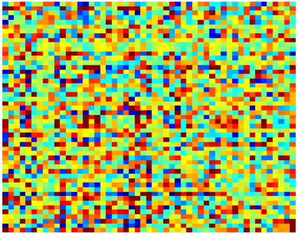
Matrix Completion: Fundamental Limits and Efficient Algorithms

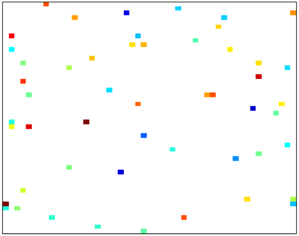
Sewoong Oh

PhD Defense
Stanford University

July 23, 2010

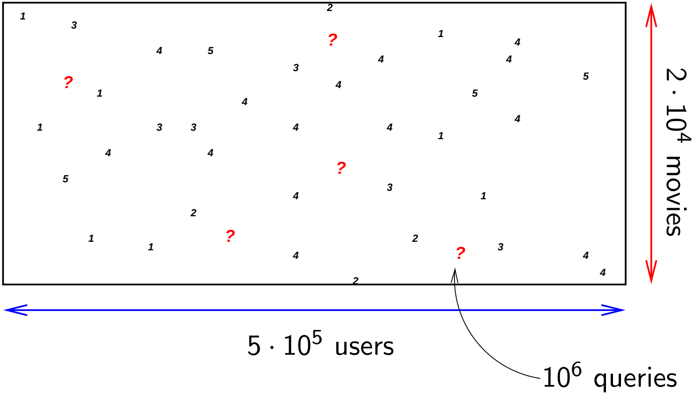
Matrix completion





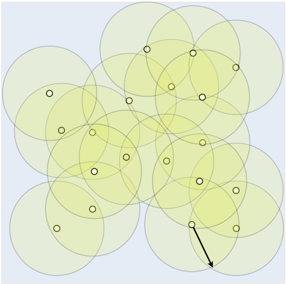
- Find the missing entries in a huge data matrix

Example 1. Recommendation systems

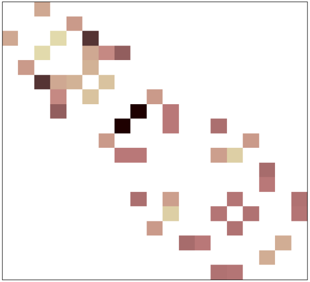


- Given less than 1% of the movie ratings
- Goal: Predict missing ratings

Example 2. Positioning

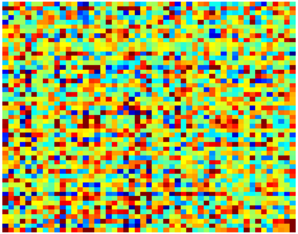


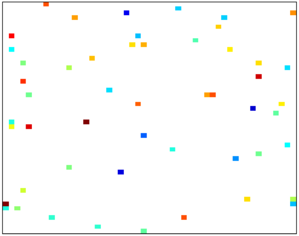
Distance Matrix



- Only distances between close-by sensors are measured
- Goal: Find the sensor positions up to a rigid motion

Matrix completion



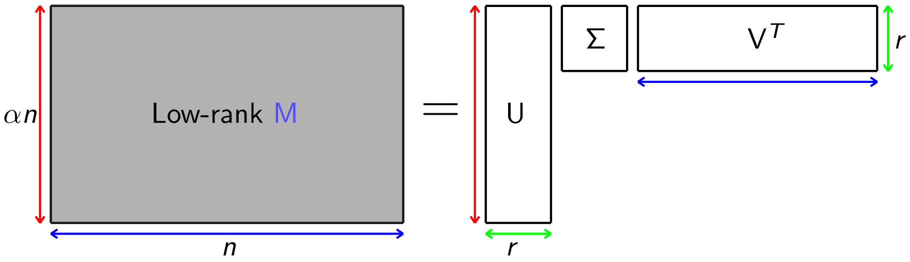


- More applications:
  - Computer vision: Structure-from-motion
  - Molecular biology: Microarray
  - Numerical linear algebra: Fast low-rank approximations
  - etc.

Outline

Background

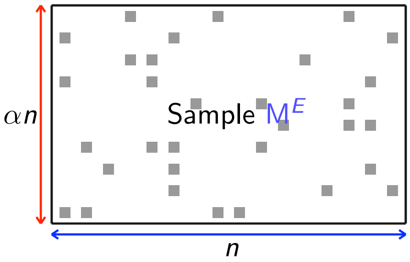
The model



- Rank-r matrix M
- Random uniform sample set E
- Sample matrix M^E

$$M^E_{ij} = \begin{cases} M_{ij} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

The model



- Rank-r matrix M
- Random uniform sample set E
- Sample matrix M^E

$$\mathsf{M}^E_{ij} = \begin{cases} \mathsf{M}_{ij} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Which matrices?

