A Time-varying Partial Correlation Network Analysis of Price Change in Intraday Stock Market

Shanshan Li

Department of Applied Mathematics and Statistics

Stony Brook University, NY, 11794-3600

SUM UP

Keywords: Partial correlation; Network analysis; Market stability; Market turbulence; Localized smoother

1 Introduction

Nowadays, giant datasets are collected with lots of empirical information about the functioning of almost every field of study, at a cost much lower than a few decades ago, for instance biotechnology(McBride 2012), medical science(Groves 2013), and in particular business and economics study(Einav and Levin 2013). One can be interested in the existing linkages between the different elements that included in a collection of the dataset. And that is the reason why Network Analysis has emerged in recent years. Network Analysis is used to help interpret the hidden interconnections between different elements in large dataset. With the application of proper statistical tools, analysts can not only get the statistical results about the data, but also plot the interconnections of large multivariate time series system in a graphical representation that eases the interpretation of the real market observation. That is to say, Network Analysis allow us to construct graphs representing the reality behind those complex empirical datasets.

Stock markets behave as complex dynamic systems, and as such, it is critical to investigate the dependencies (interactions) between the dynamics of the system variables (stocks, bonds, etc.). It is common to associate such interactions with the notion of correlation, or similarity. Indeed, much effort is dedicated to study and understand such stock cross-correlations in an attempt to extract maximum market latent information that is embedded in the interactions between the market variables.

One prominent feature in financial markets is the presence of an observed correlation (positiveor negative) between the price movements of different financial assets. The presence of a high degree of cross-correlation between the synchronous time evolution of a set of equity returns is a well known empirical fact (Markowitz 1952, Eltonet al.2009, Campbellet al.1997). The Pearson correlation coefficient (Pearson 1895) provides information about the similarity in the price change behavior of a given pair of stocks. Much effort has been devoted to extracting meaningful information from the observed correlations in order to gain insights into the underlying structure and dynamics of financial markets (Embrechtset al.2002, Morcket al.2000,Campbellet al.2008, Krishanet al.2009, Asteet al.2010, Campbellet al.2008, Cizeauet al.2001, Lalouxet al.2000, 1999, Plerouet al.1999, Podobniket al.2009, Pollet and Wilson 2010,Tumminelloet al.2010, Huanget al.2013, Forbes and Rigobon 2002).

Despite the meaningful information provided by investigating the correlation coefficient, it lacks the capacity to provide information about whether a different stock(s) eventually controls the observed relationship between other stocks. To overcome this issue we introduce the use of the partial correlation coefficient (Baba et al.2004), and its applications.

The presence of significant cross-correlations between the synchronous time evolution of a pair of equity returns is a well-known empirical fact. The Pearson correlation is commonly used to indicate the level of similarity in the price changes for a given pair of stocks, but it does not measure whether other stocks influence the relationship between them. To explore the influence of a third stock on the relationship between two stocks, we use a partial correlation measurement to determine the underlying relationships between financial assets.

1.1 Network Analysis

In mathematics, the traditional way of representing networks is using graphs, which can be generally defined as a collection of nodes connected by lines. Here we will introduce a bit of graph theory. A graph is an ordered pair as following:

$$G = (V; \epsilon) \tag{1}$$

The first one represents nodes while the second one is edges connecting nodes. We consider two main characteristics of a network: directionality and weight. First, If an edge from node i to node j is different from an edge from j to i, then the graph is directed. On the other hand, if all the links between nodes don't have a particular direction, then the graph is undirected. Next, The difference between weighted and unweighted networks has to do with the relative weight of each edge. In weighted networks, the thickness of the edge depends on the intensity of the correlation between two nodes. Due to the specific resources and purpose of our study, we use undirected and weighted graphs.

As briefly introduced in the above section, the final aim of network analysis is to represent large data collection as a network with which would be easy to interpret the linkages between different elements. In this article, our data collection is a multivariate time series. That is to say, we will apply network analysis using values of a determined number of variables taken in successive periods of time. In our study, the stock prices and trading volumes at different time points for each stock can be regarded as a time series. Without loss of generality, we suppose that we have N variables for T periods of time.

In this article, the final output of the network analysis of the particular multivariate time series could be displayed in a graph, in which all the target stocks are represented by nodes and the interconnections between them are plotted as the edges linking the nodes.

1.2 Partial Correlation Network

1.2.1 Partial Correlation Matrix

To study the relationship between two stock returns, say y_i and y_j, the common method is to calculate the Pearson correlation coefficient:

$$\rho_{raw}(i,j) = \frac{(y_i - \mu_i) * (y_j - \mu_j)}{\sigma_i * \sigma_j} \tag{2}$$

where μ_* represents average and σ_* denotes the standard deviation.

However, in some cases, a strong correlation not necessarily means strong direct relation between two stocks. For example, two stock in the same market can be influenced by common microeconomic factor and investors' herd behavior. To study the direct correlation of the performance of these two stocks, we need to get rid of the common driving factors, which here are represented by the market index. Partial correlation quantifies the correlation between variables, for instance stock returns, when conditioned on one or several other variables (Kennet 2010). Thus, we introduce partial correlation network in this section.

One particular assumption of the partial correlation network model is the sparsity of the dataset. Sparsity refers to the fact that the given network is not complete. In other words, not every node is connected with any other node in the dataset. Recall that a zero element in the correlation matrix implies the absence of an edge between two variables, therefore the correlation matrix of a sparse network contains a great number of zeros. In fact, this kind of sparse networks have been studied in many areas, e.g. genetics networks, social networks and so on. Thus, it is reasonable to assume that the data from stock market can also be considered as sparse.

