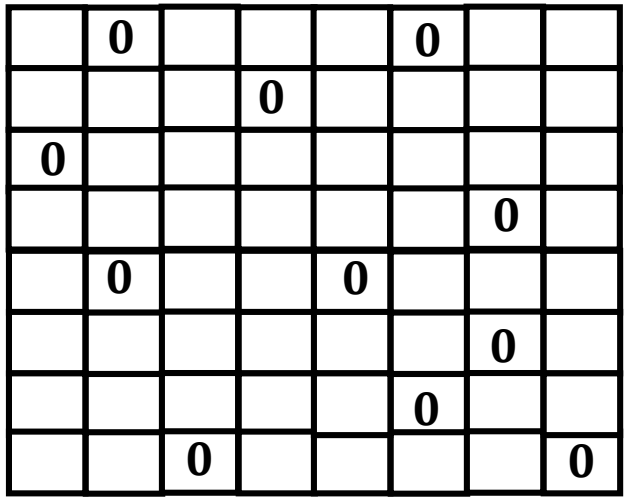
Again decouples into multiple least squares problems



Need for Regularity Assumptions



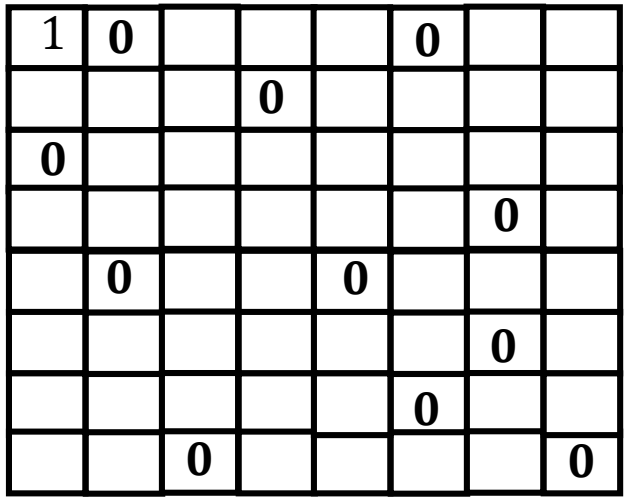
Need for Regularity Assumptions



Need for Regularity Assumptions

?	**0**				**0**		
			0				
0							
						0	
	0			**0**			
						0	
					0		
		0					**0**

Need for Regularity Assumptions



Need for Regularity Assumptions

2	0				0		
			0				
0							
						0	
	0			0			
						0	
					0		
		0					0

Need for Regularity Assumptions

100	**0**				**0**		
			0				
0							
						0	
	0			**0**			
						0	
					0		
		0					**0**

- Prediction hard since user unique

- For good prediction, require no user to be unique

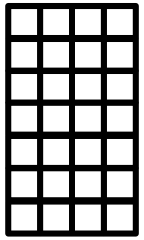
Need for Regularity Assumptions

100	**0**				**0**		
			0				
0							
						0	
	0			**0**			
						0	
					0		
		0					**0**

- Prediction hard since user unique
- For good prediction, require no user to be unique

Non-uniqueness $\cong$ Incoherence : if $M = U\Sigma V^T$ SVD then

U

$\max_i$  $\} \rightarrow$ $\|U_i\|_2 \le \mu\sqrt{\frac{r}{n}}$ $\left(1 \le \mu \le \sqrt{\frac{n}{r}}\right)$

Similarly V

Incoherence: $\|U_i\|_2 \leq \mu\sqrt{\frac{r}{n}}$

$r = 2$

$\frac{n}{2}$

1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

Singular vectors

$\sqrt{\frac{2}{n}} *$

1	0
1	0
1	0
1	0
0	1
0	1
0	1
0	1

- No user very unique
- $\mu = 1 \Leftrightarrow$ Very incoherent

Incoherence: $\|U_i\|_2 \leq \mu\sqrt{\frac{r}{n}}$

$r = 2$

$\frac{n}{4}$

1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

Singular vectors

$\sqrt{\frac{2}{n}} *$

$\sqrt{2}$	0
$\sqrt{2}$	0
0	$\sqrt{2/3}$
0	$\sqrt{2/3}$
0	$\sqrt{2/3}$
0	$\sqrt{2/3}$
0	$\sqrt{2/3}$
0	$\sqrt{2/3}$

- Some users belong to a smaller group
- $\mu = \sqrt{2}$ $\Leftrightarrow$ Still very incoherent, but a little less than earlier

Incoherence: $\|U_i\|_2 \leq \mu\sqrt{\frac{r}{n}}$

$r = 2$

1 ↕

1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1

Singular vectors →

$\sqrt{\frac{2}{n}} *$

$\sqrt{n}/\sqrt{2}$	0
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$
0	$\sqrt{1/2}$

- User 1 very unique
- $\mu = \sqrt{\frac{n}{2}}$ $\Leftrightarrow$ Not incoherent

Regularity Assumptions

1. Low-rank matrix is **incoherent**: if $M = U\Sigma V^T$ SVD then

$$\max_i \; U \;\rightarrow\; \|U_i\|_2 \leq \mu\sqrt{\frac{r}{n}} \qquad \left(1 \leq \mu \leq \sqrt{\frac{n}{r}}\right)$$