- Pathological example

$$M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

Which matrices?

- Pathological example

$$M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

- [Candès, Recht '08] $M = U \Sigma V^T$ has coherence μ if

$$A0. \qquad \max_{1 \le i \le \alpha n} \sum_{k=1}^{r} U_{ik}^2 \le \mu \frac{r}{n}, \quad \max_{1 \le j \le n} \sum_{k=1}^{r} V_{jk}^2 \le \mu \frac{r}{n}$$

$$A1. \qquad \max_{i,j} \left| \sum_{k=1}^{r} U_{ik} V_{jk} \right| \le \mu \frac{\sqrt{r}}{n}$$

- Intuition
  - μ is small if singular vectors are well balanced
  - We need low-coherence for matrix completion

Previous work

Rank minimization

$$\begin{array}{ll} \text{minimize} & \mathrm{rank}(\mathsf{X}) \\ \text{subject to} & \mathsf{X}_{ij} = \mathsf{M}_{ij},\ (i,j) \in E \end{array}$$

- NP-hard

Previous work

Rank minimization

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & X_{ij} = M_{ij},\ (i,j) \in E \end{array}$$

- NP-hard

Heuristic [Fazel '02]

$$\begin{array}{ll} \text{minimize} & \|X\|_* \\ \text{subject to} & X_{ij} = M_{ij},\ (i,j) \in E \end{array}$$

- Convex relaxation
- Nuclear norm

$$\|X\|_* = \sum_{i=1}^{n} \sigma_i(X)$$

- Can be solved using Semidefinite Programming(SDP)

Previous work

- [Candès, Recht '08]
  - Nuclear norm minimization reconstructs M exactly with high probability, if
    $$|E| \geq C\,\mu\,r\,n^{6/5}\log n$$
  - Surprise?

Previous work

- [Candès, Recht '08]
  - Nuclear norm minimization reconstructs M exactly with high probability, if
    $$|E| \geq C\,\mu\, r\, n^{6/5} \log n$$
  - Degrees of freedom $\simeq (1+\alpha)rn$
  - Open questions
    - ⋆ Optimality: Do we need $n^{6/5} \log n$ samples?
    - ⋆ Complexity: SDP is computationally expensive
    - ⋆ Noise: Can not deal with noise

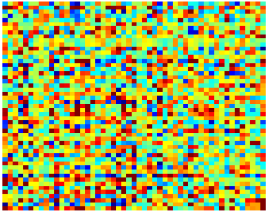
Previous work

- [Candès, Recht '08]
  - Nuclear norm minimization reconstructs M exactly with high probability, if
    $$|E| \geq C\,\mu\,r\,n^{6/5}\log n$$
  - Degrees of freedom $\simeq (1+\alpha)rn$
  - Open questions
    - ★ Optimality: Do we need $n^{6/5}\log n$ samples?
    - ★ Complexity: SDP is computationally expensive
    - ★ Noise: Can not deal with noise

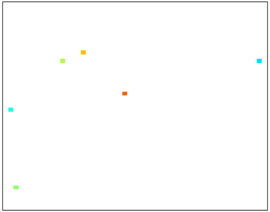
A new approach to Matrix Completion: OptSpace