1.2.2 Space

In this section, we take advantage of Lasso for detecting pairs of stocks having nonzero partial correlations among a large dataset. Lasso, stands for Least Absolute Shrinkage and Selection Operator, has been a very effective tool to obtain the estimations of partial correlations since Year 1996. It allows to shrink a number of estimated coefficients enough to end up with a sparse network. Lasso estimators are calculated as following:

$$\theta_\lambda = \underset{\theta}{argmin} \sum_{i=1}^{n} (Y_i - X_i^T \theta)^2 + \lambda \sum_{j=1}^{N} \mid \theta_j \mid \quad \lambda \geq 0 \tag{3}$$

The great thing about Lasso estimation is that, with proper λ, it selects the variables which better explain the linkage between them. It shrinks the parameters corresponding to the variables which are not so explanatory to exact zeros, keeping the parameter of the worthy variables different from zero. Therefore, we introduce sparse regression model by imposing the l_1 penalty on a suitable loss function to solve the high-dimension-low-sample-size problem.

Suppose that $(y_1, ...y_p)^T$ has a joint distribution with mean μ and covariance $\sum$, where $\sum$ is a p by p positive definite matrix. Denote the partial correlation between y_i and y_j by $\rho_{ij}(1 \leq i \leq j \leq 1)$. Also, here we define concentration matrix $\sum^{-1}$ by $(\sigma_{ij})_{p*p}$. It is known that $\rho_{ij} = -\frac{\sigma_{ij}}{\sqrt{\sigma_{ii} * \sigma_{ij}}}$. Thus, we denote $\beta_{ij} = -\frac{\sigma_{ij}}{\sigma_{ii}} = \rho_{ij} * \sqrt{\sigma_{jj}/\sigma_{ii}}$. Recall that we have several time periods, we suppose that $Y^k = (y_1^k, y_2^k, ..., y_p^k)^T$ for $k = 1, ..., T$. Denote the sample of the ith variable as $Y_i = (y_1^i, y_2^i, ..., y_p^i)^T \quad i = 1, ..., T$. Thus, we estimate the partial correlation θ by minimizing the penalized loss function:

$$\begin{aligned} L_n(\theta, \sigma, Y) &= \frac{1}{2}(\sum_{i=1}^{p} \omega_i \| Y_i - \sum_{j \neq i} \beta_{ij} Y_j \|^2) + \lambda \sum_{1 \leq i < j \leq p} \mid \rho_{ij} \mid && (4) \\ &= \frac{1}{2}(\sum_{i=1}^{p} \omega_i \| Y_i - \sum_{j \neq i} \rho_{ij} * \sqrt{\sigma_{jj}/\sigma_{ii}} Y_j \|^2) + \lambda \sum_{1 \leq i < j \leq p} \mid \rho_{ij} \mid && (5) \end{aligned}$$

where $\sigma = \{\sigma^{ii}\}_{i=1}^p$, $Y = \{Y_k\}_{k=1}^n$ and $\omega = \{\omega^i\}_{i=1}^p$. The outcome of the application of this model is the matrix containing the estimations of the partial correlations between variables. However, the performance of this model depends on the choice of the tuning parameter λ.

1.2.3 Tuning Parameter

The tuning parameter λ controls for the amount of shrinkage in the estimation procedure. First, in the case λ takes a value equal to zero, no shrinkage is produced and the estimators are exactly the same as in the ordinary least square case. Second, in the case λ takes a value big enough, all the lasso estimators might be shrunk to zeros so that there is no estimator different from zero. Thus, only when we pick a suitable λ, we could have a proper number of parameters been shrunk to zero.

In practice, different values of λ are estimated for the partial correlation networks, and afterwards, information criteria like AIC or BIC are applied to determine the optimal value of λ. In general, BIC is preferred to AIC due to its simplicity and computational easiness.

The rest of the article is organized as follows. In section 2, we will give full details about the empirical dataset and how we pre-process with it in our project. In Section 3, we describe the time-varying partial correlation network with our kernel smoothing approach. Once we have theoretically explained how the partial correlation network estimation method works, an illustrated simulation is going to be carried out in the first part of Section 4. Afterwards, we present our results with a number of plots in Section 4.2. In Section 5, a summary of the main results and proposal about the future work are given.

2 Data

The S&P 500 index is probably the most commonly referenced U.S equity benchmark for determining the state of the overall economy. S&P Dow Jones Indices updates the components of the S&P 500 periodically, typically in response to acquisitions, or to keep the index up to date as various companies grow or shrink in value. Between 1/1/2005 and 1/1/2015, 188 index components were replaced by other components. In our paper, we propose to track the performance of the largest and most dominant American companies included in the index. So, we pick 233 stocks as our target stocks, which have been included in the index for 12 years, from the year 2002 till the year 2013.

Through Daily TAQ(Trade and Quote), which provides us with FTP access to all trades and quotes for all issues traded on NYSE for all the trading days, we download the 1-minute stock data using our own target stock lists. We chose September 2013 as our test sample. So our dataset includes associated key financials such as time of trading, stock price, market capitalization of that month.

Before we apply the model to our data, we shall ensure that supplied data is clean, correct and useful. In our raw data, there are included 1-minute price, time of trading and trading volume of all the 233 stocks. However, there are some issues worthy of our attention. Firstly, we have, inevitably, lost some data on the certain trading time points during one month. Secondly, the 1-minute trading volume should be within a certain range, to make sure that the future calculation will not spill over. So before the introduction of our theoretical method, we have to process with data loss and data cleaning.