Similarly V

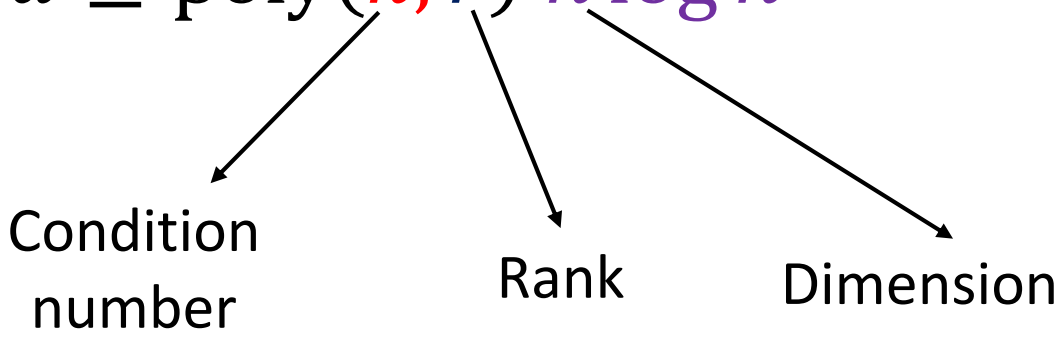
2. d entries observed uniformly at random from all entries

Result

Theorem

AltMin recovers the full matrix if:

$$d \geq \text{poly}(\kappa, r)\, n \log n$$



[Keshavan 2012]

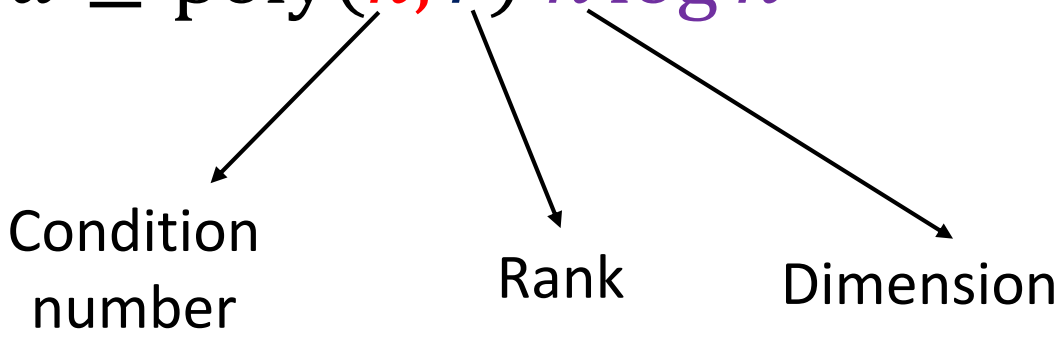
[Jain, Netrapalli, Sanghavi 2013]

Result

Theorem

AltMin recovers the full matrix if:

$$d \geq \text{poly}(\kappa, r)\, n \log n$$



Runtime	
AltMin	$\text{poly}(\kappa, r) n \log n$
Conv. Relaxation	$O(n^3)$

[Keshavan 2012]

[Jain, Netrapalli, Sanghavi 2013]

Further Improvements

Resampling	**[Keshavan 2012] [Jain, Netrapalli, Sanghavi 2013]**	$\mathrm{poly}(\kappa, r)\, n \log n$
	[Hardt 2014] [Hardt, Wootters 2014]	$\log \kappa\, \mathrm{poly}(r)\, n \log n$
No resampling	**[Sun, Luo 2015]**	$\mathrm{poly}(\kappa, r) n$
	[Jain, Netrapalli 2015]	$\mathrm{poly}(r) n \log^3 n$

Proof Outline (rank-1 case)

$$\min_{u,v} \sum_{(i,j)\ observed} \left(M_{ij} - u_i v_j\right)^2$$

AltMin

Approximate power method

Proof Outline (rank-1 case)

$$\min_{u,v} \sum_{(i,j)\ observed} \left(M_{ij} - u_i v_j\right)^2$$

AltMin → Approximate power method

$\Downarrow$ If all entries observed

$$\min_{u,v} \|M - uv^T\|_F^2$$

Proof Outline (rank-1 case)

$$\min_{u,v} \sum_{(i,j)\ observed} \left(M_{ij} - u_i v_j\right)^2$$

AltMin → Approximate power method

= If all entries observed

$$\min_{u,v} \|M - uv^T\|_F^2$$

AltMin

Power method

$$u_{t+1} = \frac{Mv_t}{\|v_t\|^2} \qquad v_{t+1} = \frac{M^T u_{t+1}}{\|u_{t+1}\|^2}$$

Proof Outline (rank-1 case)

$$\min_{u,v} \sum_{(i,j)\ observed} \left(M_{ij} - u_i v_j\right)^2$$

AltMin

Approximate power method

Some entries observed

If all entries observed

$$\min_{u,v} \|M - uv^T\|_F^2$$

Power method

AltMin

$$u_{t+1} = \frac{Mv_t}{\|v_t\|^2} \qquad v_{t+1} = \frac{M^T u_{t+1}}{\|u_{t+1}\|^2}$$

