# Low-rank Matrix Completion under Monotonic Transformation

Laura Balzano, with Ravi Sastry Ganti and Rebecca Willett

University of Michigan and University of Wisconsin, Madison

Michigan Communications and Signal Processing Seminar
May 2016

# Low-rank Matrix Completion under Monotonic Transformation

Two common hurdles for handling high-dimensional data:

Our observations are incomplete: missing data.

Our observations are indirect: we observe only some unknown transformation of some true phenomenon of interest.

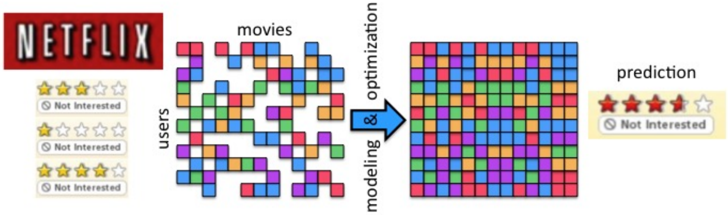
Can we recover the matrix of interest?

YES! We leverage low-rank structure in the true signal and the transformation's smoothness and monotonicity.

# Overview


1. Motivation
2. Background
3. Problem Formulation
4. Our Algorithm
5. Experiments
6. Conclusion

# Example 1: Recommender Systems



# Example 1: Recommender Systems

Mixture of hundreds of models, including gradient descent

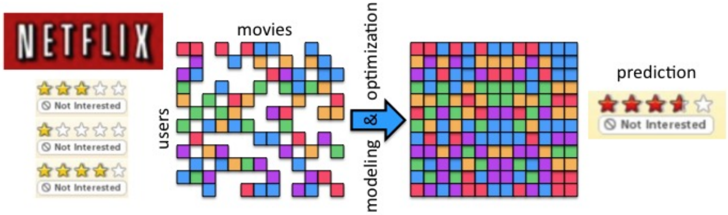
Gradient descent on low-rank parameterization

Netflix Prize

Leaderboard

| Rank | Team Name | Best Score | % Improvement | Last Submit Time |
|---|---|---|---|---|
| -- | No Grand Prize candidates yet | -- | -- | -- |
| Grand Prize - RMSE <= 0.8563 | | | | |
| -- | No Progress Prize candidates yet | -- | -- | -- |
| Progress Prize - RMSE <= 0.8625 | | | | |
| 1 | When Gravity and Dinosaurs Unite | 0.8675 | 8.82 | 2008-03-01 07:03:35 |
| 2 | BellKor | 0.8682 | 8.75 | 2008-02-28 23:40:45 |
| 3 | | 0.8708 | 8.47 | 2008-02-06 14:12:44 |
| 2007 - RMSE = 0.8712 - Winning Team: KorBell | | | | |
| 4 | KorBell | 0.8712 | 8.43 | 2007-10-01 23:25:23 |
| 5 | acmehill | 0.8720 | 8.35 | 2008-03-02 05:08:12 |
| 6 | Dan Tillberg | 0.8727 | 8.27 | 2008-03-02 08:42:29 |
| 7 | basho | 0.8729 | 8.25 | 2007-11-24 14:27:00 |
| 8 | Just a guy in a garage | 0.8740 | 8.14 | 2008-02-06 12:16:40 |
| 9 | BigChaos | 0.8748 | 8.05 | 2008-03-01 17:26:06 |
| 10 | Dinosaur Planet | 0.8753 | 8.00 | 2007-10-04 04:56:45 |
| ⋮ | | ⋮ | ⋮ | ⋮ |
| 50 | amgl | 0.8897 | 6.49 | 2007-12-23 18:44:03 |
| 51 | Remco | 0.8899 | 6.46 | 2007-04-04 06:16:56 |
| 52 | mxlg | 0.8900 | 6.45 | 2007-12-23 18:54:46 |
| 53 | JustWithSVD | 0.8900 | 6.45 | 2008-02-14 16:17:54 |
| 54 | | 0.8900 | 6.45 | 2008-02-28 09:56:20 |
| 55 | | 0.8901 | 6.44 | 2008-02-29 05:53:11 |
| | Bozo_The_Clown | 0.8902 | 6.43 | 2007-09-06 17:24:48 |

