Studying, Implementing and Exploring Use Cases for the Paper "Identifying long-term precursors of financial market crashes using correlation patterns"

Submitted by

Suhas Gumma - 1700278C203

Batch - 2018

Course
Data Science of Complex Systems.

BML MUNJAL UNIVERSITY™
FROM HERE TO THE WORLD

SCHOOL OF ENGINEERING & TECHNOLOGY
BML MUNJAL UNIVERSITY

December 2020

1. Introduction

A financial market crash is loosely defined as an abrupt drop in stock prices. But, it is more than a drop in stock prices. The event will most often have a significant impact on the economy. So, identifying long term precursors for a financial crisis could be used to avoid the crash by taking relevant measures or taking advantage of the crash. To have an idea about the situation of the financial market, there are various financial metrics to be tracked. Instead of keeping track of all those, researchers focused their attention on the **information of co-movements and correlations among the stocks of the market**. Various empirical studies have already shown that the mean correlation between the stocks increases during market crashes compared to normal periods. In the same way, there are some correlation structures which seem to appear more often than by pure randomness. Identifying and classifying these structures can be used to track the movement of the market which will consequently help in identifying the precursors for market crashes.

In the following sections, we will try to explain how the paper “Identifying long-term precursors of financial market crashes using correlation patterns” tackled this problem. We present our implementation of the methodology and explore some use cases of the methodology outside the domain of financial markets.

2. Objective

The objective of the paper is to identify long term warning signs for the financial market crashes using the correlation patterns among the stocks in the financial market.

3. Data

3.1 Data Used in the Paper:

Time series of adjusted closure prices of two financial markets.

1. S&P 500 index (USA). The prices of 195 stocks are taken over a period of 8068 days from 02-Jan-1985 to 30-Dec-2016.
2. Nikkei 225 index (JPN). The prices of 165 stocks were taken over a period of 7998 days from 04-Jan-1985 to 30-Dec-2016.

3.2 Data Used in our Implementation

Time series of adjusted closure prices of S&P 500 index (USA). The data consists of prices of 165 stock values spanned over 4277 days from 03-Jan-2000 to 30-Dec-2016.

4. Overview of Methodology:

Data is divided into equal short time intervals. Then, for each time interval, noise suppressed correlation matrices are built. These matrices are organized into different clusters using k-means clustering. Each cluster represents a market state and one of them is a critical state. Transition probabilities are calculated in the end through which we can identify the precursor state to the critical state.

5. Challenges:

- ***Determining the length of the time interval (epoch).*** After calculating the return time series, long intervals will smoothen the fluctuations but will suffer from non-stationarity. If short intervals are chosen, the correlation matrices will be noisy.

- ***Suppressing the noise in the correlation matrices calculated for each epoch.*** The paper used a recent and effective method called power map method to suppress the noise.

- ***Choosing the value of noise suppression parameter (ε) in the power map method***.

- ***Visualizing the data points.*** The shape of the correlation matrix will be n X n (n is the number of stocks). It can be visualized using the heatmap. But, the visualisation of data points relative to each other could not be done using that shape. So, the MDS algorithm is used to reduce the dimensions to n X 2 (or n X 3 if we wish to plot on a 3D plane) to plot them relative to each other on a 2D plot.

  A distance matrix is generated using the metric $d_{ij} = \sqrt{2(1 - C_{ij})}.$ and passed to the MDS algorithm for visualising the correlation matrices.

- ***Finding the optimal number of states to be clustered into.*** Further details will be discussed in the methodology section.

6. Methodology

6.1 Converting the time series of adjusted closure price to log-return time series.

Log returns are used for various reasons in the financial time series analysis. One of the main reasons and most relative use here is that it makes the time series more stationary especially when divided into short time periods.

