Applied Tensor Technology to Recommender System in Social Network

September 2016

MSc Data Science
Current, recommender systems typically use matrix factorisation to predict the most preferred item to particular users. However, they typically consider this preference as fixed over time, and do not take into account the temporal dependencies (i.e., users can change their preferences over time). Given this, we extend the state of the art prediction techniques to capture the temporal dependency. In particular, instead of considering recommender systems as a matrix completion problem (where the dimensions are the users and items) we add a third dimension to the model, which captures the temporal dependency between the preferred items of different users. Therefore, we face a tensor completion problem instead. To tackle this problem, we apply state-of-the-art completion algorithms which are proven to be efficient in completion error. We compare the performance of the algorithms on a syntactically generated data which follows behavioural influence of the underlying social network of the users.
## Contents

Acknowledgements ................................................................. ii

1 Introduction ................................................................. 1
   1.1 Background .............................................................. 1
   1.2 Research aim and objectives ........................................... 2
   1.3 Notation ............................................................... 3

2 Related Work ........................................................................ 5
   2.1 Preliminaries ............................................................. 6
      2.1.1 Basic Knowledge of Tensor ......................................... 6
      2.1.2 Tensor Matricization: ................................................ 6
      2.1.3 Trace Norm .......................................................... 7
      2.1.4 Tensor Completion .................................................. 7

3 Main Algorithm .................................................................... 9
   3.1 Simple Low Rank Tensor Completion ................................... 9
   3.2 High Accuracy Low Rank Tensor Completion ......................... 11

4 Test Model ......................................................................... 13
   4.1 Parameter Determine ..................................................... 14
   4.2 Visual testing ............................................................. 16
      4.2.1 Figure size: 225x225x3 ............................................. 17
      4.2.2 Figure size: 858x536x3 ............................................. 19
      4.2.3 Figure size: 1920x1200x3 .......................................... 22
      4.2.4 Comparison the three consequences ............................... 25
      4.2.5 Speculation ........................................................... 25
   4.3 Simulated data testing ..................................................... 26
      4.3.1 The consequence of 225x225x3 ................................... 27
      4.3.2 Analysis of the consequence ...................................... 28

5 Application ...................................................................... 30
   5.1 Simulated Social Network Data Generated ......................... 34
   5.2 Analyse of the results ................................................... 34
6 Conclusion

6.1 Future work ......................................................... 38
  6.1.1 Low rank tensor completion ................................. 38
  6.1.2 Continue social network recommender system .......... 39

Appendix A  RSE values of Simulated Data 40

Appendix B  Matlab Code of simple low rank tensor completion 43

Appendix C  Matlab Code of high accuracy low rank tensor completion 44

References 46
# List of Figures

2.1 1-D, 2-D, 3-D, 4-D, 5-D, 6-D tensor .............................................. 6  
4.1 RSE value of $\beta$ ................................................................. 14  
4.2 iteration step of $\beta$ ............................................................ 14  
4.3 RSE value of $\rho$ ................................................................. 15  
4.4 iteration step of $\rho$ ............................................................ 15  
4.5 10% rank data rho value of $\rho$ ............................................... 15  
4.6 40% rank data rho value of $\rho$ ............................................... 15  
4.7 70% rank data rho value of $\rho$ ............................................... 16  
4.8 full rank data rho value of $\rho$ ............................................... 16  
4.9 Original figure of 115x115x3 ...................................................... 17  
4.10 10%,30%,50%,70%,90% missing pixels completion results of 225x225x3 . 17  
4.11 Relative square error value ...................................................... 18  
4.12 Original figure of 858x536x3 ...................................................... 20  
4.13 10%,30%,50%,70%,90% missing pixels completion results of 858x536x3 . 20  
4.14 Relative square error value of 858x536x3 .................................... 21  
4.15 Original figure of 1920x1200x3 .................................................. 22  
4.16 10%,30%,50%,70%,90% missing pixels completion results of 1920x1200x3 . 23  
4.17 Relative square error value of 1920x1200x3 ................................ 24  
4.18 Relative square error value ...................................................... 26  
4.19 Relative square error value of 225x225x3 .................................... 27  
5.1 Social Network ................................................................. 31  
5.2 High accuracy RSE ............................................................... 35  
5.3 Simple RSE ................................................................. 35  
5.4 High accuracy ANE ............................................................... 36  
5.5 Simple ANE ................................................................. 36
## List of Tables

4.1 RSE value of image testing (225x225x3) ........................................ 19
4.2 RSE value of image testing (858x536x3) ........................................ 21
4.3 RSE value of image testing (1920x1200x3) ................................. 24
4.4 RSE value of 10% rank ............................................................. 27
4.5 RSE value of 40% rank ............................................................. 28
4.6 RSE value of 70% rank ............................................................. 28
4.7 RSE value full rank ................................................................. 28

A.1 High accuracy RSE ................................................................. 40
A.2 Simple RSE .......................................................................... 41
A.3 High accuracy ANE ............................................................... 41
A.4 Simple ANE .......................................................................... 42
Chapter 1

Introduction

1.1 Background

In the era of data explosion, the capacity of data and contents in data becomes larger and larger. The information on data has thousands of features compare to the previous information systems. Therefore, developing efficient way to extract useful features and analyse the intrinsic between features is hot research area for data scientists.

Tensor is defined a generalisation of matrices, that is presented as a multidimensional array. Tensor is a N-way or N-order arrays. A first order tensor is a vector, a second order tensor is a matrix, three order or higher than three tensor are called higher order tensors. In this data explosion era, tensor is to express the data in high dimension, which means with the dimension of data itself increasing, tensor technique can be new technique to process data. Within the research field of tensors, higher order tensor decomposition and completion are extensive with plentiful applications in many areas, like data mining(3), learning latent variable models (5), and etc.

For recommender system, the definition can be found in Recommender System Handbook, which is summarised in a completed explanation to solve the Information Overload problem using information filtering technique that to suggest needs with users’ information(2). In recent years, the use of recommender system becomes extremely common, which utilised in various fields, for instance, music, movies, books and e-commerce. The use of Recommender System becomes core competence in many major companies, AMAZON, NETFLIX and etc. The recommender systems is achieved by two techniques, typically, collaborative filtering or personality-based approach. Data are recorded in a matrix which is named as user-item(movie) matrix: columns are referred to items(movies) and rows to user filled with the preference for a user to the item(movie). In general case, the preference of users to items(movies) is much less than the total size of matrix. Therefore, the core issue in RS is the matrix is very sparse. The technique to full fill and predict the sparse matrix is the key to the solution.
Chapter 1 Introduction

However, the limited of previous RS recommender system is that size of data is in a matrix, which is 2-D data, user-items(movie). For instance, previous user-movie model can predict the person who has the same preference of movies to the user. A person’s preference may change with time changing. And preference of the user may be influenced by their friends or the person close to them. The question becomes find intrinsic relation in a three dimension data. Therefore, a high dimension data recommender system could be designed to solve the problem.

In this era, another key word is social network. The modern life is inundated with social network, which influences the people behaviour. For modern person, they receive and change information from social network. The influence of social network to daily life becomes larger and larger. Also, it increase the human interaction is also increasing by social network. If the user to user influential is know, on the basis of this concept, to create a recommender system. It can recommend the items, according to users influential, which is more reasonable.

1.2 Research aim and objectives

The main aim of this project is to build a 3-way tensor recommender system which is based on social network. The project can be separated into two parts, which are tensor completion and applied to recommender system. The first goal is tensor completion. It’s the technique that can fill and predict the missing values of the sparse tensor. We attach importance to the accuracy and performance of the model. Efficiency is not the core aspect that we consider to build the model. Second part is applied to recommender system, which in other words is to disposal of the data that we want to research and check the performance of the model. Below lists explain the detail description of two sections:

- For the first part, discover an appropriate tensor completion technique which is used to predict the missing value of sparse tensor. Utilise Liu’s work as reference to generate two completion techniques, simple low rank tensor completion and high accuracy low rank tensor completion.
  To demonstrate the performance of two techniques, we use visual data. Visual data is using the RGB data, which is a 3-way tensor data of an image. We randomly remove different percentage of pixel and utilise the completion techniques to recover the image to compare the performance. The consequence can be intuitively. As a consequence, the threshold value can be chosen, which can be used to test the simulated data.

