2021

Predicting Stock Market Indices Using LRMC

Pragyat Agrawal
University of Pennsylvania

Predicting Stock Market Indices Using LRMC

Abstract

Through this paper we have been able to study several Low Rank Matrix Techniques, looking especially at their applications to the stock market space and how efficiently they can be used to predict stock indices. Several techniques have been used as well as key data used in previous studies to compare different approaches in this review paper. Some of these approaches include using Singular Value Decomposition techniques, analyzing its usage as per the different sources that can be used to make it more efficient. We also include the Low Rank Matrix by ASD (Alternating Steepest Descent) method to analyze how the efficient image inpainting algorithm may be applied to the stock markets. These approaches, after being analyzed on runtime, computational space and the availability of data amongst many factors, indicate to us that the Singular Value Decomposition using the Order Method is the most efficient method amongst all those methods analyzed.

Keywords

Low rank matrix completion, singular value decomposition

Disciplines

Applied Mathematics | Business | Business Analytics | Corporate Finance

SPUR - Low Rank Matrix Completion

Pragyat Agrawal

September 2021

1 Summary

Through this paper we have been able to study several Low Rank Matrix Techniques, looking especially at their applications to the stock market space and how efficiently they can be used to predict stock indices. Several techniques have been used as well as key data used in previous studies to compare different approaches in this review paper. Some of these approaches include using Singular Value Decomposition techniques, analyzing its usage as per the different sources that can be used to make it more efficient. We also include the Low Rank Matrix by ASD (Alternating Steepest Descent) method to analyze how the efficient image inpainting algorithm may be applied to the stock markets. These approaches, after being analyzed on runtime, computational space and the availability of data amongst many factors, indicate to us that the Singular Value Decomposition using the Order Method is the most efficient method amongst all those methods analyzed.

2 Introduction

In today's world we have been able to come across a lot information in an efficient manner for any event that we are concerned with. The event may come from different sources and may record different times at which the event took place. However, in many real world applications of such information, we have also been facing the problem of coming across incomplete information as well which jeopardizes the entire process of drawing knowledge from the information. This is why Low Rank Matrix Completion techniques have been viewed as the next big step in the paradigm of recovering unknown entries of a matrix from partial observations, generating a great deal of interest over the years. Though most of the studies have been in a theoretical nature, some studies have been done to expand upon this technique in a contemporary manner. One of these have been used in Netflix show recommendation system where the Netflix system, based on the aggregate reviews given to a particular show, and a user's particular choice decides to suggest the next shows to the user which he/she may prefer. LRMC techniques are also used heavily in removing dirt from images and video deraining, by which corrupted images can be taken out in an easy manner. It does so by decomposing the pixels in the image to basis vectors based on the sparse representation. It further classifies those vectors into actual or dirt pixels, hence helping in recovering an image which is free of dirt.

One important aspect about the low-rank matrix is that the essential information (expressed in terms of degree of freedom) in a matrix is smaller than the total number of entries hence giving us a pretty good chance to recover the whole matrix. This situation, where the unknown is a huge factor times greater than the known, is much common that it seems. This is seen in a variety of scenarios like in recommendation systems, where users do not want to leave a rating, hence leading to missing entries in the rating matrix. We use Nuclear Norm Minimization techniques along with algorithms to solve these Low Rank Matrices. One thing that needs to be touched upon is the idea of randomness i.e. the set of missing entries is random. This assumption of randomness (that it is uniform across the entire matrix), helps researchers prove several theorems though it may not be always true. Our approach would only look at deterministic methods here. We also need to expand upon the idea of independence - without assumptions about the level of independence we have on the matrix, we can't determine the observations. What if all the values are independent? In this case it is impossible to reveal the unknown matrix. This is the reason we assume that the matrix has a low rank - which makes entries dependent on each other and hence capable of being calculated. One interesting question we might get - why only a Low Rank Structure and not a structure where the entries are dependent of each other without this Low Rank Structure? The answer is that LRMC techniques are used for events which have a strong dependence between event attributes which in turn leads to a Low Rank Matrix structure for the matrices.