For every one of our target stocks, there are no more than 2 percent of data has been left blank. To deal with the data loss, we use linear interpolation to fill the gap in our raw data. For instance, if there is no record at the certain time during a trading day, then we will use two closest stock prices and trading volume to get the estimated value. To calculate the weighted average as unknown prices and trading volume, the weights are inversely related to the distance from the known points to the unknown point. In this way, we successfully get all the data at all the time points filled.

In our methodology, we will use an iterative algorithm to calculate the parameters of partial correlation matrix. So we have to get rid of those trading volume records which are beyond the limit of acceptability or fairness. First, all the trading volume data should be positive. Secondly, the value of trading volume should not exceed the certain range, since unreasonable large value will limit the calculation of partial correlation matrix. Without loss of common sense, we eliminate those 1-minute trading volumes whose values are negative or more than 1 million.

After processing with the raw data, we have more concerns before application of real data to our model. In our article, we will use 30 minute as interval of two neighbor records. That is to say, we calculate the average values of stock prices and trading volume in half an hour. During one trading day, whose trading period last from 9:30 in the morning till 4:00 in the afternoon, we have 14 records of prices and trading volumes for every target stock. For September 2013, we have 20 trading days. Thus, we have 280 records of stock prices and trading volumes for 223 target stocks in the whole.

To have a better understanding of the macrostructure of the stock market, we choose two measurements to do the quantitative research: stock returns and Volume-price trend indicators. Stock return is a traditional measure of a company's performance over time. Here we use single period return. Next, to measure the balance between a stock' demand and supply, we introduce Volume-price trend (VPT), which is an technical analysis indicator to relate price and volume in the stock market. VPT is based on a running cumulative volume that adds or subtracts a multiple of the percentage change in share price trend and current volume, depending upon the investment's upward or downward movements. We have the formula as following:

$$VPT = VPT_{prev} + volume * \frac{price_{now} - price_{prev}}{price_{prev}} \tag{6}$$

$$VPT = VPT_{prev} + volume * return \tag{7}$$

Thus, we use stock returns and VPT indicators as our measurement to the performance of target stocks and stock market balance.

3 Time-varying partial correlation network analysis

3.1 Local Polynomial Regression

In this section, we introduce a class of regression techniques that achieve flexibility in estimating the regression function, say $f(X)$, by using only those observations close to the target point x_0. In such a way, the resulting estimated function $\widehat{f}(X)$ is smooth. This localization is achieved via a weighting function or kernel $K_\lambda(x_0, x_i)$, which assigns a weight to x_i based on its distance from x_0. Recall Nadaraya-Watson kernel-weighted average

$$\widehat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)} \tag{8}$$

with the Epanechnikov quadratic kernel

$$K_\lambda(x_0, x) = D(\frac{\mid x - x_0 \mid}{\lambda}) \tag{9}$$

with

$$D_t = \begin{cases} \frac{3}{4}(1 - t^2) & if \mid t \mid \leq 1 \\ 0 & otherwise \end{cases} \tag{10}$$

Thus, we have progressed from the raw moving average to a smoothly varying locally weighted average by using kernel weighting. Furthermore, locally weighted regression solves a separate weighted least square problem at each target point x_0:

$$\underset{\alpha(x_0), \beta(x_0)}{min} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0) x_i]^2 \tag{11}$$

The estimate is then $\widehat{f}(x_0) = \widehat{\alpha}(x_0) + \widehat{\beta}(x_0) x_0$. Without stopping local linear fits, we can fit local polynomial fits of any degree d,

$$\min_{\alpha(x_0),\beta_j(x_0),\ j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0,x_i)[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0)x_i]^2 \tag{12}$$

with solution $\widehat{f}(x_0) = \widehat{\alpha}(x_0) + \sum_{j=1}^{d} \widehat{\beta}_j(x_0)x_0^j$. Since local linear fits can help bias dramatically at the boundaries at a modest cost in variance, we will apply this kernel smoother to joint sparse regression model as noted in the previous section.

3.2 Proposed Method

In this section, we will deduct our own methodology: apply kernel smoothers to the sparse regression model. Thus, the penalized regression problem will be as following:

$$min \sum_{i=1}^{N} D(\frac{\mid t - t_0 \mid}{h_\lambda(x_0)})[y_i - \beta_0 - \beta_1 x_{i1} - ... - \beta_j x_{ij} - ... - \beta_d x_{id}]^2 \quad s.t. \parallel \beta \parallel \leq \lambda \tag{13}$$

which can also be written as

$$min[\sum_{i=1}^{N} D(\frac{\mid t - t_0 \mid}{h_\lambda(x_0)})(y_i - \beta_0 - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \mid \beta_j \mid] \tag{14}$$

Consider a coordinate descent step for solving the above problem. That is, suppose we have estimates $\tilde{\beta}_0$ and $\tilde{\beta}_l$ for $l \neq j$, and we wish to partially optimize with respect to β_j. Denote by $R(\beta_0, \beta)$, the objective function in (14). We would like to compute the gradient at $\beta_j = \tilde{\beta}_j$, which only exists if $\tilde{\beta}_j \neq 0$. If $\tilde{\beta}_j > 0$, then

$$\frac{\partial R}{\partial \beta_j} \mid_{\beta=\tilde{\beta}} = -2 * \sum_{i=1}^{N} D(\frac{\mid t - t_0 \mid}{h_\lambda(x_0)}) x_{ij}(y_i - \beta_0 - X_i^T \beta) + \lambda \tag{15}$$