- Apply the tensor completion technique to recommender. Basically, due to the limited of permission of using real data, generated simulated data is the way
to check the performance of consequences. The simulated data is generated by imitation of social network. The main concept is to build a user-movie-time recommender system model. We also want to to research that, in social network, how is the influence of others given to the user. The recommender system is to discover that, in a social network, with the influence of others to the user, what’s the preference of a user of the movie in the different time period. The generated data also combined by two parts. User-movie-time performance tensor, it displays the time users score their preference to the movie. From the social network, we obtain the users’ influence value, which composed by self-influenzial and others-influenzial. Multiply the influential value by the tensor, the new tensor can be defined by the influential of preference of users to movies. The recommender system is built on the basis of this tensor.

1.3 Notation

In this thesis, 1-D tensor, vector is using lower case characters (e.g. x, y and etc), 2-D tensor, matrix is using upper case characters (e.g. X, Y) and for 3-D tensor or more than 3-D, using greek letter to be expressed (e.g. χ, τ). The algorithm is based on visual data which is three order tensor(three way arrays).

For an n-order tensor, χ ∈ ℝI1×I2×⋯×In, it should unfold a tensor into matrix. For example, to unfold k-th mode on a tensor χ is defined as Unfoldk(χ) = ℝI1×⋯×Ik−1×Ik+1×⋯×In. In this project, use 3-order tensor χ as test data. The unfold matrix could be expressed as followings:

χ(1) = ℝI1×I2I3  
χ(2) = ℝI2×I1I3  
χ(3) = ℝI1I2×I3

χ(1), χ(2) and χ(3) are named as 1-mode, 2-mode and 3-mode unfolding matrix of χ.

To express the elements in tensor, we use lower case letter with index, xij. Let Ω be an index set, which serves to distinguish the missing value in a matrix. Therefore, XΩ shows the exist value.

The Frobenius norm of matrix is ∥X∥F = (∑i,j |xij|^2)^1/2 and the spectral norm is ∥X∥ = σ1(X). Therefor, the trace norm of matrix is ∥X∥tr = ∑i σi(X).

For a matrix X, it singular value decomposition is X = UΣV. Cai, Cande’s and Shen propose a theory, ”shrinkage” operator(10)
\[ D_{\tau}(X) = U\Sigma_{\tau}V \]

\( D_{\tau}(X) \) is "shrinkage" operator. \( \Sigma_{\tau} = diag(max(\sigma_i - \tau, 0)) \)
Chapter 2

Related Work

As above mentioned, the challenge of RS is to find a way to predict the missing values of the tensor. There exist mature techniques, for instance, SVD, low rank matrix factorisation and etc. However, with adding a new dimension, the size of data becomes 3-D, therefore, the aforementioned matrix completion technique should be modified. As a consequence, a novel technique named low rank tensor completion is provided, which is used to solve 3-way tensor or more than 3-way tensor completion problems. The low rank and approximately low rank problems are widely used in many fields, such as machine learning (7) computer vision (4), signal processing(6).

The estimation of missing value relies on finding the relationship between the known entries and unknown ones. An assumption is that the missing values may depend on the data close to them. The main idea is to capture the global information from data to predict the missing values, which is achieved by using the rank of the matrix. The rank of a matrix is calculated by singular value decomposition(SVD). However, in the general case, to calculate the ‘rank’, the aforementioned is a non-convex optimisation problem so it’s hard to reconstruct the matrix by rank. To overcome this issue, researchers have introduced a convex technique for low rank completion, which is known as trace norm(10).

While the convex optimisation technique has been widely used, not so many solutions have been prepared for tensor. A notable exception is the work of Liu(1), where the author introduces two low rank tensor completion technique, which named as simple low rank tensor completion and high accuracy low rank tensor completion.
2.1 Preliminaries

2.1.1 Basic Knowledge of Tensor

2.1.1.1 Tensor

For tensors, the mathematical definition is that all the components follow certain rules. The components are defined by the rank of tensor in multiple dimension. A $n^{th}$-rank tensor in $m$-dimension, which has $n$ indices and $m^n$ components. Tensor can use the number of indices to be generalised, scalar (no index), vector (one index), matrix (two indices) and high order tensor (three and more than three indices). The index describes the number of dimensions of space.

![Figure 2.1: 1-D, 2-D, 3-D, 4-D, 5-D, 6-D tensor](image)

2.1.1.2 Matricization:

Matricization: A very important process of reordering a tensor into $N$ modes matrix form in tensor decomposition, which is called as unfolding or flattening. It finds the intrinsic relation from the unfolding matrix. For example, consider the following 3-way tensor

$$
\chi = \begin{bmatrix}
1 & 3 & 5 & 7 \\
2 & 4 & 6 & 8 \\
\end{bmatrix}
$$

The 3 mode-$n$ unfoldings are


2.1.2 Trace Norm

For a matrix, the singular value decomposition is $X = U\Sigma V$. The vector of the singular value is $\hat{\lambda} = \text{diag}(\Lambda)$ (the diagonal entries of $\Lambda$). The trace norm of a matrix $X$ is defined as the $L_1$ norm of the sum of all singular values.

$$
\|X\|_{tr} = \left\| \hat{\lambda} \right\|_1 = \sum_i |\lambda_i|
$$  (2.1)

In many fields, factorisation, approximation, estimation technique, minimising the nuclear norm is to minimise the rank of the matrix. This ensures the problem to be convex so we can use convex minimisation methods.

2.1.3 Tensor Completion

The trace norm completion for matrices is to solve the following convex problem (8).

$$
\min_X : \|X\|_{tr}, \text{s.t.} : X_\Omega = M_\Omega
$$  (2.2)

The $X$ and $M$ are matrices, both having the same size. $X$ is the matrix that the rank should be minimised during the completion process. $\Omega$ is the index set of matrix $M$, which indicates the observed values. All entries that do not belong to $\Omega$ is the missing values.

Liu (1) found the following trace norm formula for the n-way tensor $\chi$ completion problem.

$$
\min_X : \sum_{l=1}^n \alpha_l \|\chi(l)\|_{tr}
\text{s.t.:} \chi_\Omega = \tau_\Omega
$$  (2.3)
In n-way tensor, $\chi$ is the target, which should be completed. The problem of this formulation is, $\chi_l, l = 1, ..., n$, are independent of each other.

For the above equation, it solves the minimum value of sum of multiple matrix trace norms. Because the unfolding tensor, $\chi_l, l = 1, ..., n$, determines the entries are the same ones so that they are all independent. Therefore, using trace norm completion cannot be optimised independently. The main thought is to separate them and solve each unfolding matrix independently. We introduce other matrices to the formula, $M_l, l = 1, ..., n$, and implement $M_i$ to be identical. It can change the problem to be the following optimisation problem.

\[
\min_{\chi, M_l} : \sum_{l=1}^{n} \alpha_l \|M_l\|_{tr} \\
\text{s.t.:} \chi = \tau \Omega \\
\chi_l = M_l, i = 1, ..., n
\] (2.4)

We use $\|M_l - \chi_l\|_F^2 \leq d_i$ to replace $\chi_l - M_l$. Thus, we have the following:

\[
\min_{\chi, M_l} : \sum_{l=1}^{n} \alpha_l \|M_l\|_{tr} \\
\text{s.t.:} \chi = \tau \Omega \\
\|M_l - \chi_l\|_F^2 \leq d_i, i = 1, ..., n
\] (2.5)

When $d_i$ is a threshold, which is defined by users. The threshold value, $d_i$ should be determined by users in different conditions. In tensor trace norm completion, it could be replaced by a positive value $\beta_l$

\[
\min_{\chi, M_l} : \sum_{l=1}^{n} \alpha_l \|\chi_l\|_{tr} + \frac{\beta_l}{2} \|\chi_l - M_l\|_F^2 \\
\text{s.t.:} \chi = \tau \Omega
\] (2.6)
Chapter 3

Main Algorithm

3.1 Simple Low Rank Tensor Completion

The core algorithm idea is from Liu’s work (1). He proposes a BCD (block coordinate descent) solution, which is to optimise one group with fixing the others groups. Due to the equation (2.6) ensured the separability of convex and non-smooth terms. The global optimal solution can be guaranteed by BCD as follows.

The pseudo-algorithm shows the steps of tensor completion, which follows the concept of BCD. It can be divided into two parts, computing $M_i$ and $\tau_{in}$.

