LOW RANK MATRIX COMPLETION: CONVEX, NON-CONVEX AND GREEDY APPROACHES

Jieping Ye

University of Michigan

Joint work with Zheng Wang, Ming-Jun Lai, Zhaosong Lu, Wei Fan, and Hasan Davulcu

First International Workshop on Machine Learning Methods for Recommender Systems, Vancouver, British Columbia, Canada, May 2nd, 2015.

Outline

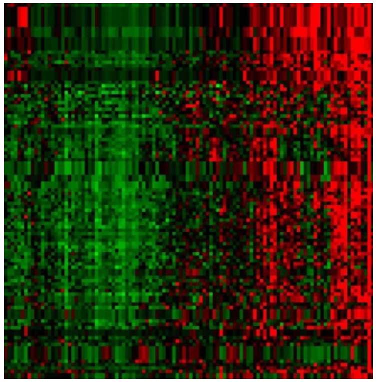
- Background
- Trace Norm Formulation
- Matrix factorization
- Orthogonal Rank-One Matrix Pursuit
- Evaluation
- Summary

Matrix Completion

Image

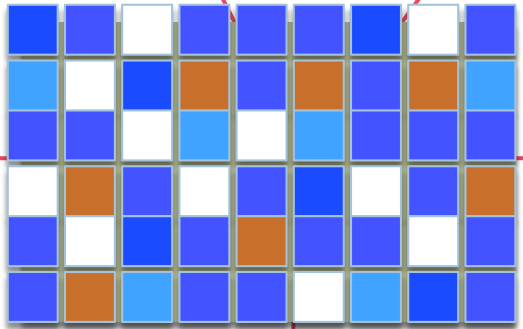
Inpainting

Microarray data imputation



Video

Recovery

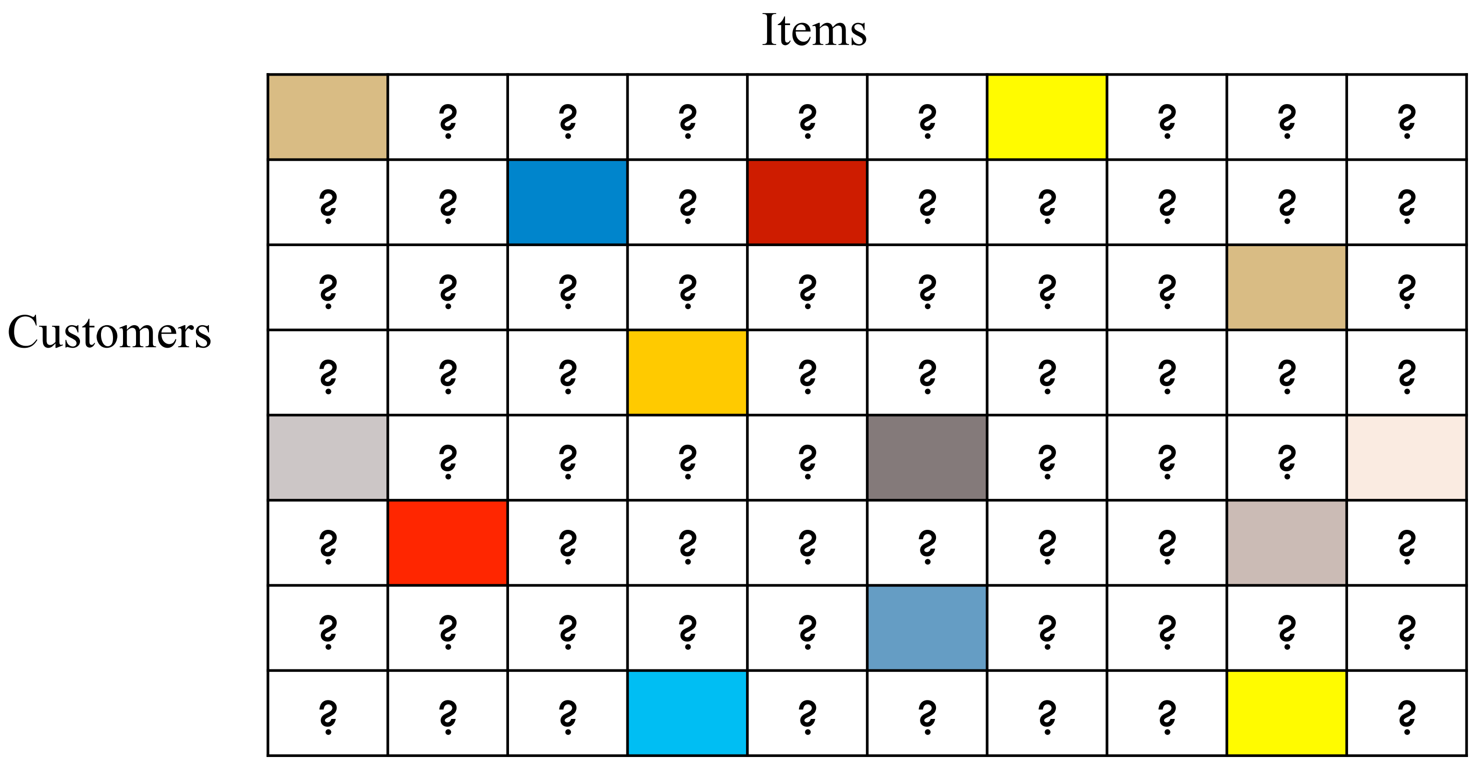


Collaborative filtering

Matrix Completion

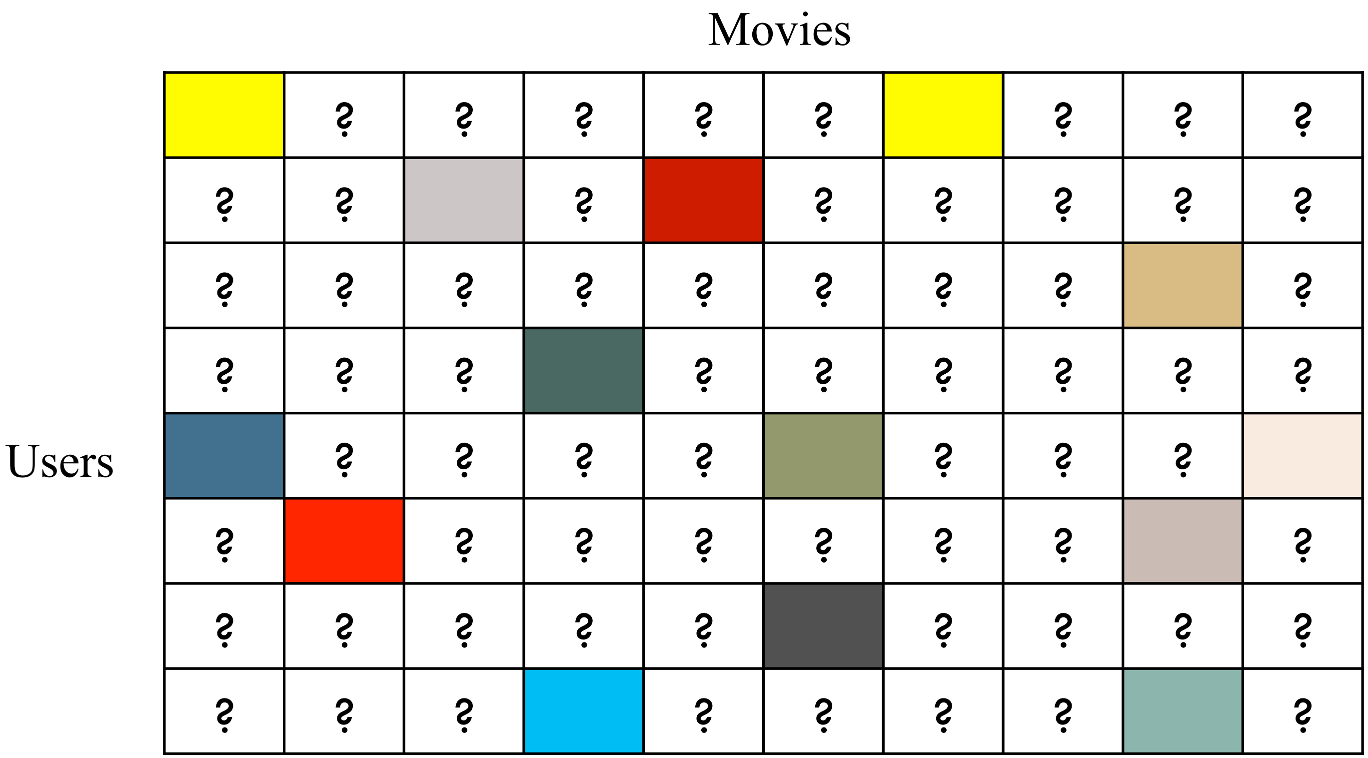
SDN

Collaborative Filtering



- Customers are asked to rank items
- Not all customers ranked all items
- Predict the missing rankings (98.9% is missing)

The Netflix Problem

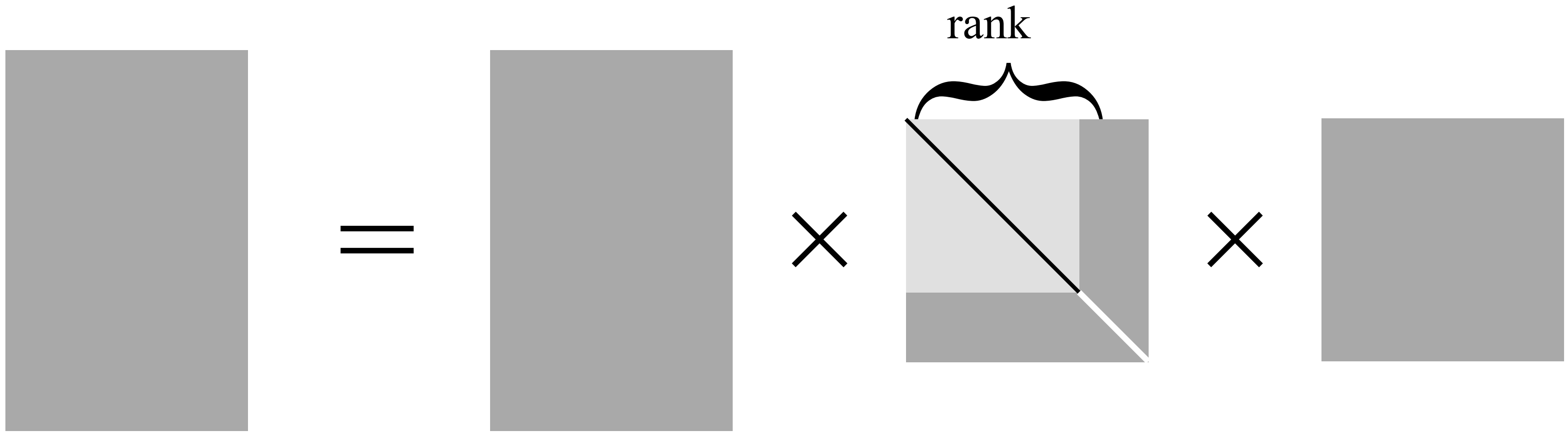


- About a million users and 25,000 movies
- Known ratings are sparsely distributed

Preferences of users are determined by a small number of factors → low rank

Matrix Rank

- The number of independent rows or columns
- The singular value decomposition (SVD):



Low Rank Matrix Completion

- Low rank matrix completion with incomplete observations can be formulated as:

$$\min_{X} \quad rank(X)$$
$$s.t. \quad P_\Omega(X) = P_\Omega(Y)$$

with the projection operator defined as: $P_\Omega(X) = \begin{cases} x_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$

Other Low-Rank Problems

- Multi-Task/Class Learning
- Image compression
- System identification in control theory
- Structure-from-motion problem in computer vision
- Low rank metric learning in machine learning
- Other settings:
  - low-degree statistical model for a random process
  - a low-order realization of a linear system
  - a low-order controller for a plant
  - a low-dimensional embedding of data in Euclidean space

Two Formulations for Rank Minimization

$$\min \quad \text{loss}(X) + \lambda * \text{rank}(X)$$

$$\begin{array}{ll} \min & \text{rank}(X) \\ \text{subject to} & \text{loss}(X) \leq \varepsilon \end{array}$$

Rank minimization is NP-hard