Power Method for Computing Top Eigenvector

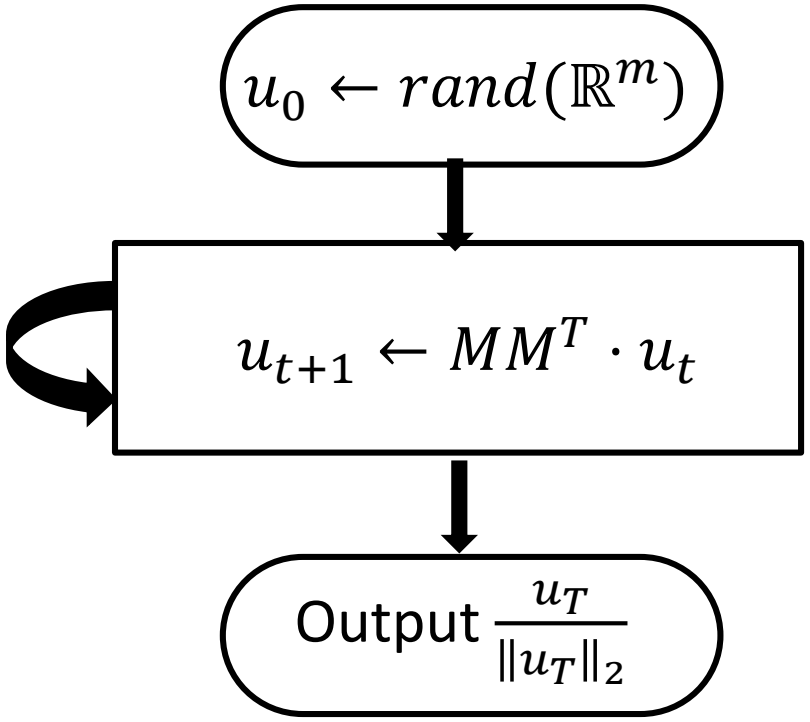
m

n M

- Eigval(MM^T): $\lambda_1 \geq \lambda_2 \geq \cdots$
- Eigvec(MM^T) : $w_1, w_2, \cdots$
- Eigengap : $\delta \overset{\text{def}}{=} \frac{\lambda_1 - \lambda_2}{\lambda_1}$ (SNR)

Task

Output u such that $\|u - w_1\|_2 \leq \epsilon$



error ϵ $\Rightarrow O\left(\frac{1}{\delta}\log\frac{m}{\epsilon}\right)$ iterations

Summary – Matrix Completion

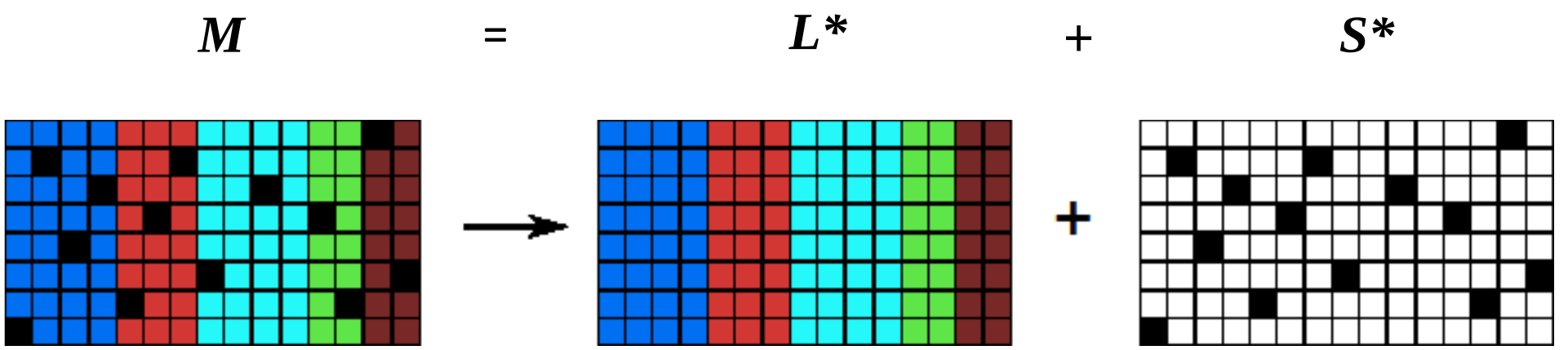
- AltMin can provably solve low rank matrix completion
- AltMin performs approximate power method
- Improved understanding inspired the use of AltMin for many other problems

Summary – Matrix Completion

Phase Retrieval	Netrapalli-Jain-Sanghavi 2013 Candes-Li-Soltanolkotabi 2014
Learning Sparsely Used Dictionaries	Arora-Ge-Moitra 2013 Agarwal-Anandkumar-Jain-Netrapalli 2013 Arora-Bhaskara-Ge-Ma 2014 Arora-Ge-Ma-Moitra 2014 Sun-Qu-Wright 2015
Mixed Linear Regression	Yi-Caramanis-Sanghavi 2014
Tensor decomposition	Anandkumar-Ge-Janzamin 2014 Jain-Oh 2014

Part II: Robust PCA

Robust PCA



Task: Given *M*, find *L** and *S**

Motivation: PCA with missing/corrupted entries

Foreground-Background Separation

Frames in a video → Background (Low rank) + Foreground (Sparse)

AltMin [Netrapalli et al. 2014]

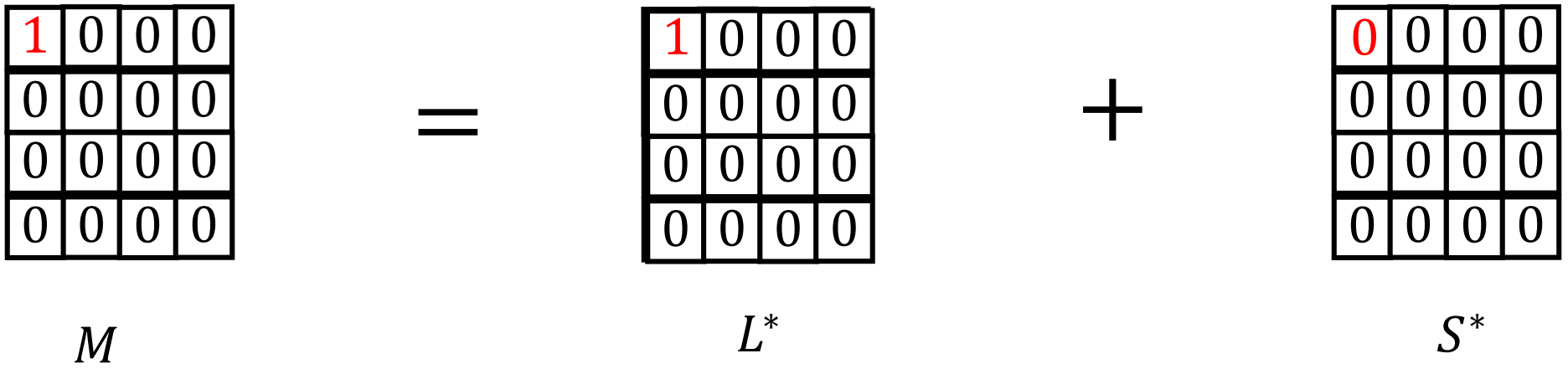
$$\min_{L,S} \|M - L - S\|_F^2$$
$$\text{s.t. } \mathrm{rk}(L) = r$$
$$S \text{ is sparse}$$