- **Calculate $M_i$**: The optimal $M_i$ can be computed by "shrinkage" operator

  $$M_i = D_{\frac{\alpha_i}{\beta_i}}(X_i)$$

  Other researchers have already proved that the formula above can be used instead of equation (2.6)(10), which is shown below.

  $$\min_{M_i} : \alpha_i \| M_{(i)} \|_{tr} + \frac{\beta_i}{2} \| \chi_{(i)} - M_i \|_F^2$$

  $$\equiv \frac{\alpha_i}{\beta_i} \| M_{(i)} \|_{tr} + \frac{1}{2} \| \chi_{(i)} - M_i \|_F^2$$

- **Calculate $\tau_{in}$**: To solve the optimal $\chi$ with all other variables fixed problem, which converts to solve the following subproblem:

  $$\min_{\chi} : \sum_{i=1}^{n} \frac{\beta_i}{2} \| \chi_{(i)} - M_i \|_F^2$$

  The solution of above equation can be solved and is given by
\[ \tau_{in} = \begin{cases} \frac{\sum_i \beta_i \text{fold}_i(M_i)}{\sum_i \beta_i} i_1, i_2, \ldots, i_n \notin \Omega \\ \chi_{i_1, i_2, \ldots, i_n} \in \Omega \end{cases} \]

Where \( \Omega \) stores the index, which is used to distinguish the missing value. \( \chi_{in} \) keep the original value and only fill the missing value with predict consequence.

**Algorithm 1: Simple Low Rank Tensor Completion**

**Input**:
- \( \chi \): Input tensor
- \( \alpha \): The coefficient of the objective function
- \( \beta \): The relaxation parameter. The larger, the closer to the original problem.
- \( \Omega / \text{Omega} \): The index set indicating the observed elements
- \( K \): The maximum iteration steps

**Output**: \( \tau \): Completion tensor

```plaintext
for k=1 to K do
    for i=1 to n do
        \( \chi_i = \text{Unfolding}(\chi) \)
        \( D_i = \text{ShrinkageTraceNorm}(\chi_i, \frac{\alpha}{\beta_i}) \)
        \( M_i = \text{Fold}(D_i) \)
    end
    if \( i_1, i_2, \ldots, i_n \notin \Omega \) then
        \( \tau_{in} = \frac{\sum_i \beta_i \text{fold}_i(M_i)}{\sum_i \beta_i} \)
    else
        \( \tau_{in} = \chi_{in} \)
    end
end
```

The pseudo algorithm is achieved by Matlab. The detail of codes are shown in AppendixB.
3.2 High Accuracy Low Rank Tensor Completion

Liu also gives a basic idea of high accuracy low rank tensor completion, which uses the ADMM algorithm. We use the framework of ADMM algorithm to give an implementation with simple low rank tensor, which solves the noiseless problem directly. Here, we use the original definition of matrix trace norm and use tensor to instead of matrix.

\[
\min_{\chi, M_1, \ldots, M_n} \sum_{i=1}^{n} \alpha_i \| M_{i(i)} \|_{tr} \\
\text{s.t.: } \chi \Omega = \tau \Omega \\
\chi = M_i, i = 1, \ldots, n
\]

Therefore, the augmented Lagrangian function is:

\[
L_\rho(\chi, M_1, \ldots, M_n, Y_1, \ldots, Y_2) = \sum_{i=1}^{n} \alpha_i \| M_{i(i)} \|_{tr} + \langle \chi - M_i, Y_i \rangle + \frac{\rho}{2} \| M_i - \chi \|_F^2
\]

According to the framework of ADMM, the high accuracy low rank tensor completion could be designed on the basis of simple low rank tensor completion. The pseudo algorithm is shown in algorithm 2.

The pseudo algorithm is achieved by Matlab. The detail of codes are shown in AppendixC.
Algorithm 2: High Accuracy Low Rank Tensor Completion

**Input:**
- $\chi$: Input tensor
- $\alpha$: The coefficient of the objective function
- $\rho$: the initial value of the parameter; it should be small enough
- $\Omega$/Omega: The index set indicating the observed elements
- $K$: The maximum iteration steps

**Output:** $\tau$: Completion tensor

1. Update $Y$
2. For $k=1$ to $K$ do do
3.     For $i=1$ to $n$ do do
4.         $\chi_i = Unfolding(\chi - \frac{1}{\rho} M_i)$
5.         $D_i = ShrinkageTraceNorm(\chi_i, \frac{\alpha_i}{\beta_i})$
6.         $Y_i = Fold(D_i)$
7.     end
8. end
9. Update $X$
10. $\chi = \sum_{i=1}^{n} M_i + \rho \sum_{i=1}^{n} Y_i$
11. Update $M$
12. For $i=1:n$ do do
13.     $M_i = M_i + \rho \ast (Y_i - \chi)$
14. end
Chapter 4

Test Model

In this section, visual data and simulated data are used to test two models to give comparison. Here, we uses relative square error (RSE) to compare the performances between the models.

\[ RSE = \frac{\|\tau - \chi\|_F}{\|\chi\|_F} \quad (4.1) \]

Where \( \tau \) is the completion tensor and \( \chi \) is the original tensor. The relative square error (RSE) defines the relation of prediction values to actual values, which ranges from zero to infinite. Zero indicates the predicted result is an ideal result. The closer the consequence is to zero, the better the result is.

To test the performance of completion techniques, here, we uses visual data and simulated data. We aim to define the threshold value of RSE, which is used to check whether the result is good or not.

- **Visual data:** The reason to use visual data is that visual data can show the performance of completion figure result intuitionally. Therefore, the threshold value could be easily determined. It is the standard way to calculate tensor completion methods.

- **Simulated data:** The reason to measure simulated data is that it could discover the performance to complete the data with discrete ranks.

All the data used in this chapter are measured by ten times and calculate ten values average value.
4.1 Parameter Determine

The 3 parameters that are used in the calculation are, $\alpha$, $\beta$ and $\rho$.

- $\alpha$: It is defined by the coefficient of the objective function, $\|X\|_r := \alpha_i \|X_{i(i)}\|_r$
- $\beta$: It is the initial value of the parameter, which is used to determine the threshold value of shrinkage operator with $\alpha_i \frac{\beta_i}{\beta_i}$
- $\rho$: It is the relaxation parameter. The larger, the closer to the original problem.

Visual data are used as test data to identify the parameter values, among which $\alpha$ is a small constant value. We change the value of $\beta$ and $\rho$ to check the total iteration steps and RSE value. The parameters are determined by iteration steps and RSE values, and usually, we want less iteration steps and small RSE value.

The above two figures, figure 4.1 and figure 4.2, show the curves of RSE values and total iteration steps as $\beta$ changes. The parameter, $\beta$ changes from 0.001 to 1 by 0.01 each interval.

As previously mentioned, a lower RSE value is the ideal expectation. Although, less iteration steps can save more time, we try to find an appropriate $\beta$ value, which could have relatively less RSE value and iteration steps.

From the curves, it is noticed that when $\beta$ is between 0.005 to 0.1, the RSE values stay smaller and the iteration steps stay lower between 0.001 to 0.01. Once $\beta$ is beyond 0.01, the total iteration steps increase rapidly. Therefore, by combining the two situations, the value of $\beta$ can vary from 0.005 to 0.01.
The above two figures, figure 4.3 and figure 4.4, show the curves of RSE values and total iteration steps as $\rho$ changes. The parameter $\rho$ is changes from $10^{-6}$ to $10^3$ by multiples of ten each interval.

To select an appropriate value of $\rho$, we will compare the two plots. Discover the value of $\rho$, which have less iteration steps and lower RSE value. When $\rho$ is below $10^{-4}$, the RSE value stay in low value level. For iteration steps, when $\rho$ is $10^{-4}$, iteration steps reaches a relatively lower level case. From figure 4.4, we can observe that some iteration steps is 1. In this case, iteration step is 1, which means that the high accuracy low rank tensor completion only converge once. The result is very high. Therefore, we will not use the value of $\rho$, once the iteration step is 1. We choose $10^{-4}$ as the parameter value of $\rho$.

However, the previous $\rho$ can only be used to test the visual data. Because, the influence of $\rho$ to RSE is changing with the size of data changing, which means when we change the size of data, the value $\rho$ should be recalculated. We use three groups of simulated data, 20x20x20, 50x50x50 and 100x100x100. We use high order singular value decomposition to get discrete percentage rank of original data rank. We use the discrete percentage rank to regenerate new tensors and use new tensors to discover the relationship between rank and RSE.
The four above figures illustrate the RSE value in discrete size of data, different rank of data and different \( \rho \) value. We keep the data rank constant. Observes the performance of different size of data with values of \( \rho \) changing. Considering various conditions, the best performance of RSE in each figure happens at \( \rho = 0.01 \). Therefore, we use the 0.01 as the value of \( \rho \).

