Characterizing Deep Dynamic Financial Networks using the Convolutional AutoEncoder

Rui Sun[1], Keyu Li[2], and Yuhang Jiao[3]

The Central University of Finance and Ecomomics, Beijing, China
{2016312336,2016312338,jiaoyuhang}@email.cufe.edu.cn

Abstract. Many real-world data can be represented by the graph structures, but we find that in the field of analyzing time series data, previous works utilizing graphs for analysis are less or not effective. These previously methods have altogether led to the loss of part of intrinsic information, which indicates unsatisfying suitability between extracted features and real-world theories. In this paper, we propose a Convolutional AutoEncoder(CAE) Correlation Graph Kernel method, which includes a) CAE to extract features from the graph, b) the approximate von Neumann entropy and the Shannon entropy to measure the consistency in the reconstructed graph. Our method can achieve more effective extraction of potential features from graphs of time series data by imitating human vision. In the experiment, we choose historical data of Chinese stock market as the dataset. The experimental result shows that our model performs better than previous works.

Keywords: Chinese Financial Market · Convolution Autoencoder · Shannon Entropy · Neumann Entropy

1 Introduction

Graph is a kind of data structures that uses nodes and edges to model a set of objects and their relationships. And representation based on graphs is a useful tool to analyze complex real-world data. For instance, Duck Hoon Kim et al. [1] have used graphs to present both 2-D and 3-D shapes, aiming at establishing a new shape decomposition scheme. Hamilton et al. [2] have adopted graphs to represent online social networks to forecast which community that different posts belong to.

Compared with simple structures like vectors, graphs can glean more complex features of real-world data, such as time series, social networks [3], physical systems [4], knowledge graphs [5] and many other research fields [6]. But if extraction of features in graph is not adequately effective, some intrinsic information would be lost during this process. E.g., vectors cannot represent the correlations between pairwise financial time series [7].

In aspect for time series data, there have been a large number of previous works to analyze and process it [8–10]. However, there are merely a couple of works using graphs to represent time series data, all of which usually exploit well-selected data instead of using the whole dataset to perform experimental evaluation.

In this paper, we propose a method for analyzing Chinese financial time series data by using deep learning and kernel methods. We intend to convert the raw data (i.e., time series) to images and the CAE is used to extract its features. Then these extracted features are used to reconstruct graphs to describe deep dynamic financial network.

Overall, our contribution is to provide a distinct framework that combines the Convolutional AutoEncoder, which stimulates human vision, followed by the graph kernel method to deal with time series data. The later experiments on Chinese financial stock market show impressive performance of our model that it can extract the intrinsic information of the financial stock market better than other previous methods do.

2 Related Work

Deep Feature Learning These years, CNN [11] is the most widely applied model in supervised learning method. Since AlexNet [12] won the championship in ImageNet 2012, with the rapid development of GPU recent years, CNN has been used in almost every computer vision task. In recent years, many researchers have put forward different CNN models that have high performance growth, such as Inception [15], VGGNet [16], and recent ResNet [17].

On the other hand, AutoEncoders and its varients are important parts of unsupervised learning methods. The AutoEncoder can learn an valid representation of the input without labels. This representation can denote all the features in the input. Like Variational AutoEncoder [13] and Denoising AutoEncoder [14] , they show strong feature learning ability in many tasks.

However, the original AutoEncoder cannot handle image very well, so we apply a Convolutional AutoEncoder(CAE) [18], which is the developed version of the original AutoEncoder, and uses convolution and pooling layers instead of the fully-connected layers to handle this drawback. Thus the space for storing parameters is greatly reduced and the model can extract features from images effectively.

Graph Kernel Method Kernel method bases on statistical learning theory and kernel method theory [19], and it is an efficient way to deal with non-linear pattern analysis problem. However, it not only limits the length of the vector must be fixed, but also can not reflect the structural information between the data. So, the traditional kernel function cannot be applied to the graph structure data.

To overcome the aforementioned drawback, graph kernel functions have been proposed [22] and is under heated discussion these years [20, 21]. It retains all the advantages of vector-based kernel functions and also reflects the structural information of graph data in high-dimensional Hilbert space. Using it, we can measure the similarity between a pair of graphs effectively [23]. At present, most of the graph kernel functions are based on the R-convolution theory put forward by professor Haussler of Stanford University in 1999 [22], which is the current method for constructing graph kernel functions.

