PU Learning for Matrix Completion

Cho-Jui Hsieh
Dept of Computer Science
UT Austin

ICML 2015

Joint work with N. Natarajan and I. S. Dhillon

Matrix Completion

- Example: movie recommendation
- Given a set Ω and the values M_Ω, how to predict other elements?

movies

users

		6.8			1.9	
		7.9				4.4
5.1			?		2.3	
6.8						5.3
				9.3		
	8.8			8.0		
5.1			7.7		1.9	

Matrix Completion

- Assumption: the underlying matrix M is low rank.
- Recover M by solving

$$\min_{\|X\|_* \le t} \sum_{i,j \in \Omega} (X_{ij} - M_{ij})^2,$$

$\|X\|_*$ is the nuclear norm (the best convex relaxation of $\text{rank}(X)$).

0.9	2.7
2.1	1.8
2.9	1.1
3.4	1.7
1.6	1.6
2.1	0.7
1.9	1.6

*

1	3.3	2.2	1.8	2.8	0.6	0.8
2	2.7	1.8	2.7	3.0	0.5	1.5

=

		6.8			1.9	
		7.9				4.4
5.1			8.2		2.3	
6.8						5.3
				9.3		
	8.8			8.0		
5.1			7.7		1.9	

One Class Matrix Completion

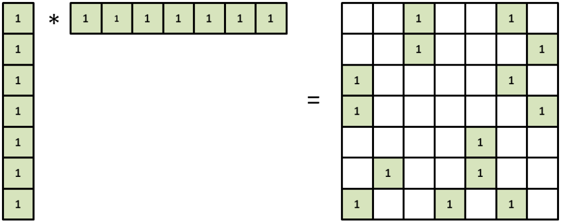
- All the observed entries are 1's.
- Examples:
  - Link prediction using social networks (only friend relationships)
  - Product recommendation using purchase networks.
  - "Follows" in Twitter, "like" in Facebook, ...

users

users							
			1			1	
			1				1
	1	?	?	?	?	1	?
	1						1
					1		
		1			1		
	1			1		1	

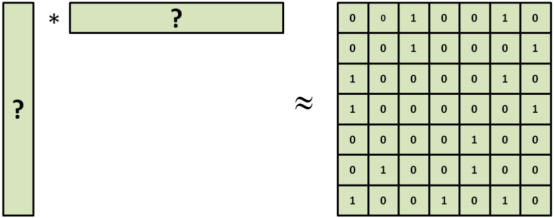
Can we apply matrix completion?

- Minimizing the loss on the observed 1's.
- Will get a trivial solution.



Can we apply matrix completion?

- Treat all the missing entries as zeroes, and minimizing the loss on all the entries.
- 99% elements are zero $\Rightarrow$ tend to fit zeroes instead of ones.



Challenges

- All the observed entries are 1's.
- '0' is unlabeled entries: can be either 0 or 1 in the underlying matrix.
- PU (Positive and Unlabeled) Matrix Completion:
  - How to formulate the problem?
  - How to solve the problem?
  - What's the sample complexity?
  - What's the time complexity?

Outline

- Non-deterministic setting – Shifted Matrix Completion
- Deterministic setting – Biased Matrix Completion
- Extension to PU matrix completion with features.
- Experimental Results

Non-deterministic setting

- $M_{ij} \in [0, 1]$, M is low-rank.
- The generating process: M (underlying) $\to Y$ (0-1 matrix) $\to \Omega_1$.
- An underlying $0-1$ matrix Y is generated by

$$Y_{ij} = \begin{cases} 1 & \text{with prob. } M_{ij} \\ 0 & \text{with prob. } 1 - M_{ij}. \end{cases}$$

- Ω_1 sampled from $\{(i, j) \mid Y_{ij} = 1\}$, the sample rate is $1 - \rho = |\Omega_1| / \|Y\|_0$

.17	.24	.30	.33	.19
.30	.41	.52	.56	.32
.11	.16	.20	.21	.12
.38	.53	.67	.72	.41
.13	.18	.23	.24	.14

M
(rank 1)

$\Rightarrow$

1		1		1
	1			1
1	1	1		1
	1			1
1		1	1	1

Y
(0-1 matrix)

$\Rightarrow$

	1			
1	1			1
	1			

A (Ω_1)