Example: 2000×2000 rank-8 random matrix

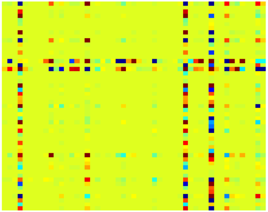
low-rank matrix M



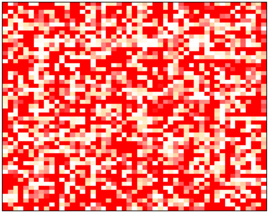
sampled matrix M^E



OptSpace output $\widehat{M}$

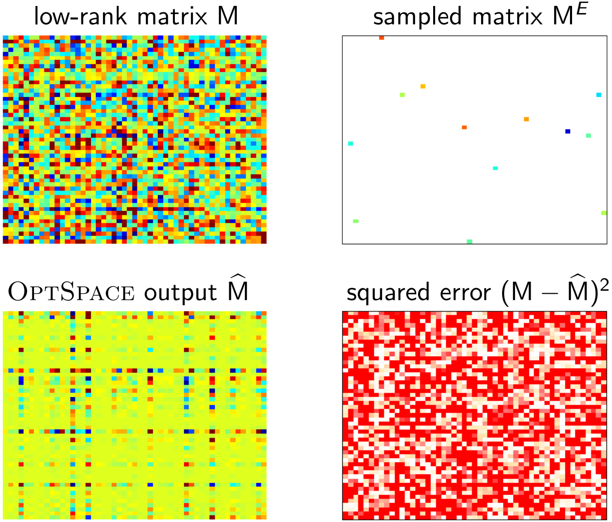


squared error $(M - \widehat{M})^2$



0.25% sampled

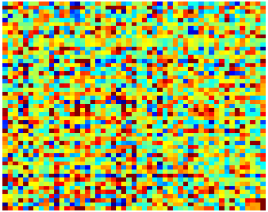
Example: 2000×2000 rank-8 random matrix



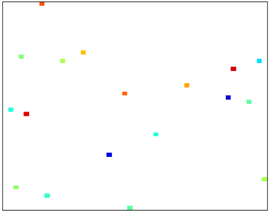
0.50% sampled

Example: 2000×2000 rank-8 random matrix

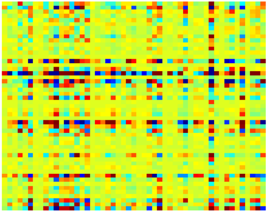
low-rank matrix M



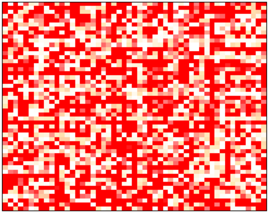
sampled matrix M^E



OptSpace output $\widehat{M}$

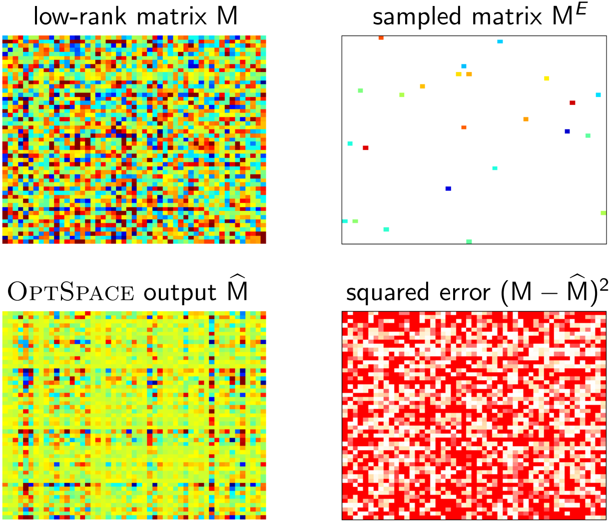


squared error $(M - \widehat{M})^2$



0.75% sampled

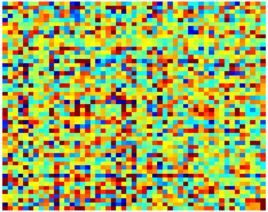
Example: 2000×2000 rank-8 random matrix



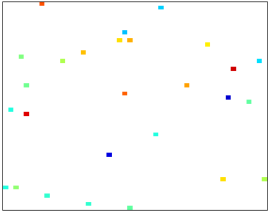
1.00% sampled

Example: 2000×2000 rank-8 random matrix

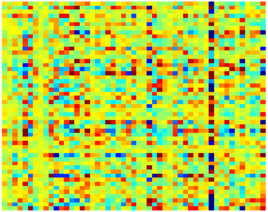
low-rank matrix M



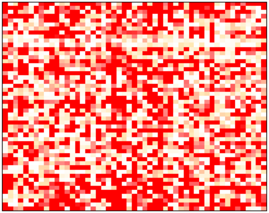
sampled matrix M^E



OptSpace output $\widehat{M}$



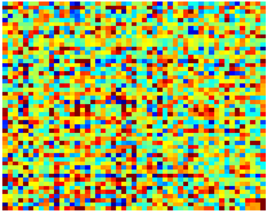
squared error $(M - \widehat{M})^2$



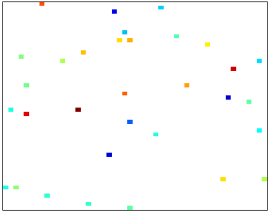
1.25% sampled

Example: 2000×2000 rank-8 random matrix

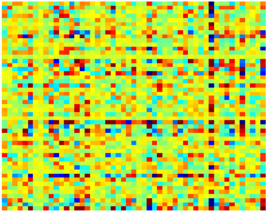
low-rank matrix M



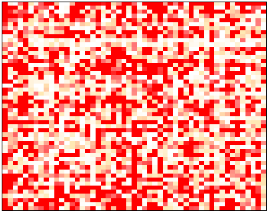
sampled matrix M^E



OptSpace output $\widehat{\mathsf{M}}$



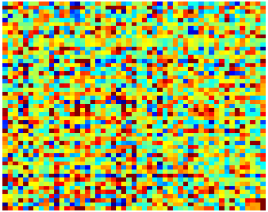
squared error $(\mathsf{M} - \widehat{\mathsf{M}})^2$