AltMin [Netrapalli et al. 2014]

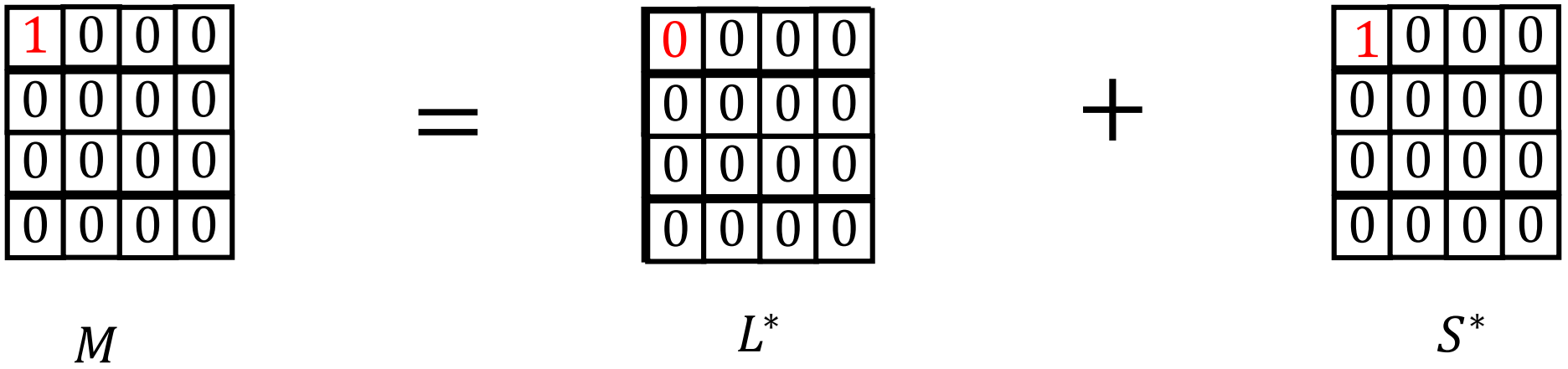
$$\min_{L,S} \|M - L - S\|_F^2 \quad \text{s.t. } \mathrm{rk}(L) = r, \; S \text{ is sparse}$$

Key Challenge: Provable convergence for non-convex projections

Regularity Assumptions

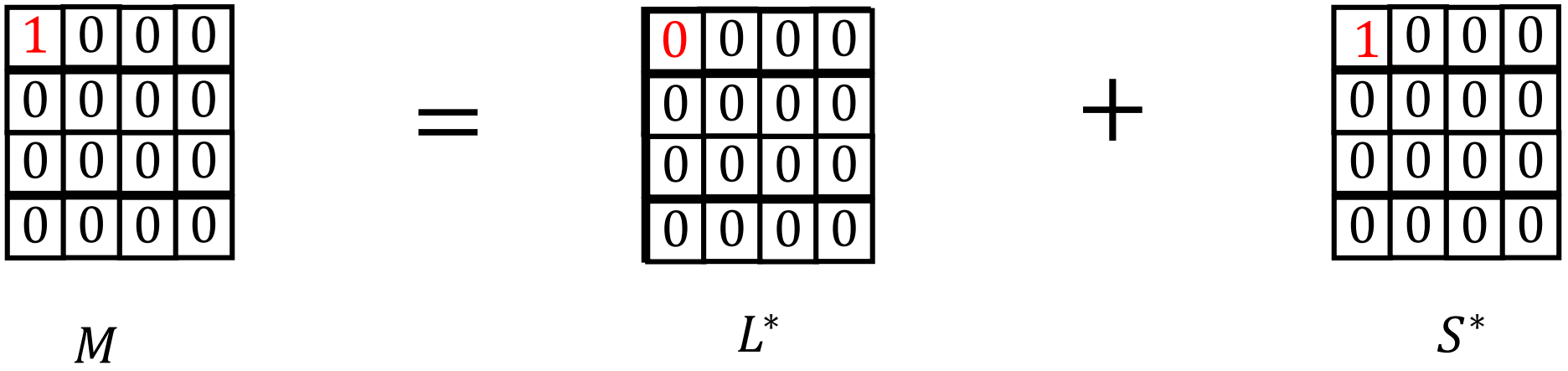


Regularity Assumptions



M L^* S^*

Regularity Assumptions

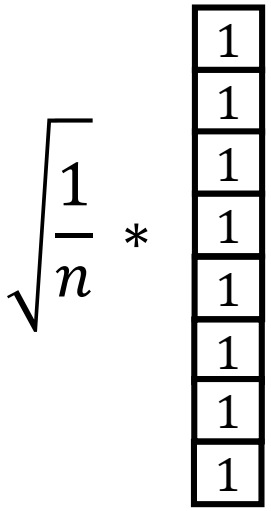


Issue: Low rank matrix is also sparse

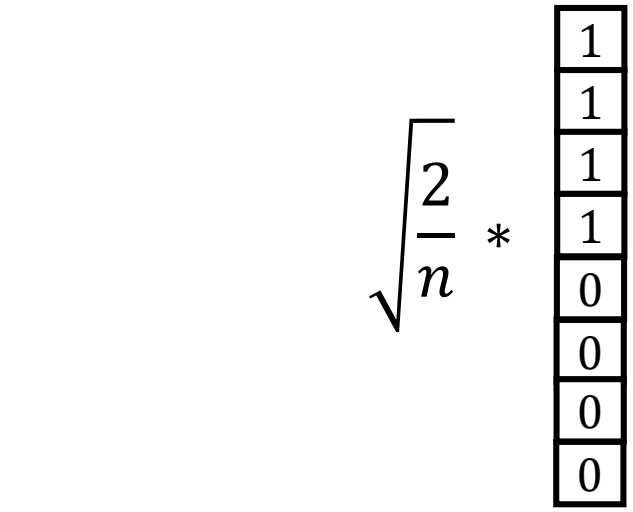
Solution: Ensure PCA directions not sparse – known as incoherence

