Image Restoration by Matrix Completion Using Schatten Capped p-Norm

K Kamalesh Kumar
Akash Njattuvetty

Indian Institute Of Technology, Madras

May 2022

What is Matrix Completion ?

- Matrix Completion is the task of recovering a matrix with only partially observed entries.

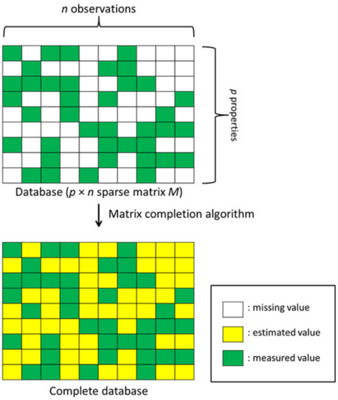

Figure 1: Matrix Completion

Figure 2: Restoring damaged images

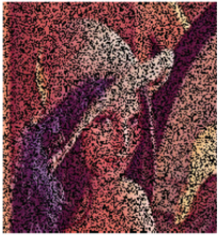

Figure 3: Restoring corrupted images

Low Rank Matrix Completion Formulation

- Let $D \in \mathbb{R}^{m \times n}$ denote the incomplete matrix with partially observed entries. $X \in \mathbb{R}^{m \times n}$ is the matrix to be restored,then

$$\min_{X} rank(X) \tag{1}$$

$$\text{s.t. } \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(D) \text{ (equivalently } X_\Omega = D_\Omega)$$

- where Ω is the observed index set of D and $\mathcal{P}_\Omega$ is the observation operator.

$$[\mathcal{P}_\Omega(D)]_{ij} = \begin{cases} D_{ij}, & (i,j) \in \Omega \\ 0, & otherwise \end{cases}$$

- Rank minimisation is done since the underlying matrix takes a low-rank structure in most real-world applications.

Problem with the Rank and Existing Surrogates

- The rank minimisation formulation is NP-hard, due to the non-convexity and discontinuity of the rank function
- Thus many convex and non-convex surrogates have been given as relaxations.
- Existing matrix completion algorithms can be broadly classified into two categories:
  - Spectral Regularisation Based Methods
  - Matrix Factorisation Based Methods
- The former utilizes the singular values and their variations to recover the missing entries of a matrix.
- The latter decomposes the latent matrix into two or more small sized matrices. However the rank has to be known in advance.

Formulations of Some Existing Surrogates to the Rank

Nuclear Norm Minimisation	Truncated Nuclear Norm Regularisation	Schatten p-Norm	Low Rank Matrix Factorisation
$\|X\|_* = \sum_{i=1}^{min(m,n)} \sigma_i$	$\|X\|_r = \sum_{i=r+1}^{min(m,n)} \sigma_i$	$\|X\|_{S_p} = (\sum_{i=1}^{min(m,n)} \sigma_i^p)^{\frac{1}{p}}$	$X = AB^T$ $A \in \mathbb{R}^{m\times r}, B \in \mathbb{R}^{r\times n}$
$\min_X \|X\|_*$ $s.t.\ X_\Omega = D_\Omega$	$\min_X \|X\|_r$ $s.t.\ X_\Omega = D_\Omega$	$\min_X \|X\|_{S_p}$ $s.t.\ X_\Omega = D_\Omega$	$\min_{A,B} \left\|\mathcal{P}_\Omega(AB^T - D)\right\|_F$ unconstrained
Convex surrogate	Non-convex surrogate	Non-convex surrogate	Non-convex surrogate

- Drawbacks of the existing methods and challenges posed:
  - Least rank solution is sub-optimal.
  - Thus a balance between the rank and the nuclear norm is desired.
  - Dominant singular values should be preserved, smaller singular values should be punished.
  - Matrix factorisation methods needs rank to be known a priori.

The Schatten Capped p-Norm Based Matrix Completion

- To strike a balance between the rank and the nuclear norm, the Schatten Capped p-Norm (SC_p) is formulated as:

$$\|X\|_{SC_{p,\tau}} = \left(\sum_{i=1}^{min(m,n)} min(\sigma_i, \tau)^p \right)^{\frac{1}{p}} \tag{2}$$

- where $\tau \geq 0$ and $p \in (0, 1]$
- for optimisation purposes, (2) is reformulated as:

$$\|X\|^p_{SC_{p,\tau}} = \sum_{i=1}^{min(m,n)} min(\sigma_i, \tau)^p \tag{3}$$

- Thus the SC_p norm can effectively preserve large singular values and filter out smaller singular values

The SC_p Matrix Completion Method

- The matrix completion problem is posed as:

$$\min_{X} \|X\|_{SC_{p,\tau}}^{p} \quad \text{s.t } X_\Omega = D_\Omega \tag{4}$$

- Transforming (4) into an unconstrained optimization problem,

$$\min_{X} \|X_\Omega - D_\Omega\|_F^2 + \gamma \|X\|_{SC_{p,\tau}}^{p} \tag{5}$$

