Matrix Completion Under Monotonic Single Index Models

Ravi Ganti1, Laura Balzano2, Rebecca Willett3
1Wisconsin Institutes for Discovery, University of Wisconsin-Madison
2Department of Electrical Engineering and Computer Science, University of Michigan at Ann Arbor
3Department of Electrical and Computer Engineering, University of Wisconsin-Madison

Introduction

Most recent results in matrix completion assume that the matrix under consideration is low-rank or that the columns are in a union of low-rank subspaces. In real-world settings, however, the linear structure underlying these models is distorted by a (typically unknown) nonlinear transformation. We address the challenge of matrix completion in the face of such nonlinearities. Given a few observations of a matrix that are obtained by applying a Lipschitz, monotonic function to a low-rank matrix, our task is to estimate the remaining unobserved entries. We propose a novel matrix completion method that alternates between low-rank matrix estimation and monotonic function estimation to estimate the missing matrix elements. Mean squared error bounds provide insight into how well the matrix can be estimated based on the size, rank of the matrix and properties of the nonlinear transformation.

Single index model for matrix completion

- There is some unknown matrix $Z \in \mathbb{R}^{n \times m}$ with $m \leq n$ and of rank $r \ll m$.
- An unknown, non-linear, monotonic, $L$-Lipschitz function $g^*$ is applied to each element of the matrix $Z$ to get another matrix $M^*$.
  \[ M^*_{ij} = g^*(Z_{ij}), \quad \forall i \in [n], j \in [m] \]
- Task: Use the observations of $X$ on $\Omega$ to predict the remaining elements, assuming that the elements are missing at random.
- Applications: Recommender systems, random dot product graphs.

Non-linear link functions and rank of the matrix

Given an $\epsilon \in (0, 1)$, define the effective rank of $X$ as follows:
\[ r_\epsilon(X) = \min \left\{ k \in \mathbb{N} : \sqrt{\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2}} \leq \epsilon \right\}. \]

![Figure: The plot shows the $r_\epsilon(X)$ defined in equation (1) obtained by applying a non-linear function $g^*$ to each element of $Z$, where $g^*(z) = \min(0, z)$. $Z$ is a 30 × 20 matrix of rank 5.](image)

Non-linearities can destroy the low-rank information in the latent matrix. We propose two algorithms to simultaneously learn both the low-rank matrix $Z^*$, and the link function $g^*$.

A least squares based algorithm

An natural approach to the monotonic matrix completion problem is to learn $g^*$, $Z^*$ via squared loss minimization. In order to do this we need to solve the following optimization problem:
\[ \min g \in \mathbb{R}^n \sum \{ g(Z_{ij}) - X_{ij} \}^2 \quad : \quad g : \mathbb{R} \rightarrow \mathbb{R} \text{ is } L\text{-Lipschitz and monotonic} \]

- Problem is non-convex in each parameter. We will try to solve it via alternating minimization.
- Gradient updates for $Z$:
  \[ Z_{ij} \leftarrow Z_{ij}^{(t-1)} - \eta (g^{-1}(Z_{ij}^{(t-1)}) - X_{ij}) (g^{-1})'(Z_{ij}^{(t-1)}) \quad \forall (i,j) \in \Omega \]
  \[ Z_{ij}^{(t)} \leftarrow P_1(Z_{ij}^{(t)}) \]
  where $P_1$ is a projection on the rank $r$ cone.
- Noise in the estimates of $g^{-1}$ will also lead to noise in the estimates of the derivative $(g^{-1})'$. Hence, we would expect MMC–LS to be less accurate than a learning algorithm that does not have to estimate $(g^{-1})'$.
- Update for $g$:
  Learning 1-d monotonic, Lipschitz functions can be cast as a simple quadractic program [1] (called LPAV) that minimizes squared loss subject to linear constraints corresponding to monotonicity and Lipschitzness.

Algorithm based on calibrated loss

Consider the following optimization problem:
\[ \min \Phi, Z \mathcal{L}(\Phi, Z, \Omega) = \min \Phi, Z \sum_{(i,j) \in \Omega} \Phi(Z_{ij}) - X_{ij}Z_{ij} \]
where $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ is a convex function, with $\Phi' = g$ and $Z \in \mathbb{R}^{n \times m}$ is a low-rank matrix.

- If $g^*$ is known then we can fix $\Phi$ to $\Phi^*$ in the above optimization problem. In that case, the minimizer of the above optimization problem is the MMC model.
- Since $g^*$ is unknown, we shall minimize simultaneously over both $\Phi$ and $Z$.

Updates based on calibrated loss

- The main difference from an approach based on least squares is that the updates for $Z$ do not involve gradients of the link function. In round $t$, gradient update step for $Z$ looks as follows:
  \[ Z_{ij}^{(t)} \leftarrow Z_{ij}^{(t-1)} - \eta (g^{-1}(Z_{ij}^{(t-1)}) - X_{ij}) (g^{-1})'(Z_{ij}^{(t-1)}) \quad \forall (i,j) \in \Omega \]
  \[ Z_{ij}^{(t)} \leftarrow P_1(Z_{ij}^{(t)}) \]
- The update for $\Phi$ (equiv $g$) uses the LPAV algorithm.

Mean squared error guarantees

Let $||Z^*|| = O(\sqrt{n})$ and $\sigma_{i+1}(X) = O(\sqrt{n})$ with high probability. Furthermore, let $E|\|N\|_2 = O(\sqrt{n})$, $E|\|N\|_2^2 = O(n)$. Let $||M^* - Z^*|| = O(\sqrt{n})$. Then, assuming that the elements of $Z^*$ and $X$ are bounded in absolute value by 1, the MSE of the estimator output by MMC with $T = 1$ is given by
\[ MSE(M) = O \left( \frac{T}{m} + \frac{\min \{ 1, 1 + n \} }{ ||Z^*||^2 } \right) \]

Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimensions</th>
<th>Cameraman</th>
<th>PaperReco</th>
<th>Jester-3</th>
<th>LRs1-10k</th>
<th>UCI-10k</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCI</td>
<td>34224 × 50</td>
<td>32494</td>
<td>4026</td>
<td>4247</td>
<td>0.2965</td>
<td>0.2536</td>
</tr>
<tr>
<td>Jester</td>
<td>24958 × 100</td>
<td>23494</td>
<td>4346</td>
<td>3728</td>
<td>0.2536</td>
<td>0.2965</td>
</tr>
<tr>
<td>LRs-10k</td>
<td>1682 × 943</td>
<td>60000</td>
<td>43101</td>
<td>4388</td>
<td>1.1543</td>
<td>1.1543</td>
</tr>
<tr>
<td>UCI</td>
<td>1536 × 512</td>
<td>157016</td>
<td>10754</td>
<td>10656</td>
<td>0.06885</td>
<td>0.06885</td>
</tr>
</tbody>
</table>

References