### 4.2 Visual testing

The visual testing is developed to discover the performance of two completion techniques in different missing pixels cases. The performance of completion results that can utilise the value of RSE to define. In visual data, the results of completion can be observed visually. We choose three different size of figures to draw the curves of RSE value changing in various percentage remove pixels. We compare the three plots to detect the relationship between RSE and missing percentage of pixels. From the curves, determine a threshold value of RSE value, which shows the better performance of the result of completion figure. The three discrete size of figures are, 225x225x3, 858x536x3 and 1920x1200x3 which indicate small figure size, medium figure size and large figure size. The contents of three figures are.

- The small image, 225x225x3, is Lena who is a famous model in image processing
- The medium image, 858x536x3, is an animal.
- The large image, 1920x1200x3, is a car.

We want to discover the performance of completion techniques in discrete image content.
4.2.1 Figure size: 225x225x3

The size of the test figure is 225x225x3. Randomly remove the pixels of the image at an increasing percentage from 10% to 90% with 10% each time. The original figure is given as follows.

\[ \text{Figure 4.9: Original figure of 115x115x3} \]

The results of 10%, 30%, 50%, 70% and 90% missing pixels are shown as follows.

\[ \text{Figure 4.10: 10%,30%,50%,70%,90% missing pixels completion results of 225x225x3} \]
The first column shows the image after removing pixels. The second column indicates the high accuracy low rank tensor completion and the third column indicates the simple low rank tensor completion result. We compare the completion figures to original figure and observe how close the completion figures to original figure to define the performance of results.

From the figure 4.10, it can be discovered that under 60% (include 60%), the result of high accuracy low rank tensor is better than the result of simple low rank tensor completion. The performance comparison to the original is acceptable. However, if percentage missing is more than 60%, the result of high accuracy can become worse than the simple low rank tensor completion. The simple low rank tensor completion image can still give the image an outline of image even if it is not obvious. Therefore, we map the value of RSE under 60% , which can be the threshold value to distinguish the performance of completion techniques. The detailed data of RSE value and the curve is given below.

![Relative square error value](image)

**Figure 4.11:** Relative square error value
We compare the RSE value for scenarios with missing percentage varying from 10% to 90%. Figure 4.11 demonstrates that RSE changes with different missing percentage of pixels.

The blue line describes the curve of high accuracy low rank tensor completion RSE with changing percentage. The red line describes the simple tensor completion. When the missing percentage is lower than 60%, the performance of high accuracy low rank tensor completion is better than simple low rank tensor completion by comparing the completion figures to original figure. However, once the missing percentage beyond 60%, the high accuracy RSE goes steep increasing. The RSE of simple low rank tensor completion is changing relatively stable. For RSE of high accuracy low rank tensor completion, when the missing percentage is below 60%, value of RSE is below 0.1. The value of RSE is below 0.1, which is the ideal performance of completion result. When the missing percentage beyond 60%, RSE steep increases to 0.8, which the performance is not good enough that cannot be used. The RSE value of simple low tensor completion, it remains stably increasing. When the missing percentage is 60%, its RSE value is 0.18. The value of RSE, which is below 0.2 can also give a relatively good result. Comparing the completion images to original image, when the value is between 0.2 to 0.3, the completion images are acceptable.

### 4.2.2 Figure size: 858x536x3

The size of the test figure is 858x536x3. Randomly remove the pixels of the image at an increasing percentage from 10% to 90% with 10% each time. The original figure is given as follows.
The results of 10%, 30%, 50%, 70% and 90% missing percentage are shown below.

Figure 4.12: Original figure of 858x536x3

Figure 4.13: 10%, 30%, 50%, 70%, 90% missing pixels completion results of 858x536x3
We compare the completion 858x536x3 image to original image. When the missing percentage reaches 70%, the result is still good. Aforementioned, when the missing percentage reaches 70%, for 225x225x3 image, the result of high accuracy low rank tensor completion is unacceptable. Therefore, we can define that the result of 858x536x3 image completion result if better. Thus, we map the value of RSE, which is 70%, and plot RSE curve.

**Table 4.2: RSE value of image testing (858x536x3)**

<table>
<thead>
<tr>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Accuracy</td>
<td>0.0991</td>
<td>0.0407</td>
<td>0.0581</td>
<td>0.0765</td>
<td>0.0962</td>
</tr>
<tr>
<td>Simple</td>
<td>0.0321</td>
<td>0.0620</td>
<td>0.0943</td>
<td>0.1283</td>
<td>0.1628</td>
</tr>
</tbody>
</table>

![Relative square error value of 858x536x3](image)

**Figure 4.14:** Relative square error value of 858x536x3

The blue line denotes the curve of high accuracy low rank tensor completion RSE with changing percentage. The red line is simple tensor completion RSE with changing percentage. For high accuracy RSE, from the RSE plot, when the missing percentage is below 80%, we can see that the RSE is less than 0.2. The missing percentage once becomes 90%, the RSE goes very high that can reach 0.9. For the RSE value is 0.9, which means the result cannot be acceptable. There is also a strange point in this plot. We can observe that when missing percentage is 10% that the RSE closes to 0.1. It is much
higher than 20% RSE. The point RSE closes to 0.1 is when the missing pixels reaches 50%. The RSE value of simple low tensor completion, it remains stably increasing. The RSE plot becomes close to the straight line. The plot of 858x536x3 simple RSE is close to 225x225x3 simple RSE plot.

4.2.3 Figure size: 1920x1200x3

The size of the test figure is 1920x1200x3. Randomly remove the pixels of the image at an increasing percentage from 10% to 90% with 10% each time. The original figure is given as follows.

![Image of a car](image.png)

**Figure 4.15:** Original figure of 1920x1200x3

The results of 10%, 30%, 50%, 70% and 90% missing percentage are shown below.
We compare to the previous visual image, 225x225x3 and 858x536x3. The results of high accuracy completion is unexpectedly excellent. We can discover from figure 4.16, when the missing percentage reaches 90%. The completion figure, basically, recover the original image. In the aforementioned two tests, the completion results of 90% missing percentage are not ideal results. For the simple low rank tensor completion results, only from the figure, it cannot tell whether the performance of simple low rank tensor completion is better or not.

The details and RSE curves are shown below.
### Table 4.3: RSE value of image testing (1920x1200x3)

<table>
<thead>
<tr>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Accuracy</td>
<td>0.0191</td>
<td>0.0206</td>
<td>0.0236</td>
<td>0.0288</td>
<td>0.0363</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0474</td>
<td>0.0640</td>
<td>0.0897</td>
<td>0.1369</td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td>0.0649</td>
<td>0.1212</td>
<td>0.1737</td>
<td>0.2219</td>
<td>0.2667</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3096</td>
<td>0.3514</td>
<td>0.3932</td>
<td>0.4370</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.17:** Relative square error value of 1920x1200x3

The blue line denotes the curve of high accuracy low rank tensor completion RSE with changing percentage. The red line is simple tensor completion RSE with changing percentage. For high accuracy RSE, as previously mentioned, the performance is excellent, even when the missing percentage reaches 90%. Therefore, we can record RSE in 90% missing pixels case. When the missing percentage is below 80% (include 80%), RSE is below 0.1, which generate the completion results ideally. For simple RSE value, the results are worthy compared between the previous two tests. The curve of simple low rank tensor completion RSE is a straight line. However, for previous tests, the simple RSE value goes higher than 0.3 when the missing percentage reaches 90%. In this test, the RSE value goes higher than 0.3 when the missing percentage is 60%.
4.2.4 Comparison the three consequences

In this section, the results of 3 different image data are compared. The changing trend of RSE of different data in different missing percentage pixels case is covered. Therefore, we can determine the appropriate threshold value that could be used to define the performance of other data.

For high accuracy low rank tensor completion, in general, the performance is better than simple low rank tensor completion. However, once the missing percentage reaches a threshold value, it will become rather worse, whose result cannot be used. Further, also in some lower percentage pixel missing cases, the performance of high accuracy low rank tensor completion is not better than the performance of simple low rank tensor completion. The cause of this situation is to be discovered by further research.

For simple low rank tensor completion technique, the plots of RSE follows the line regression. When the size of data is small, it is a curve. Once the data size becomes larger, the plot becomes closer to straight line, which means the simple low rank tensor completion is more stable.

We combine the three consequences of visual data testing and compare the completion image with the original image to determine the performance. Then we map the corresponding value to the RSE value so the threshold value can be determined.