The basic idea which differentiates normal matrices with low rank matrices is the idea of rank. Rank, using linear algebra, refers to the number of independent vectors contained within the rows of a matrix. The idea of having a low rank means that a matrix's unknown values could be found by creating a simple linear relationship between the vectors in the matrix or even converting the matrix to the row reduced echelon form. The structure of the matrix is also of high importance. Usually the rows and columns are chosen in such a way that the data becomes dependent on each other. For example: we can take the Low Rank Matrix of a hypothetical economic crisis which affects the entire world. Over here our event matrix could be X i.e. some economic indicator of dimensions $R \times C$ where R is the number of rows, where each row is represented by one country and C is the number of columns, where each column is represented by a specific time during the economic crisis. Both of these combine to give us the value of the indicator at a particular time for a particular country making our work a little convenient.

These Low Rank Matrix techniques have been used in several places like image compression and restoration where there is dirt or scribble in a two dimensional image which can be cleared through using LRMC techniques. The problem of recovering a low-rank matrix is a fundamental problem with applications in machine learning as well.

How do we solve Low Rank Matrices? A naive way to solve Low Rank Matrices would be to use the combinatorial search technique. This means fixing the rank of the matrix to 1 and then solving for other unknowns by doing $m_i = \alpha m_j$ where m_i and m_j represent two separate rows/vectors in the matrix. If no solution could be found, then we increase the rank by one to 2 and re-do the process until we find the solution. This is a very naive approach as the system would take a very high time to compute with complexity of $O(n2^n)$. A better way to solve this problem would be to convert this combinatorial search problem into a convex optimization problem where a local optimum solution is globally optimum and there are many efficient polynomial time convex optimization solvers.

Sometimes a low rank matrix can be recovered using lesser entries than the matrix actually has. A natural question arises over here, how many entries? This question can be answered by the notion of Degree of Freedom (DOF). The DOF of a matrix is the number of freely chosen variables in the matrix. This also implies that if the number of observed entries is smaller than the degree of freedom then no algorithm can recover the entries of the matrix. The formula for the degree of freedom a $n_1 \times n_2$ is given by $(n_1 + n_2)r - r^2$. One more thing to note is coherence i.e. the density of the known entries. If the known entries are concentrated in a specific area in a large matrix then we need more entries whereas if we were given a matrix with widely spread entries which means we would need lesser known entries to find the entire matrix. This means if we are given an n by n matrix, and if we know the top $\frac{n}{2}$ entries in the matrix, then it doesn't mean we will be able to decipher all entries of the matrix. We should note that the matrix should remain dense in any form of re-labelling of the matrix o.w. we could re-label the matrix and make it very dense in one area of the matrix and sparse in the other. Also given the fact that there are multiple Low Rank Matrix algorithms, the natural question arises which algorithm should we use. The answer is simple: the one which has the highest accuracy and low computational complexity. These two factors are both important, but their weightage depends on the size of the matrix with larger matrices probably going well with algorithms which have a low computational complexity while accuracy might be the better reason behind smaller matrices.

A lot of studies have been conducted to retrieve the values of low rank matrices using stable and non-stable arguments. What do these mean? Stability means that we approximate the values in the matrices (close to the actual ones) while non-stable retrieval implies retrieving exact values. After some studies (A Deterministic Theory of Low Rank Matrix Completion - Sourav Chatterjee) it has been established that stable recovery is impossible from a small set of revealed entries if there is no assumption of randomness.

One open question we get over here is: What if we drop the condition of stability and want the exact values? How will our approach and consequently, our theory change?

One more question we get is: to properly understand the level of sparsity allowable for a matrix of a given size, one needs a non-asymptotic result.

The problem on matrix completion that we face comes under the category of NP-hard optimization problem, due to the non-convexity (meaning the non - linear nature and more possible set of solutions for the problem) and combinatorial nature of the rank function. Many existing algorithms rely on the convex relaxation (this means that the solution revolves around taking a non-convex problem that we want to solve, and replacing it with a convex problem which we can actually solve — the solution to the convex problem gives information about (usually a lower bound) the solution to the original program.). Nevertheless, these involve a high computational cost and memory space, making them unsuitable for handling large size problems.

One open question we get over here if we could find a convex relaxation solution for our LRMC problems which can find a solution within our complexity and space restrictions.