$$\text{loss}(X) = \frac{1}{2} \left\| P_\Omega(\mathrm{X}) - P_\Omega(\mathrm{Y}) \right\|_F^2$$

Trace Norm (Nuclear Norm)

Trace norm of a matrix is the sum of its singular values:

$$X = U \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix} V^T$$

$$\| X \|_* = \sum_{i=1}^{k} \sigma_i$$

- **trace norm ⇔ 1-norm of the vector of singular values**
- **trace norm is the convex envelope of the rank function over the unit ball of spectral norm ⇒ a convex relaxation**

Two Convex Formulations

$$\min \quad \text{loss}(X) + \lambda \times \|X\|_*$$

$$\begin{array}{ll} \min & \|X\|_* \\ \text{subject to} & \text{loss}(X) \leq \varepsilon \end{array}$$

Trace norm minimization is convex

- Can be solved by semi-definite programming
  - Computationally expensive
- Recent more efficient solvers:
  - Singular value thresholding (Cai et al, 2008)
  - Fixed point method (Ma et al, 2009)
  - Accelerated gradient descent (Toh & Yun, 2009, Ji & Ye, 2009)

Trace Norm Minimization

- Trace norm convex relaxation

$$\min_{X} \quad \|X\|_* \qquad s.t. \quad P_\Omega(X) = P_\Omega(Y) \quad \xrightarrow{\text{noisy case}} \quad \min_{X} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(Y)\|_F^2 + \lambda\|X\|_*$$

It can be solved by the sub-gradient method, the proximal gradient method or the conditional gradient method.

Convergence speed: sub-linear

Iteration: truncated SVD or top-SVD (Frank-Wolfe)

Ref: 1. Candes, E. J. and Recht, B. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6):717–772, 2009.
2. Jaggi, M. and Sulovsky, M. A simple algorithm for nuclear norm regularized problems. In ICML, 2010.

Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

min $f(x)$= loss(x) + λ×penalty(x)

Model

$$\mathcal{M}(x_i, \gamma_i) = \underbrace{[\text{loss}(x_i) + \langle \text{loss}'(x_i), x - x_i \rangle]}_{\text{1st order Taylor expansion}} + \underbrace{\frac{1}{2\gamma_i}\|x - x_i\|_2^2}_{\text{Regularization}} + \lambda \times \underbrace{\text{penalty}(x)}_{\text{Nonsmooth part}}$$

1st order Taylor expansion

Regularization

Nonsmooth part

Repeat

$$x_{i+1} = \arg\min \mathcal{M}(x_i, \gamma_i)$$

Until "convergence"

Convergence rate $O(1/N)$

Can the proximal operator be computed efficiently?