The RSE values can be separated into 4 level, below 0.1, 0.1-0.2, 0.2-0.3 and above 0.3. We can regard the value of RSE, 0.3, as the threshold value, which means the completion tensor is acceptable to use. The RSE value above 0.3 should be re-calculated. The ideal value of RSE is below 0.1 and the value that the more closer to zero, the better performance of the completion result. The other two level, 0.2-0.3 and 0.1-0.2, we can define that 0.1-0.2 is relatively good and 0.2-0.3 is not good enough. However, both two levels are the acceptable result in completion technique.

4.2.5 Speculation

From previous visual data test, the performance of high accuracy low rank tensor completion becomes better as the size of data increase. The performance of simple low rank tensor completion, on the contrary, becomes worse as the size of data increase.

Here, we make some assumption:

- High accuracy low rank tensor completion is used for large size data. Simple low rank tensor completion is used for small size data.
• We conjecture that we can improve the performance of two completion techniques by changing the parameter? Furthermore, for large missing percentage data, could we change the parameter to improve the completion consequence.

4.3 Simulated data testing

As previously mentioned, the simulated data are used to further test the techniques. Therefore, we can generate the testing data randomly. The size was chosen to be the same with visual data, 225x225x3, which is easier to compare between RSE values. Due to the limits of time and equipment, the larger size 858x538 is not tested.

The above flow chart explains the process to generate the types of simulated data in discrete rank. The first step is to generate a 225x225x3 tensor, which is fully filled with random numbers from 0 to 9 then unfold the tensor into three matrices. Next step is using singular value decomposition to decompose three matrices and get four distinct groups of singular values. In the process of SVD, we only keep the numbers of rank we want. The first singular value keeps 10 percentage of original rank. Next keep 40 percentage and 70 percentage. The last group has the full rank of the original. The re-construct data have the particular rank we want. We use four new singular values, which are all deleted, of each group to generate the new tensor, respectively. Therefore, the four new re-construct tensors have corresponding ranks, which are 23x23x1, 90x90x2,
158x158x3, 225x225x3. We correlate the RSE values of two completion technique, and identify the performance of two models in different rank data and distinct missing percentage.

4.3.1 The consequence of 225x225x3

![Figure 4.19: Relative square error value of 225x225x3](image)

**Table 4.4: RSE value of 10% rank**

<table>
<thead>
<tr>
<th></th>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Accuracy</td>
<td></td>
<td>1.30E-05</td>
<td>1.66E-05</td>
<td>1.59E-05</td>
<td>1.57E-05</td>
<td>1.63E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.65E-05</td>
<td>1.61E-05</td>
<td>3.61E-03</td>
<td>5.86E-02</td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td></td>
<td>1.30E-05</td>
<td>1.66E-05</td>
<td>1.59E-05</td>
<td>1.56E-02</td>
<td>2.10E-02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.84E-02</td>
<td>4.00E-02</td>
<td>5.83E-02</td>
<td>9.05E-02</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.5: RSE value of 40% rank

<table>
<thead>
<tr>
<th></th>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td></td>
<td>2.03E-06</td>
<td>2.67E-06</td>
<td>1.70E-04</td>
<td>6.98E-02</td>
<td>1.35E-01</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td></td>
<td>1.94E-01</td>
<td>2.45E-01</td>
<td>2.90E-01</td>
<td>3.31E-01</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: RSE value of 70% rank

<table>
<thead>
<tr>
<th></th>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td></td>
<td>1.33E-01</td>
<td>2.04E-01</td>
<td>2.61E-01</td>
<td>3.11E-01</td>
<td>3.55E-01</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td></td>
<td>3.94E-01</td>
<td>4.30E-01</td>
<td>4.63E-01</td>
<td>4.96E-01</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7: RSE value full rank

<table>
<thead>
<tr>
<th></th>
<th>Missing Percentage(%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td></td>
<td>3.05E-01</td>
<td>3.42E-01</td>
<td>3.76E-01</td>
<td>4.06E-01</td>
<td>4.35E-01</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td></td>
<td>4.62E-01</td>
<td>4.88E-01</td>
<td>5.13E-01</td>
<td>5.37E-01</td>
<td></td>
</tr>
</tbody>
</table>

4.3.2 Analysis of the consequence

Figure 4.19 describes the behaviour of two models in various missing percentage in different rank cases. The red line gives explicit curve for 10% rank data. The green line gives 40% rank and the blue line is for 70%. The black line gives the curve for full rank data. The four tables gives the precise RSE value.
From visual data testing, RSE value, the conclusion, which is under 0.1 shows a ideally performance. For the lower rank data, 10% rank data, RSE are all under 0.1. Both completion techniques perform excellent in low rank data case.

As the rank increase to 40% of the original data, the performance is still well in lower remove percentage(0.4) situation. Even when the missing percentage reaches to 90%. For both two RSE values, are a little bit higher than 0.3. For ideally consequences, the RSE value is below 0.1, when the missing percentage is lower than 40% (include 40%). The two RSE value increase relatively stable in the same cases, which is the missing percentage between 50% to 90%.

The performance for high rank data, 70% rank, is not well enough. There are only three points less than threshold value, which the result is acceptable. For the rest of points, all larger than 0.3. And for full rank case, the value of RSE are all not good enough. All point are higher than 0.3. There is not point that RSE is lower than threshold value.

In summary, we compare the two completion techniques, we can see that the both completion techniques perform well in low rank data. We can also observe that, high accuracy low rank tensor completion performs better than simple low rank tensor completion. In the low rank data as missing percentage increase, high accuracy low rank tensor completion RSE is less than simple low rank tensor completion RSE. For the same missing percentage data, when the rank goes high, high accuracy also gives better implementation than simple one.
In this chapter, the following ideas will be introduced:

- The concept of combination of social network and recommender system.
- Give description of how to build the input tensor data. And the meaning of input data.
- Apply two completion techniques to the simulated data of social network recommender system and analyse the results.

To build a recommender system with social network, the key concept is generate a influential tensor data, which is determined by user to user influence and user’s like to movies. Here, use three processes to build the input tensor data.

- From social network, build user-user influential matrix.
- Use user-movie matrix data to expand to user-movie-time tensor.
- Multiply user-user influential matrix to user-movie-time tensor.

Therefore, the new generated tensor is the aim tensor data, which we are gonna applied to.

**User to User influence:** The user to user influential matrix is generated by social network. The simple example is given. Figure 5.1 describe a simple social network.
From figure 5.1, we can see that the point indicates a user and the lines between points mean the users know each other. Therefore, a matrix can be built by the social network.

\[
\begin{array}{ccccc}
& A & B & C & D & E \\
A & 0 & 0 & 0 & 1 & 1 \\
B & 0 & 0 & 1 & 1 & 0 \\
C & 0 & 1 & 0 & 1 & 0 \\
D & 1 & 1 & 1 & 0 & 0 \\
E & 1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

We convert the social network into matrix. The value 1 in this table means the two person know each other in social network. The value 0 is the two persons who have no relation. Here, we introduce two concepts, self-influential and others-influential.

- **Self-influential**: Self-influential is how the preference of items/movies influenced by user himself/herself.

- **Others-influential**: Others-influential is how the preference of items/movies influenced by others.