A similar expression exists if $\tilde{\beta}_j < 0$, and $\tilde{\beta}_j = 0$ is treated separately. Simple calculation shows (Donoho 1994) that the coordinate-wise update has the form

$$S(-2 * \sum_{i=1}^{N} D(\frac{\mid t - t_0 \mid}{h_\lambda(x_0)}) x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda) \rightarrow \tilde{\beta}_j \tag{16}$$

where

$$\tilde{y}_i^{(j)} = \tilde{\beta}_0 + \sum_{l \neq j} x_{il}\tilde{\beta}_l \tag{17}$$

is the fitted value excluding the contribution from x_{ij}, and hence $y_i - \tilde{y}_i^{(j)}$ the partial residual for fitting β_j. $S(z, \gamma)$ is the soft-thresholding operator with value

$$\begin{cases} z - \gamma & if\ z > 0\ and\ \gamma <\mid z \mid \\ z + \gamma & if\ z < 0\ and\ \gamma <\mid z \mid \\ 0 & if\ \gamma \geq \mid z \mid \end{cases} \tag{18}$$

Thus with smoother on time period, we compute the simple least-squares coefficient on the partial residual, apply soft-thresholding to take care of the lasso contribution to penalty, and then apply a proportional shrinkage for the ridge penalty.(Van der Kooij 2007)

4 Numerical Results

4.1 Illustrated Simulation

In order to check the performance of the joint sparse model to estimate partial correlation networks, we process with an illustrated simulation. Based on the assumption of sparsity, we create a true network with 20 elements and a total of 19 edges. Specifically, there are 3 hubs with three or more edges. Therefore the concentration matrix is a $20 * 20$ matrix. Though it doesn't represent a very big dataset, it is enough to perform a practical simulation to analyze and get useful results.

First, we make n random observations from a multivariate normal distribution. The number of N is 20 and their variance-covariance matrix used to generate those n observation s is the inverse of the concentration matrix created in the first step. After that, the join sparse regression is used to generate a concentration matrix from the random observations obtained previously taking λ as the value of the penalty. Finally, we will get the adjacency matrix. We then evaluate each method at a series of different values of the tuning parameter λ.

In our simulation, we study for four different sample sizes $n = \{50; 100; 500; 1000\}$. The tuning parameter λ, always proportional to the sample size, takes values from $\lambda = 0.00 * n$ to $\lambda = 1.00 * n$ in a sequence of 0.05. Since the simulation is a random process, it must be done many times in order to get accurate and unbiased results. To reach that end, for each pair of λ and sample size values, we process with the network estimation for 1000 times.

In order to analyze the results obtained from the simulation, one graph will be showed to study the features of the joint sparse regression model. Figure 1 will show how changes in the penalty value affect the sparsity of the estimated network.

From Figure 1, we can see that there is a positive relationship between the penalty value λ and sparsity. When λ increased, the total number of edges found decreases, so the estimated network becomes sparser, moving from far more above the number of true edges to below it when the penalization is too big. When the penalty value is zero, the number of edges found is 190 for every sample size, which is the maximum possible number of existing edges in a network of 20 variables. On the other hand, when the penalty is $1.00 * n$, sparsity converges again for all sample sizes at a level below the true number of the edges in the network. For different sample sizes, it is observed that for large values of n, the estimated networks become very sparse and the number of edges estimated fall close to the number of true edges. That is to say, in the case n is lower, the estimated networks become sparser following a not-to-steep path until the true number of edges is reached, and once in there, the number of edges estimated fall below the optimal value faster than with large sample sizes as λ increases.