Proximal Operator Associated with Trace Norm

Optimization problem

$$\min_X f(X) = \text{loss}(X) + \lambda \|X\|_*$$

Associated proximal operator

$$X^* = \pi_{tr}(V) = \arg\min_X \frac{1}{2}\|X - V\|_2^2 + \lambda \times \|X\|_*$$

Closed form solution: $X^* = P\text{diag}(\tilde{\sigma})Q^{\mathrm{T}}$,

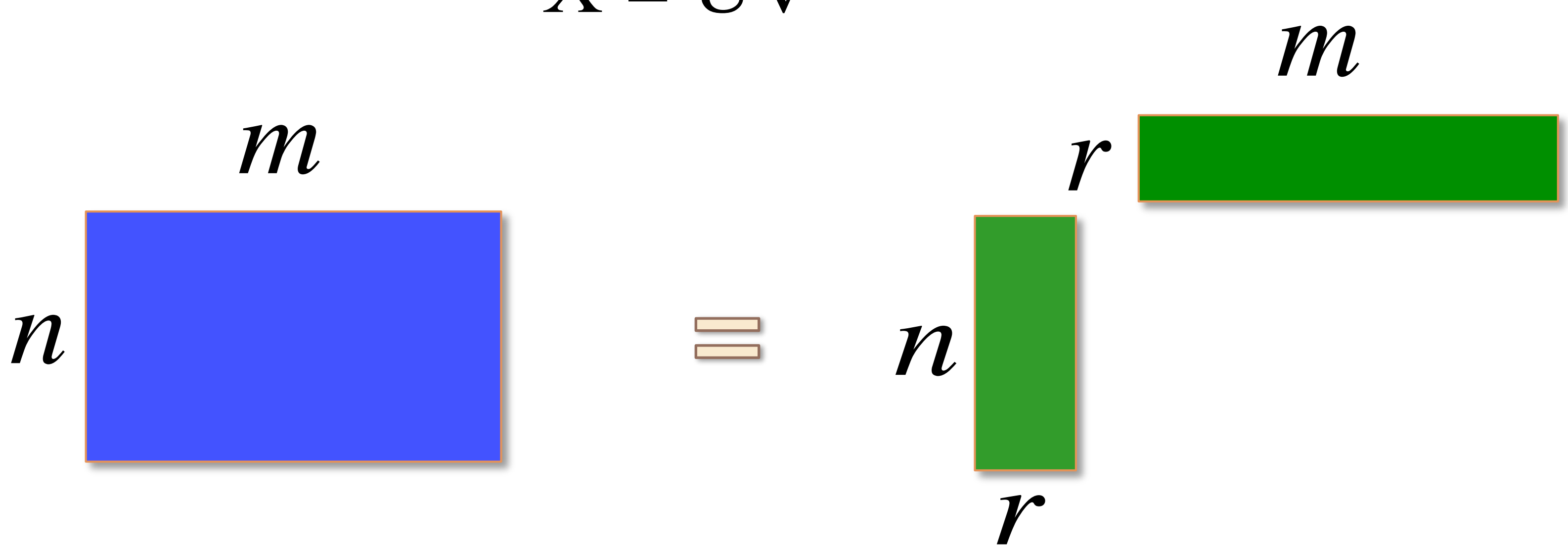
where $V = P\text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)Q^{\mathrm{T}}$ is the SVD of $V \in \mathbb{R}^{m\times n}$, $k = \min(m, n)$, $P \in \mathbb{R}^{m\times k}$, $Q \in \mathbb{R}^{n\times k}$, and

$$\tilde{\sigma}_i = \begin{cases} v_i - \lambda & \sigma_i > \lambda \\ 0 & \sigma_i \le \lambda \end{cases}$$

A Non-convex Formulation via Matrix Factorization

- Rank-r matrix X can be written as a product of two smaller matrices U and V

$$\mathrm{X} = \mathrm{U}\mathrm{V}^T$$



$$\|\mathrm{X}\|_* = \min_{\mathrm{X}=\mathrm{U}\mathrm{V}^T} \frac{1}{2}\left(\|\mathrm{U}\|_F^2 + \|\mathrm{V}\|_F^2\right)$$

Alternating Optimization

$$\min_{U,V} \quad \left\| P_\Omega(UV^T) - P_\Omega(Y) \right\|_F^2 + \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2 \right)$$

Non-convex

- Can be solved via
  - Alternating minimization (Jain et al, 2012)
  - Augmented Lagrangian (Wen et al, 2007)

Summary of Two Approaches

- Trace norm convex relaxation

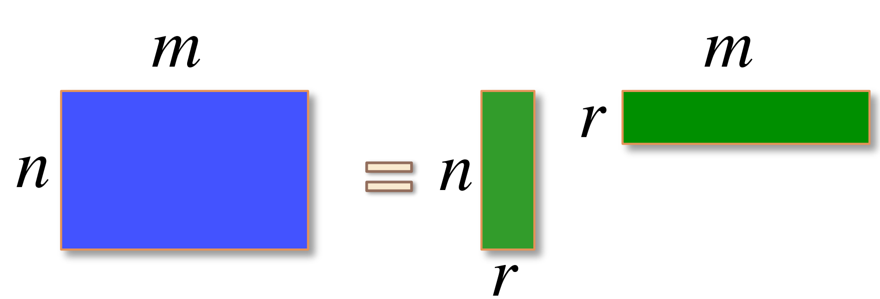
$$\begin{aligned} \min_{\mathrm{X}} \quad & \|\mathrm{X}\|_* \\ s.t. \quad & P_\Omega(\mathrm{X}) = P_\Omega(\mathrm{Y}) \end{aligned} \quad \xrightarrow{\text{noisy case}} \quad \min_{\mathrm{X}} \ \|P_\Omega(\mathrm{X}) - P_\Omega(\mathrm{Y})\|_F^2 + \lambda \|\mathrm{X}\|_*$$

Projection operator: $$P_\Omega(\mathrm{X}) = \begin{cases} x_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$$

- Bilinear non-convex relaxation

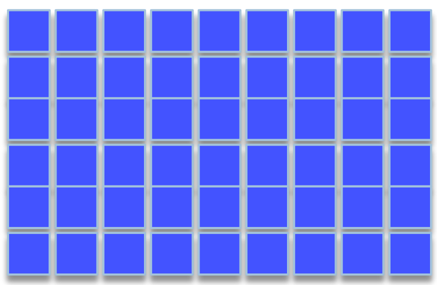
$$\mathrm{X} = \mathrm{U}\mathrm{V}^T$$

$$\min_{\mathrm{U},\mathrm{V}} \quad \|P_\Omega(\mathrm{U}\,\mathrm{V}^T) - P_\Omega(\mathrm{Y})\|_F^2$$

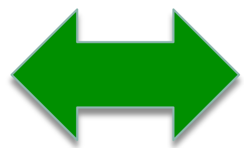


Rank-One Matrix Space

$$\mathrm{X}$$

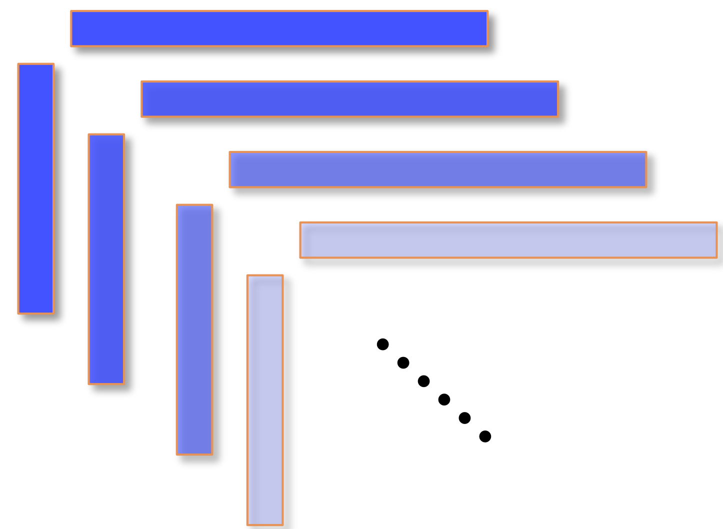


$$\mathrm{X} = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$



SVD

$$\sum_{i \in I} \theta_i \mathrm{M}_i$$



Rank-one matrices with unit norm as *Atoms*

$\mathrm{M} \in \Re^{n \times m}$ for $\mathrm{M} = uv^T \quad u \in \Re^n \quad v \in \Re^m$