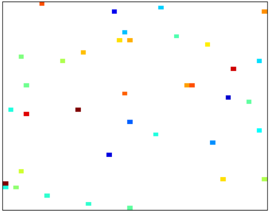
1.50% sampled

Example: 2000×2000 rank-8 random matrix

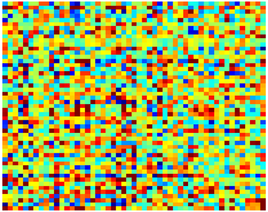
low-rank matrix M



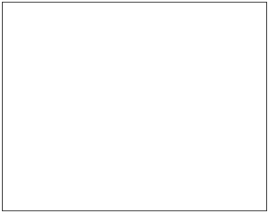
sampled matrix M^E



OptSpace output $\widehat{M}$



squared error $(M - \widehat{M})^2$



1.75% sampled

Algorithm

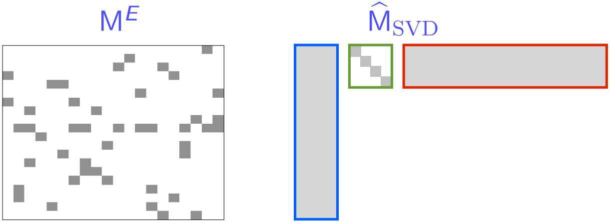
Naïve approach

- Singular Value Decomposition (SVD)

$$\mathsf{M}^E = \sum_{i=1}^{n} \sigma_i x_i y_i^T$$

- Compute rank-r approximation $\widehat{\mathsf{M}}_{\mathrm{SVD}}$

$$\widehat{\mathsf{M}}_{\mathrm{SVD}} \triangleq \frac{\alpha n^2}{|E|} \sum_{i=1}^{r} \sigma_i x_i y_i^T$$



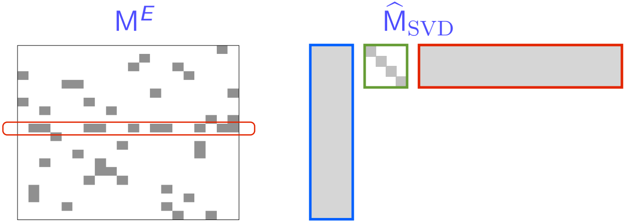
Naïve approach fails

- Singular Value Decomposition (SVD)

$$\mathsf{M}^E = \sum_{i=1}^{n} \sigma_i x_i y_i^T$$

- Compute rank-r approximation $\widehat{\mathsf{M}}_{\mathrm{SVD}}$

$$\widehat{\mathsf{M}}_{\mathrm{SVD}} \triangleq \frac{\alpha n^2}{|E|} \sum_{i=1}^{r} \sigma_i x_i y_i^T$$



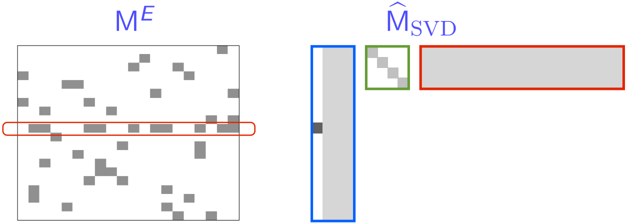
Naïve approach fails

- Singular Value Decomposition (SVD)

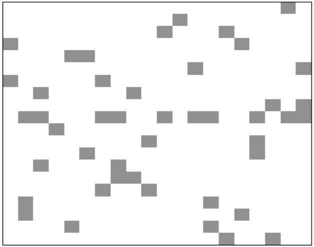
$$\mathsf{M}^E = \sum_{i=1}^{n} \sigma_i x_i y_i^T$$

- Compute rank-r approximation $\widehat{\mathsf{M}}_{\mathrm{SVD}}$

$$\widehat{\mathsf{M}}_{\mathrm{SVD}} \triangleq \frac{\alpha n^2}{|E|} \sum_{i=1}^{r} \sigma_i x_i y_i^T$$



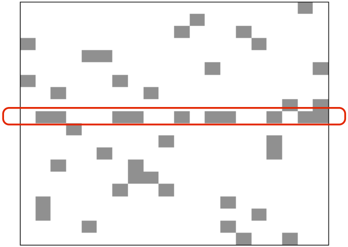
Trimming

$\mathrm{M}^E =$ 

$$\widetilde{\mathrm{M}}^E_{ij} = \begin{cases} 0 & \text{if } deg(row_i) > 2|E|/\alpha n \\ 0 & \text{if } deg(col_j) > 2|E|/n \\ \mathrm{M}^E_{ij} & otherwise \end{cases}$$