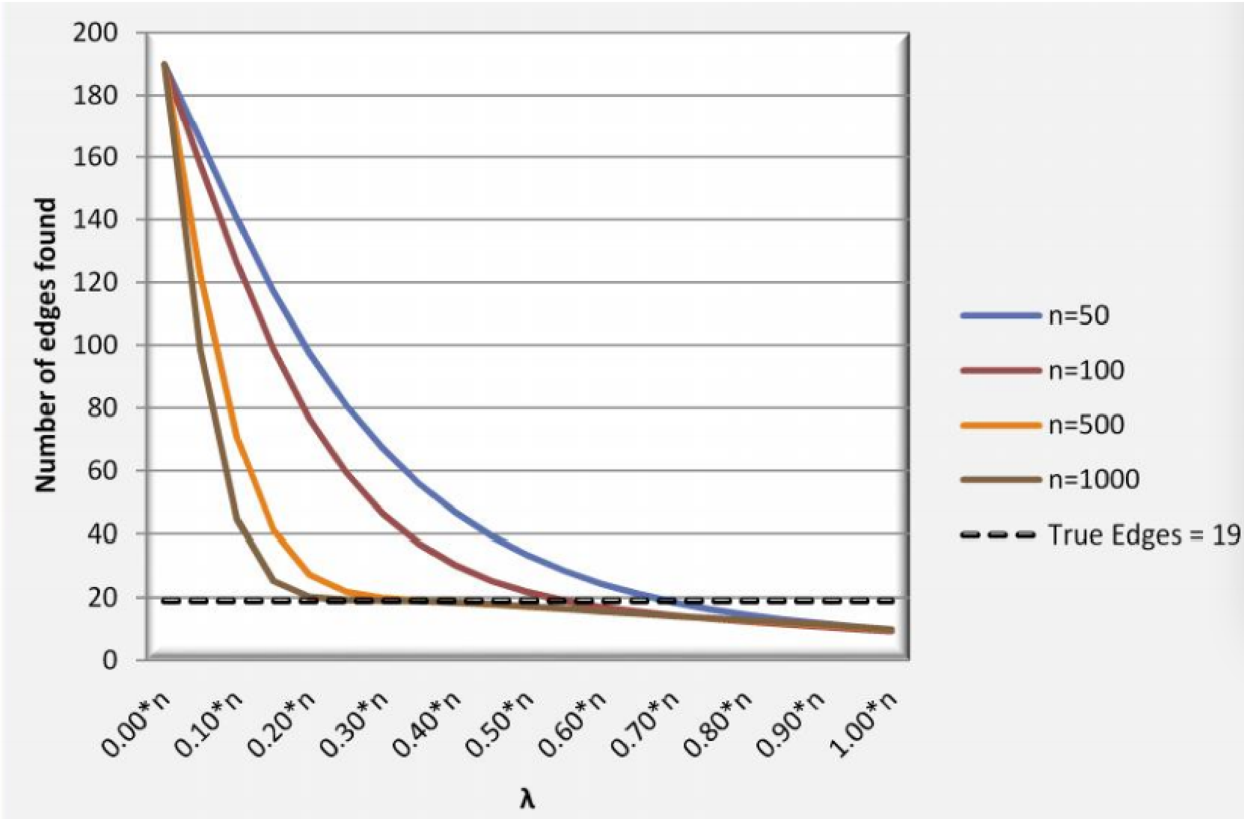


Figure 1: How does λ affect sparsity

4.2 Real Data

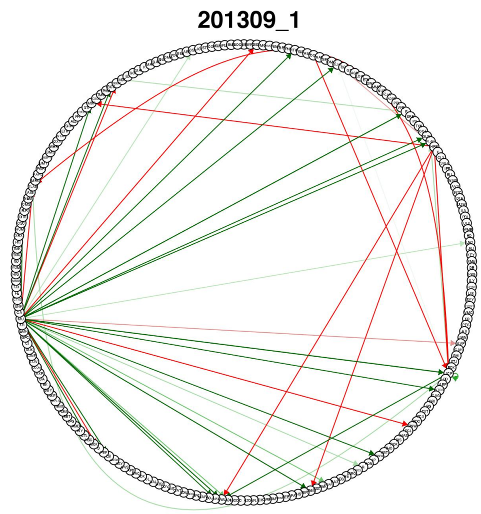
In the previous section, we have achieved the associated trading data of 233 target stocks. For September 2013, we have 279 30-minute stock returns and 280 30-minute VPTs for every stock. Thus, our real dataset is composed of 233 variables containing 279 time points for each one, standing as a reasonable size for our estimations to end up with significant results. The procedure to reach the estimated network is based on calculating the partial correlation matrices of the dataset and plotting its network. It is performed by a script in R, since we used the Space package developed by Peng(2009).

Firstly, we import the 233*279 matrix, where each column is named with the time points and each row is named with the ticker stock symbol. Go through each time point, we apply kernel smoother on time to get the partial correlation matrix is generated by running the space.joint command. Here, we choose 30 as the moving window size. Afterwards, we use Qgraph package in R to plot each network on each time point. Thus, we have 279 plots of partial correlation networks of stock returns. Here in this article, we picked 4 of them at different Tuesday in September 2013. The interval of these 4 figures are the same, which is $14 * 5 = 70$. See Figure 2. In the pictures, each node represents a stock and each edge represents a partial correlation between two stocks. Green edges indicate positive correlations, red edges indicate negative correlations, and the width and color of the edges correspond to the absolute value of the correlations: the higher the correlation, the thicker and more saturated is the edge.

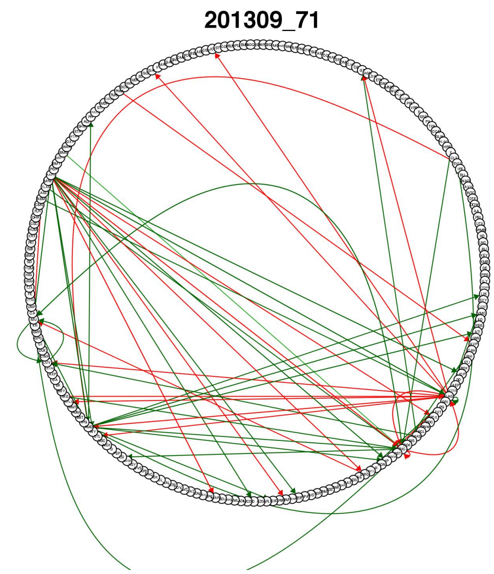
As mentioned in previous section, there are quantitative methods as BIC and AIC which enable to find out the optimal λ for the model. In practice, the penalty value of the real data application has been chosen on the results obtained from several of tryings. That is to say, the tuning process has been based on observing the networks from a wide range of λ values and choosing the one that looks more efficient for our study. For instance, if we choose $\lambda = 7e^{-6}$, the partial correlation network at time 9:30 am on September 3rd, 2013 is shown in Figure 3a; while if $\lambda = 7 * e^(- 5)$ is chosen, then the corresponding partial correlation network at exact the same time, is shown in Figure 3b

From Figure 3a, we can see that the choice of $\lambda = 7e^{-6}$ must be too small, since the shrinkage of Lasso doesn't have much effect on partial correlation. We can almost read no useful information from the redundant partial correlation network in Figure 3a. For Figure 3b, we can read clear information of the significant partial correlations between two stocks. So the choice of $\lambda = 7e^{-5}$ is a good one. In practice, remind that the analysis of the causes behind the nature of the partial correlations are not a goal of the project, therefore the selection of the optimal λ is not a crucial point in our study, as long as we are working in a close interval from it.