Matrix Completion in Rank-One Matrix Space

- Matrix completion in rank-one matrix space

$$\min_{\boldsymbol{\theta}\in\Re^{I},\{M_i\}} \quad \|\boldsymbol{\theta}\|_0$$
$$s.t. \quad P_\Omega(\mathrm{X}(\theta)) = P_\Omega(\mathrm{Y})$$

with the estimated matrix in the rank-one matrix space as $\mathrm{X}(\boldsymbol{\theta}) = \sum_{i\in I}\theta_i \mathrm{M}_i$

- Reformulation in the noisy case

$$\min_{\mathrm{X}(\boldsymbol{\theta})} \quad \|P_\Omega(\mathrm{X}(\theta)) - P_\Omega(\mathrm{Y})\|_F^2$$
$$s.t. \quad \|\boldsymbol{\theta}\|_0 \le r$$

We solve this problem using an orthogonal matching pursuit type greedy algorithm. The candidate set is an infinite set composed by all rank-one matrices

$$\mathrm{M} \in \Re^{n\times m}$$

Orthogonal Matching Pursuit

- Greedy algorithm to iteratively solve an optimization problem with a solution spanned by the bases in a given (over-complete) dictionary

$$D = \{d^{(1)}, d^{(2)}, \ldots, d^{(T)}\}$$

$$\min_{\hat{x}} \quad \|x - \hat{x}\|^2$$
$$s.t. \quad \hat{x} = \sum_{i=1}^{r} \theta_i d_i$$

Iteration *k*:

Step 1: basis selection

$$d_i = \operatorname*{argmax}_{d \in D} \left|\langle r, d \rangle\right| \qquad r = x - \sum_{i=1}^{k-1} \theta_i d_i$$

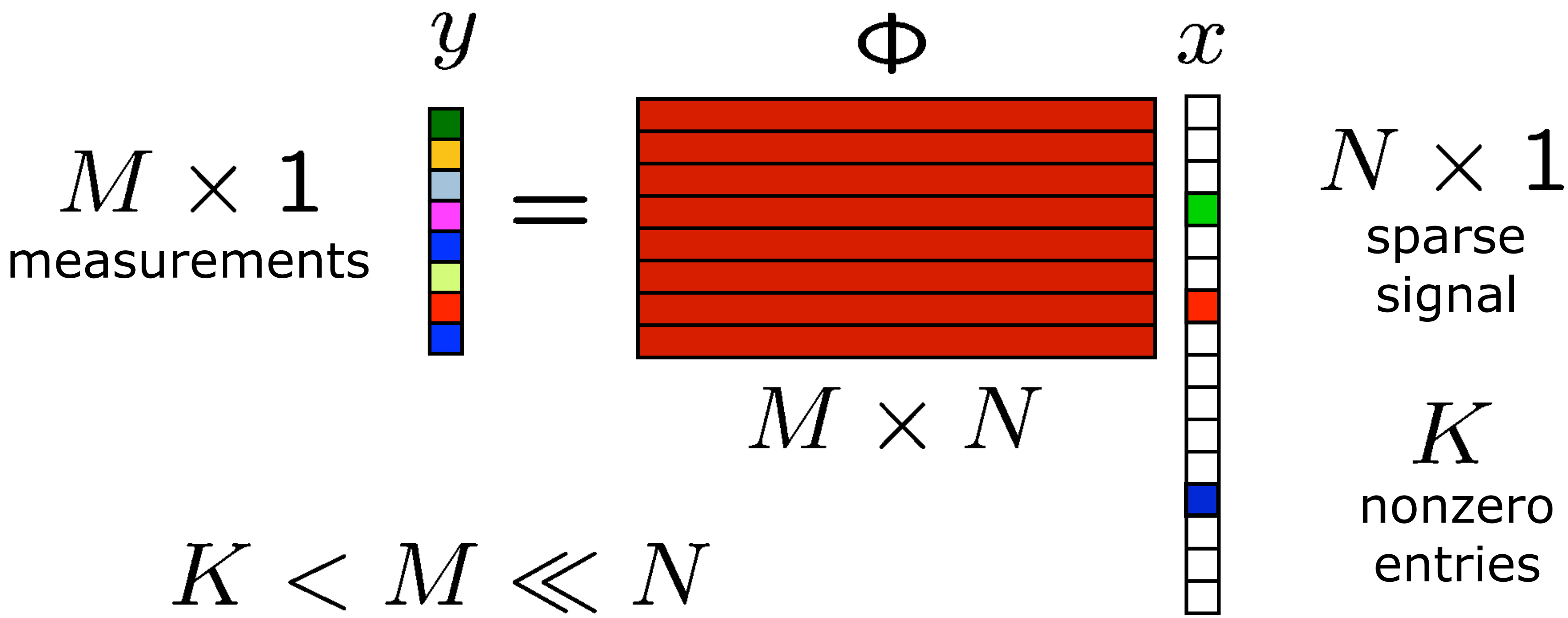
Step 2: orthogonal projection

$$\theta = \operatorname*{argmax}_{\theta} \left\| x - \sum_{i=1}^{k} \theta_i d_i \right\| \qquad \hat{x} = \sum_{i=1}^{k} \theta_i d_i$$

Compressive Sensing

- When data is sparse/compressible, can directly acquire a ***condensed representation***