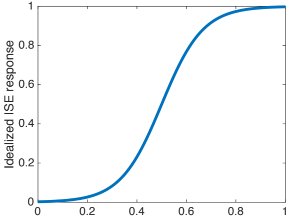
# Example 1: Recommender Systems

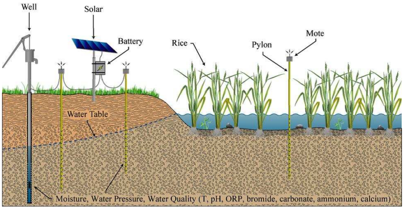


# Example 2: Blind Sensor Calibration


# Example 2: Blind Sensor Calibration

Ion Selective Electrodes have a nonlinear response to their ions (pH, ammonium, calcium, etc)




# Background

- Single Index Model
- Low-rank Matrix Completion

# Single Index Model

Suppose we have predictor variables $x$ and response variables $y$, and we seek a transformation $g$ and vector $w$ relating the two such that

$$\mathbb{E}[y|x] = g\left(x^T w\right) .$$

- Generalized Linear Model: $g$ is known, $y|x$ are RVs from an exponential family distribution parameterized by $w$.
    - Includes linear regression, log-linear regression, and logistic regression
- Single Index Model: Both $g$ and $w$ are unknown.

# Single Index Model Learning

We seek a transformation $g$ and vector $w$ such that

$$\mathbb{E}[y|x] = g\left(x^T w\right) .$$

**Theorem ([Kalai and Sastry, 2009], [Kakade et al., 2011])**

*Suppose* $(x_i, y_i) \in \mathbb{B}_n \times [0,1]$, $i = 1, \ldots, p$ *are draws from a distribution where* $\mathbb{E}[y|x] = g(x^T w)$ *for monotonic* $G$*-Lipschitz* $g$ *and* $\|w\| \leq 1$*. There is a* $poly(1/\epsilon, \log(1/\delta), n)$ *time algorithm that, given any* $\delta, \epsilon > 0$*, with probability* $\geq 1 - \delta$ *outputs* $h(x) = \hat{g}(\hat{w}^T x)$ *with*

$$err(h) = \mathbb{E}_{y|x}[(g(x^T w) - h(x))^2] < \epsilon$$

# Single Index Model Learning

**Algorithm 1** Lipshitz-Isotron Algorithm [Kakade et al., 2011]

Given $T > 0$, $(x_i, y_i)_{i=1}^p$;

Set $w^{(1)} := 1$;

**for** $t = 1, 2, \ldots, T$ **do**

Update $g$ using Lipschitz-PAV: $g^{(t)} = LPAV\left((x_i^T w^{(t)}, y_i)_{i=1}^p\right)$.

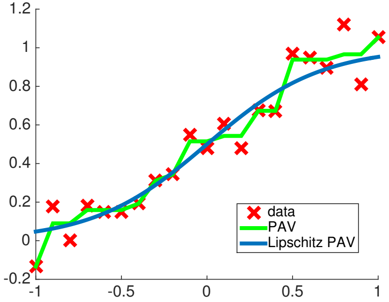
Update $w$ using gradient descent:

$$w^{(t+1)} = w^{(t)} + \frac{1}{p}\sum_{i=1}^{p}\left(y_i - g^{(t)}(x_i^T w^{(t)})\right) x_i$$

**end for**

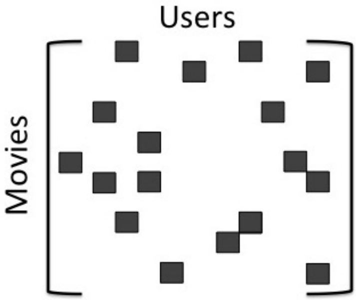
# Lipschitz Pool Adjacent Violator

- The Pool Adjacent Violator (PAV) algorithm pools points and averages to minimize mean squared error $g(x_i) - y_i$. PAV
- L-PAV adds the additional constraint of a given Lipschitz constant.



# Low-rank Matrix Completion

We have an $n \times m$, rank $r$ matrix $X$. However, we only observe a subset of the entries, $\Omega \subset \{1, \dots, n\} \times \{1, \dots, m\}$.



# Low-rank Matrix Completion

We have an $n \times m$, rank $r$ matrix $X$. However, we only observe a subset of the entries, $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, m\}$.