One more way we have to solve a Low Rank Matrix, other than convex optimization, is to apply matrix factorization methods to approximate the matrix as close to the actual matrix. Hence our main matrix can be represented as a product of two low-rank matrices. This approach is fast but these are effective only for *easy* problems and *not hard* problems. The weakness in this approach mainly comes from the fact that we need a good approximation of the unknown rank of the original matrix.

We have also been presented with Least Mean Squares (LSM) methods which can help do low rank matrix completion. These were first introduced by Widrow and Hoff. Subsequently, these LSM methods have been applied to a plethora of areas which include noise cancellation, adaptive control, equalization etc.

There have been many algorithms suggested for completing low rank matrices. We won't go in the mathematical foundations of these algorithms but some of them include The Least Mean Squares Algorithm, Adaptive Singular Value Thresholding, Singular Value Regularized LMS, Proximal LMS etc. and it has been seen that in worse case instances - where as less as 30% entries are known, the maximum relative error any of these algorithms reach is 0.30% with some algorithms reducing it to 0.05% when given sufficient running time (like the SVR - LMS algorithm).

For applications of the LRMC to the realm of finance, it will be naive to consider that the variables have no correlation to each other. Hence the sparsity property (the property which states that many entries in our Low Rank Matrix are 0 or near 0) is not applicable over here. For example, financial returns depend on common equity market risks, housing prices depend on the general level of economic health etc. Because of the presence of common factors, we can't assume that a lot of these factors are uncorrelated. In the case of financial studies of an asset's returns, if we study assets over a long period of time, their dynamics can vary substantially. Therefore, to capture the current market condition, financial analysts may wish to use a short time horizon to calculate all the risk factors and hence the low rank matrix.

3 Proposal

There is a huge unclarity about the stock market being unpredictable in an exact sense. We propose otherwise. Our proposal would seek to find the price movement of stocks using Web news and social media postings. Along with that we would also seek to predict the stock using the logarithmic differences in daily return values. At the end, we would be comparing different techniques as well as different sources we can use to predict stocks for ourselves. We would also end up calculating the various correlations between different stocks as well as propose a self learning system which can find which stocks to use on the basis of the degree of correlation between them.

4 Applications to Stocks

4.1 Background

Financial market plays an important role in facilitating the growth of an economy. So the operational state of a financial market often attracts the most attention of both market participants as well as policy makers. Among them, stock market is of sovereign and utmost importance and is considered to be a clear indicator of one country's economic power and advancement. However, some well-known financial crises and market crashes in the history have put the stock market at the forefront of suspicion as people have started questioning it on the principle of traditional financial theory.

In Financial Markets, the efficient market hypothesis (EMH) was proposed by Fama which occupies a central place and has a huge interest amongst the scientific community. The market efficiency, according to this, can be classified into 3 categories of efficiency: weak form of efficiency, semi-strong form of efficiency and strong form of efficiency. A market is said to be weak if the current market price has fully reflected history information; in this case no amount of technical/mathematical analysis would be useful. The semi-strong state says that the current price has fully reflected all publicly available information and renders fundamental analysis worthless. In strong form of efficiency, current price is also reflected, which reflects all the inside information and no one can make excess profit in the market. EMH states that it is impossible to predict the market principally because market efficiency causes existing share prices to fully incorporate and reflect all relevant information.

When talking about stocks, the mean-variance (MV) model by Markowitz, which selects portfolios by minimizing the risk (measured by variance) under constraints on the expected return (measured by mean), laid the foundation of modern financial investment theory. This MV model has been extended and modified in numerous researches but the MV model hasn't been used to construct a large scale portfolio. This is because of computational difficulty in solving a large quadratic system like this with a very dense covariance matrix. With time though, there have been seen new algorithms which can potentially solve this problem in an efficient way (by efficient we mean algorithms which are stable and can be used for the entire stock market at a time with a computation time of around one minute).

One algorithm used by Sharpe and Rosenberg considers using multi-factor model to estimate the return vector $c_0 = Uf + d$ where $c_0 \in R^n$ is our required return vector of n stocks, $f \in R^n$ is the vector of return of r factors, $U \in R^{n \times r}$ is the factor loading matrix, $d \in R^n$ is a random vector whose elements are specific to individual stocks and are assumed to be uncorrelated with each other and with the factors. Takehara used this special structure of c_0 which is the summation of a diagonal matrix and a low rank modification, to design an algorithm for the large scale portfolio optimization problem.