Incoherence

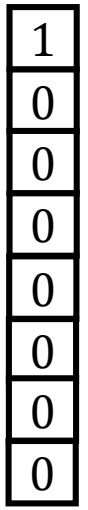
PCA directions



Very incoherent



Incoherent



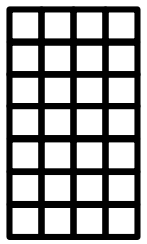
Not incoherent

Regularity Assumptions

(deterministic) $n \times n$ matrix $M = L^* + S^*$ such that:

1. Rank-r matrix is μ-**incoherent**: if $L^* = U\Sigma V^T$ then

U



$$\max_i \rightarrow \|U_i\|_2 \leq \mu\sqrt{\frac{r}{n}} \qquad \left(1 \leq \mu \leq \sqrt{\frac{n}{r}}\right)$$

Similarly V

2. Sparse matrix is **degree bounded**:

Any row or column of S^* has at most d non-zero elements

Guarantee for AltMin

[Netrapalli et al. 2014]: L^* and S^* recovered by a non-convex algorithm if

$$d \leq \frac{n}{512\mu^2 r}$$

Information theoretically optimal up to constant factors

Runtime: $O\left(r^2 n^2 \log\frac{1}{\epsilon}\right)$

Factor r away from that of regular PCA

[Hsu, Kakade, Zhang 2011]: *L*,S** recovered by a convex program if

$$d \leq c\,\frac{n}{\mu^2 r}$$

Runtime: $O\left(\frac{n^3}{\epsilon^2}\right)$

Key Technical Result

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Need to show $L \sim L^*$

Incoherence of L^*
Sparsity of S^*

$$\Rightarrow \quad \|L - L^*\|_\infty \lesssim \frac{1}{10}\|S^*\|_\infty$$

Standard Matrix Perturbation: $\|L - L^*\|_2 \leq \|S^*\|_2$
Does not use incoherence of L^*; not sufficient

Key Technique Contd.

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Rank 1: $L^* = u^* u^{*T}$ and let $L = uu^T$

Key Technique Contd.

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Rank 1: $L^* = u^* u^{*T}$ and let $L = uu^T$ ➡ $(L^* + S^*)u = u$

Rearranging, $u \approx (I - S^*)^{-1} L^* u \approx \left(I + S^* + S^{*2} + \cdots\right) u^*$

Key Technique Contd.

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Rank 1: $L^* = u^* u^{*T}$ and let $L = uu^T$ ➡ $(L^* + S^*)u = u$

Rearranging, $u \approx (I - S^*)^{-1} L^* u \approx \left(I + S^* + S^{*2} + \cdots\right) u^*$

Incoherence of L^*
Sparsity of S^*

➡ $\|u - u^*\|_\infty \lesssim d\|S^*\|_\infty \|u^*\|_\infty$

Key Technique Contd.

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Rank 1: $L^* = u^* u^{*T}$ and let $L = uu^T$ ➡ $(L^* + S^*)u = u$

Rearranging, $u \approx (I - S^*)^{-1} L^* u \approx \left(I + S^* + S^{*2} + \cdots\right) u^*$

Incoherence of L^*
Sparsity of S^*

➡ $\|u - u^*\|_\infty \lesssim d\|S^*\|_\infty \|u^*\|_\infty$

➡ $\|L - L^*\|_\infty \lesssim \frac{1}{10}\|S^*\|_\infty$

Key Technique Contd.

$$L \leftarrow \mathcal{P}_r(M) = \mathcal{P}_r(L^* + S^*)$$

Rank 1: $L^* = u^* u^{*T}$ and let $L = uu^T$ ➡ $(L^* + S^*)u = u$

Rearranging, $u \approx (I - S^*)^{-1} L^* u \approx \left(I + S^* + S^{*2} + \cdots\right) u^*$