Unbiased Estimator of Error

- Find the best X to minimize the mean square error on M:

$$\min_X \sum_{i,j} (X_{ij} - M_{ij})^2 = \min_X \sum_{i,j} \ell(X_{ij}, M_{ij})$$

$$\sum_{i,j} (X_{ij} - M_{ij})^2$$

.17	.24	.30	.33	.19
.30	.41	.52	.56	.32
.11	.16	.20	.21	.12
.38	.53	.67	.72	.41
.13	.18	.23	.24	.14

M

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

Y

(0-1 matrix)

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

A

Unbiased Estimator of Error

- Find the best X to minimize the mean square error on M:

$$\min_X \sum_{i,j} (X_{ij} - M_{ij})^2 = \min_X \sum_{i,j} \ell(X_{ij}, M_{ij})$$

$$\sum_{i,j} (X_{ij} - M_{ij})^2 = E[\sum_{i,j} (X_{ij} - A_{ij})^2] + C$$

.17	.24	.30	.33	.19
.30	.41	.52	.56	.32
.11	.16	.20	.21	.12
.38	.53	.67	.72	.41
.13	.18	.23	.24	.14

M

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

Y
(0-1 matrix)

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

A

Unbiased Estimator of Error

- Find the best X to minimize the mean square error on M:
- The unbiased estimator [Natarajan et al., 2013]:

$$\tilde{\ell}(X_{ij}, A_{ij}) = \begin{cases} \frac{(X_{ij}-1)^2-\rho X_{ij}^2}{1-\rho} & \text{if } A_{ij} = 1 \\ X_{ij}^2 & \text{if } A_{ij} = 0. \end{cases}$$

$$\boxed{\sum_{i,j}(X_{ij} - M_{ij})^2} = \boxed{E[\sum_{i,j}(X_{ij} - A_{ij})^2] + C} = \boxed{E[\sum_{i,j}\tilde{\ell}(X_{ij} - A_{ij})] + C}$$

.17	.24	.30	.33	.19
.30	.41	.52	.56	.32
.11	.16	.20	.21	.12
.38	.53	.67	.72	.41
.13	.18	.23	.24	.14

M

$\Rightarrow$

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

Y

(0-1 matrix)

$\Rightarrow$

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

A (Ω_1)

Shifted Matrix Completion

- Shifted Matrix Completion.
- Solve

$$\min_X \sum_{i,j} \tilde{\ell}(X_{ij}, A_{ij}) \text{ s.t. } \|X\|_* \leq t, 1 \geq X_{ij} \geq 0.$$

Where

$$\tilde{\ell}(X_{ij}, A_{ij}) = \begin{cases} \frac{(X_{ij}-1)^2 - \rho X_{ij}^2}{1-\rho} & \text{if } A_{ij} = 1 \\ X_{ij}^2 & \text{if } A_{ij} = 0. \end{cases}$$

- Equivalent to

$$\min_X \|X - \hat{A}\|_F^2 + \lambda \|X\|_* \text{ s.t. } 1 \geq X \geq 0,$$

where

$$\hat{A}_{ij} = 1/(1-\rho) \text{ if } A_{ij} = 1$$
$$\hat{A}_{ij} = 0 \text{ if } A_{ij} = 0.$$

Error Bound for Shifted MF

- Measure the error by $R(X) = \frac{1}{n^2} \sum_{i,j} (M_{ij} - X_{ij})^2$.

Theorem: error bound for Shifted MF

Let $\hat{X}$ be the solution of the Shifted MF, then with probability at least $1-\delta$,

$$R(\hat{X}) \leq \frac{3\sqrt{\log(2/\delta)}}{n(1-\rho)} + Ct\frac{2\sqrt{n} + \sqrt[4]{s}}{(1-\rho)n^2}$$
$$= O(\frac{1}{n(1-\rho)}),$$

where C is a constant.

Deterministic Setting

- $M_{ij} \in [0,1]$ (can be generalized to other bounded matrix).
- With some threshold $q \in [0,1]$,

$$Y_{ij} = \begin{cases} 1 & \text{if } M_{ij} > q \\ 0 & \text{if } M_{ij} \le q, \end{cases}$$

- Ω_1 sampled from $\{(i,j) \mid Y_{ij} = 1\}$.
- Given Ω_1, impossible to recover M:
  for example, $M = \eta \boldsymbol{e}\boldsymbol{e}^T$ will generate $Y = \boldsymbol{e}\boldsymbol{e}^T$ for all $\eta > q$.
- So our goal is to recover Y.

