Tensor Completion on Estimating Missing Values

Anderson Chiu *

3 December 2022

Abstract

Data science and machine learning has become a hot topic recently. Many models have been developed to make prediction on future data. The data science field has been working hard in developing a quick and accurate model in estimating missing values. Tensor completion is one of the ways being developed to estimate the missing values. This project aims to understand a highly cited paper on Tensor Completion and implement the algorithms on using Tensor Completion to estimate missing data values. Applications including repairing missing values on images and make estimations on multidimensional arrays including movie rating and product rating, etc. The code will be uploaded to github. https://github.com/cychiuak/Math4992-final-project/tree/main

1 Introduction

Tensor is a multi-dimensional array. Tensor completion means filling in missing values in tensors. Due to the complexity of data sets nowadays, many data sets can be formulated into a multi-dimensional array. A typical examples will be the movie rating data. There are multiple dimensions of movie rating data including user age, user gender, etc. Therefore, the movie rating data can be modelled into a tensor and algorithms in this paper could be applied to estimate the desired missing values. Another application will be repairing broken images. The image's colour of a pixel is represented by R, G, B values. To estimate the missing pixels, we can model the image into a 3-D tensor with each dimension corresponding to values in R, G, B. Then we apply tensor completion to approximate the missing values in it. The next part of tensor completion is to find a criteria for selecting the missing values. An intuitive way is to select the values based on its neighbours. However, there are some cases that the values are from some distanced items. Therefore, a good criteria is to make estimation based on the global information of the whole tensor. For a matrix completion problem, it is well known to use the matrix rank as a criteria to capture the global information of a matrix. The missing values will be selected such that the rank is as low as possible, which means as similar to the global values of the matrix as possible. However, there are two problems for such kind of estimation criteria. Firstly, rank minimization problem is a NP-hard problem and secondly, how to define rank in tensors. The following paragraphs will address these problems. In the final part, this report will explain the algorithms, methodology used to code the algorithm and present the results using examples of data with high number of missing values.

2 Notations

We use the italic upper case letter χ to denote tensors. We use uppercase X to denote matrix and lower case $x_{i,j}$ to denote matrix or tensor entries. A n-dimension tensor is defined as

*Department of Mathematics, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. Email: **cychiuak@connect.ust.hk**

$\chi \in \mathbb{R}^{I_1 \times I_2 \times ... \times In}$. Its elements is denoted by $x_{i1,i2,...,in}$. Unfolding along dimension k is defined as $\text{unfold}_{(k)}(\chi) = X_{(k)} \in \mathbb{R}^{I_k \times (I_1 ... I_{k-1} I_{k+1} ... In)}$. The opposite of unfolding which is fold is defined as $\text{fold}_{(k)}(X_{(k)}) = \chi$. Denote the SVD of X as $X = U\Sigma V^T$. Denote the singular values of X as σ_i where $\sigma_i \in diag(\Sigma)$.The trace norm of a matrix is denoted as follows, $\|X\|_* = \sum_i \sigma_i$. The Frobenius norm of tensor is denoted as $(\sum_{i1,i2,...,in} |x_{i1,i2,...,in}|^2)^{\frac{1}{2}}$. Which is summation of the square of all entries. The spectral norm of a matrix is defined as $\|X\|_s = \max(\sigma_i)$. Then we define the shrinkage operator. $D_\tau(X) = U\Sigma_\tau V^T$ where $\Sigma_\tau = \text{diag}(\max(\sigma_i - \tau))$. We define the truncate operator. $T_\tau(X) = U\Sigma_{\bar{\tau}} V^T$ where $\Sigma_{\bar{\tau}} = diag(min(\sigma_i, \tau))$

3 Problem formulation

As mentioned above, matrix rank minimization problem is a NP-Hard problem. Fortunately, The trace norm can be used to approximate the matrix rank and it has been justified to be the tightest convex approximation of matrix rank[**?**]. Moreover, Cande's and Recht [**?**], B. Recht, M. Fazel, and P.A. Parrilo [**?**], and Cande's and Tao [**?**]showed that under certain conditions, the trace norm can be a good approximate to the optimal solution of rank minimization. Therefore, we aim to use trace norm to estimate the matrix rank. We need to first define the trace norm for a n-dimension tensor.

$$\sum_{i=1}^{n} \alpha_i \|X_{(i)}\|_*$$

It is the summation of trace norm of each matrix unfolded along each mode. Where α are constants satisfying $\sum_{i=1}^{n} \alpha_i = 1$. Under this definition, we obtained a minimization problem on tensor trace norm denoted as follows.

$$\min_{\chi} : \sum_{i=1}^{n} \alpha_i \|X_{(i)}\|_*$$
$$s.t. \chi_\Omega = T_\Omega$$

Where T_Ω are existing values.