- By introducing $E_\Omega = X_\Omega - D_\Omega$ and $W = X$, (5) can be rewritten as

$$\min_{X, E_\Omega, W} \|E_\Omega\|_F^2 + \gamma \|W\|_{SC_{p,\tau}}^{p} \quad \text{s.t. } E_\Omega = X_\Omega - D_\Omega \text{ and } W = X \tag{6}$$

The SC_p Matrix Completion Method Continued

- The augmented Lagrangian function $L(X, E_\Omega, W, Y, Z, \mu)$ for (6) is found and (4) is optimised through ADMM iteratively.
- By fixing E_Ω and W in (6), the optimal X can be computed as

$$X = N_{\Omega^c} + \frac{K_\Omega + N_\Omega}{2}$$

$$\text{where } \mathrm{K}_\Omega = E_\Omega + D_\Omega + \tfrac{1}{\mu} Z \textit{ and } N = W - \tfrac{1}{\mu} Z \tag{7}$$

- By fixing X and W in (6), E_Ω can be computed as

$$E_\Omega = \frac{\mu}{\mu + 2} H_\Omega \; ; \; H_\Omega = X_\Omega - D_\Omega - \frac{1}{\mu} Y_\Omega \tag{8}$$

- Finally, fixing X and E_Ω, (6) is simplified into

$$\underset{W}{min} \frac{1}{2} \|W - G\|_F^2 + \lambda \|W\|_{SC_{p,\tau}}^p \; ; \; G = X + \frac{1}{\mu} Z, \lambda = \frac{\gamma}{\mu} \tag{9}$$

How to solve for W ?

- The non-convex term $\|W\|^p_{SC_{p,\tau}}$ in (9) cannot be solved analytically.

Theorem 1:

The optimal solution W for (9) is :

$$W = Q\Sigma R^T \tag{10}$$

where Q and R are singular vector matrices of G, Σ is diagonal with entries σ_i^* which is solved by :

$$\min_{\sigma_i^* \geq 0} \frac{1}{2}(\sigma_i^* - \delta_i)^2 + \lambda \min(\sigma_i^*, \tau)^p \tag{11}$$

where δ_i is the ith singular value of G, thus

$$W = Qdiag(\sigma_1^*, \sigma_2^*, \cdots, \sigma_r^*, 0, \cdots, 0)R^T \tag{12}$$

- To solve for σ_i^*, the following theorem is adopted.

Theorem 2:

The minimal solution σ_i^* to (11) is given by :

$$\sigma_i^* = \begin{cases} \delta_i, & \tau \in [0, \tau^*] \\ x_i', & \tau \in (\tau^*, \infty) \end{cases} \tag{13}$$

where $\tau^* = \left(\frac{1}{2\lambda}(x_i' - \delta_i)^2 + (x_i')^p \right)^{\frac{1}{p}}$ and x_i' is given by:

$$x_i' = \begin{cases} 0, & \delta_i \in [0, v'] \\ x_i^*, & \delta_i \in (v', \infty) \end{cases} \tag{14}$$

where $v' = v + \lambda p v^{p-1}$ and $v = (2\lambda(1-p))^{\frac{1}{2-p}}$ and x_i^* is solved by classical gradient descent.

The Final Algorithm

Algorithm 1 The Schatten Capped p Norm Based Matrix Completion Method

Input: $D_\Omega, W, Y, Z, p, \mu > 0, \tau, \rho \in (1,2), \lambda > 0, k = 1, Iter$

Output: X^*

```
while k < Iter and not converged do
    update X by Eq. (7)
    update E_Ω by Eq. (8)
    update W by Eq. (12)
    Y_Ω = Y_Ω − μ(X_Ω − E_Ω − D_Ω)
    Z = Z − μ(W − X)
    μ = ρμ
    k = k + 1
end while
```

Our Implementations and Results

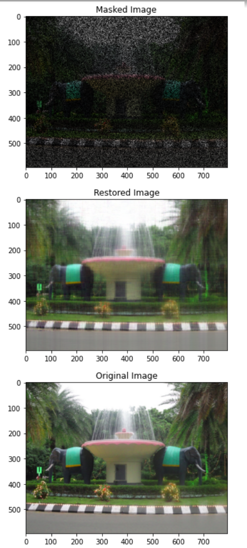

Figure 4: Random Mask Experiment

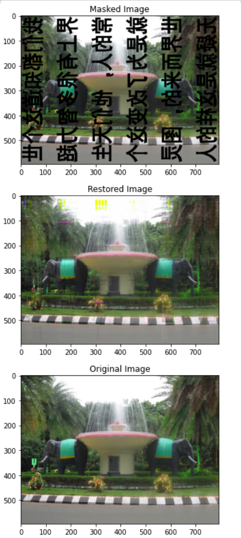

Figure 5: Text Mask Experiment

THANK YOU !

For implementation details and result, please visit:

https://github.com/KKamaleshKumar/EE5120_Course_Project