Code Snippet:

```
def getLogDiffAllStocks(matrix):
  logDiffMatrix = []

  for row in matrix:
    logDiffMatrix.append(np.diff(np.log(row)))

  return logDiffMatrix
```

6.2 Dividing data into short time- intervals (epochs):

The data is divided into short time periods. For the data of S&P 500 used in the paper, the authors selected an **epoch length of 20** days with **slide distance of 10** days. Each epoch generated is **50% overlapping** with the previous epoch. For **8060 days** in S&P 500 data, **805 epochs** are obtained. The shape of each epoch is **194 X 20** (Because of 194 stocks in S&P data).

Code Snippet:

```
def getEpochs(matrix, size = 20, slide = 10):
  finalEpochs = []

  startIdx = 0

  while startIdx+size < len(matrix[1]):
    epoch = []

    for row in matrix:
      epoch.append(row[startIdx: startIdx+size])

    finalEpochs.append(epoch)

    startIdx+= slide

  return finalEpochs
```

6.3 Get Cross-Correlation Matrix for each epoch.

For each epoch, build a cross-correlation matrix among the stocks present. For each epoch obtained for S&P 500 data, we get a cross-correlation matrix of size **195 X 195**. The **cell (i, j)** represents the **correlation between stock i and stock j**.

6.4 Noise Suppression in the Cross-Correlation Matrices

Due to the short length of the epoch selected, the resulting cross-correlation matrices are noisy. To **suppress the noise**, a powerful and more recent technique called **power mapping** is used. In this technique, a **non-linear distortion** is applied to **each element** in the cross correlation matrix. The function is: $C_{ij} \rightarrow (\text{sign } C_{ij})|C_{ij}|^{1+\epsilon}$. Here, C_{ij} represents each cross-correlation coefficient in the matrix and '$\boldsymbol{\varepsilon}$' denotes the noise suppression parameter. A suitable noise suppression parameter is chosen to keep in mind the determination of an optimum number of market states. The authors chose 0.6 for both S&P 500 and Nikkei 225 data.

Code Snippets:

```
def noiseSuppressSingleEl(el, epsilon):
  sign = -1 if el < 0 else 1

  return sign* math.pow(abs(el), 1+ epsilon)
```

```
def noiseSuppressMatrix(matrix, epsilon):
  newMatrix = np.zeros(shape = (len(matrix), len(matrix)))

  for i in range(len(matrix)):
    for j in range(len(matrix)):
      newMatrix[i][j] = noiseSuppressSingleEl(matrix[i][j], epsilon)

  return newMatrix
```

6.5 Build a Similarity Matrix

A similarity matrix is built in which **cell (i, j)** represents the **similarity** between **cross-correlation matrices of epoch i and epoch j**.

The similarity measure between to cross-correlation matrices is:

$\zeta(\tau_1, \tau_2) \equiv \langle |C_{ij}(\tau_1) - C_{ij}(\tau_2)| \rangle$, . Here, |...| denotes that the absolute value is taken and <..> denotes the mean.

Code Snippet:

```
def getSimilarityMatrix(corrMatrices):
  similarityMatrix = np.zeros(shape = (len(corrMatrices), len(corrMatrices)))

  for i in range(len(corrMatrices)):
    for j in range(len(corrMatrices)):
      similarityMatrix[i][j] = np.mean(np.abs(corrMatrices[i] - corrMatrices[j]))

  return similarityMatrix
```

6.6 Determine the optimal K for K-Means clustering:

- First, we select a very large group (say 500) of different initial conditions. Each condition is a random choice of k initial points.
- If clusters are wide apart, we get the same final results even for different initial conditions which leads to a small variance in intra-cluster distances. Vice versa, a small variance in intra-cluster distances signifies that the clusters are wide apart.
- If the clusters are too close or overlapping, it becomes difficult to assign a point to the cluster and the inaccuracy increases.
- That is why a small variance in intra-cluster distances denotes the validity of the clustering.

- When choosing K, the authors chose the maximum K with minimum variance in intra-cluster distances because of the above reasons. For S&P 500, K turned out to be 4 and for Nikkei 225, the K is 5.