4 Algorithm one: Simple Low Rank Tensor Completion Algorithm (SiLRTC)

The first algorithm aims to use block coordinate descend to solve the minimization problem. So we introduce blocks $M_1, M_2, ..., M_n$. The minimization problem becomes as follows.

$$\min_{\chi, M_i} : \sum_{i=1}^{n} \alpha_i \|M_i\|_*$$
$$s.t.\ X_{(i)} = M_i,\ for\, i = 1, 2, ..., n$$
$$\chi_\Omega = T_\Omega$$

However, notice that the terms are interdependent as when we try to minimize M_i, $X_{(i)}$ will be changed due to the constraint and changing the $X_{(i)}$ changes everything. Therefore, to solve this problem, the constraint is relaxed.

$$\min_{\chi, M_i} : \sum_{i=1}^{n} \alpha_i \|M_i\|_*$$
$$s.t.\ \|X_{(i)} - M_i\|_F^2 \le d_i,\ for\, i = 1, 2, ..., n$$

$$\chi_\Omega = T_\Omega$$

The minimization problem then can be written in the following lagrangian form.

$$\min_{\chi, M_i} : \sum_{i=1}^{n} \alpha_i \|M_i\|_* + \frac{\beta_i}{2} \|X_{(i)} - M_i\|_F^2$$

$$s.t.\ \chi_\Omega = T_\Omega$$

The extra terms d_i and also β_i can be self-defined by the user. A minimization problem with independent terms is obtained and we can apply the block coordinate descend algorithm. Keeping M_i as constants, the solution of $X_{(i)}$ is verified as follows: Since we are minimizing a Frobenius norm and it is piece wise disjoint, so we can minimize each entry of tensor χ individually.

$$\min_{\chi, M_i} : \sum_{i=1}^{n} \alpha_i \|M_i\|_* + \frac{\beta_i}{2} \|X_{(i)} - M_i\|_F^2$$

$$s.t.\ \chi_\Omega = T_\Omega$$

$$\min_{\chi_{(i1,i2,\ldots,in)}} : \sum_{i=1}^{n} \frac{\beta_i}{2} (X_{(i1,i2,\ldots,in)} - fold_{(i)}(M_i)_{(i1,i2,\ldots,in)})^2$$

$$s.t.\ \chi_\Omega = T_\Omega$$

Taking the derivative w.r.t $X_{(i1,i2,\ldots,in)}$ and equate it to zero we have,

$$X_{(i1,i2,\ldots,in)} = \begin{cases} (\frac{\sum_{i=1}^{n} \beta_i fold_{(i)}(M_i)}{\sum_{i=1}^{n} \beta_i})_{(i1,i2,\ldots,in)}, & \text{if } (i1, i2, ..., in) \notin \Omega \\ T_{(i1,i2,\ldots,in)}, & \text{if}(i1, i2, ..., in) \in \Omega \end{cases} \tag{1}$$

The solution of each M_i is the solution to the following problem.

$$\min_{M_i} : \frac{\alpha_i}{\beta_i} \|M_i\|_* + \frac{1}{2} \|X_{(i)} - M_i\|_F^2$$

It is proved [**?**] to have a solution $D_\tau(X_{(i)})$ where $\tau = \frac{\alpha_i}{\beta_i}$ Then we obtained the following algorithm.

Input: χ, T, Ω, β_i *with* $\chi_\Omega = T_\Omega$
For k = 1 to k = MaxIteration
  For i = 1 to i = n
    $M_i = D_{\frac{\alpha_i}{\beta_i}}(X_{(i)})$
  end
  update χ by (**??**)
end

Notice that if we compute the shrinkage of $X_{(i)}$,which is usually a matrix with very large number of column, by computing the SVD of $X_{(i)}$, it is costly in storage space as V^T will be occupying a large space. Therefore, it is necessary to perform some computation tricks to do the shrinkage operation. I propose the following trick which make use of the orthogonality of U in SVD. Let svd of $X_{(i)} = \mathrm{U}\Sigma V^T$

- compute the svd of $XX^T = \mathrm{U}(\Sigma\Sigma^T)U^T$
- Multiply U^T *to* $X_{(i)}$
- Do the shrinkage operation on Σ by multiplying another matrix with diagonal values = $\frac{diag(max(\sigma_i - \tau))}{\sigma_i}$ and the remaining entries equal to zero.