We may find a solution by solving the following NP-hard optimization:

$$\begin{aligned} \underset{M}{\text{minimize}}\ & \text{rank}(M) \\ \text{subject to}\ & M_\Omega = X_\Omega \end{aligned}$$

# Low-rank Matrix Completion

We have an $n \times m$, rank $r$ matrix $X$. However, we only observe a subset of the entries, $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, m\}$.

Or we may solve this convex problem:

$$\underset{M}{\text{minimize}} \; \|M\|_* = \sum_{i=1}^{n} \sigma_i(M)$$
$$\text{subject to } M_\Omega = X_\Omega$$

Exact recovery guarantees: $X$ is exactly low-rank and incoherent.
MSE guarantees: $X$ is nearly low-rank with bounded $(r+1)^{th}$ singular value.

# Low-rank Matrix Completion Algorithms

There are a plethora of algorithms to solve the nuclear norm problem or reformulations.

- LMaFit, APGL, FPCA
- Singular value thresholding: iterated SVD, SVT, FRSVT
- Grassmannian: OptSpace, GROUSE

# High-rank Matrices

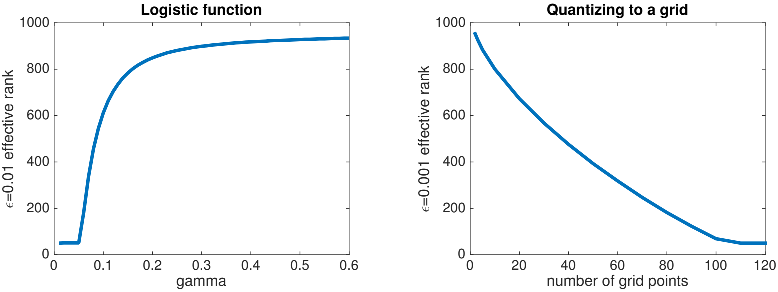
For $Z$ low-rank,

$Y_{ij} = g(Z_{ij}) = \frac{1}{1+\exp^{-\gamma Z_{ij}}}$, $Y$ has full rank.

$Y_{ij} = g(Z_{ij}) = \texttt{quantize\_to\_grid}(Z_{ij})$, $Y$ has full rank.

# High-rank Matrices: Effective rank

These matrices even have high effective rank.
For a rank-50, 1000x1000 matrix:




erank

# Problem Formulation

Our model is as follows:

- **Low-rank matrix** $Z^* \in \mathbb{R}^{n \times m}$ with $m \leq n$ and (for now, known) rank $r \ll m$.
- **Lipschitz link function** $g^* : \mathbb{R} \to \mathbb{R}$, monotonic, Lipschitz
- **Noise matrix** $N \in \mathbb{R}^{n \times m}$ with iid entries $\mathbb{E}[N] = 0$.
- **Samples of matrix entries** $\Omega \in \{1, \dots, n\} \times \{1, \dots, m\}$ is a multiset, sampled independently with replacement.

We observe $Y_{ij} = g^*(Z^*_{ij}) + N_{ij}$ for $(i, j) \in \Omega$

and we wish to recover $g^*$, $Z^*$.

# Optimization Formulation

$$\begin{aligned} \min_{g,Z} \quad & \sum_{\Omega} (g(Z_{i,j}) - Y_{i,j})^2 \\ \text{subj. to} \quad & g : \mathbb{R} \to \mathbb{R} \text{ is Lipschitz and monotone} \\ & \operatorname{rank}(Z) \leq r \end{aligned}$$

Non-convex in each variable, but we can alternate the standard approaches:

- Use gradient descent and projection onto the low-rank cone for $Z$.
- Use LPAV for $g$.

We call this algorithm MMC-LS.

# MMC-LS Algorithm

**Algorithm 2** MMC-LS

Given max iterations $T > 0$, step size $\eta > 0$, rank $r$, data $Y_\Omega$

Init $\hat{g}^{(0)}(z) = \frac{|\Omega|}{mn} z$, $\hat{Z}^{(0)} = \frac{mn}{|\Omega|} Y_0$, where $Y_0$ zero-filled $Y_\Omega$.