4.2 Singular Value Decomposition

Singular Value decomposition has been a very innovative technique used for the calculation for the Low Rank Matrices. Their use is not only limited to matrices but other applications as well which include calculating the pseudoinverse and homogeneous linear equations. With the presence of a lot of academic material in this area, we have been able to enhance the technique for our use in the LRMC process with several algorithms being proposed to calculate the Singular Value Threshold.

One of the earliest algorithms was presented by Frieze which involves independently sampling s rows and columns from A to from an $s \times s$ matrix S. They showed that when the columns and rows are sampled with probability proportional to the column and row norms respectively, and the sample size s, the approximation can be bounded with a high probability. Aclipotas obtained a similar result by designing a random matrix R with entries entirely dependant on other values in matrix A, such that matrix $A + R$ is sparse and $E[R] = 0$ (Kernel Based Clustering of Big Data).

This approach was further enhanced by nullifying the entries which have sufficiently low magnitudes. This is an innovative way which we may also use in calculating our stock returns.

4.3 Approach 1: Media and Web Pages

One way we can predict stocks is through the use of social media. In this, we extract the events from Web news and the users' sentiments from social media and investigate their impacts on the stock price movements via a coupled matrix and tensor factorization matrix. We will use the same logic with Low Rank Matrices that we discussed earlier, using highly correlated stock simultaneously through their commonalities, which are enabled via sharing the collaboratively factorized low rank matrices between matrices and the tensor (which are n dimensional matrices). Web information though very extensive usually extracts in events which usually reside in unstructured texts that are difficult to extract.

Also a lot of information i.e. presented on the web can be presented in a different manner on different websites. Another issue we face that is even if we have successfully extracted the information on the events of a stock, it is still difficult to determine whether the event will cause a positive or negative impact on the stock value. Eg: Microsoft acquisition of LinkedIn leads to fall in Microsoft's stock price, while Intel acquiring Altera leads to a rise in its stock price. In addition to the events, emotions also play a significant role in decision making. Previous studies in behavioral economics show that financial decisions are significantly driven by emotion and mood. However, relying on the sentiments alone is not sufficient for prediction either. For example, in holidays, people's mood tends to be positive yet it may not really reflect their investment opinions.

This method is not new. Various studies have been done on this type of methods with people using various tools to analyse the companies. Some include using a deep neural model to measure the content of financial news. This method has its own drawbacks. These works only consider the impact of news events but fail to consider the moods of investors that may lead to fluctuations in stock prices. One more method is the stock sentiment analysis method where the analysis is solely based on news about the company. It's such that, if there is a good news, it would be advisable to buy the stock whereas if there's any bad news, it would be advisable to sell the stock. Another method encompasses the use of social media especially platforms like Twitter where each of the users tweets can be tagged as one of many emotional indicators and that can be used to predict the stock market. This technique is accurate as it shows that the ratio of emotional tweets present a negative correlation with the NASDAQ. The drawback of a lot of these models is that they don't consider events when looking up stocks. Hence a methodology may be the use of tensors where we use the sentiments as one dimension and events as the other dimension. This Tensor method, however, still has its drawbacks as it forces us to use a parameter heavy approach (as tensors tend to be dense). There is also the issue of data sparsity as we move over to more minute intervals while calculating the data like from weeks to days or from days to hours. Lastly, tensors don't take into consideration the relations with stocks, making the algorithmic approach a little uncertain.

One way we can substantially reduce the drawbacks in the tensor method is by effectively integrating it with historical quantitative (along with using news events and sentiments, naturally), to achieve a higher quality of output i.e. a better stock prediction.

4.3.1 Steps for Media and Web Pages:

Before writing the steps behind our calculation, we should write about the attributes inside the tensor. We would be considering inter-coupled similarities between different attributes along with the usual Inter-coupled similarities. Intra-coupled similarities for example can reveal the similarities in various values and intra-coupled similarities can reveal a relation between two or more attributes.