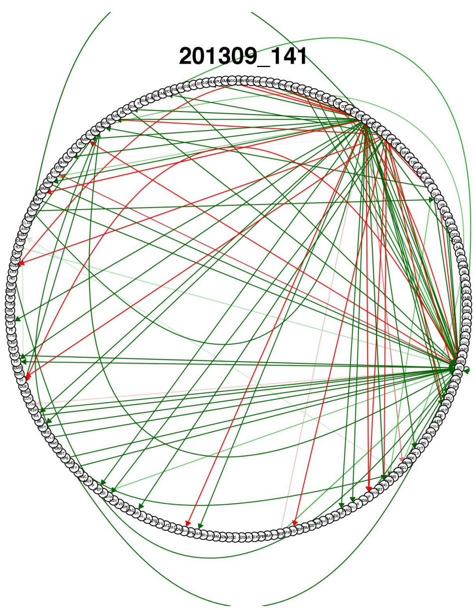
Furthermore, we are interested in maximum eigenvalue in the partial correlation matrix at the certain time point. For the time being, the change of maximum eigenvalues is shown in



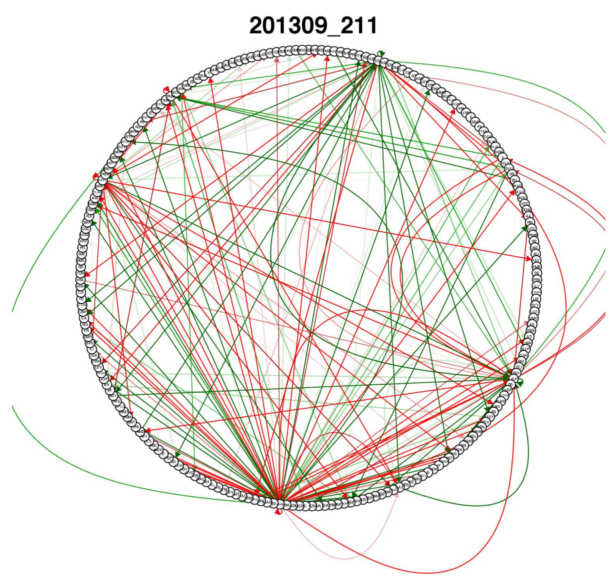
(a) Network 1



(b) Network 2

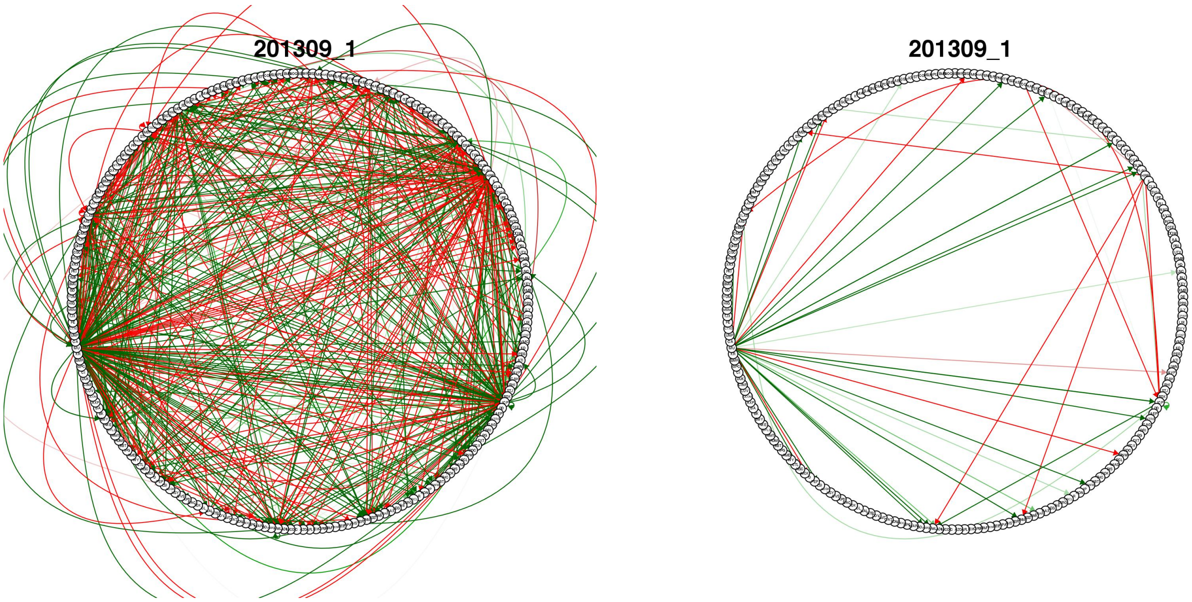


(c) Network 3



(d) Network 4

Figure 2: Partial Correlation Network of stock return at 9:30am on (a)September 3rd, 2013; (b)September 10th; (c)September 17th; (d)September 24th. (From left to right, top to down)