**for** $t = 1, 2, \ldots, T$ **do**

Update $\hat{Z}$ using gradient descent:

$$\hat{Z}_{i,j}^{(t)} = \hat{Z}_{i,j}^{(t-1)} - \eta \left( \hat{g}^{t-1} \left( \hat{Z}_{i,j}^{(t-1)} \right) - Y_{i,j} \right) (\hat{g}^{t-1})'(\hat{Z}_{i,j}^{(t-1)}) \mathbb{I}_{(i,j) \in \Omega}$$

Project: $\hat{Z}^{(t)} = \mathcal{P}_r(\hat{Z}^{(t)})$

Update $\hat{g}$: $\hat{g}^{(t)} = LPAV \left( \{ (\hat{Z}_{i,j}^{(t)}, Y_{i,j}) \text{ for } (i,j) \in \Omega \} \right)$.

**end for**

# Optimization of Calibrated Loss

Let $\Phi : \mathbb{R} \to \mathbb{R}$ be a differentiable function that satisfies $\Phi' = g^*$. Since $g^*$ is monotonic, $\Phi$ is convex. Consider:

$$L(\Phi, Z) = \sum_{(i,j)\in\Omega} \Phi(Z_{i,j}) - Y_{i,j} Z_{i,j}$$

Differentiating with respect to $Z$ we get that a minimizer satisfies $\sum_{(i,j)\in\Omega} g^*(Z_{i,j}) - Y_{i,j} = 0$; in other words, $Z^*$ is a minimizer in expectation. So $L(\Phi, Z)$ is a calibrated loss for our problem.

# MMC-c Algorithm

**Algorithm 3** MMC-calibrated

Given max iterations $T > 0$, step size $\eta > 0$, rank $r$, data $Y_\Omega$

Init $\hat{g}^{(0)}(z) = \frac{|\Omega|}{mn} z$, $\hat{Z}^{(0)} = \frac{mn}{|\Omega|} Y_0$, where $Y_0$ zero-filled $Y_\Omega$.

**for** $t = 1, 2, \ldots, T$ **do**

Update $\hat{Z}$ using gradient descent:

$$\hat{Z}_{i,j}^{(t)} = \hat{Z}_{i,j}^{(t-1)} - \eta \left( \hat{g}^{t-1} \left( \hat{Z}_{i,j}^{(t-1)} \right) - Y_{i,j} \right) \mathbb{I}_{(i,j) \in \Omega}$$

Project: $\hat{Z}^{(t)} = \mathcal{P}_r(\hat{Z}^{(t)})$

Update $g$: $g^{(t)} = LPAV \left( \{ (\hat{Z}_{i,j}^{(t)}, Y_{i,j}) \text{ for } (i,j) \in \Omega \} \right)$.

**end for**

# Remarks

MMC consists of three steps: gradient descent, projection, and LPAV.

- The gradient descent step requires a step size parameter $\eta$; we chose a small constant stepsize by cross validation.
- The projection requires rank $r$. For our implementation, we started with a small $r$ and increased it, in the same vein as [Wen et al., 2012].
- LPAV is the solution of a QP. Ravi developed an ADMM implementation as well.

# MSE Analysis of MMC-c

Let $\hat{M} = \hat{g}(\hat{Z})$ and $M^* = g^*(Z^*)$.
Define the MSE as

$$MSE(\hat{M}) = \mathbb{E}\left[\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\hat{M}_{i,j} - M^*_{i,j}\right)^2\right]$$

# MSE Analysis of MMC-c

**Theorem (MSE of MMC-c after one iteration [Ganti et al., 2015])**

*Let* $\|Z^*\| = O(\sqrt{n})$ *and* $\sigma_{r+1}(Y) = \tilde{O}(\sqrt{n})$ *with high probability. Let* $\alpha = \|M^* - Z^*\|$*. Furthermore, assume that elements of* $Z^*$ *and* $Y$ *are bounded in absolute value by 1.*
*Then the MSE of one step of MMC* $(T = 1)$ *is bounded by*

$$MSE(\hat{M}) \leq O\left(\sqrt{\frac{r}{m}} + \frac{mn}{|\Omega|^{3/2}} + \sqrt{\frac{r\alpha}{m\sqrt{n}}\left(1 + \frac{\alpha}{\sqrt{n}}\right)}\right) .$$

# MSE Analysis of MMC-c

**Theorem (MSE of MMC-c after one iteration [Ganti et al., 2015])**

*In addition to the previous assumptions, let*

$$\alpha = \|M^* - Z^*\| = O(\sqrt{n}) \; .$$

*Then the MSE of one step of MMC is bounded by*

$$MSE(\hat{M}) \leq O\left(\sqrt{\frac{r}{m}} + \frac{mn}{|\Omega|^{3/2}}\right) \; .$$

# Synthetic Data

$Z^*$ is $30 \times 20$ and rank 5.