The influential table is shown below, which is transferred by above social network table.

\[
\begin{array}{ccccc}
& A & B & C & D & E \\
A & P_{A\rightarrow A} & 0 & 0 & P_{D\rightarrow A} & P_{E\rightarrow A} \\
B & 0 & P_{B\rightarrow B} & P_{C\rightarrow B} & P_{D\rightarrow B} & 0 \\
C & 0 & P_{B\rightarrow C} & P_{C\rightarrow C} & P_{D\rightarrow C} & 0 \\
D & P_{A\rightarrow D} & P_{B\rightarrow D} & P_{C\rightarrow D} & P_{D\rightarrow D} & 0 \\
E & P_{A\rightarrow E} & 0 & 0 & 0 & P_{E\rightarrow E} \\
\end{array}
\]

\(P_{\text{user}_A\rightarrow \text{user}_B}\) indicate others-influential, the influence of User A to User B. \(P_{A\rightarrow A}, P_{B\rightarrow B}\), etc. are self-influential. The character of self-influential and others-influential is that the sum of a user’s influential value should be equal to one. Because we want to research whether a user will be influenced by others to watch a movie. A person who has large self-influential value, the others-influential value should be small. In other words, person
who has a large self-influential value will not be easy to be influenced by others into watching a movie. On contrary, if a user has a smaller self-influential value, the chances are bigger for this user to be influenced into watching a movie. For instance, there are two persons, A and D, if they have a strong relationship, like siblings or couples, the others-influential value could be a large value. B has a larger probability to make impacts on A. If someone also has a large self-influential value, which indicates that this person is very confident that he receives relatively less suggestion from others. The above influential table can be converted by the equations shown below:

\[
\begin{align*}
P_{A\rightarrow A} + P_{D\rightarrow A} + P_{E\rightarrow A} &= 1 \\
P_{B\rightarrow B} + P_{C\rightarrow B} + P_{D\rightarrow B} &= 1 \\
P_{B\rightarrow C} + P_{C\rightarrow C} + P_{D\rightarrow C} &= 1 \\
P_{A\rightarrow D} + P_{B\rightarrow D} + P_{C\rightarrow D} + P_{D\rightarrow D} &= 1 \\
P_{A\rightarrow E} + P_{E\rightarrow E} &= 1
\end{align*}
\]

**User-Movie tensor:** The user-movie matrix is generally used in recommender system, which provides users’ preferences to movies. To generate the wanted user-movie-time influential tensor, here, we generate a random user-movie matrix, which the number of user is the same with number of users in influence matrix.

<table>
<thead>
<tr>
<th></th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
<th>$M_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

The range of a user’s preference of a movie is 0-5, where 0 indicates the user does not watch the movie and 1-5 indicates how a user feels about a movie (1 is the minimum point and 5 is the maximum point). To extend the user-movie matrix to user-movie-time tensor, on the basis of this matrix, add another axis, time T. It can express when the user watches the movie. The user-movie matrix can be regarded as the sum of user-movie-time tensor. Here is an example that the user-movie-time tensor is converted by the above user-movie matrix.
Matrix $T_1$, $T_2$ and $T_3$ expound the user’s preference of a movie in different time period. If user watches the movie at time $T_2$, the previous time $T_1$ and next time $T_3$ should be 0.

**User-MovieTime influential tensor:** From the social network, we can obtain user-user influential matrix, which accords self-influential value and the influence value of others given to the user. From user-movie-time tensor, we obtain the user’s evaluate of movies and when the user gives the evaluation. At last, we combine the matrix and tensor to generate a new tensor, which explain the user-movie-time influence.
We give the influence vector of user A and user’s movie evaluation in various time axis. We generate a new tensor, which describes the influence of a movie given to user in a various time axis.

\[
\begin{array}{c|c}
T_1 & M_1 \\
A & 5^*P_{A\rightarrow A} + 4^*P_{E\rightarrow A} \\
\end{array}
\quad
\begin{array}{c|c|c}
T_2 & M_1 & T_3 \\
A & 0 & P_{D\rightarrow A} \\
\end{array}
\]

5.1 Simulated Social Network Data Generated

The social network data and user-movie-time tensor are generated by MatLab. Applied simple low rank tensor completion and high accuracy low rank tensor completion to simulated data. We generated 1 groups of data, 50x10x5, 50x20x5, 50x30x5, 100x10x5, 100x20x5, 100x30x5, 150x10x5, 150x20x5, 150x30x5, 200x10x5, 200x20x5, 200x30x5(user x movie x time). The size of user utilise four sets(50, 100, 150 and 200) and size of movie also has three sets(10, 20 and 30). We make the time axis as constant, 5. We choose a size of user from three sets, changing size of movie to record corresponding result. Relative square error(RSE) is utilised, which is easy to compare simulated data with previous results.

\[
RSE = \frac{\| \tau - \chi \|_F}{\| \chi \|_F}
\]

Propose a new concept used to evaluate the performance, average norm error(ANE).

\[
ANE = \frac{\| \tau - \chi \|_F}{\text{size of tensor}}
\]

The ANE denotes the percentage error of total size of tensor. RSE defines the relation of prediction values to actual values. ANE can reflect how prediction values close to actual values.

5.2 Analyse of the results

From previous outcomes, the RSE values, which under 0.1 can be considered as ideally consequence. And the range from 0.1 to 0.2, which is relatively good consequence, is also well result. The range from 0.2 to 0.3 is acceptable. The threshold value is 0.3, when the consequence is higher than 0.3. It indicates that the consequence is unacceptable. The result cannot be used. The RSE plots is given, which are high accuracy low rank tensor completion RSE and simple low rank tensor completion RSE.
The above two figures give the high accuracy RSE and simple RSE with different missing percentage. We compare the two RSE plots, figure 5.2 and figure 5.3. The RSE value depends on the size of each axis of tensor. We keep one axis size constant and change another axis size. For four sizes of user(50, 100, 150 and 200), keep one of four remain constant. RSE goes smaller, when the size of movies goes higher ($RSE_{10} > RSE_{20} > RSE_{30}$). For sizes of movie(10, 20 and 30), keep one of three remain constant. RSE goes smaller, when the size of users goes higher.

Then, we are searching the relationship between total size of tensor and RSE. For the groups of data, we have some same total size of tensor(50x20x5 and 100x10x5). Contrast the four groups(50x20, 100x10; 50x30, 150x10; 100x20, 200x10; 100x30, 150x20). The result is ambiguous, which cannot give the conclusion that there is relationship between the total size of tensor and the RSE.

From previous results, the RSE, 0.3, can be regarded as a threshold. The RSE threshold
line shifts to right side as the size increase, which means both two completion techniques perform well in large missing percentage situation. We can see that the performance of high accuracy RSE reaches 0.3 at 0.3 missing percentage in the smallest size 50x10x5. When the size increase to 200x30x5, RSE reaches 0.3 at 0.7 missing percentage. The same situation appears at simple low rank tensor RSE.

The above two figures illustrates high accuracy ANE and simple ANE with different missing percentage. The ANE depends on the size of each axis of tensor. We keep one axis size constant and change another axis size. For four sizes of user (50, 100, 150 and 200), keep one of four remain constant. ANE goes smaller, when the size of movies goes higher ($RSE_{10} > RSE_{20} > RSE_{30}$). For sizes of movie (10, 20 and 30), keep one of three remain constant. ANE goes smaller, when the size of users goes higher. However, the difference is perhaps that ANE value depends more on the total size of
tensor. For instance, the data 50x20 and 100x10, the two data has the same total size. However, the curves of two data, 50x20 and 100x20, coincide. So is the same as curves, 150x10 and 100x20.

Further more what differs from RSE is that the changing rate is not as large as RSE. The growth rate is much smaller than RSE when it does not reach the threshold. The threshold here, for most groups of data, is 0.8 missing percentage. When it is less than 0.8 percentage, it grows a little higher each time. It increases suddenly after the percentage grows to 0.9 percentage.
Chapter 6

Conclusion

In this paper, we use Liu’s work to develop low rank matrix completion technique to low rank tensor completion, which deploys the definition of trace norm convex technique. Here, we research two completion techniques, simple low rank tensor completion and high accuracy low rank tensor completion, and give the performance and corresponding characters. Visual data and simulated are used to define the performance of two techniques in discrete size of data and different rank data. In visual data test, the high accuracy low rank tensor completion performs well in visual data completion, which is valuable to have further research on video data completion.

We also introduce the concept of combining the social network with recommender system. Test the performance of social network recommender system applied with tensor completion techniques. The social network data is generated by simulation. The result is better than we expected, which can be used for social network recommender system.

6.1 Future work

The further work, it can be separated into two parts.

- Research on trace norm, low rank tensor completion
- Research on social network recommender system.

6.1.1 Low rank tensor completion

For the trace norm, low rank tensor completion technique, the performance is discussed. However, the suspecting is provided in previous thesis.
• Test the performance of two completion techniques in discrete size data, and observe which technique performs well in large size data and which performs well in small size data.

• More efforts should be put into parameter tests. As previously mentioned, for high accuracy low rank tensor, it influences the consequence a lot when the parameter changes. The relation between the parameter and various size data/discrete missing percentage are in demand to be tested.

• For both two completion techniques, the disadvantage is that the completion process is very slow. For instance, to complete a 90% missing of 1920x1200x3 image data, it spent 8063.373 seconds in using two completion techniques. It took 7 days running the programme to calculate the average RSE value. Therefore, develop the fast low rank tensor completion is necessary.

6.1.2 Continue social network recommender system

The idea of social network is theoretical. The model is only tested with simulation data. Real data test is necessary to discover more useful characters. Therefore, to continue the research, several processes should be paid more attention.