6.7 Cluster into K clusters using K-means Algorithm:

- Now, we have 'n' noise suppressed cross-correlation matrices (n is the number of epochs). All we have to do is cluster them into an already determined k number of clusters.
- We measure the distance between two cross-correlation matrices as the absolute difference between the mean correlations.

Code Snippets:

K-Means Clustering

In [131]:
```
meanCorrelVector = []

for matrix in corrMatrices:
    meanCorrelVector.append(np.mean(matrix))

meanCorrelVector = np.array(meanCorrelVector)
```

In [133]:
```
kmeans = KMeans(n_clusters=4)
kmeans.fit(meanCorrelVector.reshape(-1,1))
```

Out[133]:
```
KMeans(n_clusters=4)
```

In [134]:
```
labels = kmeans.labels_
```

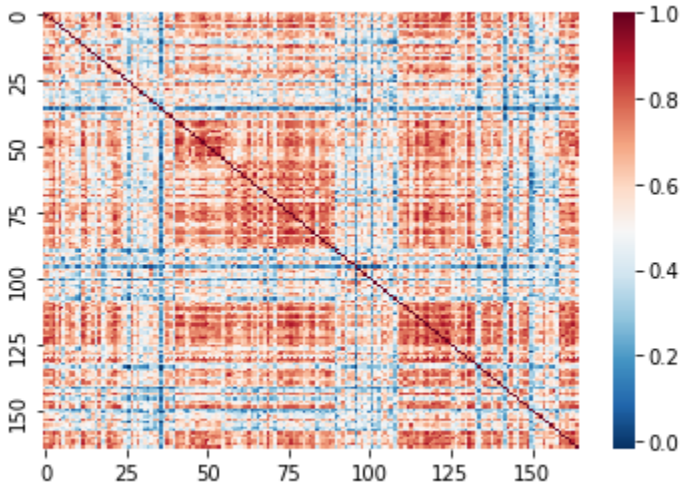
6.8 Identify the critical state using Cross-correlation Heatmaps

- Pick a random epoch from each cluster and visualize the cross-correlation matrix using a heatmap.
- One of them will be intense. That one is the critical state. If it is not apparent visibly, the state with the highest mean correlation is the critical state.
- This is so because the cross-correlation among stocks increases significantly when there is trouble in the market.

Code Snippet:

```
In [145]: #Cluster 1
corrMatrix = corrMatrices[295]
ax = sns.heatmap(corrMatrix, cmap = "RdBu_r", xticklabels =25, yticklabels = 25)


plt.show()
```

HeatMap of one of cross-correlation matrix in the critical state:



6.9 Calculate the Transition Probabilities Between States:

- After labelling each epoch, track each movement of market state and build a transition probability matrix.

Code Snippet:

Transition Probabilities

In [136]:
```
transitionProbMatrix = np.zeros(shape = (4, 4))

for i in range(1, len(labels)):
    currLabel = labels[i]
    prevLabel = labels[i-1]

    transitionProbMatrix[prevLabel][currLabel]+=1

print(transitionProbMatrix)
```

6.10 Identify the precursor market state:

- Identify the market state which has the highest probability to transition into the critical state. We need to be alert when the market is in that state.

6.11 3D MDS Visualization with different Clusters:

- Reduce the dimensions of the similarity matrix from (n X n) to (n X 3) using MDS algorithm.

- Plot a 3D plot using the MDS algorithm and color categorize different labels.

