MC-Kit Manual

Stephen Tierney

April 5, 2014

1 Introduction

The problem of Matrix Completion (MC) is to recover a matrix $\mathbf{A}$ from only a small sample of its entries. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be the matrix we would like to know as precisely as possible while only observing a subset of its entries $(i, j) \in \Omega$. It is assumed that observed entries in Ω are uniform randomly sampled. Low-Rank Matrix Completion is a variant that assumes that $\mathbf{A}$ is low-rank. The tolerance for "low" is dependant upon the size of $\mathbf{A}$ and the number of sampled entries.

For further explanation let us define the sampling operator $\mathcal{P}_\Omega : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ as

$$[\mathcal{P}_\Omega(X)]_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We also define the opposite operator $\mathcal{P}_{\bar{\Omega}}$, which keeps those outside Ω unchanged and sets values inside Ω (i.e. $\bar{\Omega}$) to 0. Since we assume that the matrix to recover is low-rank, one could recover the unknown matrix by solving

$$\begin{aligned} &\min_{\mathbf{A}} \ \text{rank}(\mathbf{A}) \\ &\text{s.t. } \mathcal{P}_\Omega(\mathbf{A}) = \mathcal{P}_\Omega(\mathbf{M}) \end{aligned} \tag{2}$$

where matrix $\mathbf{M}$ is the partially observed matrix. In practice this problem is intractable therefore we use the closest convex relaxation i.e. the nuclear norm

$$\begin{aligned} &\min_{\mathbf{A}} \ \tau \|\mathbf{A}\|_* \\ &\text{s.t. } \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) \end{aligned} \tag{3}$$

We also consider the case where our observed entries may contain a limited amount of noise. Our corresponding objective is the following

$$\begin{aligned} &\min_{\mathbf{A}} \ \tau \|\mathbf{A}\|_* + \frac{\lambda}{2} \| \mathcal{P}_\Omega(\mathbf{E}) \|_F^2 \\ &\text{s.t. } \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) + \mathcal{P}_\Omega(\mathbf{E}) \end{aligned} \tag{4}$$

2 Singular Value Shrinkage Operator

Central to this work is the **singular value shrinkage operator**. Consider the singular value decomposition (SVD) of a matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ with rank r

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}, \quad \mathbf{\Sigma} = \text{diag}(\{\sigma_i\}_{i=1}^{r}). \tag{5}$$

Then we define the singular value shrinkage operator for any $\tau \geq 0$ as

$$\mathcal{D}_\tau(\mathbf{Y}) = \mathbf{U}S_\tau(\mathbf{\Sigma})\mathbf{V}^T, \quad S_\rho(\mathbf{\Sigma}) = \text{diag}(\{\max(\sigma_i - \tau, 0)\}). \tag{6}$$

It has been shown [1] that the operator $\mathcal{D}_\tau(\mathbf{Y})$ is the solution to the proximal nuclear norm problem i.e.

$$\mathcal{D}_\tau(\mathbf{Y}) = \underset{\mathbf{X}}{\text{argmin}} \ \tau\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 \tag{7}$$

The implementation of the singular value singular shrinkage operator is implemented by the function $[\mathbf{X}, \mathbf{s}] = \text{nn\_prox}(\mathbf{Y}, \tau)$, where $\mathbf{s}$ is the vector of the singular values of $\mathbf{X}$.

3 Function Listing

Problem	Implementation	Section
$\min_{\mathbf{A}} \ \tau\|\mathbf{A}\|_* + \frac{1}{2}\|\mathbf{A}\|_F^2$ s.t. $\mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A})$	Singular Value Thresholding (SVT)	4.1
$\min_{\mathbf{A}} \ \tau\|\mathbf{A}\|_*$ s.t. $\mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A})$	Inexact ALM Linearised ALM	4.2 4.3
$\min_{\mathbf{A}} \ \tau\|\mathbf{A}\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega(\mathbf{E})\|_F^2$ s.t. $\mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) + \mathcal{P}_\Omega(\mathbf{E})$	Linearised ALM	5.1