$deg(\cdot)$ is the number of samples in that row/column

Trimming

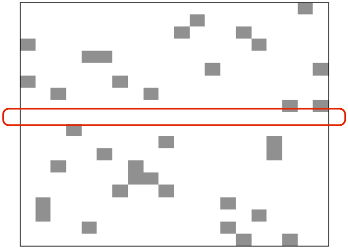
$\mathsf{M}^E =$ 

$$\widetilde{\mathsf{M}}^E_{ij} = \begin{cases} 0 & \text{if } deg(row_i) > 2|E|/\alpha n \\ 0 & \text{if } deg(col_j) > 2|E|/n \\ \mathsf{M}^E_{ij} & otherwise \end{cases}$$

$deg(\cdot)$ is the number of samples in that row/column

Trimming

$\widetilde{\mathsf{M}}^E =$



$$\widetilde{\mathsf{M}}^E_{ij} = \begin{cases} 0 & \text{if } deg(row_i) > 2|E|/\alpha n \\ 0 & \text{if } deg(col_j) > 2|E|/n \\ \mathsf{M}^E_{ij} & otherwise \end{cases}$$

$deg(\cdot)$ is the number of samples in that row/column

Algorithm

OptSpace

Input : sample indices E, sample values M^E, rank r

Output : estimation $\widehat{\mathsf{M}}$

1: Trimming

2: Compute $\widehat{\mathsf{M}}_{\rm SVD}$ using SVD

3: Greedy minimization of the residual error

Algorithm

OptSpace

Input : sample indices E, sample values M^E, rank r

Output : estimation $\widehat{\mathsf{M}}$

1: Trimming

2: Compute $\widehat{\mathsf{M}}_{\rm SVD}$ using SVD

- $\widehat{\mathsf{M}}_{\rm SVD}$ can be computed efficiently for sparse matrices

Main results

Theorem

For any $|E|$, $\widehat{\mathsf{M}}_{\rm SVD}$ achieves, with high probability,

$$\mathrm{RMSE} \leq C\mathsf{M}_{\max}\sqrt{\frac{nr}{|E|}}$$

- $\mathrm{RMSE} = \left(\frac{1}{\alpha n^2}\sum_{i,j}(\mathsf{M} - \widehat{\mathsf{M}}_{\rm SVD})^2_{ij}\right)^{1/2}$
- $\mathsf{M}_{\max} \triangleq \max_{i,j} |\mathsf{M}_{ij}|$

Keshavan, Montanari, Oh, *IEEE Trans. Information Theory*, 2010

Main results

Theorem

For any $|E|$, $\widehat{\mathrm{M}}_{\mathrm{SVD}}$ achieves, with high probability,

$$\mathrm{RMSE} \leq C M_{\max} \sqrt{\frac{nr}{|E|}}$$

- [Achlioptas, McSherry '07]
  If $|E| \geq n(8 \log n)^4$, with high probability,
  $$\mathrm{RMSE} \leq 4 \mathrm{M}_{\max} \sqrt{\frac{nr}{|E|}}$$
- For $n = 10^5, (8 \log n)^4 \simeq 7.2 \cdot 10^7$

Keshavan, Montanari, Oh, *IEEE Trans. Information Theory*, 2010

Main results

Theorem

For any $|E|$, $\widehat{\mathrm{M}}_{\mathrm{SVD}}$ achieves, with high probability,

$$\mathrm{RMSE} \le C\mathrm{M}_{\max}\sqrt{\frac{nr}{|E|}}$$

- [Achlioptas, McSherry '07]
  If $|E| \ge n(8 \log n)^4$, with high probability,

  $$\mathrm{RMSE} \le 4\mathrm{M}_{\max}\sqrt{\frac{nr}{|E|}}$$

- For $n = 10^5, (8 \log n)^4 \simeq 7.2 \cdot 10^7$

Netflix dataset

A single user rated 17,000 movies.

"Miss Congeniality": 200,000 ratings.

Keshavan, Montanari, Oh, *IEEE Trans. Information Theory*, 2010

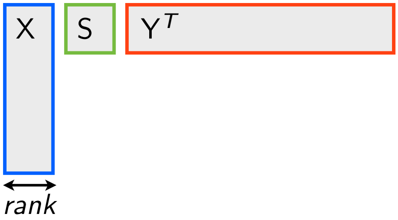
Can we do better?