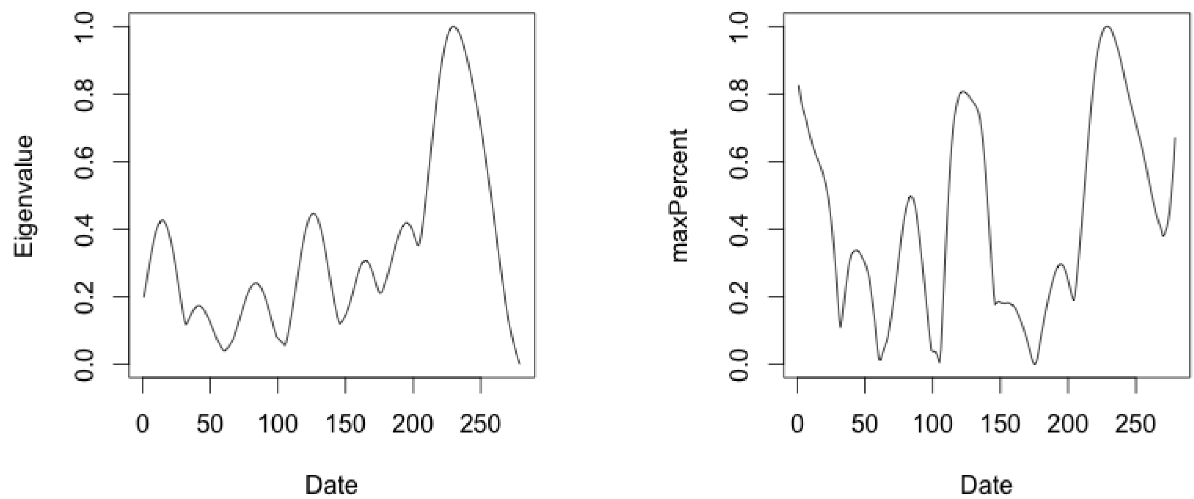
(a) (b)

Figure 3: (a)The partial correlation network at time 9:30 am on September 3rd, 2013 when $\lambda = 7*e^{-6}$; (b)The partial correlation network at time 9:30 am on September 3rd, 2013 when $\lambda = 7*e^{-5}$

Figure 4a, while the percentage of the maximum eigenvalue over the sum of all the eigenvalues in the partial correlation matrix is shown in Figure 4b.

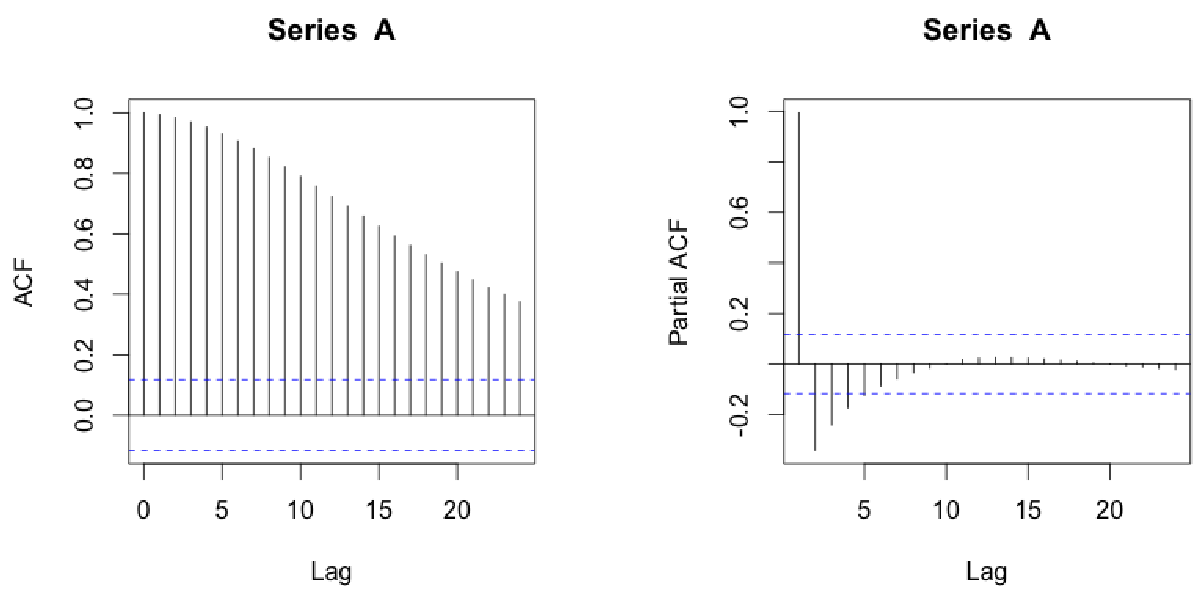
As we know, the maximum eigenvalue changes through time. So it could be regarded as a time series data. Here we fit this time series data with AR(1) model. The Autocorrelation plot is shown in Figure 5a while the partial Autocorrelation plot is shown in Figure 5b.

Moreover, we want to have a general look at the performance of the maximum eigenvalue time series with respect to different λ. We chose 20 values of λ, which is respectively $1e^{-3}$, $3e^{-3}$,$5e^{-3}$,$7e^{-3}$,$9e^{-3}$, $1e^{-4}$, $3e^{-4}$,$5e^{-4}$,$7e^{-4}$,$9e^{-4}$,$1e^{-5}$, $3e^{-5}$,$5e^{-5}$,$7e^{-5}$,$9e^{-5}$,$1e^{-6}$, $3e^{-6}$,$5e^{-6}$,$7e^{-6}$,$9e^{-6}$. Next, we draw the 3 dimension surface of the results, from different angle. See Figure 6.