Since we get the dynamic graphs after utilizing dot product and want to visualize results while retaining information of graph structure data, our strategy is utilizing graph kernel method along with Principal Component Analysis (Kernel PCA) [24]. Thus, we can achieve visual deep representation of financial time series data and analyze the stock market effectively.

3 CAE Correlation Graph Kernel

3.1 CAE Architecture

Our CAE architecture is shown in Fig 1, which is drawn by PlotNeuralNet[1].

All the convolution layers except Conv4 and Deconv4 use ReLU activation, convolution filter with size of 3 × 3 and stride of 1 as parameters. The Conv1 has 16 convolution filters first and followed by 32 convolution filters. Pool1 has 2 × 2 max pooling filter with stride 2 × 2.

The Conv2 has 64 convolution filters, followed by Pool2. The size of pooling filter in Pool2 is 4 × 4. The output size of Pool2 is 40 × 40 × 64. We apply 128 convolution filters on Conv3. Pool3 has the same parameter as Pool1.

In the last part of encoding, we apply 256, 5 × 5 convolution kernels on Conv4 to get the Compressed Representation, and the decoding process is the opposite of the encoding.

The aim of encoding is to extract the effective representation from the K-line chart step by step, and the subsequent decoding process is to use the Compressed Representation to reconstruct the K-line chart. In order to achieve a more accurate reconstruction of the picture, the Loss function is defined by

$$Loss = \arg\min_{\sigma_1,\sigma_2} (\sigma_1 - \sigma_2)^2$$

where σ_1 denotes the orgin K-line chart photo and σ_2 denotes the reconstructed corresponding photo. Our implementation is in TensorFlow [25] and we use the Adam optimizer [26] with its TensorFlow default parameters,

[1] GitHub Link: https://github.com/HarisIqbal88/PlotNeuralNet

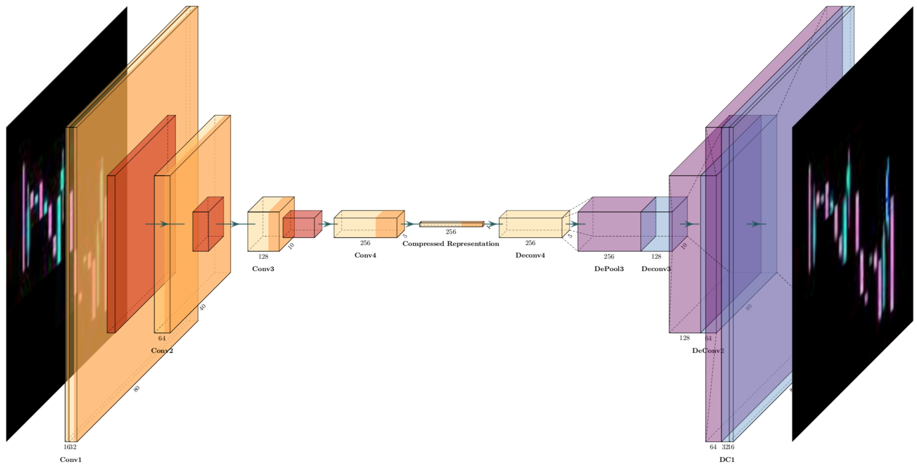


Fig. 1: Our CAE Architecture

3.2 Correlation Matrix for Stocks

We use vectors of the Compressed Representation to compute the Correlation Matrix M, which can be computed directly as

$$M_{(i,j,k)} = \begin{cases} 0 & \text{if } i = j, \\ \frac{v_{i,k} \cdot v_{j,k}}{||v_{i,k}|| \times ||v_{j,k}||} & \text{otherwise.} \end{cases}$$

where $v_{i,k}$ represents the vector of i-th stock in k-th day and $M_{(i,j,k)}$ denotes the similarity between vector i and vector j.