• The first part is real data test. All the social network data used in this project are simulated data, especially, the concept of user-influential matrix. Further work on real data to discover the appropriate matrix building is to be continued. Meanwhile for the user-movie recommender system, even it is widely applied in many companies. Converting user-movie matrix to user-movie-time tensor, it also needs more background knowledge to support to convert. Furthermore, the robustness and qualities of the user-movie-time model requires more research to be improved.
Appendix A

RSE values of Simulated Data

Table A.1: High accuracy RSE

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>50x10</td>
<td>0.1531</td>
<td>0.2262</td>
<td>0.2886</td>
<td>0.3577</td>
<td>0.4020</td>
<td>0.4604</td>
<td>0.5340</td>
<td>0.8357</td>
<td>0.9484</td>
</tr>
<tr>
<td>50x20</td>
<td>0.1486</td>
<td>0.2180</td>
<td>0.2754</td>
<td>0.3327</td>
<td>0.3888</td>
<td>0.4367</td>
<td>0.5055</td>
<td>0.5630</td>
<td>0.9491</td>
</tr>
<tr>
<td>50x30</td>
<td>0.1360</td>
<td>0.2021</td>
<td>0.2653</td>
<td>0.3175</td>
<td>0.3645</td>
<td>0.4176</td>
<td>0.4734</td>
<td>0.5429</td>
<td>0.9502</td>
</tr>
<tr>
<td>100x10</td>
<td>0.1361</td>
<td>0.2022</td>
<td>0.2541</td>
<td>0.3005</td>
<td>0.3503</td>
<td>0.4052</td>
<td>0.4574</td>
<td>0.5277</td>
<td>0.9466</td>
</tr>
<tr>
<td>100x20</td>
<td>0.1265</td>
<td>0.1891</td>
<td>0.2374</td>
<td>0.2864</td>
<td>0.3335</td>
<td>0.3744</td>
<td>0.4224</td>
<td>0.4792</td>
<td>0.9489</td>
</tr>
<tr>
<td>100x30</td>
<td>0.1225</td>
<td>0.1813</td>
<td>0.2318</td>
<td>0.2782</td>
<td>0.3218</td>
<td>0.3671</td>
<td>0.4103</td>
<td>0.4636</td>
<td>0.9487</td>
</tr>
<tr>
<td>150x10</td>
<td>0.1175</td>
<td>0.1715</td>
<td>0.2135</td>
<td>0.2568</td>
<td>0.2987</td>
<td>0.3432</td>
<td>0.3957</td>
<td>0.4800</td>
<td>0.9487</td>
</tr>
<tr>
<td>150x20</td>
<td>0.1148</td>
<td>0.1714</td>
<td>0.2143</td>
<td>0.2547</td>
<td>0.2926</td>
<td>0.3294</td>
<td>0.3727</td>
<td>0.4251</td>
<td>0.5735</td>
</tr>
<tr>
<td>150x30</td>
<td>0.1071</td>
<td>0.1575</td>
<td>0.2009</td>
<td>0.2400</td>
<td>0.2769</td>
<td>0.3127</td>
<td>0.3480</td>
<td>0.3916</td>
<td>0.4708</td>
</tr>
<tr>
<td>200x10</td>
<td>0.1087</td>
<td>0.1573</td>
<td>0.1951</td>
<td>0.2331</td>
<td>0.2681</td>
<td>0.3083</td>
<td>0.3578</td>
<td>0.4310</td>
<td>0.8813</td>
</tr>
<tr>
<td>200x20</td>
<td>0.0970</td>
<td>0.1393</td>
<td>0.17701</td>
<td>0.2097</td>
<td>0.2419</td>
<td>0.2727</td>
<td>0.3060</td>
<td>0.3518</td>
<td>0.4601</td>
</tr>
<tr>
<td>200x30</td>
<td>0.0958</td>
<td>0.1406</td>
<td>0.1783</td>
<td>0.2124</td>
<td>0.2436</td>
<td>0.2739</td>
<td>0.3064</td>
<td>0.3459</td>
<td>0.4206</td>
</tr>
</tbody>
</table>
### Table A.2: Simple RSE

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>50x10</td>
<td>0.1853</td>
<td>0.2690</td>
<td>0.3430</td>
<td>0.4304</td>
<td>0.4961</td>
<td>0.5931</td>
<td>0.7304</td>
<td>0.8704</td>
<td>0.9483</td>
</tr>
<tr>
<td>50x20</td>
<td>0.1767</td>
<td>0.2520</td>
<td>0.3150</td>
<td>0.3749</td>
<td>0.4388</td>
<td>0.5082</td>
<td>0.6288</td>
<td>0.7780</td>
<td>0.9485</td>
</tr>
<tr>
<td>50x30</td>
<td>0.1687</td>
<td>0.2379</td>
<td>0.3054</td>
<td>0.3627</td>
<td>0.4152</td>
<td>0.4796</td>
<td>0.5663</td>
<td>0.7238</td>
<td>0.9482</td>
</tr>
<tr>
<td>100x10</td>
<td>0.1539</td>
<td>0.2290</td>
<td>0.2889</td>
<td>0.3492</td>
<td>0.4170</td>
<td>0.5099</td>
<td>0.6196</td>
<td>0.7801</td>
<td>0.9466</td>
</tr>
<tr>
<td>100x20</td>
<td>0.1487</td>
<td>0.2176</td>
<td>0.2661</td>
<td>0.3189</td>
<td>0.3740</td>
<td>0.4336</td>
<td>0.5231</td>
<td>0.6772</td>
<td>0.9435</td>
</tr>
<tr>
<td>100x30</td>
<td>0.1484</td>
<td>0.2113</td>
<td>0.2626</td>
<td>0.3108</td>
<td>0.3582</td>
<td>0.4166</td>
<td>0.4910</td>
<td>0.6317</td>
<td>0.9197</td>
</tr>
<tr>
<td>150x10</td>
<td>0.1305</td>
<td>0.1890</td>
<td>0.2383</td>
<td>0.2905</td>
<td>0.3474</td>
<td>0.4189</td>
<td>0.5188</td>
<td>0.6881</td>
<td>0.9378</td>
</tr>
<tr>
<td>150x20</td>
<td>0.1304</td>
<td>0.1908</td>
<td>0.2362</td>
<td>0.2786</td>
<td>0.3252</td>
<td>0.3766</td>
<td>0.4579</td>
<td>0.5978</td>
<td>0.8893</td>
</tr>
<tr>
<td>150x30</td>
<td>0.1249</td>
<td>0.1784</td>
<td>0.2222</td>
<td>0.2612</td>
<td>0.3020</td>
<td>0.3465</td>
<td>0.4045</td>
<td>0.5129</td>
<td>0.7964</td>
</tr>
<tr>
<td>200x10</td>
<td>0.1205</td>
<td>0.1758</td>
<td>0.2209</td>
<td>0.2686</td>
<td>0.3201</td>
<td>0.3899</td>
<td>0.4917</td>
<td>0.6545</td>
<td>0.9308</td>
</tr>
<tr>
<td>200x20</td>
<td>0.1085</td>
<td>0.1539</td>
<td>0.1938</td>
<td>0.2300</td>
<td>0.2697</td>
<td>0.3143</td>
<td>0.3788</td>
<td>0.5054</td>
<td>0.8036</td>
</tr>
<tr>
<td>200x30</td>
<td>0.1099</td>
<td>0.1573</td>
<td>0.1957</td>
<td>0.2310</td>
<td>0.2657</td>
<td>0.3050</td>
<td>0.3586</td>
<td>0.4584</td>
<td>0.7320</td>
</tr>
</tbody>
</table>

