Dynamic Matrix Recovery
Journal of the American Statistical Association (2024)

Presenter: Ben-Zheng Li

June 2025

Outline

(Static) Matrix Recovery

- Goal: recover a low-rank matrix M_0 from sparse and noisy observations.
- Each observation Y is a linear measurement of M_0:

$$Y = \langle X, M_0 \rangle + \xi = \mathrm{Tr}(X^\top M_0) + \xi \tag{1}$$

- Notation:
  - X: measurement matrix of size $m_1 \times m_2$
  - M_0: the unknown static low-rank matrix
  - $\langle X, M_0 \rangle$: matrix inner product (linear measurement)
  - ξ: additive noise (zero-mean)

(Static) Matrix Recovery

- Goal: recover a low-rank matrix M_0 from sparse and noisy observations.
- Each observation Y is a linear measurement of M_0:

$$Y = \langle X, M_0 \rangle + \xi = \mathrm{Tr}(X^\top M_0) + \xi \tag{1}$$

- Notation:
  - X: measurement matrix of size $m_1 \times m_2$
  - M_0: the unknown static low-rank matrix
  - $\langle X, M_0 \rangle$: matrix inner product (linear measurement)
  - ξ: additive noise (zero-mean)
- Two examples:
  - When X is a 0-1 sampling matrix, (1) is a matrix completion problem.
  - When X is a rand Gaussian matrix, (1) is a compressed sensing problem.

Motivation

- The static model assumes the low-rank matrix M_0 is fixed across all time. But in many real-world applications, the underlying structure evolves over time.
- To model such time-varying structure, we introduce a dynamic formulation:

$$Y_t = \langle X_t, M_t^0 \rangle + \xi_t \quad (t = 1, \ldots, T) \tag{2}$$

- Goal: recover a sequence of matrices $\{M_t^0\}_{t=1}^T$ that are low-rank and vary smoothly with time.

Motivation

- The static model assumes the low-rank matrix M_0 is fixed across all time. But in many real-world applications, the underlying structure evolves over time.
- To model such time-varying structure, we introduce a dynamic formulation:

$$Y_t = \langle X_t, M_t^0 \rangle + \xi_t \quad (t = 1, \ldots, T) \tag{2}$$

- Goal: recover a sequence of matrices $\{M_t^0\}_{t=1}^T$ that are low-rank and vary smoothly with time.

Remark

Although the dynamic matrix sequence $\{M_t^0\}_{t=1}^T$ can be viewed as a third-order tensor $\mathcal{M}^0 \in \mathbb{R}^{m_1 \times m_2 \times T}$, this paper chooses not to use global tensor regularization. Instead, it applies local smoothing across time, which is more flexible and adaptive for temporal variation.

Prior Work: Static Nuclear Norm Estimation

- **Goal:** recover a sequence of low-rank matrices $\{M_t^0\}_{t=1}^T$
- Koltchinskii et al. (2011):

$$\widehat{M}_t^\lambda = \arg\min_{M \in \mathbb{M}} \frac{1}{n_t} \sum_{i=1}^{n_t} (Y_{ti} - \langle X_{ti}, M \rangle)^2 + \lambda \|M\|_*, \;\; t = 1, \cdots, T$$

- $\mathbb{M}$: convex set of matrices (bounded Frobenius norm)
- $\|M\|_*$: nuclear norm

Prior Work: Static Nuclear Norm Estimation

- **Goal:** recover a sequence of low-rank matrices $\{M_t^0\}_{t=1}^T$
- Koltchinskii et al. (2011):

$$\widehat{M}_t^\lambda = \arg\min_{M \in \mathbb{M}} \frac{1}{n_t} \sum_{i=1}^{n_t} \left(Y_{ti} - \langle X_{ti}, M \rangle\right)^2 + \lambda \|M\|_*, \;\; t = 1, \cdots, T$$

- $\mathbb{M}$: convex set of matrices (bounded Frobenius norm)
- $\|M\|_*$: nuclear norm

Error bound:

$$(m_1 m_2)^{-1/2} \|\widehat{M}_t^\lambda - M_t^0\|_F \leq C \left\{ \frac{\log(m_1 + m_2) \cdot \max(m_1, m_2) r_t}{n_t} \right\}^{1/2}$$

Koltchinskii, V., Lounici, K., Tsybakov, A. B. (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion, The Annals of Statistics, 39(5), 2302–2329.

Kernel Function

Smoothing Estimator:

$$\widetilde{M}_t^{\lambda} = \arg\min_{M} \sum_{j=1}^{T} \omega_h(j-t) \cdot \mathsf{Loss}_j(M) + \lambda \|M\|_*, \ \ t = 1, \cdots, T$$

- Weight $\omega_h(j-t) = \frac{1}{h}K\left(\frac{j-t}{h}\right)$, where $K(\cdot)$ is **kernel function**, and h is **bandwidth**.