.17	.24	.30	.33	.19
.30	.41	.52	.56	.32
.11	.16	.20	.21	.12
.38	.53	.67	.72	.41
.13	.18	.23	.24	.14

M
(rank 1)

$\Rightarrow$

		1	1	
	1	1	1	1
1	1	1	1	1

Y
(0-1 matrix)

$\Rightarrow$

	1		1	
	1		1	

A (Ω_1)

Biased Matrix Factorization

- Square loss: $\ell(x, a) = (x - a)^2$.
- Biased square loss:

$$\ell_\alpha(x, a) = \alpha 1_{a=1} \ell(x, 1) + (1 - \alpha) 1_{a=0} \ell(x, 0).$$

- Biased MF:

$$\hat{X} = \arg \min_{X : \|X\|_* \leq t} \sum_{i,j} \ell_\alpha(X_{ij}, A_{ij}).$$

- Recover Y:

$$\bar{X}_{ij} = \begin{cases} 1 & \text{if } \hat{X}_{ij} > q \\ 0 & \text{otherwise} \end{cases}$$

Sample Complexity

- Error: $\bar{R}(X) = \frac{1}{n^2} \sum_{i,j} 1_{X_{ij} \neq Y_{ij}}$.

Theorem: error bound for BiasMF

Let $\bar{X}$ be the solution of BiasMF. If $\alpha = \frac{1+\rho}{2}$, then with probability at least $1 - \delta$,

$$R(\bar{X}) \leq \frac{2\eta}{1+\rho} \left(Ct \frac{2\sqrt{n} + \sqrt[4]{s}}{n^2} + 3 \frac{\sqrt{\log(2/\delta)}}{n(1-\rho)} \right) = O(\frac{1}{n(1-\rho)}),$$

where $\eta = \max(1/q^2, 1/(1-q)^2, 8)$ and C is a constant.

Time Complexity

- Gradient can be efficiently solved using $O((\text{nnz})k)$ time:
  - For non-convex formulation: by Alternating Least Squares (ALS) or Cyclyc Coordinate Descent (CCD++) (Yu et al., 2012).
  - For convex formulation: by proximal gradient or active-subspace selection (Hsieh et al., 2014).
- One bit matrix completion: need $O(n^2)$ time.

Inductive Matrix Completion

- Proposed for matrix completion with features [Jain and Dhillon, 2013; Xu et al., 2013]
- Input: partially observed matrix A_Ω and features $F_u, F_v \in \mathbb{R}^{n \times d}$ associated with rows/columns.
- Recover the underlying matrix by solving

$$\min_{D \in \mathbb{R}^{d \times d}, \|D\|_* \le t} \sum_{i,j \in \Omega} (A_{ij} - (F_u D F_v^T)_{ij})^2$$

$$\begin{bmatrix} 0.9 & 2.7 \\ 2.1 & 1.8 \\ 2.9 & 1.1 \\ 3.4 & 1.7 \\ 1.6 & 1.6 \\ 2.1 & 0.7 \\ 1.9 & 1.6 \end{bmatrix} \begin{bmatrix} ? \end{bmatrix} \begin{bmatrix} 1 & 3.3 & 2.2 & 1.8 & 2.8 & 0.6 & 0.8 \\ 2 & 2.7 & 1.8 & 2.7 & 3.0 & 0.5 & 1.5 \end{bmatrix} \approx \begin{bmatrix} & & 6.8 & & & 1.9 & \\ & & 7.9 & & & & 4.4 \\ 5.1 & & & & & 2.3 & \\ 6.8 & & & & & & 5.3 \\ & & & & 9.3 & & \\ & 8.8 & & & 8.0 & & \\ 5.1 & & & 7.7 & & 1.9 & \end{bmatrix}$$