- multiply U back to X

Such a trick allows a much smaller storage space as for matrices with number of rows $\leq$ number of columns, the size of XX^T is much smaller.

5 Algorithm two: Fast Low Rank Tensor Completion Algorithm (FaLRTC)

As mentioned above, the trace norm function is a non-smooth function. Therefore, we can use Nesterov's[**?**] idea to solve non-smooth optimization problem. The procedure is as follows.

- Convert function into a smooth function
- Solve the problem with the smooth function and use its result to approximate the solution

The trace norm function is first converted to its dual norm which is Spectral norm. Then added to a strongly convex term $\frac{\mu}{2}\|Y\|_F^2$. The trace norm function then becomes.

$$f_\mu(\chi) = \sum_{i=1}^{n} \alpha_i \|X_{(i)}\|_* = \sum_{i=1}^{n} \max_{\|Y\|_s \leq 1} \langle X, Y \rangle - \frac{\mu}{2}\|Y\|_F^2$$

A smooth convex function is now obtained and the minimization problem can be solved by many typical approach such as gradient descend. The gradient of

$$\nabla f_\mu(\chi) = \sum_{i=1}^{n} \alpha_i \, T_1(\frac{\alpha_i}{\mu_i} X_{(i)}) = \sum_{i=1}^{n} (\frac{\alpha_i^2}{\mu_i}) \, T_1(X_{(i)})$$

Applying the gradient descend to solve the problem, we obtain the following algorithm.

Input: χ with $\chi_\Omega = T_\Omega$, K, μ_i and L, initialize Z = W = L, L' = L and B = 0
For k = 1 to K = MaxIteration
  while true do:
    $\theta = \frac{L}{2L'}(1 + \sqrt{1 + 4L'B})$, W = $\frac{\frac{\theta}{L}}{B+\frac{\theta}{L}} Z + \frac{B}{B+\frac{\theta}{L}} X$
    if ($f_\mu(\chi) \leq f_\mu(W) - \frac{\|\nabla f_\mu(W)\|_F^2}{2L'}$)
      break;
    endif
    $\chi' = W - \frac{\nabla f_\mu}{L'}$;
    if ($f_\mu(\chi') \leq f_\mu(W) - \frac{\|\nabla f_\mu(W)\|_F^2}{2L'}$;)
      $\chi = \chi'$; *break*;
    endif
    L' = L'/c;
  end while
  L = L';, Z = Z - $\frac{\theta}{L}\nabla f_\mu(W)$; B = B + $\frac{\theta}{L}$;
end for

In the code, as we don't know the actual value of $\|Y\|_F^2$. We know that$\|Y\|_s = \max(\sigma_i) \leq 1$ and $\|Y\|_F^2 = \sum_{i=1}^{n} \sigma_i^2 \leq ndims(Y)$. so it is proposed to use $ndims(Y)$ to approximate $\|Y\|_F^2$.

6 Algorithm three: High Accuracy Low Rank Tensor Completion Algorithm (HaLRTC)

It applies the ADMM algorithm to solve the problem. It is basically a block coordinate descend with more blocks. Firstly, define the augmented Lagrangian function as follows,

$$L_\rho(\chi, M_1, ..., M_n, Y_1, ..., Y_n) = \sum_{i=1}^{n} \alpha_i \|M_i\|_* + \langle \chi - M_i, Y_i \rangle + \frac{\rho}{2}\|M_i - \chi\|_F^2$$