Kernel Assumptions:

- (中心对称性) Symmetric: $K(x) = K(-x)$, defined on $[-1, 1]$
- (归一化) Integrates to 1: $\int K(x)dx = 1$
- (二阶矩有界) Finite second moment: $\alpha(K) = \int x^2 K(x)dx < \infty$
- (平方积分有界) Finite squared norm: $R(K) = \int K^2(x)dx < \infty$

Temporally Weighted Low-Rank Estimation

- **Model:** Estimate M_t^0 via temporally weighted nuclear-norm penalized regression:

$$\widetilde{M}_t^\lambda = \arg\min_M \underbrace{\sum_{j=1}^{T} \omega_h(j-t) \left[\frac{1}{n_j} \sum_{i=1}^{n_j} (Y_{ji} - \langle X_{ji}, M \rangle)^2 \right]}_{\text{(核化动态保真项)}} + \lambda \|M\|_* \quad (3)$$

- **Notation:**
  - Y_{ji}: Observation i at time j
  - X_{ji}: Corresponding measurement matrix
  - $\omega_h(j-t)$: Kernel weight centered at t (bandwidth h)
  - n_j: Number of samples at time j
  - $\|M\|_*$: Nuclear norm regularization

Optimization Algorithm

FISTA-style Dynamic Estimation Algorithm

Input: $\{X_{ji}, Y_{ji}\}_{j=1}^{T}$, bandwidth h, steps K, tolerance $\texttt{tor}$
for $t = 1$ to T **do**
  Initialize $M_t^{(-1)} \leftarrow M_{t-1},\ M_t^{(0)} \leftarrow M_{t-1},\quad s_0 = 1,\ s_{-1} = 1$
  for $k = 0$ to K **do**
    $N_t^{(k)} \leftarrow M_t^{(k)} + \frac{s_{k-1}-1}{s_k}(M_t^{(k)} - M_t^{(k-1)})$
    Compute weighted gradient $\nabla F_t(N_t^{(k)})$
    $G_t^{(k)} \leftarrow N_t^{(k)} - \frac{1}{L_f}\nabla F_t(N_t^{(k)})$
    $[U, D, V] \leftarrow \text{svd}(G_t^{(k)})$
    $M_t^{(k+1)} \leftarrow U \cdot (D - \frac{2\lambda}{L_f})_+ \cdot V^\top$ (SVT step)
    $s_{k+1} \leftarrow \frac{1+\sqrt{1+4s_k^2}}{2}$
    if $|F_t(M_t^{(k+1)}) - F_t(M_t^{(k)})| \leq \texttt{tor}$ **then**
      break
    end if
  end for
  $M_t \leftarrow M_t^{(k+1)}$
end for
Return: $\{M_t\}_{t=1}^{T}$

Main Theoretical Results

$$\mathbb{E}\left[\|\widehat{M}_t^{\lambda} - M_t^0\|_F^2\right] \leq \inf_{M}\left\{\|M - M_t^0\|_F^2 + \lambda\|M\|_*\right\} + C \cdot \frac{\log(m_1 + m_2)}{n_t} \quad \text{(Thm 1)}$$

$$\|\widehat{M}_t^{\lambda} - M_t^0\|_F^2 \leq \inf_{M}\left\{\|M - M_t^0\|_F^2 + \lambda\|M\|_*\right\} + C \cdot \frac{\log(m_1 + m_2)}{n_t},$$

with probability at least $1 - \delta$ (Thm 2)

$$\|\widehat{M}_t^{\lambda} - M_t^0\|_F \leq C \cdot \sqrt{\frac{\log(m_1 + m_2) \cdot \max(m_1, m_2) \cdot r_t}{n_t}}$$

with probability at least $1 - \delta$ (Thm 3)

$$\max_{1 \leq t \leq T}\|\widehat{M}_t^{\lambda} - M_t^0\|_F \leq C \cdot \sqrt{\frac{\log(m_1 + m_2) \cdot \max(m_1, m_2) \cdot r}{n_{\min}}}$$

with probability at least $1 - \delta$ (Thm 4)