We will be using a 3rd order tensor for this study.

Step 1: First take out the Inter-Coupled Value similarity i.e. δ which is the aggregate relative similarity for all the other attributes excluding itself. The simple formula for the Inter-Coupled Value similarity of two random values x and y would be (Improving Stock Market Prediction via Heterogeneous Fusion)

$$\delta_j(x,y) = \Sigma_{k=1,k\neq j}^{n} \alpha_k * \delta_{j|k}(x,y)$$

Here α_k is nothing but the weight parameter for attribute j

Step 2: Then we choose the four most quantitative features, that is the price to earning ratio, the price to book ratio, the price to cash flow ratio and the share turnover. These will constitute the historical part of the data we want to input in our Low Rank Matrices. Share turnover is calculated by dividing the sum total of shares traded over a particular period by the average shares outstanding for that same period. These 4 ratios are chosen as they are the most quantitative and hence the most representative of all. The best of all these 4 ratios would be the P/E ratio as the P/E ratio tells us the amount of earnings the investors can expect per dollar they put in the company.

These indicators will be used after we factorize the matrix. Each of these stocks will be represented by a vector.

Step 3: In stock market, stocks are usually not independent of each other and can have correlations from various perspectives. This correlation could be easily used in Low Rank Matrices as mentioned before and hence we will be building a stock correlation matrix. We do this by calculating the following: (Approach given in Improving Stock Market via Heterogeneous Information Fusion)

Stock Correlation $CSS(i,j) = \Sigma_k \delta_k^{I_a}(V_{ik}, V_{jk}) * \delta_k^{I_e}(V_{ik}, V_{jk})$

Here V_{ik} and V_{jk} are the values of feature k on stocks i and j. δ_a^I represents the intra-coupled attribute values while δ_a^I is the inter-coupled attribute values.

One more tool we use here is the theory of co-evolving stocks. This means that if stocks' closing prices go up (or go down) at the same closing day, this means that the stocks are co-evolving. This also means that, the more days the two stocks co-evolve, the higher degree to which they are correlated. We will use the ratio of the days the two stocks co-evolve to the total number of trading days there are to calculate the stock correlation matrix.

The last tool we will use is the Perceived Correlation Index. We will be extracting user perceivedness using social media platforms like twitter. These perceptions will be about correlations between stocks as stocks which are talked about together on twitter show a higher degree of correlation. We will ignore the tweet if more than 5 stocks are mentioned on it.

Step 4: Now, having found the correlation coefficients of each pair of stock, we will normalize them and fill them in the stock correlation matrix. Here higher values at positions i and j indicate a higher correlation between stocks i and j.

Step 5: Now we will fill in the tensor with dimensions for the stock quantitative matrix being $N \times K$ i.e. Stock ID times the Stock Quantitative Features. The Stock Correlation matrix would be $N \times N$. The stock movement tensor would finally be of dimension $N \times M \times L$ where M is the number of type of events.

Step 6: The final step would involve doing a coupled matrix and tensor factorization. This would help us complete our problem and would return the 4 factored low rank matrices, denoting the low rank latent factors for stocks, events and sentiments with the final one representing the combination of stocks and quantitative features i.e. the quantitative matrix.