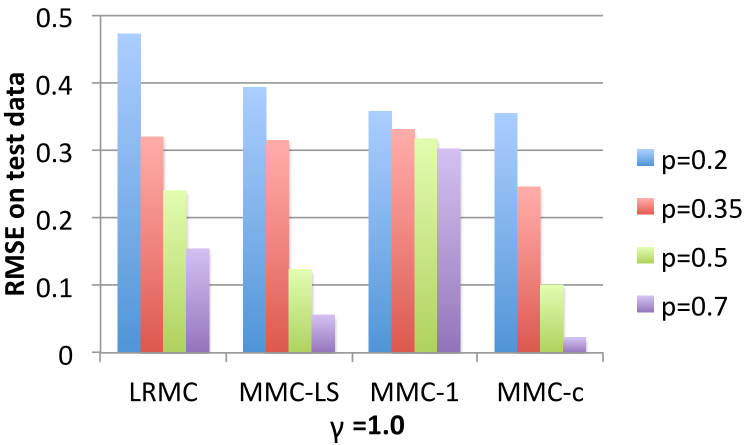
$N = 0$

Toy ISE calibration function: $g^*(z) = 1/(1 + \exp^{-\gamma z})$

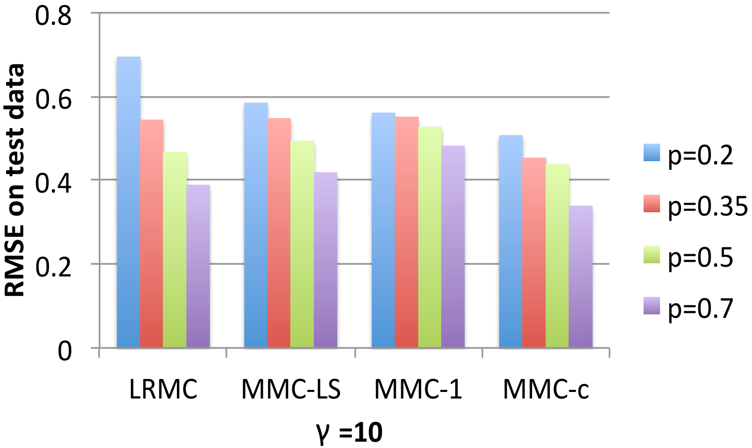
Vary $\gamma = 1, 10, 40$.

Vary probability of observation $p = .2, .35, .5, .7$.

# Synthetic Data

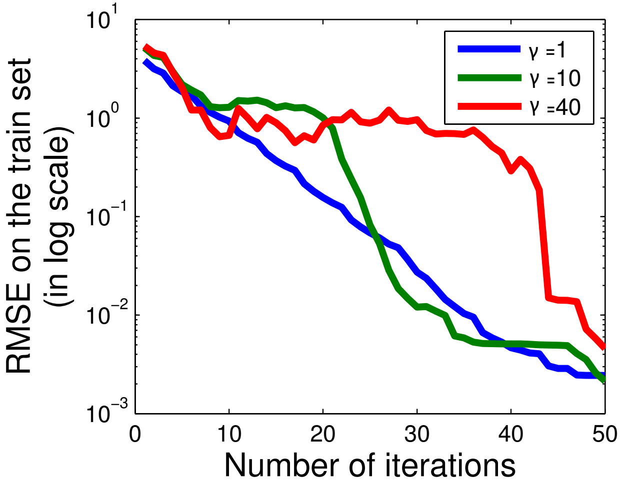


# Synthetic Data




$\gamma = 40$

# Synthetic Data



# Real Data

- Paper recommendation: 3426 features from 50 scholars' research profiles.
- Jester: 4.1 Million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users.
- Movie lens: 100,000 ratings from 1000 users on 1700 movies.
- Cameraman: Dictionary learning on patches of the image.