Synthetic Experiments: Recovery Accuracy

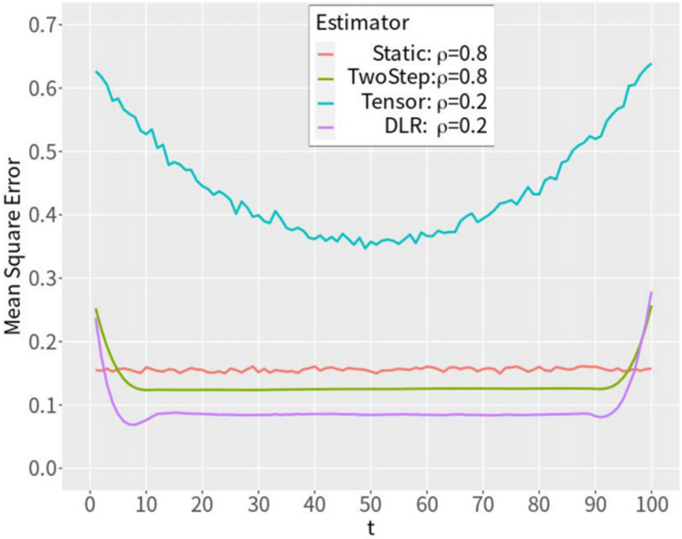
- **Used Kernel: Epanechnikov Kernel**

$$K(x) = \frac{3}{4}(1 - x^2), \quad |x| \leq 1$$

- Setup:
  - Generate dynamic low-rank matrices $M_t^0 = A_t B_t^\top$ with $\mathrm{rank} = r$
  - Observations Y_{ti} generated by partial noisy linear measurements: $Y_{ti} = \langle X_{ti}, M_t^0 \rangle + \xi_{ti}$
- Compare proposed method with the following baselines:
  - **Static:** Classical nuclear-norm estimator $\widehat{M}_t^\lambda$, applied independently at each time t
  - **TwoStep:** Perform static estimation first, then apply temporal kernel smoothing over $\widehat{M}_t^\lambda$
  - **Tensor:** Tensor completion model using the sum of nuclear norms (SNN):

$$\widehat{N}^\lambda = \arg \min_{N \in \mathbb{R}^{T \times m_1 \times m_2}} \|\mathcal{P}_\Omega(N) - Y\|_2^2 + \lambda \sum_{i=1}^{3} \|N_{(i)}\|_*$$

Synthetic Experiments: Recovery Accuracy

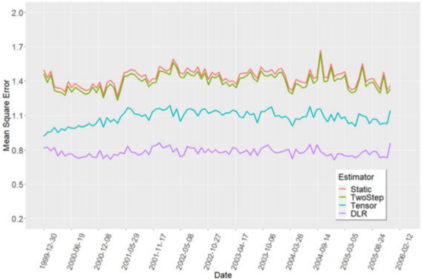


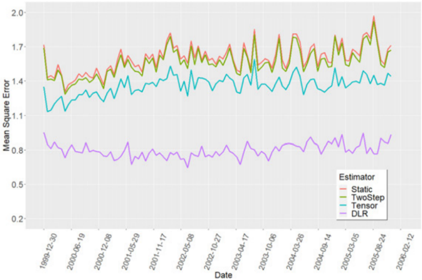
Real-World Experiment: Recommendation System

Dataset: Netflix Movie Ratings (https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data)

- **Task:** Recover user-movie ratings dynamically from sparse observations.
- **Dataset Description:**
  - Netflix dataset: 3036 users, 1034 movies, total ratings ~ 2.34M.
  - Time span divided into $T = 100$ intervals.
- **Methods Compared:**
  - Proposed (Dynamic Low-Rank)
  - Static
  - Two-Step
  - Tensor Completion (sum of nuclear norms)
- **Performance Metric:** Test Mean Squared Error (MSE)

Real-World Experiment: Recommendation System





Conclusion: Proposed method significantly outperforms benchmarks in predicting dynamic user ratings.

Real-World Experiment: Video Data

Dataset: DAVIS 2017 Video Dataset
(https://davischallenge.org/davis2017/code.html#unsupervised)

- **Task:** Recover dynamic low-rank video from compressed video frames.
- **Dataset Description:**
  - Video resolution: 480×854 pixels per frame.
  - Compression rate: Observations sampled at $\sim 15\%$ per frame.
- **Methods Compared:**
  - Proposed (Dynamic Low-Rank)
  - Static Matrix Completion
  - Two-Step Smoothing

Real-World Experiment: Video Data

Figure 6. The top five pictures are the 5th, 25th, 45th, 65th, and 85th original frames in the lions video, the second rows are the corresponding estimates of the proposed DLR method and the third and fourth rows are the benchmarks Static and TwoStep, respectively.

Future works

- Adaptive weights (e.g., learn an one-dimensional kernel)?
- Incorporate autoregressive or recurrent temporal structures (e.g., VAR, RNN) to better model temporal dependency.
- slice-wise nuclear norms VS tensor nuclear norm?