Greedy minimization of residual error

- Starting from (X_0, Y_0) for $\widehat{M}_{\mathrm{SVD}} = X_0 S_0 Y_0^T$, use gradient descent methods to solve

$$\begin{array}{ll} \text{minimize} & F(X,Y) \\ \text{subject to} & X^T X = \mathbb{I},\ Y^T Y = \mathbb{I} \end{array}$$

$$F(X,Y) \triangleq \min_{S \in \mathbb{R}^{r\times r}} \sum_{(i,j)\in E} \left(M_{ij}^E - (XSY^T)_{ij} \right)^2$$



- Can be computed efficiently for sparse matrices

Algorithm

OptSpace

Input : sample indices E, sample values M^E, rank r
Output : estimation $\widehat{\mathsf{M}}$
1: Trimming
2: Compute $\widehat{\mathsf{M}}_{\rm SVD}$ using SVD
3: Greedy minimization of the residual error

Main results

Theorem (Trimming+SVD)

$\widehat{\mathrm{M}}_{\mathrm{SVD}}$ *achieves, with high probability,*

$$\mathsf{RMSE} \le C\mathrm{M}_{\max}\sqrt{\frac{nr}{|E|}}$$

Theorem (Trimming+SVD+Greedy minimization)

OptSpace *reconstructs* M *exactly, with high probability, if*

$$|E| \ge C\,\mu\,r\,n\max\{\mu r, \log n\}$$

Keshavan, Montanari, Oh, *IEEE Trans. Information Theory*, 2010

OptSpace is order-optimal

Theorem

If μ and r are bounded, OptSpace *reconstructs* M *exactly, with high probability, if*

$$|E| \geq C\, n \log n$$

- Lower bound (coupon collector's problem):
  If $|E| \leq C'\, n \log n$, then exact reconstruction is impossible

OptSpace is order-optimal

Theorem

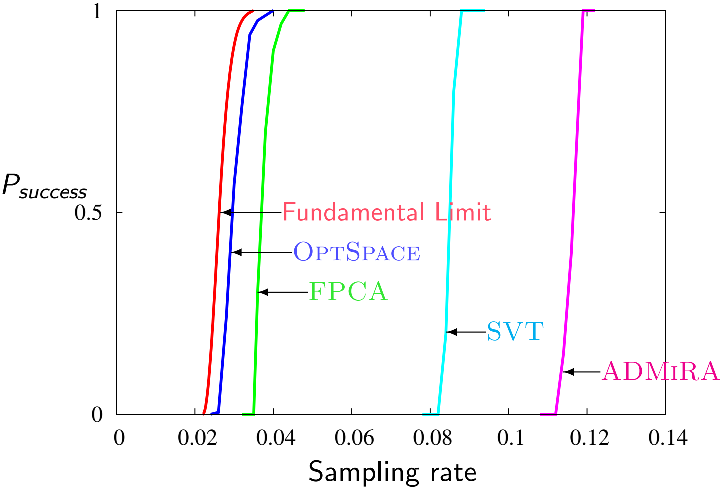
If μ and r are bounded, OptSpace *reconstructs* M *exactly, with high probability, if*

$$|E| \geq C\, n \log n$$

- Lower bound (coupon collector's problem):
  If $|E| \leq C' n \log n$, then exact reconstruction is impossible
- Nuclear norm minimization:
  [Candès, Recht '08, Candès, Tao '09, Recht '09, Gross et al. '09]
  If $|E| \geq C'' n (\log n)^2$, then exact reconstruction by SDP

Comparison

- 1000×1000 rank-10 matrix M

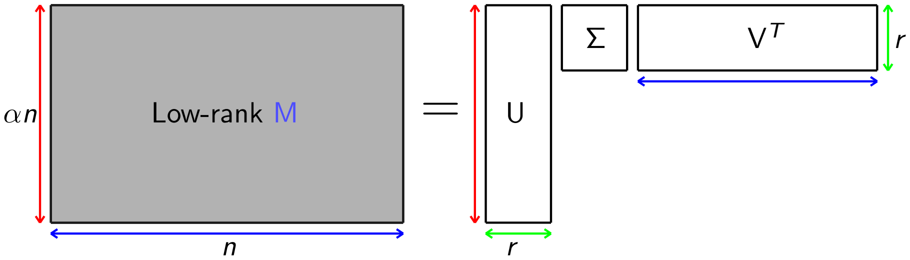


Fundamental Limit [Singer, Cucuringu '09], FPCA [Ma, Goldfarb, Chen '09], SVT [Cai, Candès, Shen '08], ADMiRA [Lee, Bresler '09]

Story so far

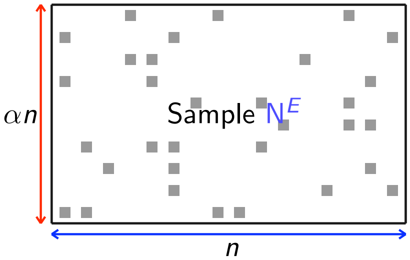
- OPTSPACE reconstructs M from a few sampled entries, when M is exactly low-rank and samples are exact
- In reality,
  - M is only approximately low-rank
  - samples are corrupted by noise

The model with noise



- Rank-r matrix M
- Random sample set E
- Sample noise Z^E
- Sample matrix $N^E = M^E + Z^E$

The model with noise



- Rank-r matrix M
- Random sample set E
- Sample noise Z^E
- Sample matrix $N^E = M^E + Z^E$

Main results

Theorem

For $|E| \geq C\mu r n \max\{\mu r, \log n\}$, OptSpace achieves, with high probability,

$$\text{RMSE} \leq C' \frac{n\sqrt{r}}{|E|} ||Z^E||_2 \, ,$$

provided that the RHS is smaller than $\sigma_r(\mathrm{M})/n$.