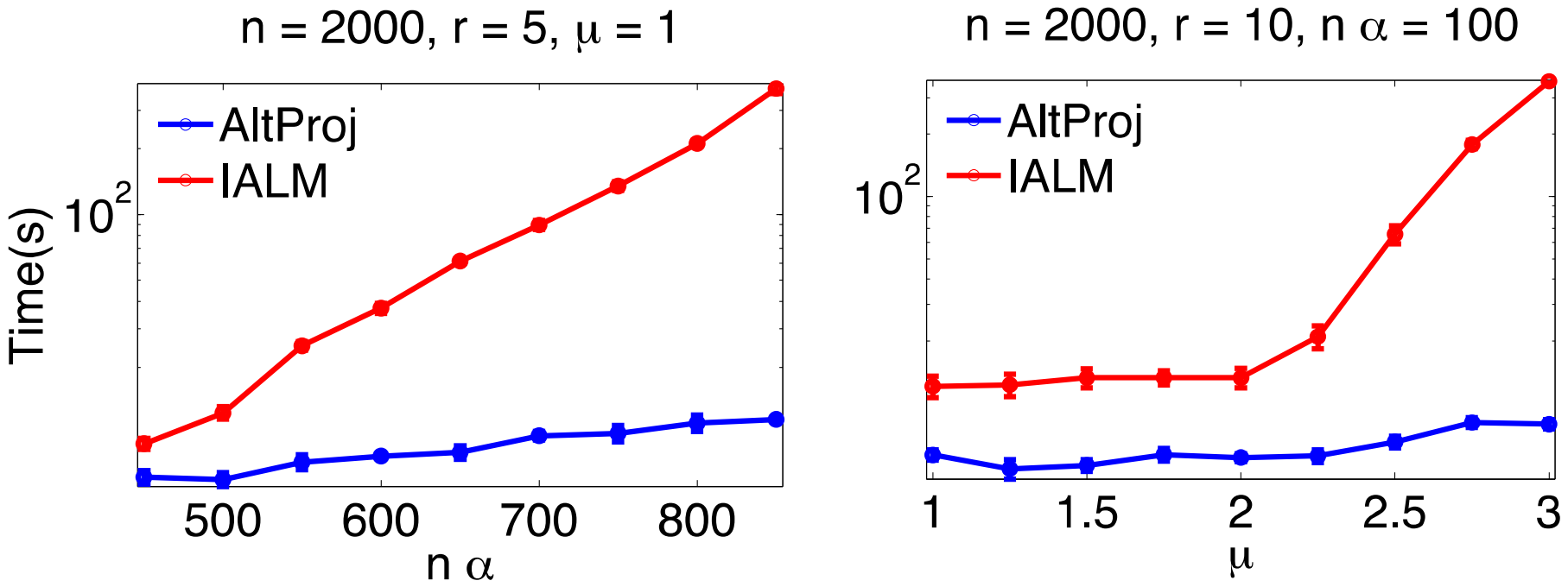
⬇

- Technique originally invented to study eigenvectors of random matrices: Erdos et al. 2011

Comparison to Convex Solvers

IALM [Lin-Chen-Ma] : fastest / most popular special-purpose convex algorithm

Speed comparison, synthetic example. Single laptop.



Empirical Results

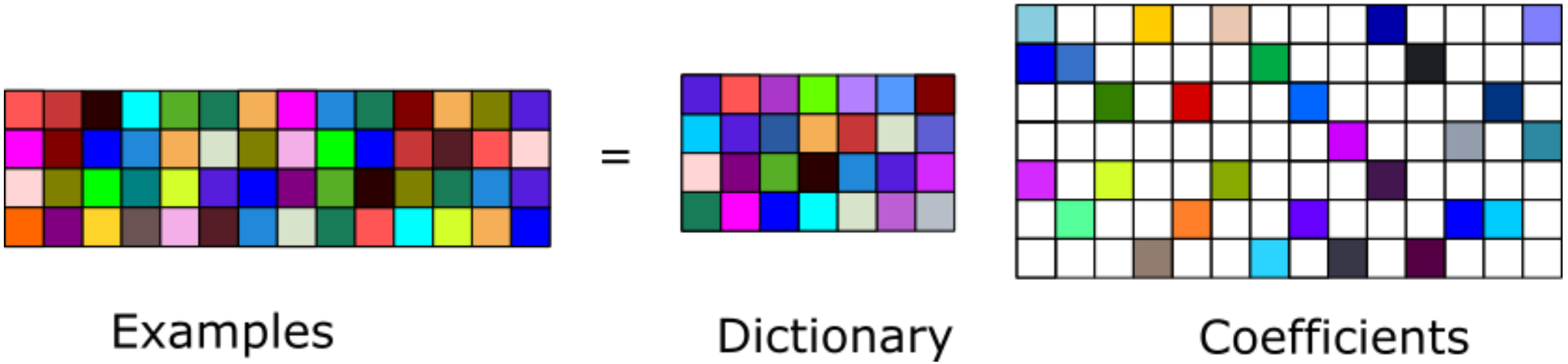
Convex Method. Runtime: 1700 sec

AltMin. Runtime: 70 sec

Part III: Sparse Coding

Sparse coding

Y A^* X^*



Applications: Image compression, denoising, inpainting etc.

Image reconstruction

50 % missing pixels

Learned reconstruction
Average # coeffs: 4.0202
MAE: 0.012977
RMSE: 0.029204

Haar reconstruction
Average # coeffs: 4.7677
MAE: 0.022833
RMSE: 0.071107

OverComplete DCT reconstruction
Average # coeffs: 4.7694
MAE: 0.015719
RMSE: 0.037745

Image credit: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, Aharon et al. IEEE TSP 54(11) 2006