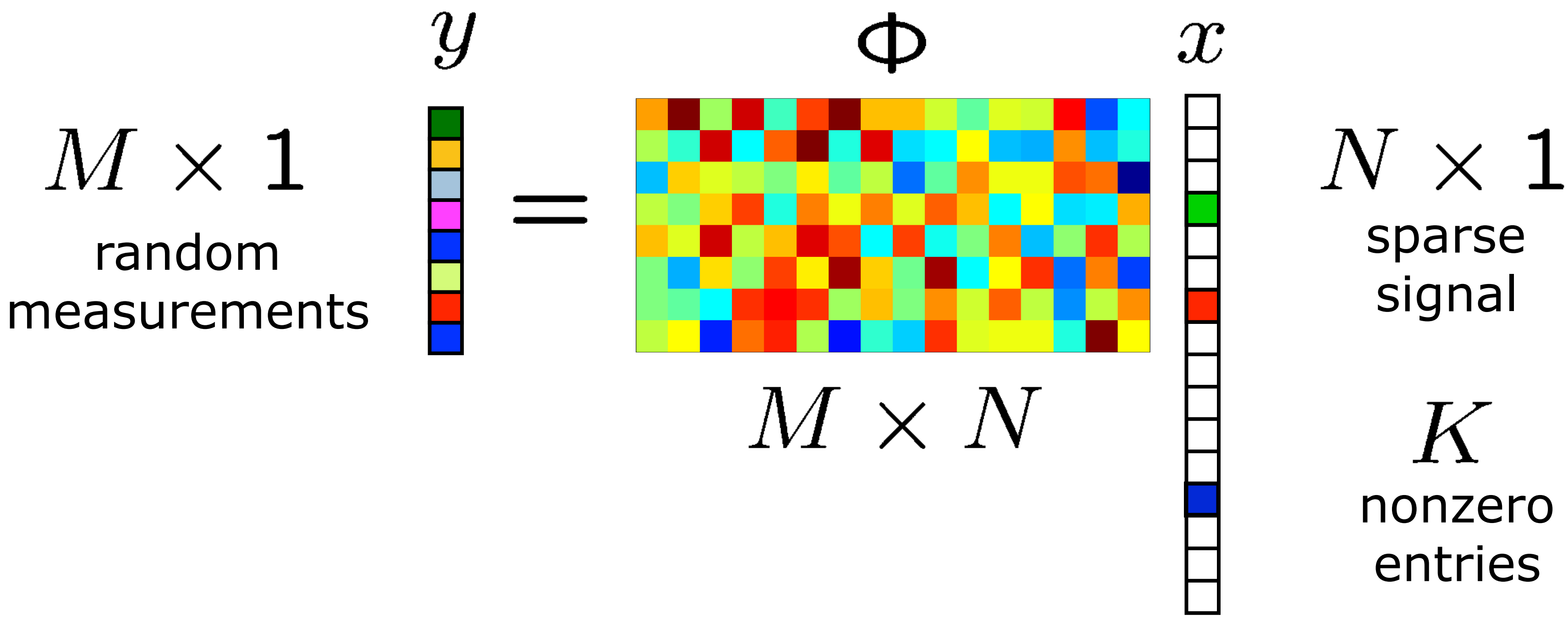
$$y = \Phi x$$



$$K < M \ll N$$

Baraniuk et al., 2012

Convex Formulation

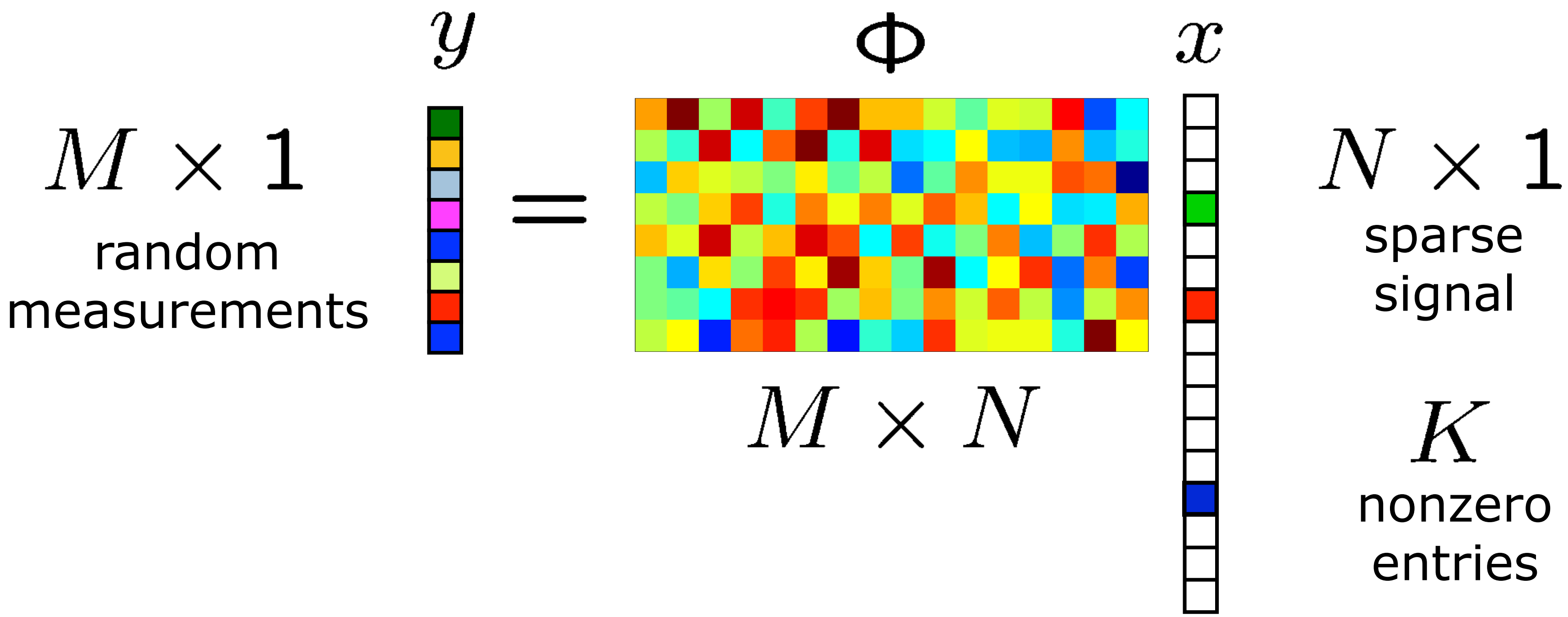


- Signal **recovery** via ℓ_1 optimization
[Candes, Romberg, Tao; Donoho]

$$\widehat{x} = \arg\min_{y=\Phi x} \|x\|_1$$

Baraniuk et al., 2012

Greedy Algorithms

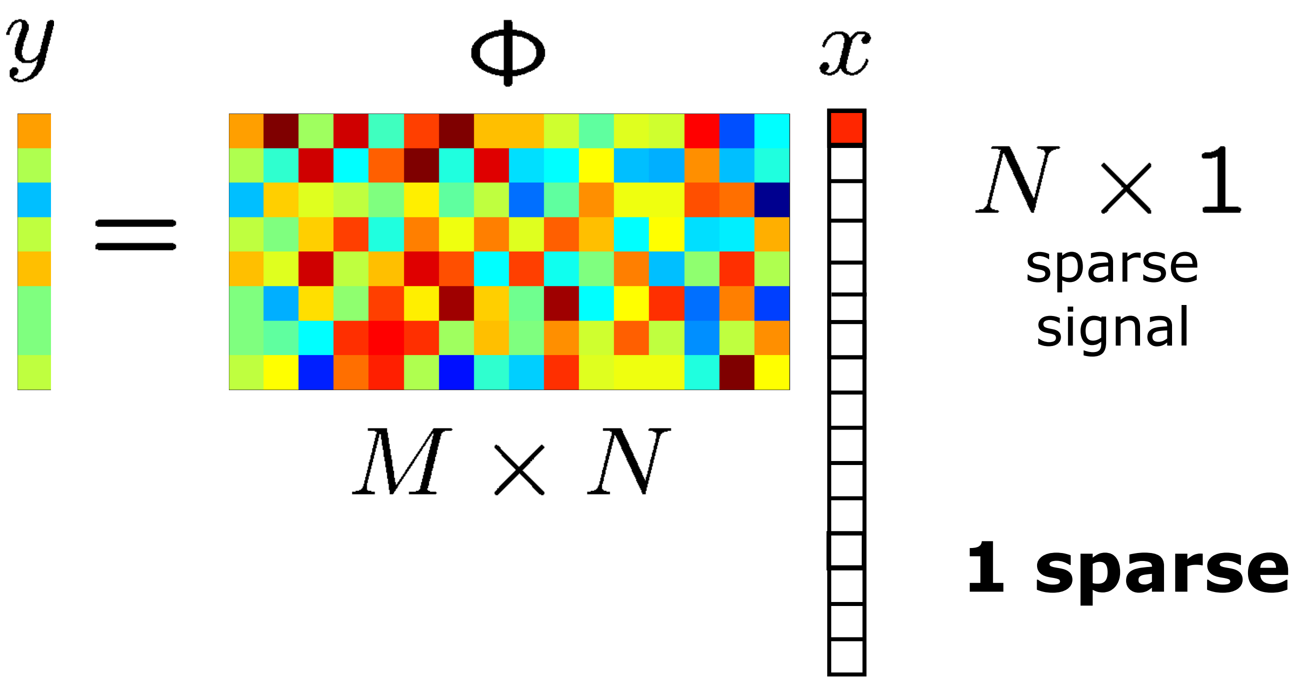


- Signal **recovery** via iterative greedy algorithms
  - (orthogonal) matching pursuit [Gilbert, Tropp]
  - iterated thresholding [Nowak, Figueiredo; Kingsbury, Reeves; Daubechies, Defrise, De Mol; Blumensath, Davies; …]
  - CoSaMP [Needell and Tropp]

Baraniuk et al., 2012

Greedy Recovery Algorithm (1)