- $\| \cdot \|_2$ is the spectral norm

Keshavan, Montanari, Oh, *Journ. Machine Learning Research*, 2010

OptSpace is order-optimal when noise is i.i.d. Gaussian

Theorem

For $|E| \geq C\mu r n \max\{\mu r, \log n\}$, OptSpace *achieves, with high probability,*

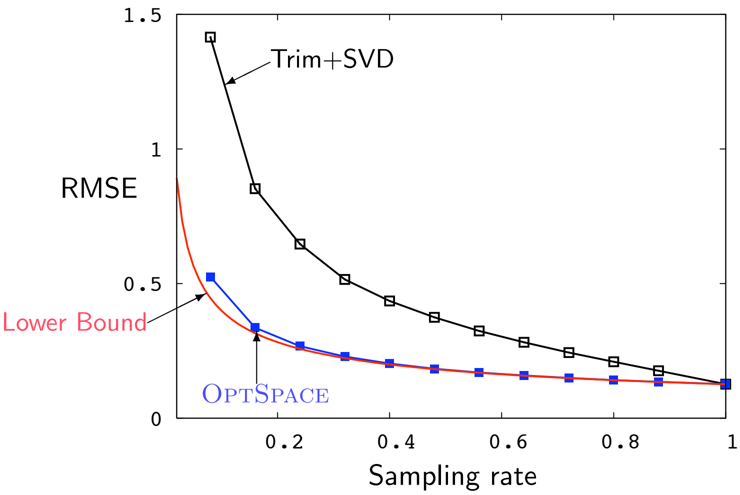
$$\text{RMSE} \leq C' \sigma_z \sqrt{\frac{r\,n}{|E|}}$$

provided that the RHS is smaller than $\sigma_r(\mathrm{M})/n$.

- Lower bound: [Candès, Plan '09]
  $\text{RMSE} \geq \sigma_z \sqrt{\frac{2\,r\,n}{|E|}}$
- Trimming + SVD
  $$\text{RMSE} \leq \underbrace{C\mathrm{M}_{\max}\sqrt{\frac{r\,n}{|E|}}}_{\text{missing entries}} + \underbrace{C'\sigma_z\sqrt{\frac{r\,n}{|E|}}}_{\text{sample noise}}$$

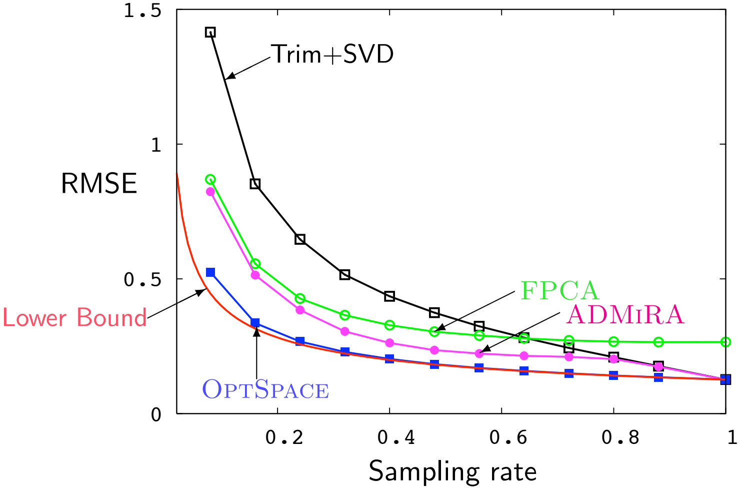
Comparison

- 500 $\times$ 500 rank-4 matrix M, Gaussian noise with $\sigma_z = 1$
- Example from [Candès, Plan '09]



Comparison

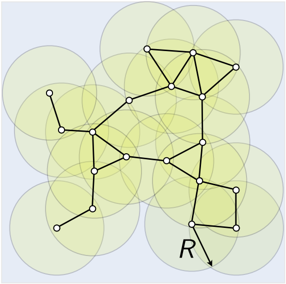
- 500×500 rank-4 matrix M, Gaussian noise with $\sigma_z = 1$
- Example from [Candès, Plan '09]



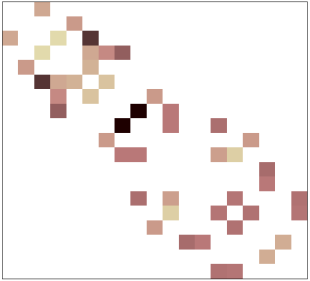
FPCA [Ma, Goldfarb, Chen '09], ADMiRA [Lee, Bresler '09]