### Table A.3: High accuracy ANE

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>50x10</td>
<td>0.00158</td>
<td>0.00240</td>
<td>0.00282</td>
<td>0.00349</td>
<td>0.00420</td>
<td>0.00482</td>
<td>0.00549</td>
<td>0.00835</td>
<td>0.00981</td>
</tr>
<tr>
<td>50x20</td>
<td>0.00158</td>
<td>0.00240</td>
<td>0.00282</td>
<td>0.00349</td>
<td>0.00420</td>
<td>0.00482</td>
<td>0.00549</td>
<td>0.00835</td>
<td>0.00981</td>
</tr>
<tr>
<td>50x30</td>
<td>0.00083</td>
<td>0.00123</td>
<td>0.00162</td>
<td>0.00194</td>
<td>0.00222</td>
<td>0.00255</td>
<td>0.00289</td>
<td>0.00331</td>
<td>0.00581</td>
</tr>
<tr>
<td>100x10</td>
<td>0.00084</td>
<td>0.00125</td>
<td>0.00158</td>
<td>0.00187</td>
<td>0.00218</td>
<td>0.00252</td>
<td>0.00284</td>
<td>0.00328</td>
<td>0.00589</td>
</tr>
<tr>
<td>100x20</td>
<td>0.00055</td>
<td>0.00083</td>
<td>0.00104</td>
<td>0.00126</td>
<td>0.00147</td>
<td>0.00165</td>
<td>0.00186</td>
<td>0.00211</td>
<td>0.00419</td>
</tr>
<tr>
<td>100x30</td>
<td>0.00042</td>
<td>0.00062</td>
<td>0.00079</td>
<td>0.00095</td>
<td>0.00110</td>
<td>0.00126</td>
<td>0.00141</td>
<td>0.00159</td>
<td>0.00326</td>
</tr>
<tr>
<td>150x10</td>
<td>0.00063</td>
<td>0.00092</td>
<td>0.00114</td>
<td>0.00137</td>
<td>0.00160</td>
<td>0.00184</td>
<td>0.00212</td>
<td>0.00257</td>
<td>0.00509</td>
</tr>
<tr>
<td>150x20</td>
<td>0.00063</td>
<td>0.00092</td>
<td>0.00114</td>
<td>0.00137</td>
<td>0.00160</td>
<td>0.00184</td>
<td>0.00212</td>
<td>0.00257</td>
<td>0.00509</td>
</tr>
<tr>
<td>150x30</td>
<td>0.00031</td>
<td>0.00046</td>
<td>0.00059</td>
<td>0.00071</td>
<td>0.00082</td>
<td>0.00092</td>
<td>0.00103</td>
<td>0.00116</td>
<td>0.00139</td>
</tr>
<tr>
<td>200x10</td>
<td>0.00045</td>
<td>0.00066</td>
<td>0.00082</td>
<td>0.00098</td>
<td>0.00112</td>
<td>0.00129</td>
<td>0.00150</td>
<td>0.00181</td>
<td>0.00370</td>
</tr>
<tr>
<td>200x20</td>
<td>0.00030</td>
<td>0.00043</td>
<td>0.00055</td>
<td>0.00065</td>
<td>0.00075</td>
<td>0.00085</td>
<td>0.00096</td>
<td>0.00110</td>
<td>0.00144</td>
</tr>
<tr>
<td>200x30</td>
<td>0.00023</td>
<td>0.00035</td>
<td>0.00044</td>
<td>0.00053</td>
<td>0.00061</td>
<td>0.00068</td>
<td>0.00076</td>
<td>0.00086</td>
<td>0.00105</td>
</tr>
</tbody>
</table>
### Table A.4: Simple ANE

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>50x10</td>
<td>0.00190</td>
<td>0.00285</td>
<td>0.00332</td>
<td>0.00417</td>
<td>0.00509</td>
<td>0.00616</td>
<td>0.00748</td>
<td>0.00888</td>
<td>0.00981</td>
</tr>
<tr>
<td>50x20</td>
<td>0.00129</td>
<td>0.00184</td>
<td>0.00230</td>
<td>0.00273</td>
<td>0.00320</td>
<td>0.00371</td>
<td>0.00459</td>
<td>0.00568</td>
<td>0.00692</td>
</tr>
<tr>
<td>50x30</td>
<td>0.00103</td>
<td>0.00145</td>
<td>0.00186</td>
<td>0.00221</td>
<td>0.00253</td>
<td>0.00293</td>
<td>0.00346</td>
<td>0.00442</td>
<td>0.00579</td>
</tr>
<tr>
<td>100x10</td>
<td>0.00095</td>
<td>0.00142</td>
<td>0.00179</td>
<td>0.00217</td>
<td>0.00259</td>
<td>0.00317</td>
<td>0.00385</td>
<td>0.00485</td>
<td>0.00589</td>
</tr>
<tr>
<td>100x20</td>
<td>0.00065</td>
<td>0.00096</td>
<td>0.00117</td>
<td>0.00140</td>
<td>0.00165</td>
<td>0.00191</td>
<td>0.00231</td>
<td>0.00299</td>
<td>0.00417</td>
</tr>
<tr>
<td>100x30</td>
<td>0.00051</td>
<td>0.00072</td>
<td>0.00090</td>
<td>0.00106</td>
<td>0.00123</td>
<td>0.00143</td>
<td>0.00168</td>
<td>0.00217</td>
<td>0.00316</td>
</tr>
<tr>
<td>150x10</td>
<td>0.00070</td>
<td>0.00101</td>
<td>0.00128</td>
<td>0.00156</td>
<td>0.00186</td>
<td>0.00225</td>
<td>0.00278</td>
<td>0.00369</td>
<td>0.00503</td>
</tr>
<tr>
<td>150x20</td>
<td>0.00045</td>
<td>0.00066</td>
<td>0.00082</td>
<td>0.00096</td>
<td>0.00112</td>
<td>0.00130</td>
<td>0.00158</td>
<td>0.00207</td>
<td>0.00308</td>
</tr>
<tr>
<td>150x30</td>
<td>0.00037</td>
<td>0.00053</td>
<td>0.00066</td>
<td>0.00077</td>
<td>0.00089</td>
<td>0.00102</td>
<td>0.00120</td>
<td>0.00152</td>
<td>0.00236</td>
</tr>
<tr>
<td>200x10</td>
<td>0.00050</td>
<td>0.00074</td>
<td>0.00092</td>
<td>0.00113</td>
<td>0.00134</td>
<td>0.00164</td>
<td>0.00206</td>
<td>0.00275</td>
<td>0.00391</td>
</tr>
<tr>
<td>200x20</td>
<td>0.00034</td>
<td>0.00048</td>
<td>0.00060</td>
<td>0.00072</td>
<td>0.00084</td>
<td>0.00098</td>
<td>0.00119</td>
<td>0.00158</td>
<td>0.00252</td>
</tr>
<tr>
<td>200x30</td>
<td>0.00027</td>
<td>0.00039</td>
<td>0.00049</td>
<td>0.00057</td>
<td>0.00066</td>
<td>0.00076</td>
<td>0.00089</td>
<td>0.00114</td>
<td>0.00183</td>
</tr>
</tbody>
</table>
Appendix B

Matlab Code of simple low rank tensor completion

Matlab code of simple low rank tensor completion

```matlab
function [X] = SIMPLE(T, Omega, alpha, beta, maxIter, threshold, X)

normT = norm(T(:));
dim = size(T);
M = cell(ndims(T), 1);
betasum = sum(beta);
tau = alpha./ beta;

for k = 1:maxIter

    Xsum = 0;
    for i = 1:ndims(T)
        M_unfold=Unfold(X, dim, i)
        M_trace=TraceNorm(M_unfold, tau(i))
        M{i} = Fold(M_trace, dim, i);
        Xsum = Xsum + beta(i) * M{i};
    end
    Xlast = X;
    X = Xsum / betasum;
    X(Omega) = T(Omega);
    errVal = norm(X(:)-Xlast(:)) / normT;
    if (errVal < threshold)
        break;
    end
end
```

43
Appendix C

Matlab Code of high accuracy low rank tensor completion

Matlab code of high accuracy low rank tensor completion

function [X] = HighAccuracy(T, Omega, alpha, rho, thresholds, maxIter, X)

dim = size(T);
Y = cell(ndims(T), 1);
M = Y;
normT = norm(T(:));
for i = 1:ndims(T)
    Y(i) = X;
    M(i) = zeros(dim);
end
Msum = zeros(dim);
Ysum = zeros(dim);

for k = 1: maxIter
    rho = rho * 1.03;
    % update Y
    Msum = 0*Msum;
    Ysum = 0*Ysum;
    for i = 1:ndims(T)
        M_unfold = Unfold(X - M(i)/beta, dim, i);
        M_trace = TraceNorm(M_unfold, alpha(i)/rho);
        Y(i) = Fold(M_trace, dim, i);
        Msum = Msum + M(i);
        Ysum = Ysum + Y(i);
    end
    % update X
    lastX = X;
    X = (Msum + rho*Ysum) / (ndims(T)*rho);
    X(Omega) = T(Omega);
end
% update M
for i = 1:ndims(T)
    M(i) = M(i) + rho*(Y{i} - X);
end

errVal = norm(X(:)-lastX(:)) / normT;
if errVal < threshold
    break;
end
References