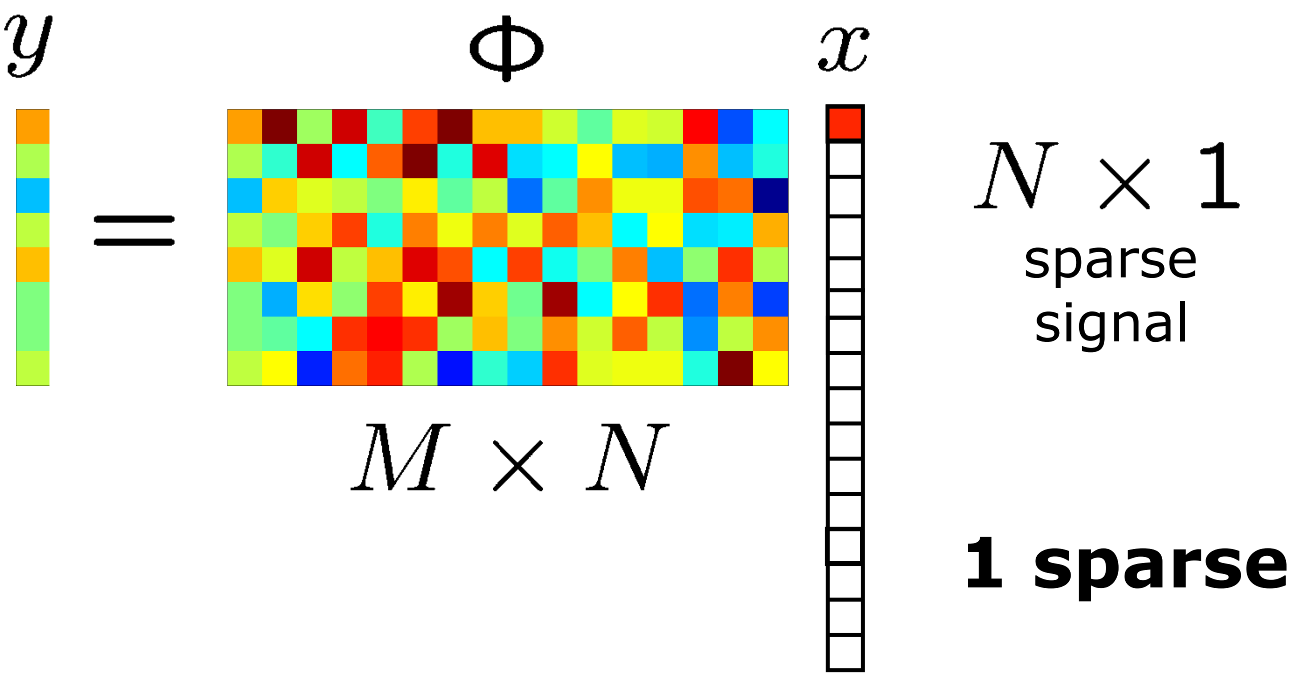
- Consider the following problem



- Can we recover the **support**?

Baraniuk et al., 2012

Greedy Recovery Algorithm (2)

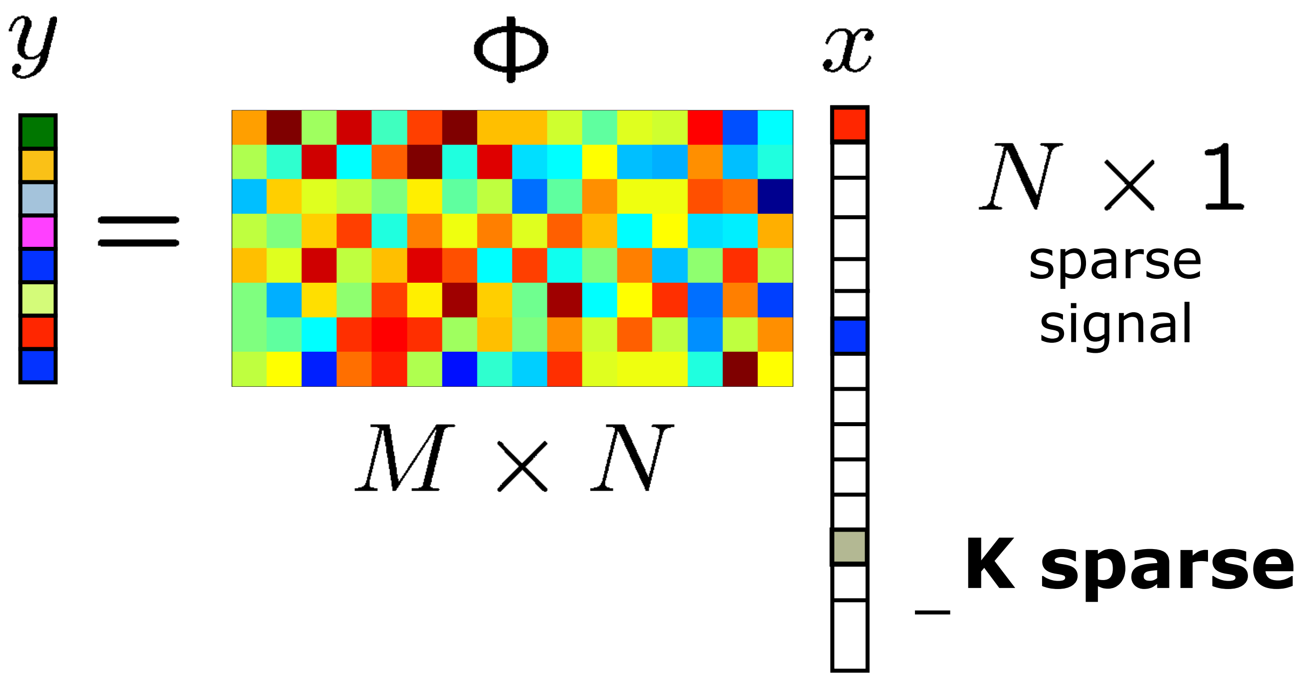


- If $\Phi = [\phi_1, \phi_2, \ldots, \phi_N]$
  then $\arg\max | \langle \phi_i, y \rangle |$ gives the support of x

- How to extend to K-sparse signals?

Baraniuk et al., 2012

Greedy Recovery Algorithm (3)



residue: $r = y - \Phi \widehat{x}_{k-1}$

find atom: $k = \arg\max \left| \langle \phi_i, r \rangle \right|$

Add atom to support: $S = S \bigcup \{k\}$

Signal estimate $x_k = (\Phi_S)^{\dagger} y$

Baraniuk et al., 2012

Orthogonal Matching Pursuit

goal:
given $y = \Phi x$, recover a sparse x
columns of Φ are unit-norm

initialize: $\widehat{x}_0 = 0, r = y, \Lambda = \{\}, i = 0$

iteration:

- $i = i + 1$
- $b = \Phi^T r$
- $\boxed{k = \arg\max\{|b(1)|, |b(2)|, \ldots, |b(N)|\}}$ **Find atom with largest support**
- $\Lambda = \Lambda \bigcup k$
- $\boxed{(\widehat{x}_i)_{|\Lambda} = (\Phi_{|\Lambda})^{\dagger} y, \ (\widehat{x}_i)_{|\Lambda^c} = 0}$ **Update signal estimate**
- $\boxed{r = y - \Phi \widehat{x}_i}$ **Update residual**