Positioning

The model

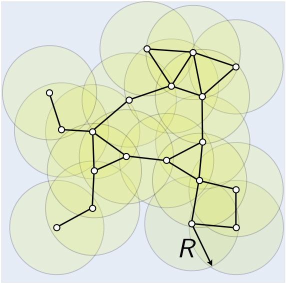


Distance Matrix D

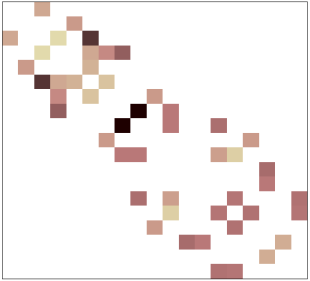


- n wireless devices uniformly distributed in a bounded convex region
- Distance measurements between devices within radio range R
- Find the locations up to a rigid motion

The model

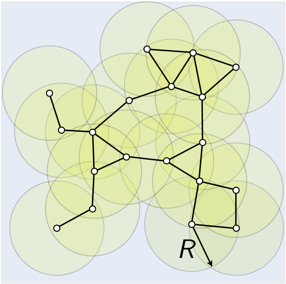


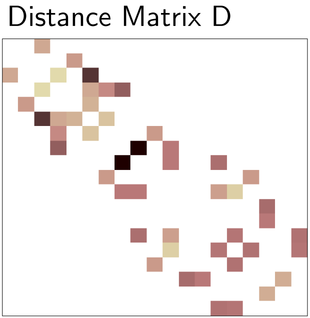
Distance Matrix D



- How is it related to Matrix Completion?
  - Need to find the missing entries
  - $\text{rank}(D) = 4$
- How is it different?
  - Non-uniform sampling
  - Rich information not used in Matrix Completion

The model





- MDS-MAP [Shang et al. '03]
  1. Fill in the missing entries with shortest paths
  2. Compute rank-4 approximation

Main results

Theorem

For $R > C\sqrt{\frac{\log n}{n}}$, with high probability,

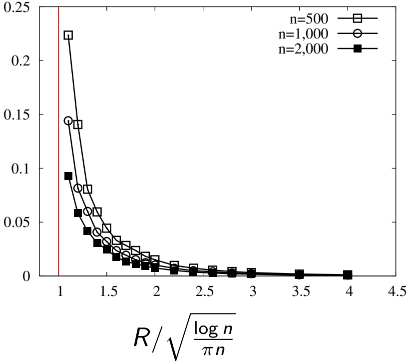
$$\text{RMSE} \le \frac{C}{R}\sqrt{\frac{\log n}{n}} + o(1)\ .$$

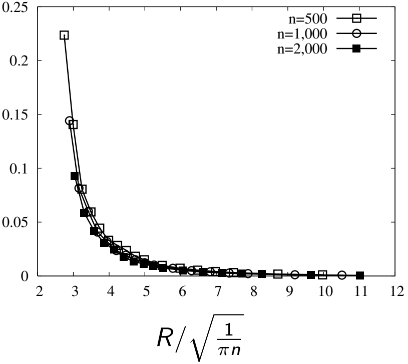
- $\text{RMSE} = \left(\frac{1}{n^2}\sum_{i,j}\left(\mathsf{D} - \widehat{\mathsf{D}}\right)^2_{ij}\right)^{1/2}$
- Lower Bound:
  If $R < \sqrt{\frac{\log n}{\pi n}}$, then the graph is disconnected
- Generalized to quantized measurements and distributed algorithms
- We can add Greedy Minimization step

Oh, Karbasi, Montanari, *Information Theory Workshop*, 2010
Karbasi, Oh, *ACM SIGMETRICS*, 2010

Numerical simulation

RMSE





Conclusion

- Matrix Completion is an important problem with many practical applications
- OptSpace is an efficient algorithm for Matrix Completion
- OptSpace achieves performance close to the fundamental limit

Special thanks to:

PMCRQOAVNBETHRDAZOKNWXAOURHLIO

Special thanks to:

PMCRQOAVNBETHRDAZOKNWXAOURHLIO

Special thanks to:

PMCRQOAVNBETHRDAZOKNWXAOURHLIO

Special thanks to:

PMCRQOAVNBETHRDAZOKNWXAOURHLIO

Special thanks to:

PMCRQOAVNBETHRDAZOKNWXAOURHLIO

Special thanks to:

- Officemates: Morteza, Yash, Jose, Raghu, Satish

- Friends: Mohsen, Adel, Farshid, Fernando, Arian, Haim, Sachin, Ivana, Ahn, Cha, Choi, Rhee, Kang, Kim, Lee, Na, Park, Ra, Seok, Song

Special thanks to:

- My family and Kyung Eun

Thank you!