(a) (b)

Figure 4: (a)Maximum Eigenvalues at different time points; (b)The percentage of Maximum Eigenvalues at different time points



(a) (b)

Figure 5: (a)ACF plots of maximum eigenvalues of Partial Correlation Matrices; (b)PACF plots of maximum eigenvalues of Partial Correlation Matrices

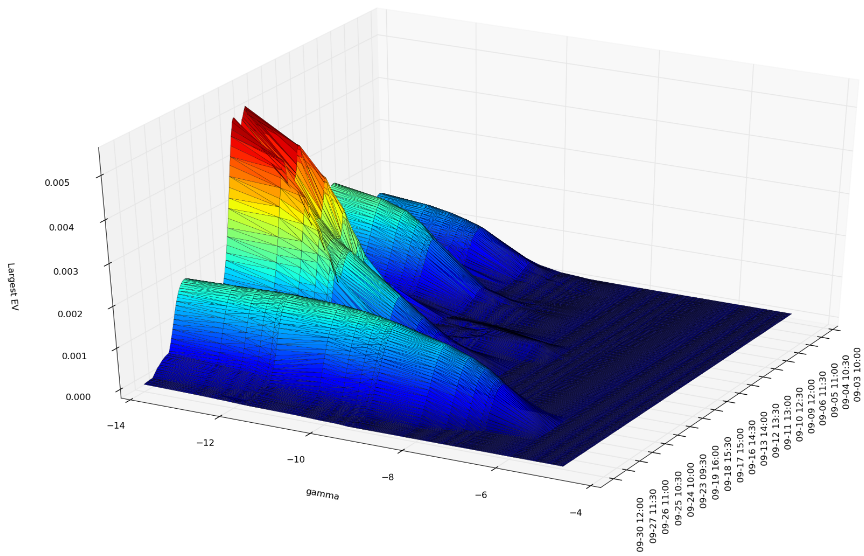


Figure 6: Three dimension plot of maximum eigenvalues with different λ

From the above 3D version plots of maximum eigenvalues with different λ, we can see that there are some peak points. Specially at the 121st time point and 165th time point. We are interested in such points so that we track back to these two exact times. For the 121st time point, it refers to 1:30pm on September 13rd, 2013; while the 165th time points refers to 2:30pm on September 18th, 2013. Tracking back to that date, a meeting of the Federal Open Market Committee was held in the offices of the Board of Governors of the Federal Reserve System in Washington, D.C., on Tuesday, September 17, 2013, at 1:00 p.m. and continued on Wednesday, September 18, 2013, at 8:30 a.m.

5 Summary and Future Work

5.1 Summary

Throughout this project, the joint sparse regression model with kernel smoother on time is proposed to estimate partial correlation networks under the assumption of sparsity. Hence, after the setting of the theoretical deduction on the method and the "space" estimation tools, we confirm the efficiency and validity of our proposed method, especially in the case of presenting cross-sectional interconnections between stocks in financial market.

5.2 Future Work

5.2.1 Vector Autoregression with One Lag

The above methods come out with the undirected partial correlation matrices. However, we may want to look into the relationship between the stocks with direction. Here we introduce VAR(1) model: define the matrix is as Y_t , then

$$Y_t = AY_{t-1} + \epsilon_t \tag{19}$$

$$\tilde{Y_t} = AY_{t-1} \tag{20}$$

$$\tilde{Y_t}^T = A^T Y_{t-1}^T \tag{21}$$

(22)

5.2.2 More time periods

We will also draw the plots of VPT in more time periods such as August 2013 and October 2013. More simulation study will be applied.

References

[1] MarketCapitalizations.com "Changes in S&P 500 components during the past ten years", 25 March 2015.

[2] Kenett DY, Tumminello M, Madi A, Gur-Gershgoren G, Mantegna RN, Ben-Jacob E (2010) Dominating Clasp of the Financial Sector Revealed by Partial Correlation Analysis of the Stock Market. PLoS ONE 5(12): e15032.

[3] Donoho, David L., and Iain M. Johnstone. "Adapting to unknown smoothness via wavelet shrinkage." Journal of the american statistical association 90.432 (1995): 1200-1224.

[4] Kooij, Anita J. Prediction accuracy and stability of regression with optimal scaling transformations. Child & Family Studies and Data Theory (AGP-D), Department of Education and Child Studies, Faculty of Social and Behavioural Sciences, Leiden University, 2007.

[5] Friedman, Jerome, et al. "Pathwise coordinate optimization." The Annals of Applied Statistics 1.2 (2007): 302-332.

[6] Peng, Jie, et al. "Partial correlation estimation by joint sparse regression models." Journal of the American Statistical Association 104.486 (2009).

[7] Epskamp, Sacha, et al. "qgraph: Network visualizations of relationships in psychometric data." Journal of Statistical Software 48.4 (2012): 1-18.

[8] W. DeSarbo et al..: Refining Initial Points for K-Means Clustering. Psychometrika, pp. 57-78. (1984)

[9] Bradley, Paul S., and Usama M. Fayyad.: Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. ICML. Vol. 98. (1998)

[10] Kanungo, Tapas, et al..: An efficient k-means clustering algorithm: Analysis and implementation. Pattern Analysis and Machine Intelligence. IEEE. 881-892 (2002)

[11] Huang, Joshua Zhexue, et al..: Automated variable weighting in k-means type clustering. Automated variable weighting in k-means type clustering. 27.5, 657-668 (2005)

[12] Modha, Dharmendra S., and W. Scott Spangler.: Feature weighting in k-means clustering. Machine learning. 52.3, 217-237 (2003)

[13] Lagarias, Jeffrey C., et al..: Convergence properties of the Nelder–Mead simplex method in low dimensions. Convergence properties of the Nelder–Mead simplex method in low dimensions. 9.1, 112-147(1998)

[14] W. Karush.: Minima of Functions of Several Variables with Inequalities as Side Constraints. University of Chicago(1939)

[15] H.W. Kuhn, and A.W. Tucker.: Nonlinear Programming. Second Berkeley Symposium on Mathematical Staitistics and Probability, University of California Press, Berkeley, pp. 481-492 (1951)

[16] Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook.: Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook. John Wiley & Sons (2009)