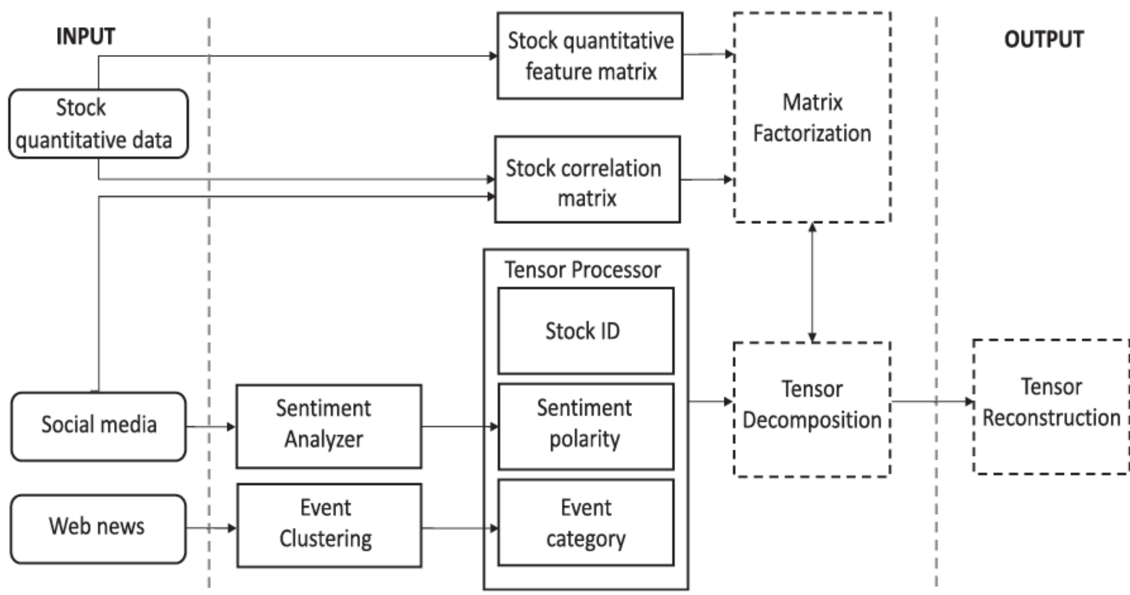

Figure 1: System framework that we will follow

4.4 Approach Two: SVD Method

Step 1: For using this approach we first have to understand trajectory matrix. Trajectory matrix can be defined by means of window-sliding and time-lagging techniques. Time-lagging technique involves looking back at the time series and sampling every observation we see. Window sliding is a common mathematical technique which involves looking at a fixed number of observations at every single turn. Hence, it provides a condition under which a smooth reconstruction attractor can retain the original topological characteristics of the entire system. (Using Rényi parameter to improve the predictive power of singular value decomposition entropy on stock market)

$$A_{n\times s} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_3 & \cdots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n+1} & \cdots & x_N \end{bmatrix}.$$

Figure 2: How a Trajectory Matrix with $(x_1, x_2, ..., x_N)$ elements looks like

Step 2: Once the trajectory matrix is completed we can decompose it using singular value decomposition. The singular values are small and non - negative.

$$A_{n\times s} = U\Lambda V' = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nk} \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_k \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1s} \\ v_{21} & v_{22} & \cdots & v_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{ks} \end{bmatrix}$$

Figure 3: How Decomposition would take place - here A is the decomposed matrix

Here U and K (using Rényi's approach) are orthonormal matrices, which finally gives us A which has singular values.

Step 3: We will then make a probability space by normalizing the singular values. Then we can derive a measure of entropy with order q given by:

$Ent(q) = \frac{1}{1-q} log \Sigma_i \bar{\lambda}_i^{\,q}, q \neq 1$

Here Λ are the singular values we were talking about and q is the weight of different probabilities. The trajectory matrix can be constructed using the form $B = A + E$ where A is a trajectory matrix and E is the noise and errors. We assume the value of E to be very small or even insignificant.

Step 4: Choose a stock index or a stock index which encompasses the entire market. Some examples may include the Dow Jones Industrial Average and the S&P 500. We can then calculate the daily return as an absolute difference or a logarithmic difference. The daily logarithmic differences are presented below.

One interesting we note is finding the dimension of the matrix series. The lower limit can be obtained by calculating $2 * d + 1$ where d is the fractal dimension of the original series.

Step 5: The final step involves testing the robustness of the results using a myriad of different tests. These include the Static Test and Dynamic Test methods.