Baraniuk et al., 2012

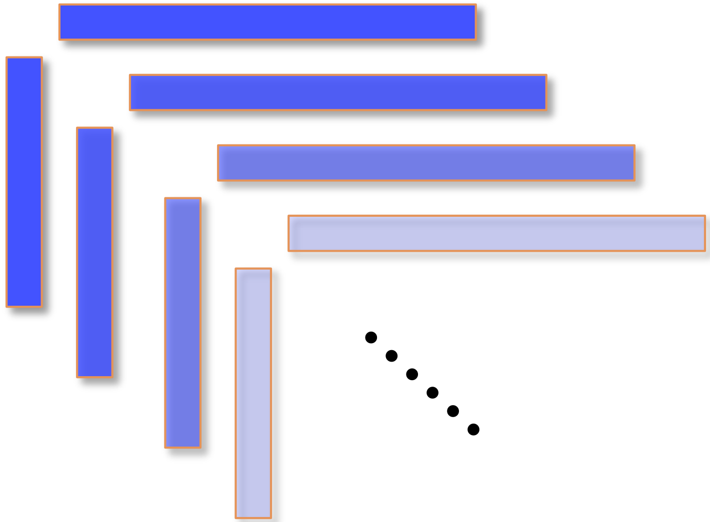
Orthogonal Rank-One Matrix Pursuit for Matrix Completion

- Matrix completion in rank-one matrix space

$$\min_{\mathrm{X}(\boldsymbol{\theta})} \quad \left\| P_\Omega(\mathrm{X}(\theta)) - P_\Omega(\mathrm{Y}) \right\|_F^2$$

$$s.t. \quad \|\boldsymbol{\theta}\|_0 \le r$$

$$\sum_{i \in I} \theta_i \mathrm{M}_i$$



$$\mathrm{X}(\boldsymbol{\theta}) = \sum_{i \in I} \theta_i \mathrm{M}_i$$

We solve this problem using an orthogonal matching pursuit type greedy algorithm. The candidate set is an infinite set composed by all rank-one matrices.

Top-SVD: Rank-One Matrix Basis

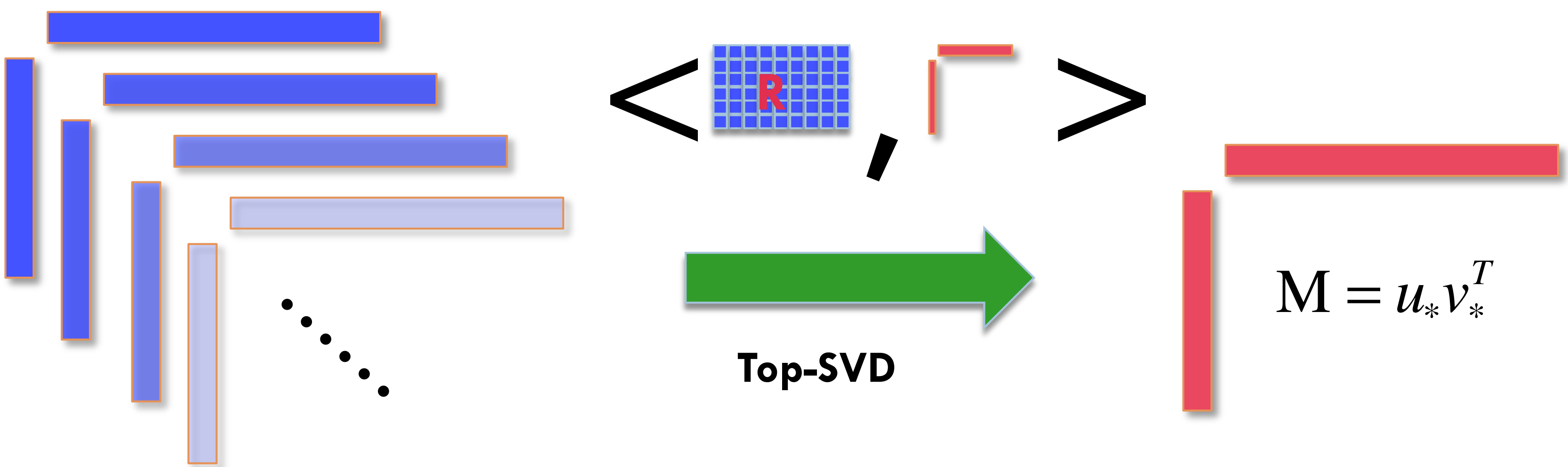
Step 1: basis construction

with residual matrix

$$[u_*, v_*] = \underset{\|u\|=1, \|v\|=1}{\operatorname{argmax}} \left\langle \mathrm{R}, uv^T \right\rangle = u^T \,\mathrm{R}\, v \qquad \mathrm{R} = \mathrm{Y}_\Omega - \mathrm{X}_\Omega$$

$\mathrm{M} = u_* v_*^T$ is selected from all rank-one matrices with unit norm.

All rank-one matrices



Infinite size

Rank-One Matrix Pursuit Algorithm

- **Step 1**: construct the optimal rank-one matrix basis

$$[u_*, v_*] = \underset{u,v}{\operatorname{argmax}} \left\langle (\mathrm{Y} - \mathrm{X}_k)_\Omega, uv^T \right\rangle \qquad \mathrm{M}_{k+1} = u_* v_*^T$$

This is the top singular vector pair, which can be solved efficiently by power method.

This generalizes OMP with *infinite* dictionary set of all rank-one matrices $\mathrm{M} \in \Re^{n \times m}$

- **Step 2**: calculate the optimal weights for current bases

$$\theta^k = \underset{\theta \in \Re^k}{\arg\min} \left\| \sum_i \theta_i \mathrm{M}_i - \mathrm{Y} \right\|_\Omega^2$$

This is a least squares problem, which can be solved incrementally.

Linear Convergence

- Linear upper bound for the algorithm to converge

Theorem 3.1. *The rank-one matrix pursuit algorithm satisfies*

$$||\mathbf{R}_k|| \leq \gamma^{k-1}\|\mathbf{Y}\|_\Omega, \quad \forall k \geq 1.$$

γ *is a constant in* $[0, 1)$.

This is significantly different from the standard MP/OMP algorithm with a finite dictionary, which are known to have a sub-linear convergence speed at the worst case.

At each iteration, we guarantee a significant reduction of the residual, which depends on the top singular vector pair pursuit step.

Z. Wang et al. **ICML'14**; **SIAM J. Scientific Computing 2015**

Efficiency and Scalability

- An efficient and scalable algorithm for matrix completion: Rank-One Matrix Pursuit
  - **Scalability**: top-SVD
  - **Convergence**: linear convergence

Z. Wang et al. **ICML'14**; **SIAM J. Scientific Computing 2015**

Related Work

- Atomic decomposition $$\mathrm{X} = \sum_{i \in I} \theta_i \mathrm{M}_i$$

  It can be solved by matching pursuit type algorithms.

- Vs. Frank-Wolfe algorithm (FW)

  Similarity: top-SVD

  Difference: linear convergence Vs. sub-linear convergence

- Vs. existing greedy approach (ADMiRA)

  Similarity: linear convergence

  Difference: 1. top-SVD Vs. truncated SVD
  2. no extra condition for linear convergence

Ref: Lee, K. and Bresler, Y. Admira: atomic decomposition for minimum rank approximation. IEEE Trans. on Information Theory, 56(9):4402–4416, 2010.

Time and Storage Complexity

- Time complexity

	R1MP	ADMiRA & AltMin	JS(FW)	Proximal	SVT
Each Iter.	$O(\|\Omega\|)$	$O(r\|\Omega\|)$	$O(\|\Omega\|)$	$O(r\|\Omega\|)$	$O(r\|\Omega\|)$
Iterations	$O(\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	$O(1/\varepsilon)$	$O(1/\sqrt{\varepsilon})$	$O(1/\varepsilon)$
Total	$O(\|\Omega\|\log(1/\varepsilon))$	$O(r\|\Omega\|\log(1/\varepsilon))$	$O(\|\Omega\|/\varepsilon)$	$O(r\|\Omega\|/\sqrt{\varepsilon})$	$O(r\|\Omega\|/\varepsilon)$

minimum iteration cost
+ linear convergence

- Storage complexity

$O(k\,|\Omega|)$ — It is large when *k* keeps increasing.

$O(|\Omega|)$ is more suitable for large-scale problems.