3.3 Entropy Measures for Graphs

In this part, we review two concepts of graph entropy measures: The approximate von Neumann entropy [27], the Shannon entropy associated with steady state random walks [28]. For a graph which is denoted as $G(V, E)$, let V to be the vertex set and $E \subseteq V \times V$ is the undirected edge set. About the adjacency matrix A of the graph $G(V, E)$, it is a $|V| \times |V|$ symmetric matrix. Each element of this matrix satisfies

$$A_{i,j} = \begin{cases} 0 & \text{if } (v_i, v_j) \in E, \\ 1 & \text{otherwise.} \end{cases}$$

. For graph G, the vertex degree matrix D is a diagonal matrix, and it's elements are defined by

$$D(v_i, v_j) = d(v_i) = \sum_{v_j \in E} (i, j)$$

Definition 1 (The Approximate Von Neumann Entropy) For the graph $G(V, E)$ which is based on the definition in [1], we can compute the approximate von Neumann entropy for the graph $G(V, E)$

$$H_{VN}(G) = 1 - \frac{1}{|V|} - \sum_{(v_i,v_j)\in E} \frac{1}{|V|^2 d(i)d(j)}$$

And each edge $(v_i, v_j) \in E$ is indicated by the adjacency matrix A.

Definition 2 (The Shannon Entropy) In the graph $G(V, E)$, for each vertex $v_i \in V$, the probability of a steady state random walk on $G(V, E)$ visiting v_i is

$$P(i) = \frac{d(i)}{\sum_{v_j \in V} d(j)}$$

From this probability distribution P, we can compute the Shannon entropy directly as

$$H_S(G) = -\sum_{i=1}^{|V|} P(i) \cdot \log P(i)$$

Based on the n^2 elements of the graph adjaceny matrix, the vertex degree statistics are computed. Thus, these two entropy measures require computational complexity $O(n^2)$ in the premise that n is the vertex number. Besides, rich edge connectivity information of these graphs in terms of the vertex degrees are shown for both of the entropy measures.

In this way, we can obtain time-varying graph based on these two entropy which are calculated in this part.

3.4 Visualization through Kernel Method

In order to achieve a unified visualization, we use dot product which is defined as

$$M_{i,j} = T(N_i) \cdot T(N_j) + T(S_i) \cdot T(S_j)$$

$$T(k_i) = \frac{k_i - min(k)}{max(k) - min(k)}$$

to recreate the graph after computing the approximate von Neumann entropy and the Shannon entropy, then we use kernel principal component analysis (Kernel PCA) [29] to embed data to 3-dimension to draw the figure. N_i denotes the approximate von Neumann entropy of i_{th} day, S_i denotes the Shannon entropy of i_{th} day.

4 Experiments

In this section, we use historical data of Chinese stock market to measure and evaluate whether our model can effectively extract features from time series data, especially during financial crisis.

4.1 Dataset Preprocessing

We select Chinese stock market data of a decade from 2009 to 2018. All original data is downloaded by Yahoo Finance API. A small portion of stock in original data which have not been traded for specific amount of days is intentionally removed. After this data processing, transaction data of 805 stocks in ten years which was used as dataset was obtained.

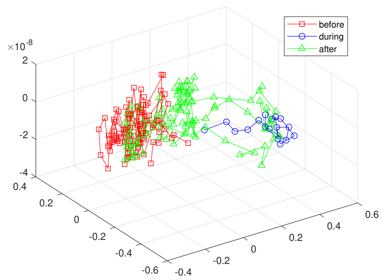
We use a time window for 20 days to draw K-line chart and move it day by day in chronological order. Thence, 2,400 K-line charts for each stock was gleaned and used as input for our model.

4.2 Financial Data Analysis

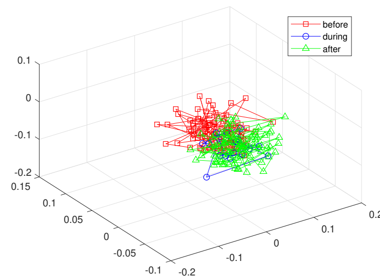
In order to reflect the effect of our model, we chose several different methods for comparison, such as Dynamic Time Warping(DTW) [30] kernel, which is a widely-used sequence kernel to analyze time series data, and Radial Basis Function(RBF) kernel which utilizes default parameters in scikit-learn [31]. In our controlled experiments, all methods exploited the kernel PCA in scikit-learn and especially, our method, along with DTW method, has used the precomputed kernel PCA as parameter.

