A Report of Matrix Completion

Yiming Zhao

June 2023

1 Introduction of Matrix Completion

This section is a brief introduction of existing completion theory and algorithms.

1.1 Matrix Completion

Matrix completion is the task of filling in the missing entries of a partially observed matrix, which is equivalent to performing data imputation in statistics. A wide range of datasets are naturally organized in matrix form. One example is the movie-ratings matrix, as appears in the Netflix problem: Given a ratings matrix in which each entry (i,j) represents the rating of movie j by customer i, if customer i has watched movie j and is otherwise missing, we would like to predict the remaining entries in order to make good recommendations to customers on what to watch next. Another example is the document-term matrix: The frequencies of words used in a collection of documents can be represented as a matrix, where each entry corresponds to the number of times the associated term appears in the indicated document.

Without any restrictions on the number of degrees of freedom in the completed matrix this problem is underdetermined since the hidden entries could be assigned arbitrary values. Thus we require **some assumption on**

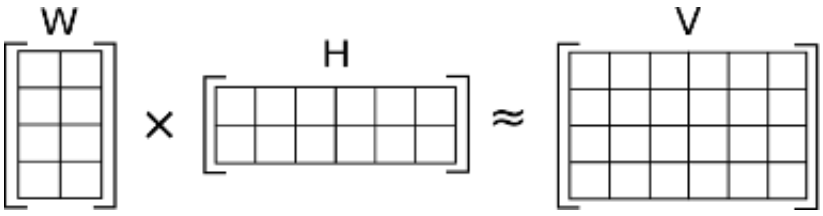

Figure 1: Matrix Completion Sample

the matrix to create a well-posed problem, such as assuming it has maximal determinant, is positive definite, or is low-rank.

For example, one may assume the matrix has low-rank structure, and then seek to find the lowest rank matrix or, if the rank of the completed matrix is known, a matrix of rank r that matches the known entries. The illustration shows that a partially revealed rank-1 matrix (on the left) can be completed with zero-error (on the right) since all the rows with missing entries should be the same as the third row. In the case of the Netflix problem the ratings matrix is expected to be low-rank since user preferences can often be described by a few factors, such as the movie genre and time of release. Other applications include computer vision, where missing pixels in images need to be reconstructed, detecting the global positioning of sensors in a network from partial distance information, and multiclass learning. The matrix completion problem is in general NP-hard, but under **additional assumptions** there are efficient algorithms that achieve exact reconstruction with high probability.

In statistical learning point of view, the matrix completion problem is an application of matrix regularization which is a generalization of vector regularization. For example, in the low-rank matrix completion problem one may apply the regularization penalty taking the form of a nuclear norm $R(X) = \lambda \|X\|_*$

1.2 Low Rank

The most common and well developed field is to find the lowest matrix X that matches sampled matrix M, which we wish to recover. An equivalent formulation, given that the matrix M to be recovered is known to be of rank r, is to solve for X where $X_{ij} = M_{ij} \ \ \forall i, j \in E$.

1.2.1 Assumptions

1. Uniform Sampling of Observed Entries. Sampled subset E should be constructed by Bernoulli sampling.

2. Lower bound on number of observed entries. Suppose the m by n matrix M we are trying to recover has rank r.There is an information theoretic lower bound on how many entries must be observed before M can be uniquely reconstructed. The set of m by n matrices with rank less than or equal to r is an algebraic variety in $\mathbb{C}^{m \times n}$ with dimension $(n + m)r - r^2$. Using this result, one can show that at least

$4nr - 4r^2$ entries must be observed for matrix completion in $\mathbb{C}^{n\times n}$ to have a unique solution when $r \leq n/2$.

3. Incoherence.The concept of incoherence arose in compressed sensing. It is introduced in the context of matrix completion to ensure the singular vectors of M are not too "sparse" in the sense that all coordinates of each singular vector are of comparable magnitude instead of just a few coordinates having significantly larger magnitudes.

1.3 Tensor Completion

The difference of tensor and matrix is the number of dimensions. The algorithms only expand the dimensions, the theory changes a little.

2 How to

Find out possible solution to control the scanning machine by direct location parameters might be the Ariadne's thread of this project. The completion theory is well developed and applications in multiple fields are persuasive to prove the effectiveness of the theory.

To implement the scanning, the first thing we need to do is shuffling the order of scanning locations. The second step is chose a number between 0 and 1, alpha, let the first alpha locations in the shuffled set be the scanning locations. Then we can push these locations to the control program and wait for the scanning data. I can't predict what error will occur in this way, but it follow the basic rule of completion theory. To scan in some order is another new story, like bi-linear or other algorithms.

3 Code Description

3.1 Before Running

Modify your input image path in the 2nd line of function init().

3.2 Running

Run HaLRTC.py by using python. Then you can find a method table printed on the screen.

1. 1 represents for total random. You can control the percentage of known pixels by change the 1st line in funtion init(), the parameter is called KownPercentage.

2. 2 stands for regular interval with certain steps in rows. You need to input the steps as you wish after you choose 2.

3. 3 stands for rays but it seems that left some bugs in the previous code. Because of the low percentage of known pixel by this method and this method is only works in random as mentioned above, I believe there is no need to fix the bug.

4. 4, in the last, will lead to random n pixels in m x m square. You can input them after you select this method.

3.3 After running

Your output will be saved in output folder.