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

Netflix Prize
Leaderboard

Mixture of hundreds of models, including gradient descent

Gradient descent on low-rank parameterization
Example 1: Recommender Systems

Motivation  |  Background  |  Problem Formulation  |  Our Algorithm  |  Experiments  |  Conclusion

Laura Balzano  
University of Michigan

Monotonic Low-Rank Matrix Completion
Example 2: Blind Sensor Calibration

Laura Balzano
University of Michigan

Monotonic Low-Rank Matrix Completion
Example 2: Blind Sensor Calibration

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

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

$$
\mathbb{E}[y|x] = g(x^Tw).
$$

- 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

$$\mathbb{E}[y|x] = g(x^Tw).$$

**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 $\mathbb{E}[y|x] = g(x^Tw)$ for monotonic $G$-Lipschitz $g$ and $\|w\| \leq 1$. There is a $\text{poly}(1/\epsilon, \log(1/\delta), n)$ time algorithm that, given any $\delta, \epsilon > 0$, with probability $\geq 1 - \delta$ outputs $h(x) = \hat{g}(\hat{w}^Tx)$ with

$$\text{err}(h) = \mathbb{E}_{y|x}[(g(x^Tw) - h(x))^2] < \epsilon$$
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: $g^{(t)} = LPAV ((x_i^T w^{(t)}, y_i)_{i=1}^p)$.
    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
The Pool Adjacent Violator (PAV) algorithm pools points and averages to minimize mean squared error $g(x_i) - y_i$.

L-PAV adds the additional constraint of a given Lipschitz constant.
We have an $n \times m$, rank $r$ matrix $X$. However, we only observe a subset of the entries, $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, m\}$.
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:

$$
\minimize_M \quad \text{rank}(M) \\
\text{subject to } M_\Omega = X_\Omega
$$
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:

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

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,

\[
Y_{ij} = g(Z_{ij}) = \frac{1}{1+\exp(-\gamma Z_{ij})}, \quad Y \text{ has full rank.}
\]

\[
Y_{ij} = g(Z_{ij}) = \text{quantize to grid}(Z_{ij}), \quad Y \text{ has full rank.}
\]
These matrices even have high effective rank. For a rank-50, 1000x1000 matrix:

- Logistic function
- Quantizing to a grid
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** $g^* : \mathbb{R} \rightarrow \mathbb{R}$, monotonic, Lipschitz
- **Noise matrix** $N \in \mathbb{R}^{n \times m}$ with iid entries $\mathbb{E}[N] = 0$.
- **Samples of matrix entries** $\Omega \in \{1, \ldots, n\} \times \{1, \ldots, m\}$ 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

\[
\min_{g, Z} \sum_{\Omega} (g(Z_{i,j}) - Y_{i,j})^2 \\
\text{subj. to } g : \mathbb{R} \rightarrow \mathbb{R} \text{ is Lipschitz and monotone} \\
\text{rank}(Z) \leq r
\]

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.
**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:

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

    Project: $\hat{Z}^{(t)} = P_r(\hat{Z}^{(t)})$

    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 \( \Phi : \mathbb{R} \rightarrow \mathbb{R} \) 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.
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
\hspace{1em} Update $\hat{Z}$ using gradient descent:

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

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
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

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

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

\[ \text{MSE}(\hat{M}) \leq O \left( \sqrt{\frac{r}{m}} + \frac{mn}{|\Omega|^{3/2}} \right) . \]
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 $p = 0.2, 0.35, 0.5, 0.7$. 
Synthetic Data

![Bar Chart]

- **LRMC**
- **MMC-LS**
- **MMC-1**
- **MMC-c**

RMSE on test data for different values of $\gamma$ and $p$.

- $\gamma = 1.0$
- $p = 0.2$
- $p = 0.35$
- $p = 0.5$
- $p = 0.7$
Synthetic Data

![Graph showing RMSE on test data for different methods and parameter values.]

- **Motivation**
- **Background**
- **Problem Formulation**
- **Our Algorithm**
- **Experiments**
- **Conclusion**

Laura Balzano
University of Michigan

Monotonic Low-Rank Matrix Completion
Synthetic Data

This graph illustrates the RMSE on the train set as a function of the number of iterations for different values of the parameter $\gamma$. The RMSE is plotted on a log scale. The lines represent different values of $\gamma$: $\gamma = 1$, $\gamma = 10$, and $\gamma = 40$. As the number of iterations increases, the RMSE decreases, indicating improved performance of the algorithm. The plot shows that higher values of $\gamma$ may lead to a faster convergence rate, but further analysis is needed to determine the optimal value.
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 | $|\Omega|$         | $r_{0.01}(Y)$ |
|---------------|-----------|-------------------|---------------|
| PaperReco     | 3426 × 50 | 34294 (20%)       | 47            |
| Jester-3      | 24938 × 100 | 124690 (5%)       | 66            |
| ML-100k       | 1682 × 943 | 64000 (4%)        | 391           |
| Cameraman     | 1536 × 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 |
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?

Matrix completion under monotonic single index models.

Efficient learning of generalized linear and single index models with isotonic regression.

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

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

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

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

![Graph showing RMSE on test data for different methods and parameter values.](image)

- **LRMC**, **MMC-LS**, **MMC-1**, **MMC-c**
- Parameters: $c=40$, $p=0.2$, $p=0.35$, $p=0.5$, $p=0.7$

Laura Balzano
University of Michigan
Monotonic Low-Rank Matrix Completion