| Dataset | Dimension | $\lvert\Omega\rvert$ | $r_{0.01}(Y)$ |
|---|---|---|---|
| PaperReco | $3426 \times 50$ | 34294 (20%) | 47 |
| Jester-3 | $24938 \times 100$ | 124690 (5%) | 66 |
| ML-100k | $1682 \times 943$ | 64000 (4%) | 391 |
| Cameraman | $1536 \times 512$ | 157016 (20%) | 393 |

# Real Data Performance

RMSE on a held-out test set:

| Dataset | $\|\Omega\|/mn$ | LMaFit-A | MMC-c $T = 1$ | MMC-c |
|---|---|---|---|---|
| PaperReco | 20% | 0.4026 | 0.4247 | **0.2965** |
| Jester-3 | 5% | 6.8728 | 5.327 | **5.2348** |
| ML-100k | 4% | 3.3101 | 1.388 | **1.1533** |
| Cameraman | 20% | 0.0754 | 0.1656 | **0.06885** |

# Conclusion

- Monotonicity of $g^*$ and low-rank structure on $Z^*$ are enough to allow joint estimation.
- A natural alternating minimization algorithm does well.
- Next steps:
  - Estimating different $g^*$ for different columns, *e.g.*, users or sensors.
  - Understanding when it is possible to recover relative differences or order information of entries of $Z^*$ instead of values of $M^* = g^*(Z^*)$.
  - Further algorithmic guarantees.

# Thank you! Questions?


Ganti, R. S., Balzano, L., and Willett, R. (2015).
Matrix completion under monotonic single index models.
In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 1864–1872. Curran Associates, Inc.

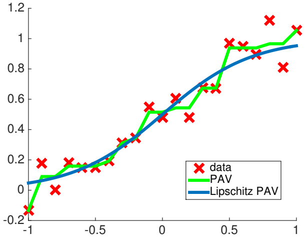
Kakade, S. M., Kanade, V., Shamir, O., and Kalai, A. (2011).
Efficient learning of generalized linear and single index models with isotonic regression.
In *Advances in Neural Information Processing Systems*, pages 927–935.

Kalai, A. T. and Sastry, R. (2009).
The isotron algorithm: High-dimensional isotonic regression.
In *COLT*.

Wen, Z., Yin, W., and Zhang, Y. (2012).
Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm.
*Mathematical Programming Computation*, 4(4):333–361.

- The Pool Adjacent Violator (PAV) algorithm pools points and averages to solve

$$\underset{\text{monotone } g}{\arg\min} \left( \frac{1}{p} \sum_{i=1}^{p} (g(x_i) - y_i)^2 \right) .$$




Back to LPAV.

## High-rank Matrices: Effective rank

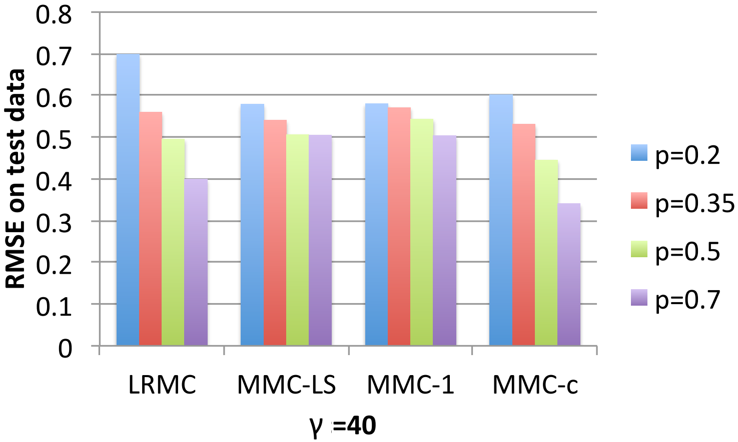
**Definition**

The **effective rank** of an $n \times m$ matrix $Y$, $m < n$, with singular values $\sigma_j$ is

$$r_\epsilon(Y) = \min \left\{ k \in \mathbb{N} : \sqrt{\frac{\sum_{j=k+1}^{m} \sigma_j^2}{\sum_{j=1}^{m} \sigma_j^2}} \leq \epsilon \right\} .$$

Back to Matrix Completion.

# Synthetic Data




Back to the experiments