On July 27_{th}, 2015, there was a huge fluctuation that the market reached more than 6% decline on that day. Therefore, we chose 100 days before and after this day to draw the figure and 8 days before and after this day was regarded as financial crisis. Days during financial crisis were marked blue on the figure while days before and after this financial crisis were marked red and green respectively. The results of 4 different methods are shown in Fig 2.

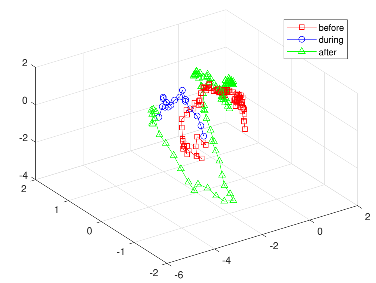
Analyzing stock market fluctuation with economic principles found that stock market retained relative stable before and after financial crisis, which suggests high similarity between features extracted from these two periods. Correspondingly, the points before and after crisis, which respectively marked red and green, in plots are supposed to be assembled. Moreover, as there were some great fluctuations in the stock market that happened suddenly, the blue area, in this case, will be more clearly separated from others. But more frequently, fluctuations in stock market occur more gradually and last longer, showing the piecemeal increasing and then declining deviation from normal, and finally stock market stabilized. Under this circumstance, red area in graph is supposed to gradually get close to blue area, and thereafter blue area is supposed to shift to the green area little by little. Namely, more red or green dots in vicinity of blue area demonstrates longer duration of a fluctuation in stock market.



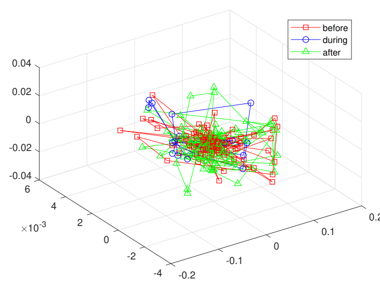
(a) CAE Correlation Graph Kernel



(b) Neighborhood Hash kernel



(c) DTW Method

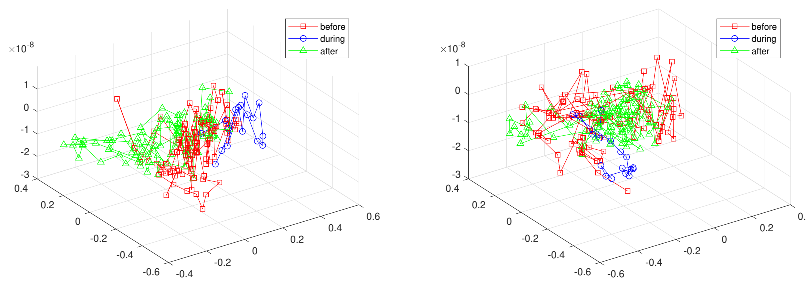


(d) RBF Kernel

Fig. 2: Kernel PCA plots of four different methods on financial crisis data in July 27_{th} 2015. Before, during and after financial crisis marked as red, blue and green respectively.

In the first plot of Figure 2, it is presented that blue areas are clearly separated, and simultaneously red and green areas share a large overlap. This result is consistent with the aforementioned expected results from economic analysis. In regard to the RBF Kernel, points in three time periods are mixed together, which means that this method cannot distinguish the characteristics of these three different time periods. Meanwhile, in spite of effective separation of blue dots by DTW Method and NHK Method, at the same time, red dots are clearly isolated from green dots as well. This phenomenon is incongruent with aforementioned expected results from economic analysis.

In Figure 3, we present the performance of our model on different stock market volatility, which are consistent with the expected results previously analyzed through economic principles. These results show that our model can extract the intrinsic correlation of the whole stock market changes from the K-line chart and correctly separate the different states of the stock market. In Figure 3, our model was applied to analysis of other two financial crisis, the result of which was consistent with aforementioned expected results from economic analysis. The satisfying performance substantiates the capability of our model in extracting underlying information from K-line chart as well as correctly separating stock market suffering from financial crisis from normal state.