Algorithm

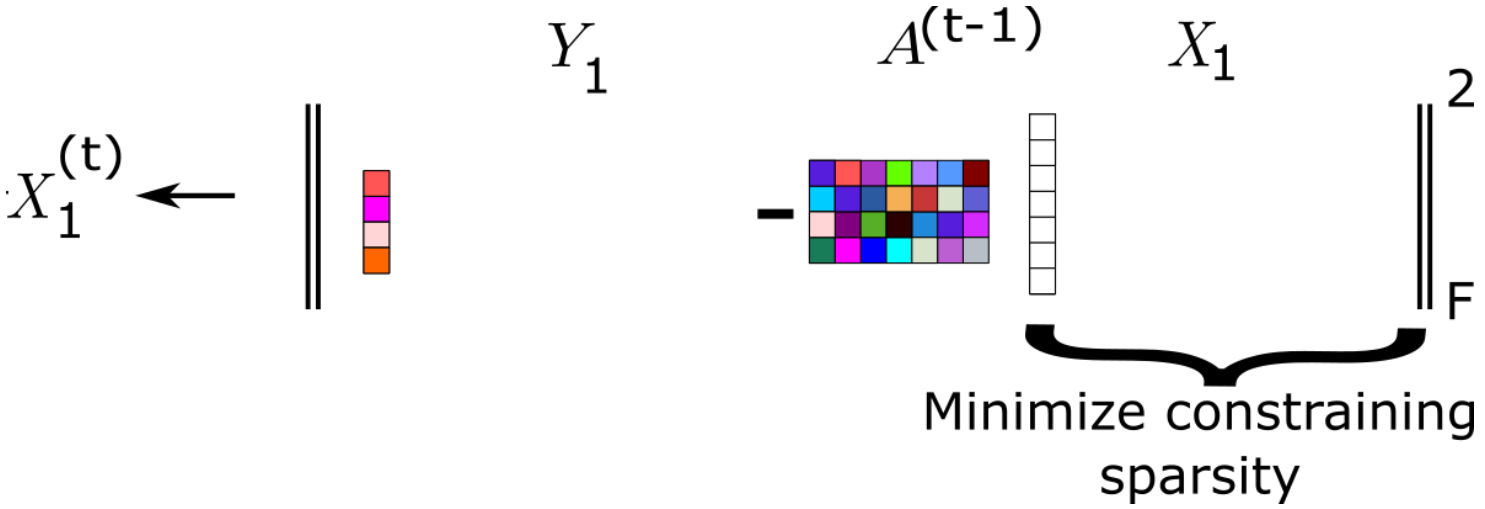
$$\min_{A,X} \|Y - AX\|_F^2 \text{ s.t. } X \text{ is sparse}$$

1. Choose an initial $A^{(0)}$ using a **clustering** algorithm
2. In iteration $t = 1, \cdots$ do
   - Estimate coefficients $X^{(t)}$ (Sparse recovery/compressed sensing)
   - Estimate dictionary $A^{(t)}$ (Least squares)

Algorithm

$$\min_{A,X} \|Y - AX\|_F^2 \text{ s.t. } X \text{ is sparse}$$

1. Choose an initial $A^{(0)}$ using a **clustering** algorithm
2. In iteration $t = 1, \cdots$ do
   - Estimate coefficients $X^{(t)}$ (Sparse recovery/compressed sensing)

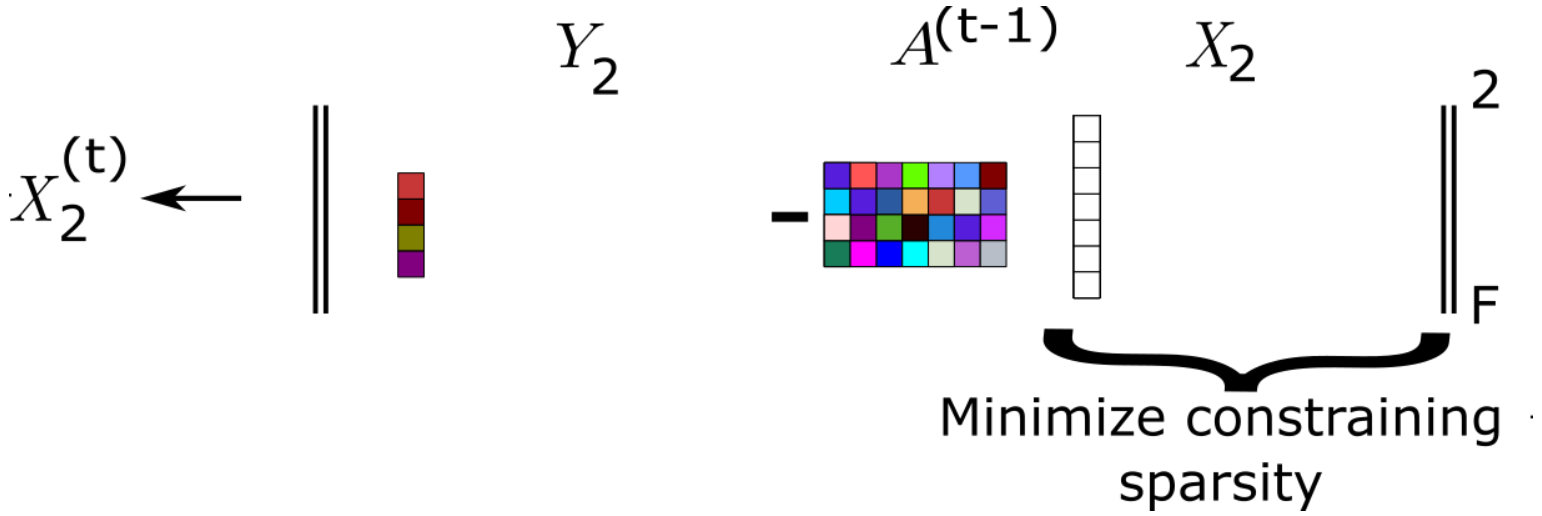


   - Estimate dictionary $A^{(t)}$ (Least squares)

Algorithm

$$\min_{A,X} \|Y - AX\|_F^2 \text{ s.t. } X \text{ is sparse}$$

1. Choose an initial $A^{(0)}$ using a **clustering** algorithm
2. In iteration $t = 1, \cdots$ do
   - Estimate coefficients $X^{(t)}$ (Sparse recovery/compressed sensing)