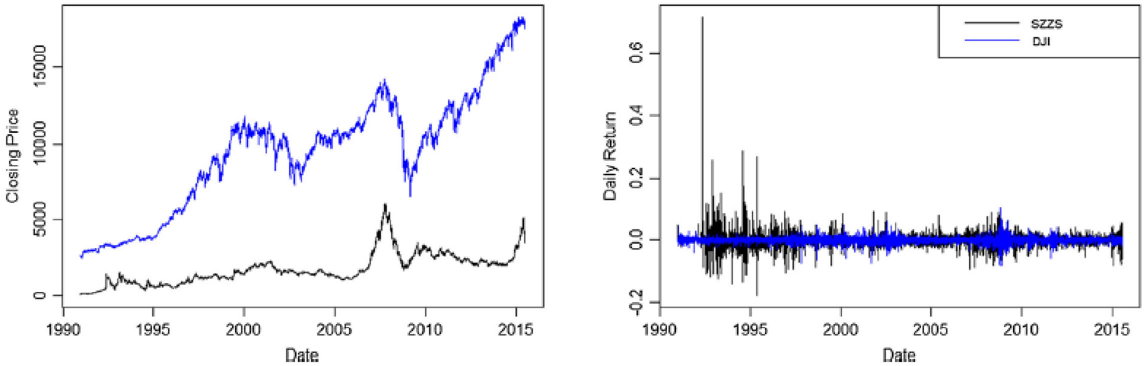

Figure 4: Calculating the daily logarithmic differences for Dow Jones and Shanghai Stock Exchange [2.]

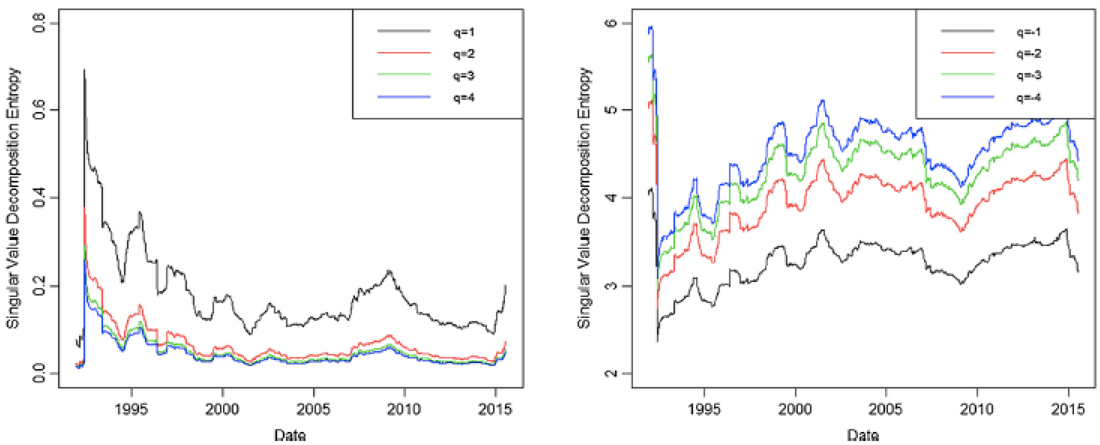

Figure 5: Entropy of SCI with varying orders [2.]

Using the static tests the singular value decomposition entropy shows a very low predicting power for the DJI series. Only when q equals -1 and -2 can entropy predict the DJI series but the significant levels are between 5% and 10% which is still very low. This is quiet understandable as the American stock market is much more older and hence mature than the Chinese stock market. It is very hard to predict the micro (small scale) trends in a developed stock market. Hence an economic model which varies according to time may be more suited for the Dow Jones Index Case.

Using the dynamic test we can try and test the investigative power of the Singular Value Decomposition method by varying various values of q. The graph for the same is above. When using the traditional method of Shannon entropy, we see that in the whole duration the market is only predictable only 6 out of 23 years in Chinese Market, and 11 out of 23 years in American Stock Market under 10% confidence interval after doing a year-by-year analysis. We also looked at the predictability by increasing the range of values for q, meaning we increased the range from -10 to 10 (assuming that different q values would lead to more accurate results). It becomes clear that the predictive power can certainly be increased by increasing the range of q values as the number of predictive years rise for the Chinese Stock Market, however no change is seen for the DJI. This is because increasing the number of orders affects the noise level i.e. it reduces it. Hence this indicates that the Chinese Stock market is more affected by noise.

The Shannon Entropy has value of $q = 1$ which is very less predictive for SCI and DJI. Hence we are able to conclude that the orders method produces more accurate results than the traditional Shannon Entropy method.