(a) 12/11/2010 (b) 25/02/2016

Fig. 3: Kernel PCA plots of the CAE Correlation Graph Kernel on other financial crises.

5 Conclusion

In this paper, we propound a method for extracting the intrinsic characteristics of a stock market from the historical data. This method includes the following steps: a) using history data of stock market to draw K-line charts, b) utilizing the CAE to extract features from the K-line charts, c) reconstructing a daily correlation matrix through daily characteristics of each stock, d) exploiting the Shannon entropy and the approximate von Neumann entropy to analyze the differences in daily stock movements, e) visualizing results by using dot product and Kernel PCA. This approach allows us to analyze the intrinsic characteristics of market's movements in a brand-new way. In the experiment above, we apply our model to a set of Chinese stock market data and contrast with other methods. It corroborates that our method harbors higher effectiveness than other methods do for analyzing the intrinsic characteristics of market trends. s well, our results accord with economic principles, which implies availability of our model.

References

1. Duck Hoon Kim, II Dong Yun, Sang Uk Lee. "A new shape decomposition scheme for graph-based representation." *Pattern Recognition*, 38(5):673-689
2. Hamilton, William L, R. Ying, and J. Leskovec. "Inductive Representation Learning on Large Graphs." *Neural Information Processing Systems* 2017:10251035.

3. W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *neural information processing systems*, pages 1024–1034, 2017
4. P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502– 4510, 2016.
5. T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence,IJCAI-17*,pages1802–1808, 2017.
6. H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. *arXiv preprint arXiv:1704.01665, 2017*
7. Bonanno, G., et al. "Networks of equities in financial markets." *The European Physical Journal* B 38.2(2004):363-371.
8. Das M., Ghosh S.K.(2017) "Spatio-Temporal Prediction of Meteorological Time Series Data: An Approach Based on Spatial Bayesian Network(SpanBN)." *Pattern Recognition and Machine Intelligence. PReMI 2017.* Lecture Notes in Computer Science, vol 10597. Springer, Cham.
9. J. Grabocka et al. "Learning time-series shapelets". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2014*, pp. 392–401
10. B. Li, N. Chen, and U. Schlichtmann. "Statistical Timing Analysis for Latch-Controlled Circuits With Reduced Iterations and Graph Transfor mations". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 31.11 (2012)*, pp. 1670–1683
11. LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE.*
12. Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.
13. Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
14. Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML. ACM.*
15. Szegedy,C.;Liu,W.;Jia,Y.;Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR.*
16. Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*
17. He,K.;Zhang,X.;Ren,S.;andSun,J.2016. Deep residual learning for image recognition. In *CVPR.*
18. Masci, J.; Meier, U.; Cires¸an, D.; and Schmidhuber,J. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011 52–59.*
19. FEEDMAN D A. "Statistical models: theory and practice, revised edition[M]." Cambridge: Cambridge University Press, 2009.
20. Kashima, H. "Marginalized Kernels between Labeled Graphs." Proceedings of the Twentieth International Conference on Machine Learning 2003:321–328.
21. Vishwanathan, S. V. N, et al. "Graph Kernels." Journal of Machine Learning Research 11.2(2008):1201-1242.
22. Bai, Lu, et al. "An Aligned Subtree Kernel for Weighted Graphs." International Conference on Machine Learning 2015:30-39.
23. Haussler, D. "Convolution kernels on discrete structures." Tech Rep 7(1999):95114.

24. Schlkopf, Bernhard, A. Smola, and K. R. Mller. ”Nonlinear component analysis as a kernel eigenvalue problem.” *Neural Computation* 10.5(1998):1299-1319.
25. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
26. Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
27. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph characterizations from von Neumann entropy. *Pattern Recogn. Lett.* 33(15), 1958–1967 (2012).
28. Bai, L., Hancock, E.R.: Depth-based complexity traces of graphs. *Pattern Recogn.* 47(3), 1172–1186 (2014).
29. Schlkopf, Bernhard, A. Smola, and K. R. Mller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10.5(1998):1299-1319.
30. Cuturi, Marco. ”Fast Global Alignment Kernels.” *International Conference on Machine Learning* 2011:929-936.
31. Pedregosa, Fabian, et al. ”Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12.10(2012):2825-2830.