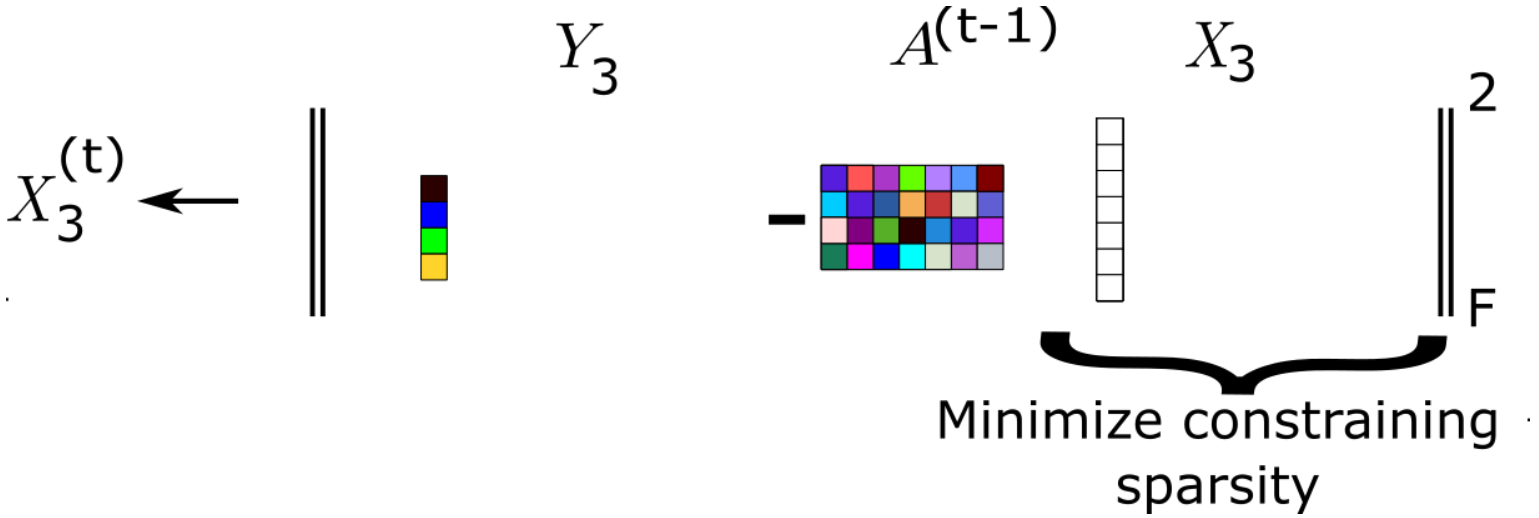


   - Estimate dictionary $A^{(t)}$ (Least squares)

Algorithm

$$\min_{A,X} \|Y - AX\|_F^2 \text{ s.t. } X \text{ is sparse}$$

1. Choose an initial $A^{(0)}$ using a **clustering** algorithm
2. In iteration $t = 1, \cdots$ do
   - Estimate coefficients $X^{(t)}$ (Sparse recovery/compressed sensing)

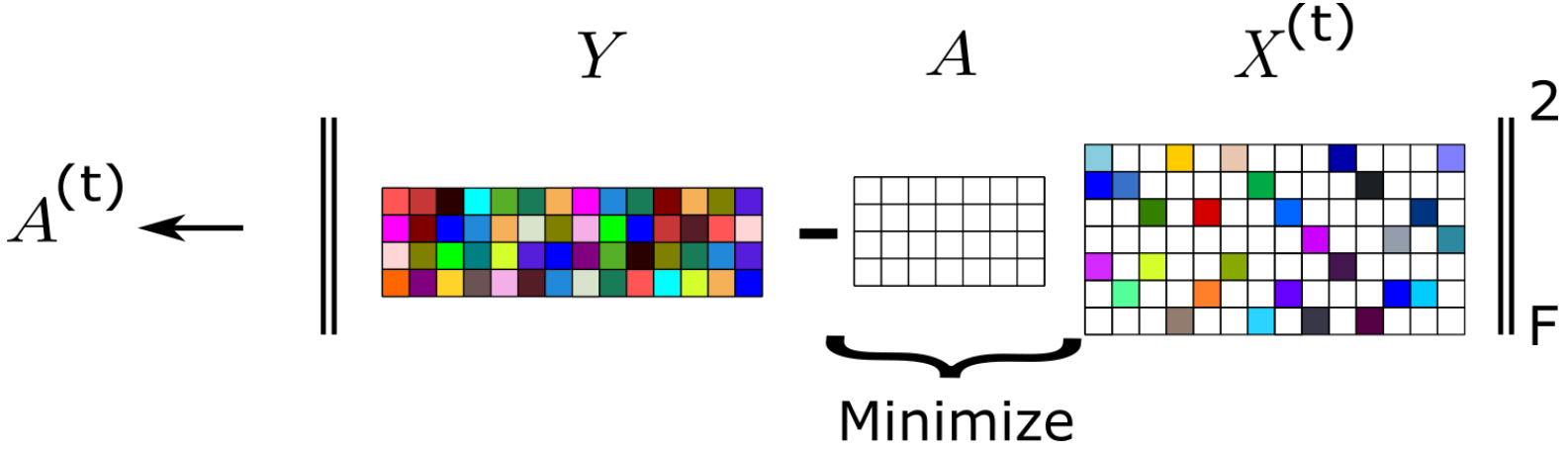


   - Estimate dictionary $A^{(t)}$ (Least squares)

Algorithm

$$\min_{A,X} \|Y - AX\|_F^2 \text{ s.t. } X \text{ is sparse}$$

1. Choose an initial $A^{(0)}$ using a **clustering** algorithm
2. In iteration $t = 1, \cdots$ do
   - Estimate coefficients $X^{(t)}$ (Sparse recovery/compressed sensing)
   - Estimate dictionary $A^{(t)}$ (Least squares)



Result

Under some nondegeneracy conditions (incoherent dictionary and bounds on sparsity), recovers the decomposition.

[Arora, Ge, Moitra 2013], [Agarwal et al. 2013]

Many more results for simpler algorithms [Arora, Ge, Ma, Moitra 2015], noisy setting, nongenerative assumptions [Bhaskara, Tai 2019] etc.

Conclusion

- Matrix decomposition is an extremely valuable tool for data analysis
- Finding the right structure is important
- AltMin is a very successful meta-algorithm
- Theoretical insights into AltMin help design new variants for new settings