4.5 Approach Three: Low Rank Matrix by Alternating Steepest Descent (ASD) method

This approach is particularly interesting as it used for various applications especially because of its flexibility and low memory usage. Empirical evaluation of ASD and Scaled ASD on both image inpainting (Image Inpainting is a task of reconstructing missing regions in an image. It is an important problem in computer vision and an essential functionality in many imaging and graphics applications, e.g. object removal, image restoration, manipulation, re-targeting, compositing, and image-based rendering) and random problems show they are competitive with other state-of-the-art matrix completion algorithms in terms of recoverable rank and overall computational time. In particular, their low per iteration computational complexity makes ASD and Scaled ASD efficient for large size problems, especially when computing the solutions to moderate accuracy such as in the presence of model misfit, noise, and/or as an initialization strategy for higher order methods.

However, ASD doesn't have enough complexity or capacity to take several factors and indices which the stock market presents, hence it may not be the best choice for predicting the stock market.

5 Result

Of the many LRMC approaches researched, Singular Value Decomposition using the Order Method turned out to be the most efficient, predicting stocks to a very high accuracy, especially when we use higher orders. This method has further scope of improvements though, especially in case of mature markets where this method sees a considerable reduction in accuracy. This can most probably be

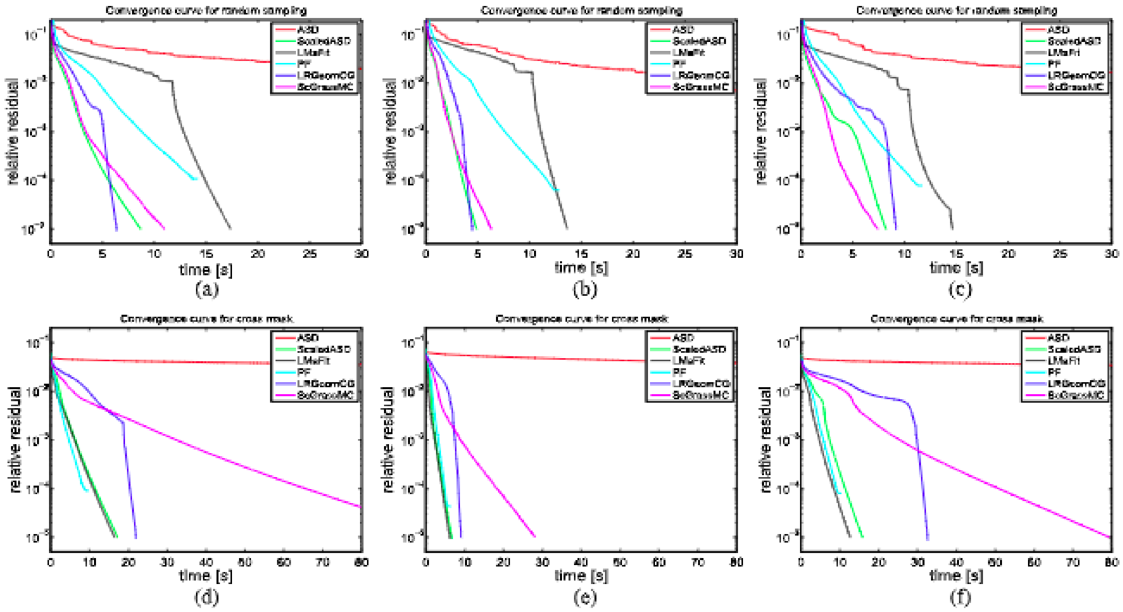

Figure 6: ASD efficiency in image inpainting

solved by making improvements in SVD algorithms especially in the case of runtime methods.

6 Sources

1. A Deterministic Theory of Low Rank Matrix Completion (Sourav Chatterjee)

2. Using Rényi parameter to improve the predictive power of singular value decomposition entropy on stock market (Jiaqi Jiang & Rongbao Gu)

3. Fast algorithms for sparse portfolio selection considering industries and investment styles (Zhi-Long Dong & Fengmin Xu & Yu-Hong Dai)

4. Adaptive Low-Rank Matrix Completion (Ruchi Tripathi, Boda Mohan, and Ketan Rajawat)

5. Low-Rank Matrix Completion: A Contemporary Survey (Luong Trung Nguyen , Junhan Kim , and Byonghyo Shim)

6. Low rank matrix completion by alternating steepest descent methods (Jared Tanner & Ke Wei)