4 Noise-Free Implementations

4.1 SVT

The function

$$[\mathbf{A}, \mathbf{f\_values}, \mathbf{stop\_vals}] = \text{solve\_svt}(\mathbf{M}, \tau, \mu, iterations, tol) \tag{8}$$

solves the following

$$\begin{aligned} \min_{\mathbf{A}} \ & \tau\|\mathbf{A}\|_* + \frac{1}{2}\|\mathbf{A}\|_F^2 \\ \text{s.t.} \ & \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) \end{aligned} \tag{9}$$

as proposed by the authors of [1].

- $\mathbf{M}$ - matrix with observed entries and zeros for all others
- τ - regularisation (optional)
- μ - step size (optional)
- *iterations* - maximum number of iterations (optional)
- *tol* - stopping criteria tolerance (optional)

4.2 Inexact ALM

The function

$$[\mathbf{A}, \mathbf{f\_vals}, \mathbf{stop\_vals}] = \text{solve\_ialm}(\mathbf{M}, \tau, \mu, iterations, tol) \tag{10}$$

solves the following

$$\begin{aligned} \min_{\mathbf{A}} \ & \tau\|\mathbf{A}\|_* \\ \text{s.t.} \ & \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) \end{aligned} \tag{11}$$

as proposed by the authors of [3].

- $\mathbf{M}$ - matrix with observed entries and zeros for all others
- τ - regularisation (optional)
- μ - step size (optional)
- *iterations* - maximum number of iterations (optional)
- *tol* - stopping criteria tolerance (optional)

4.3 Linearised ALM

The function

$$[\mathbf{A}, \mathbf{f\_vals}, \mathbf{stop\_vals}] = \text{solve\_lin\_alm}(\mathbf{M}, \tau, \mu, \rho, iterations, tol) \tag{12}$$

solves the following

$$\begin{aligned} &\min_{\mathbf{A}} \ \tau\|\mathbf{A}\|_* \\ &\text{s.t. } \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) \end{aligned} \tag{13}$$

- $\mathbf{M}$ - matrix with observed entries and zeros for all others
- τ - regularisation (optional)
- μ - step size (optional)
- ρ - linearisation step size (optional)
- *iterations* - maximum number of iterations (optional)
- *tol* - stopping criteria tolerance (optional)

5 Noise Inclusive Implemetations

5.1 Linearised ALM

The functions

$$[\mathbf{A}, \mathbf{f\_vals}, \mathbf{stop\_vals}] = \text{solve\_e\_lin\_alm}(\mathbf{M}, \tau, \lambda, \rho, iterations, tol) \quad (14)$$

$$[\mathbf{A}, \mathbf{f\_vals}, \mathbf{stop\_vals}] = \text{solve\_e\_lin\_extalm}(\mathbf{M}, \tau, \lambda, \rho, iterations, tol) \quad (15)$$

$$[\mathbf{A}, \mathbf{f\_vals}, \mathbf{stop\_vals}] = \text{solve\_e\_lin\_accalm}(\mathbf{M}, \tau, \lambda, \rho, iterations, tol) \quad (16)$$

solve the following

$$\begin{aligned} \min_{\mathbf{A}} \ & \tau\|\mathbf{A}\|_* + \frac{\lambda}{2}\| \mathcal{P}_\Omega(\mathbf{E})\|_F^2 \\ \text{s.t. } & \mathcal{P}_\Omega(\mathbf{M}) = \mathcal{P}_\Omega(\mathbf{A}) + \mathcal{P}_\Omega(\mathbf{E}) \end{aligned} \quad (17)$$

with increasing convergence speed based on [2].

- $\mathbf{M}$ - matrix with observed entries and zeros for all others
- τ - regularisation (optional)
- λ - regularisation (optional)
- ρ - linearisation step size (optional)
- *iterations* - maximum number of iterations (optional)
- *tol* - stopping criteria tolerance (optional)

References

[1] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[2] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 457–464. ACM, 2009.

[3] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.