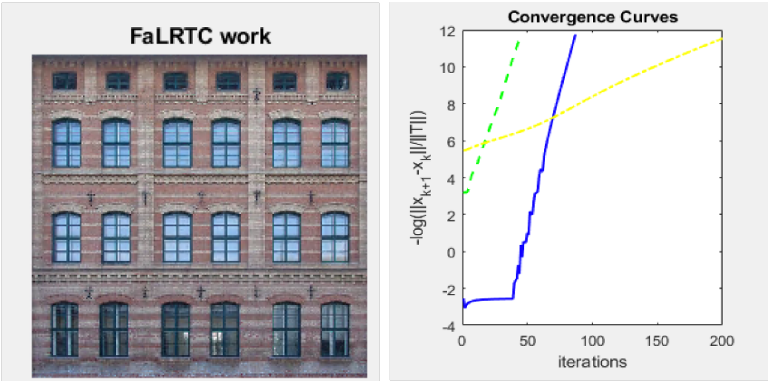
then apply the ADMM framework which keeps all blocks constant and minimizing only one block. Solving each variables we obtain the the algorithm.

Input: χ, T, Ω, ρ, *with* $\chi_\Omega = T_\Omega$
For k = 1 to k = MaxIteration
  For i = 1 to i = n
    $M_i = fold_i(D_{\frac{\alpha_i}{\rho}}(X_{(i)} + \frac{1}{\rho}Y_i))$
  end
  $\chi = \frac{1}{n}\sum_{i=1}^{n}(M_i - \frac{1}{\rho}Y_i)$, $\chi_\Omega = T_\Omega$
  $Y_i = Y_i - \rho(M_i - \chi)$
end

7 Numerical Results

We need to first choose the values of the initial variables first. For shrinkage operators $diag(\max(\sigma_i - \tau), 0)$, a too-large initial value will lead to a zero matrix and a too-small initial value will lead to no change in the matrix and thus terminated by the program quickly. The similar is for the truncate operator $diag(\min(\sigma_i, \tau))$. A value too small will lead to a near-zero matrix and a value too big will have no change on the matrix. Therefore, it is intuitive to select a suitable value. After experiment, it is suggested to select B = 0.001, $\rho = 1e-5$ and U = 0.001 with initial values of alpha all set to $\alpha_i = \frac{1}{ndims(\chi)}$

This shows the numerical results of the program. The original image has 80% missing values randomly erased and we see all three programs performed perfectly except that the time taken is actually a bit long. The running time of the quickest HaLRTC is about 10 seconds, whereas FaLRTC is about 24 seconds and the slowest SiLRTC is almost 100 seconds.



The green curve is the convergence rate of HaLRTC. The blue one is the convergence curve of FaLRTC and the yellow curve is the curve of SiLRTC. From the convergence curve we can see that the SiLRTC converges the slowest and HaLRTC converges the quickest. The HaLRTC also converges in the shortest amount of iterations. In terms of the accuracy, which is measures as $\frac{\|\chi - T\|_F}{\|T\|_F}$ where T is the ground truth, HaLRTC and FaLRTC perform equally good. Therefore, it is suggested to select HaLRTC in any case.

8 Conclusion

Indeed, the algorithm worked perfectly and is able to make very reasonable estimations with limited amount of information. Yet the run-time is still too long with over 10 seconds for the shortest ones. Therefore, it is still unrealistic to commercialise the algorithm and a future improvement will be to improve the runtime.

References

[1] G. L. Baker, & J. P. Gollub, *Chaotic Dynamics*, Cambridge University Press, 1990.

[2] B. Recht, M. Fazel, and P.A. Parrilo, *Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Mini- mization*, Proc. SIAM Rev., vol. 52, no. 3, pp. 471-501, 2010.

[3] Cande's and Recht, *Exact Matrix Completion via Convex Optimization*, Foundations of Computational Math., vol. 9, no. 6, pp. 717-772, 2009

[4] B. Recht, M. Fazel, and P.A. Parrilo, *Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Mini- mization*, Proc. SIAM Rev., vol. 52, no. 3, pp. 471-501, 2010.

[5] E.J. Cande's and T. Tao, *The Power of Convex Relaxation: Near- Optimal Matrix Completion*, IEEE Trans. Information Theory, vol. 56, no. 5, pp. 2053-2080, May 2010.

[6] Y. Nesterov, *Smooth Minimization of Non-Smooth Functions*, Math. Programming, vol. 103, no. 1, pp. 127-152, 2005.

[7] J.-F. Cai, E.J. Cande's, and Z. Shen, *A Singular Value Threshold- ing Algorithm for Matrix Completion,*SIAM J. Optimization, vol. 20, no. 4, pp. 1956-1982, 2010.