Economic Rank-One Matrix Pursuit

- **Step 1**: find the optimal rank-one matrix basis

$$[u_*, v_*] = \operatorname*{argmax}_{u,v} \langle (\mathrm{Y} - \mathrm{X}_k)_\Omega, uv^T \rangle \qquad \mathbf{M}_{k+1} = u_* v_*^T$$

- **Step 2: calculate the weights for two matrices**

$$\boldsymbol{\alpha} = \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \Re^2} \left\| \alpha_1 \boxed{\mathrm{X}_k} + \alpha_2 \boxed{\mathrm{M}_{k+1}} - \mathrm{Y} \right\|_\Omega^2$$

$$\theta_i^{k-1} = \theta_i^{k-1} \alpha_1 \qquad \theta_i^k = \alpha_2$$

- It retains the linear convergence

Theorem 4.1. *The economic rank-one matrix pursuit algorithm satisfies*

$$||\mathbf{R}_k|| \;\leq\; \tilde{\gamma}^{k-1} \|\mathbf{Y}\|_\Omega, \quad \forall k \geq 1.$$

$\tilde{\gamma}$ *is a constant in* $[0, 1)$.

Experiments

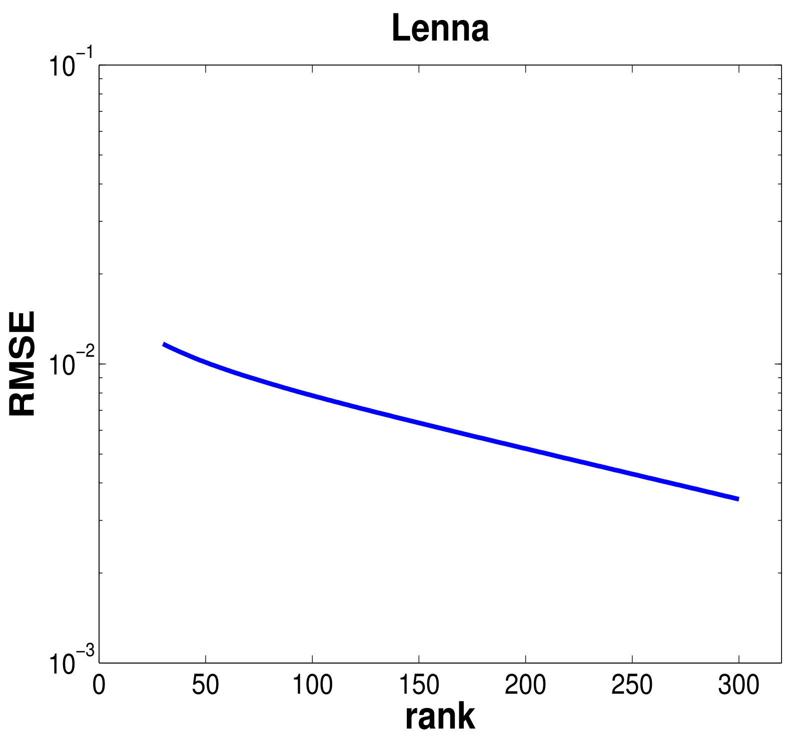
- Experiments
  - Collaborative filtering
  - Image recovery
  - Convergence property
- Competing algorithms
  - singular value projection (SVP)
  - spectral regularization algorithm (SoftImpute)
  - low rank matrix fitting (LMaFit)
  - alternating minimization (AltMin)
  - boosting type accelerated matrix-norm penalized solver (Boost)
  - Jaggi's fast algorithm for trace norm constraint (JS)
  - greedy efficient component optimization (GECO)
  - Rank-one matrix pursuit (R1MP)
  - Economic rank-one matrix pursuit (ER1MP)

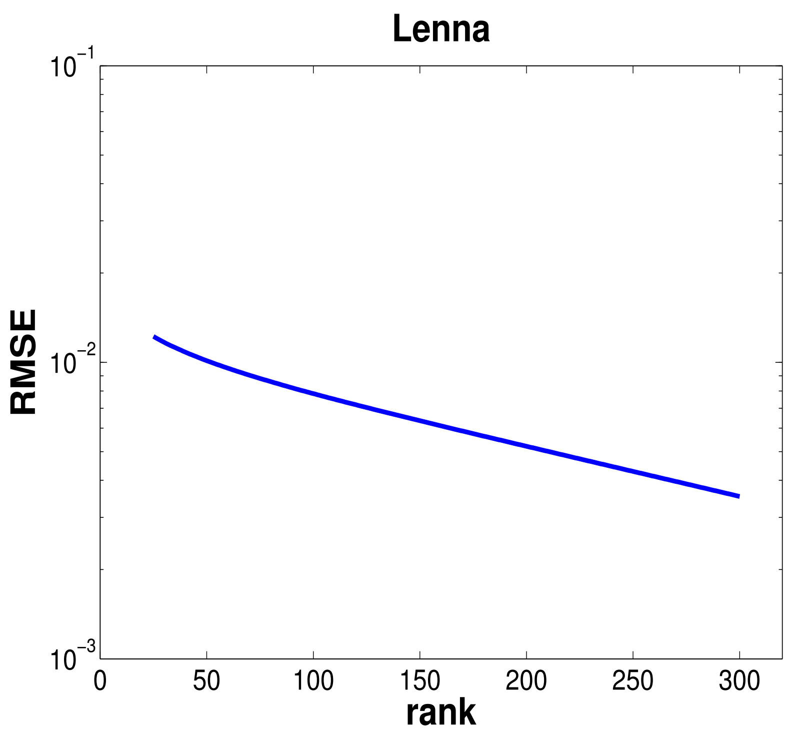
trace norm minimization

alternating optimization

atomic decomposition

Convergence





Residual curves of the Lena image for R1MP and ER1MP in log-scale

Collaborative Filtering

Running time for different algorithms

Dataset	SVP	SoftImpute	LMaFit	AltMin	Boost	JS	GECO	R1MP	ER1MP
Jester1	18.35	161.49	3.68	11.14	93.91	29.68	$> 10^4$	1.83	0.99
Jester2	16.85	152.96	2.42	10.47	261.70	28.52	$> 10^4$	1.68	0.91
Jester3	16.58	10.55	8.45	12.23	245.79	12.94	$> 10^3$	0.93	0.34
MovieLens100K	1.32	128.07	2.76	3.23	2.87	2.86	10.83	0.04	0.04
MovieLens1M	18.90	59.56	30.55	68.77	93.91	13.10	$> 10^4$	0.87	0.54
MovieLens10M	$> 10^3$	$> 10^3$	154.38	310.82	–	130.13	$> 10^5$	23.05	13.79

Prediction accuracy in terms of RMSE

Dataset	SVP	SoftImpute	LMaFit	AltMin	Boost	JS	GECO	R1MP	ER1MP
Jester1	4.7311	5.1113	4.7623	4.8572	5.1746	4.4713	4.3680	4.3418	4.3384
Jester2	4.7608	5.1646	4.7500	4.8616	5.2319	4.5102	4.3967	4.3649	4.3546
Jester3	8.6958	5.4348	9.4275	9.7482	5.3982	4.6866	5.1790	4.9783	5.0145
MovieLens100K	0.9683	1.0354	1.2308	1.0042	1.1244	1.0146	1.0243	1.0168	1.0261
MovieLens1M	0.9085	0.8989	0.9232	0.9382	1.0850	1.0439	0.9290	0.9595	0.9462
MovieLens10M	0.8611	0.8534	0.8625	0.9007	–	0.8728	0.8668	0.8621	0.8692

Summary

- Matrix completion background
- Trace norm convex formulation
- Matrix factorization: non-convex formulation
- Orthogonal rank-one matrix pursuit
  - Efficient update: top SVD
  - Fact convergence: linear rate
- Extensions
  - Tensor completion
  - Screening for matrices