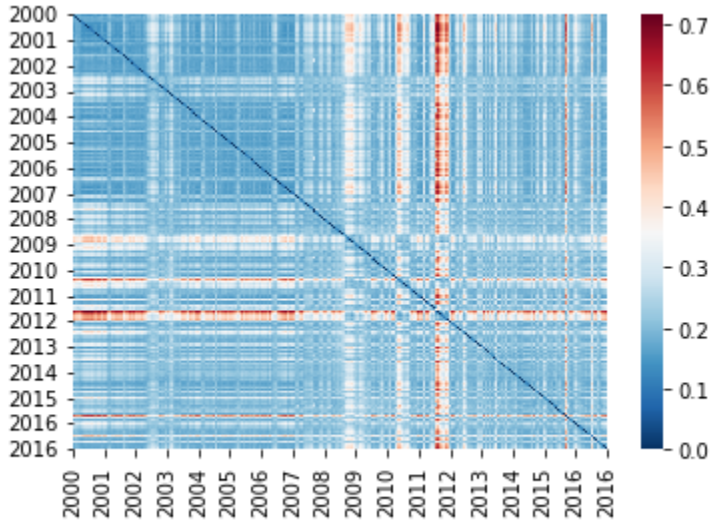
Code Snippet:

```
colors = ['#67e6ad','red','#3e4a45', 'green']
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')

for i in range(4):
    subset= fittedMDS[labels == i]
    x = [row[0] for row in subset]
    y = [row[1] for row in subset]
    z = [row[2] for row in subset]

    ax.scatter(x, y, z, c= colors[i])

ax.view_init(-140, 60)


plt.show()
```

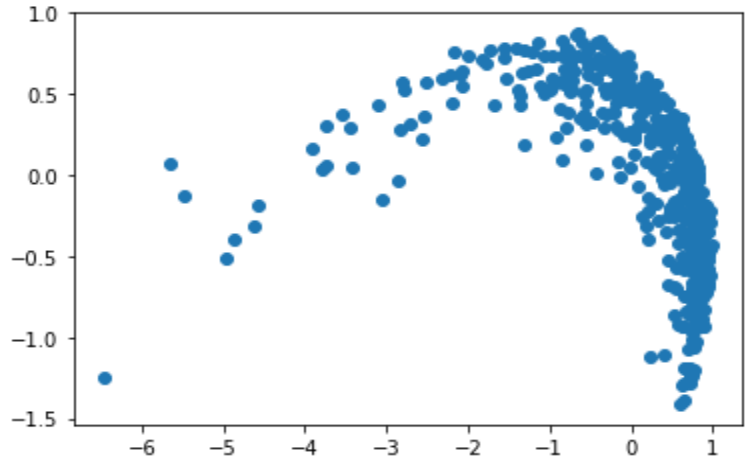
7. Results:

- Note that the size of data used by us for implementation is smaller than the dataset used in the research paper.
- The data is the time series of adjusted closure prices of S&P 500 index (USA). The data consists of prices of 165 stock values spanned over 4277 days from 03-Jan-2000 to 30-Dec-2016.

7.1 HeatMap of similarity matrix:



7.2 2D MDS of similarity matrix:



7.3 Label of each Epoch after clustering(0, 1, 2, 3):

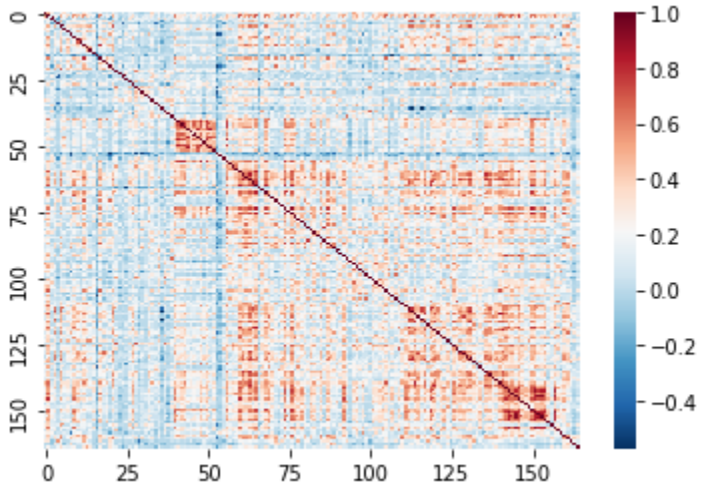
```
print((labels))
```

```
[3 3 3 3 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 0 0 0 3 3 3 3 3
 3 3 3 3 0 0 3 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 0 0 0 2 2 2 2 2 2 2 2 2 0 0 0 0
 2 2 0 0 2 2 2 0 0 0 0 0 0 3 3 3 3 3 0 0 3 3 3 0 3 3 3 3 3 0 2 0 0 0 0 0 0
 0 3 3 0 2 3 3 0 3 3 3 0 3 0 0 3 3 0 0 3 0 0 0 0 3 3 0 3 3 3 0 0 3 0 0 3 3
 3 3 0 3 3 3 3 3 0 3 3 0 0 2 2 0 0 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 0 3 3 3
 0 2 0 2 2 2 2 2 0 0 0 0 2 2 2 0 0 0 2 2 2 2 0 0 0 0 0 0 0 0 2 2 2 1 1 1 1
 1 1 1 2 2 2 2 1 1 2 2 0 0 2 2 2 2 3 0 0 0 0 2 2 2 2 0 0 3 0 2 2 0 3 3 3 2
 1 1 1 1 1 2 2 1 1 2 2 0 0 0 2 0 3 3 3 3 2 2 2 3 3 3 2 2 2 2 2 1 1 1 1 1 1
 1 1 1 1 1 2 3 3 3 0 2 2 2 0 2 1 1 2 2 0 0 0 0 0 0 0 2 2 0 2 2 3 3 2 2 0 2
 2 0 0 2 1 2 3 3 0 2 0 0 2 0 0 0 0 0 0 2 2 2 0 0 2 0 3 0 0 3 3 0 0 0 0 3 2
 2 2 0 3 0 2 2 2 0 0 0 2 0 3 0 0 0 2 2 2 0 0 1 1 1 2 0 3 0 2 2 2 2 2 2 2 0
 0 0 0 0 0 0 1 1 0 3 3 2 2 0 3 3 3 3 3]
```

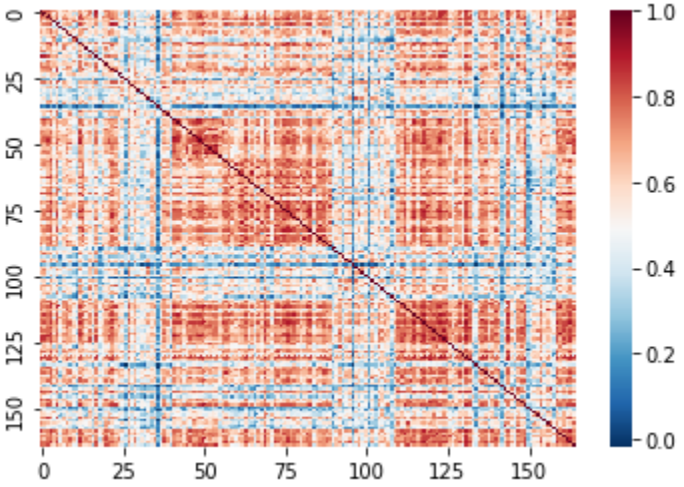
7.4 Scatter Plot of Epoch Number Vs Label:

7.5 Heatmap of a random Cross-Correlation matrix from each cluster:

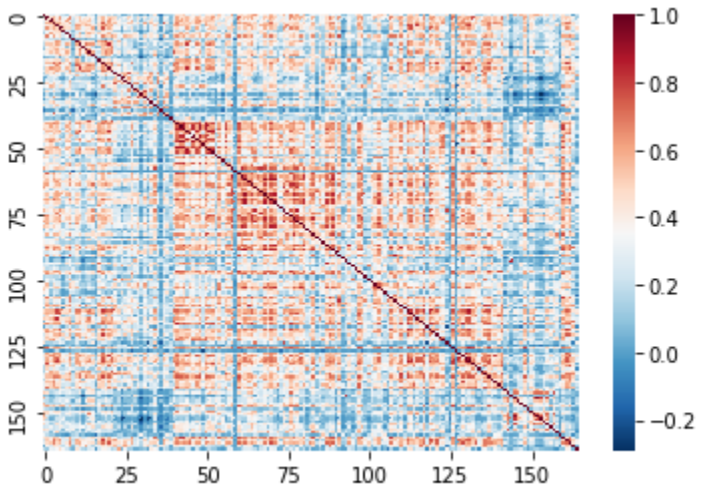
7.5.1 Cluster 1



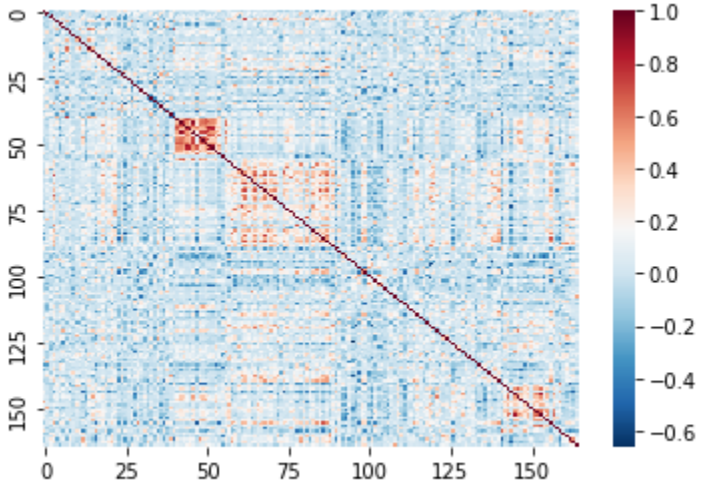
7.5.2 Cluster 2



7.5.3 Cluster 3

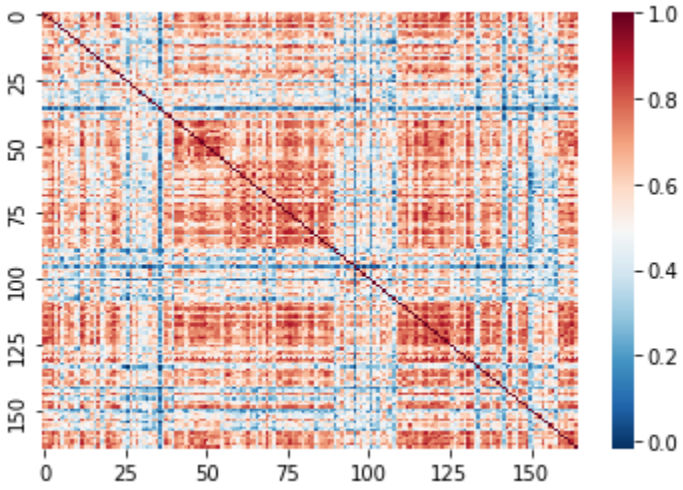


7.5.4 Cluster 4



7.6 Identifying the critical state:

- We can clearly see the heavy intensity in the cross-correlation matrix in the second cluster. Therefore, the **critical state is State 2**.



7.7Transition Probability Matrix:

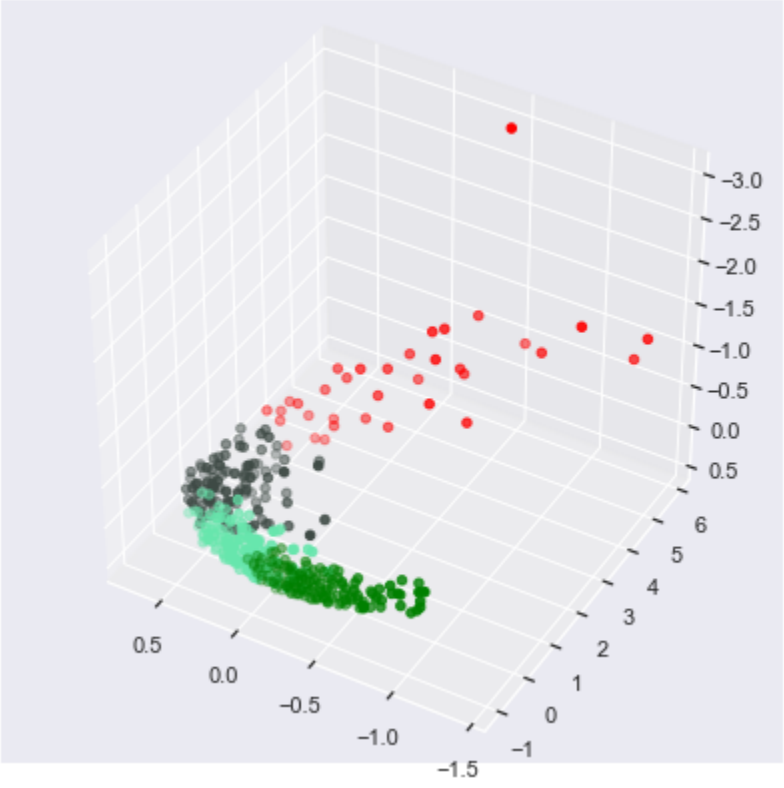
	S1	S2	S3	S4
S1	0.557	0.014	0.207	0.221
S2	0.028	0.742	0.228	0
S3	0.281	0.063	0.6	0.05
S4	0.214	0	0.05	0.735

7.8 Identifying the precursor market state to the critical state:

- **S3** has the **highest probability** to transition to S2 which is the critical state. Thankfully, it is only 0.063. There is a probability of 0.01 for S1 to transition into S2.

- The precursor for the critical state is **S3**.

7.9 3D MDS Visualization with different Clusters:



- Light Green - S1
- Red- S2
- Black- S3
- Green -S4

8. Possible Use cases outside the domain of Financial markets

Identifying precursors of epidemic critical stages(rise in active cases) using correlation patterns

Although the COVID-19 pandemic might be reaching the endemic stage in many parts of the world, it might not be our last pandemic or epidemic. Surveillance of the state of the epidemic when the economy is open (not completely locked down) is crucial. Here, we propose a model using the paper for continuous surveillance of the ongoing epidemic when there is an open economy.

Just like 194 stocks selected from S&P(USA) in the research paper, select “N” regions in an open area which would better represent the area. For example, in India, we might select 29 states. But, considering 29 is too small, we can consider a suitable number of districts which would better represent the entire country.

There are two options to consider when picking the type of data. We can consider the time series of “Daily new cases” or the time series of “Current active cases”. “Current active cases” is similar to the time series of stock value because the new value depends on the previous value just like in stock data.

Since we are assuming that the time series of current active cases is similar to the time series of stock value to design the model, it is important to identify the differences between them. In the financial market, stocks are not that correlated with each other in normal circumstances. But, we cannot say it is the same with “current active cases” of different regions in the same open area in normal circumstances. In financial markets, empirical studies show that there is a significant increase in the correlation between the stocks when there is a market crash. But, there is no evidence that the correlation increases “significantly” when there is a rise in the market. Contrary to that, we assume that there will be an increase in correlation between different regions both when the wave is rising and the wave is falling.

We can identify the “current active cases” in three different basic scenarios. When the series is a flat line, steady increase, steady decrease. Basically, when the series is a flat line, although the correlation between the regions might be good, we assume there can be little independence because they are different regions. We assume that when there is a rise in the wave and fall in the wave, the correlation between the different regions increases significantly especially when there is a rise in the wave. Unlike the model for the financial market, this model can only distinguish between the rise and fall of the wave if only there is a difference in rate of change in the correlation and probably that is not the case.