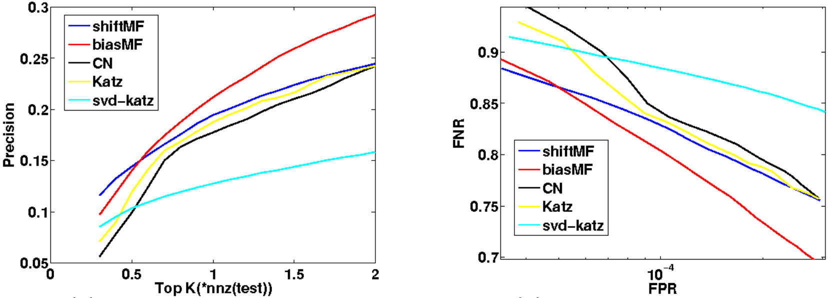
PU Inductive Matrix Completion

- Inductive Matrix Completion: recover the underlying matrix using
  1. A subset of 1s in the matrix.
  2. row and/or column features.
- Inductive shift matrix factorization—non-deterministic setting.

  Average Error $= O(\frac{1}{n(1-\rho)})$
- Inductive biased matrix factorization—deterministic setting.

  Average Error $= O(\frac{1}{n(1-\rho)})$

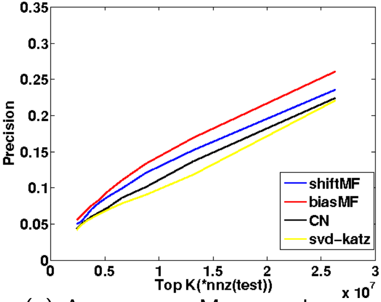
Experimental results – link prediction



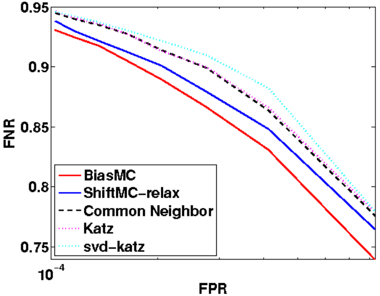
(a) Accuracy on ca-HepTh (b) FPR-FNR on ca-HepTh

Figure: Comparison of algorithms on the link prediction problem (11, 204 nodes, 235, 368 edges)

Experimental results – link prediction



(a) Accuracy on Myspace dataset.



(b) FPR-FNR on Myspace dataset.

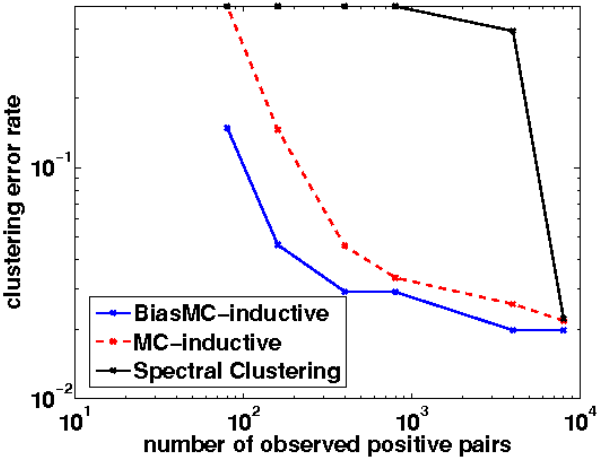
Figure: Comparison of algorithms on the link prediction problem ($2,137,264$ nodes, $90,333,122$ edges)

Application – semi-supervised clustering

- Original problem:
  - Given n samples with features $\{\boldsymbol{x}_i\}_{i=1}^n$.
  - Given partial positive and negative pairwise relationship $A \in \mathbb{R}^{n \times n}$.
  - Recover clusters (categories of samples).
  - (Yi et al, 2013) proposed to use inductive MF to solve this problem.
- Semi-supervised clustering with one class observation:
  - Only observe positive pairs Ω_1.
  - We propose a one class inductive MF to solve this problem.

Experimental results – semi-supervised clustering

- Mushroom dataset, 8142 samples, 2 clusters.



Conclusions

- Study the one class matrix completion problem.
- Proposed algorithms with nice theoretical guarantee:

  error decays with the rate of $O(1/n)$.
- Scale to large problems (millions of rows and columns).
- Applications:
  - Link prediction using social networks (only friend relationships)
  - Product recommendation using purchase networks.
  - "Follows" in Twitter, "like" in Facebook, ...