The methodology used is the same as the methodology proposed in the paper. Let's say we selected “N” regions.First, we convert the time series of each region into a log return time series. Then, we divide the time series of each region into epochs using a sliding window method with suitable window size and slide distance. We end up with “n” epochs. For each epoch, the cross-correlation matrix of size N✕ N is constructed using pearson-cross correlation coefficients(which only measures linear correlation). Then, noise suppression is done for each correlation matrix using the power map method. Now, we have “n” cross-correlation matrices. Then, we compute the optimum number of clusters “k” to divide for the k-means clustering algorithm. we cluster the epochs into “k” states. The epoch is assigned to the cluster whose mean correlation is near to the epoch’s mean correlation. Calculate the transition probabilities between different states. Finally, identify the state in which the action is happening(the rise of active cases).This can be done by examining the heatmaps of cross-correlation matrices belonging to different clusters. **Using the identified states and the transition probabilities we can monitor the evolution of the epidemic.**

9. Conclusion

The fundamental observation that the paper is based on is that the cross correlation patterns among the stocks in a financial market don't happen by pure chance. But, the challenge was to divide them into meaningful clusters. In the beginning, short epochs were chosen over long epochs for maintaining stationarity. A powerful technique called power mapping was used to remove the noise present in cross-correlation matrices. Another challenge was to determine the number of clusters to be grouped into. The authors chose to take the maximum K which will minimize the variance in intra-cluster distances and they justified it. Finally, correlation matrices were clustered into K clusters. Transition probabilities are calculated. The critical state is identified. One important observation of the result is that the market is more likely to stay in the same state rather than move to the next state. The state beside the critical state might act as the precursor for the trouble that might happen in the market and hopefully, it might be used as a **long-term** precursor. This methodology need not be limited to the domain of financial markets. It could be fitted for other complex systems as well like the example use case shown by us.

10. Acknowledgement

I, Suhas Gumma like to express our gratitude to Dr Kiran Sharma Ma’am for providing the datasets required for the implementation of the paper. We would also like to thank all the authors of the paper “ Identifying long-term precursors in financial markets using correlation patterns” for giving it open access and allowing us to study it.

11. References:

[1] Hirdesh K Pharasi, Kiran Sharma, Rakesh Chatterjee, Anirban Chakraborti, Francois Leyvraz and Thomas H Seligman. Identifying Long-Term Precursors of Financial Market Crashes Using Correlation Patterns.

[2] Pharasi H K, Sharma K, Chakraborti A and Seligman T H 2018 Complex market dynamics in the light of random matrix theory New Perspectives and Challenges in Econophysics and Sociophysics ed F Abergel et al

[3]"KMeans Sklearn", scikit-learn, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed: 12- Dec- 2021].

[4] Correlation Coefficient: Simple Definition, Formula. (2021, July 1). Statistics How To.

https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-form

ula

[5]"Visualize multidimensional datasets with MDS", Medium, 2021. [Online]. Available: https://towardsdatascience.com/visualize-multidimensional-datasets-with-mds-64d7b4c16eaa. [Accessed: 13- Dec- 2021].

[6] M. Tang, "PCA, MDS, k-means, Hierarchical clustering and heatmap for microarray data", Rstudio-pubs-static.s3.amazonaws.com, 2021. [Online]. Available: https://rstudio-pubs-static.s3.amazonaws.com/93706_e3f683a8d77244a5b993b20ad6278f4b.html. [Accessed: 15- Dec- 2021].

[7]S. Paea and R. Baird, "Information Architecture (IA): Using Multidimensional Scaling (MDS) and K-Means Clustering Algorithm for Analysis of Card Sorting DataJUS", Uxpajournal.org, 2021. [Online]. Available: https://uxpajournal.org/information-architecture-card-sort-analysis/. [Accessed: 15- Dec- 2021].