Chaos, Solitons and Fractals 000 (2017) 1–6

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Frontiers

Dynamics of cluster structure in financial correlation matrix☆

Chun-Xiao Nie

Department of Finance, School of Business, East China University of Science and Technology, Shanghai 200237, China

ARTICLE INFO

Article history:
Received 2 April 2017
Revised 15 May 2017
Accepted 28 May 2017
Available online xxx

2010 MSC:
00-01
99-00

Keywords:
Correlation dimension
Cluster algorithm
Financial network
Financial crisis
Correlation coefficient matrix

ABSTRACT

The correlation dimensions in the financial market are calculated and used as a measure to study the cluster structure in the correlation coefficient matrix. First, based on the existing model, we present a toy model. Using the model-generated data, we find that the clearer cluster structure corresponds to a smaller dimension. It implies that the correlation dimension can be used as a measure of the cluster structure in the correlation coefficient matrix. Finally, we use the algorithm to compute the clusters in the real market and verify the previous empirical evidence. The results show that the cluster structure in the financial correlation coefficient matrix may change with time. The correlation dimension is smaller after the financial crisis, indicating that the cluster structure is clearer after the financial crisis.

1. Introduction

In recent years, the clustering problem of time series has been widely studied and used in financial time series analysis. In the past studies, some clustering algorithms for time series have been proposed [1,2].

In this paper, we study the dynamics of the cluster structure in the financial correlation coefficients matrices. In our study, the correlation dimension is used to detect the change of cluster structure. Initially, the correlation dimension is introduced in chaos theory [3–6]. As a commonly used fractal dimension, it has been widely used in other fields, such as financial time series analysis [7–9] and machine learning [10,11]. In financial time series analysis, the correlation dimension is used to analyze the chaos of financial time series [7–9], and in machine learning, it is used to calculate the intrinsic dimension of the data set [10,11].

On the other hand, in recent years, the financial threshold networks have been extensively studied [12–20]. For example, for a correlation coefficient matrix, a non-diagonal element larger than a certain threshold is converted to 1, otherwise it is converted to 0 and the diagonal element is converted to 0, so that for a threshold, a 0–1 matrix is constructed and is used as the adjacency matrix of the threshold network. This approach has been used in the study of markets in different countries [12–14]. In previous studies, different structures, such as degree distributions [12–20] and cliques [12,18,20], are studied. In the threshold method, the method of converting the correlation coefficient matrix or distance matrix into an adjacency matrix is similar to the method in the correlation dimension.

In particular, recently, the author has used the correlation dimension to analyze the dimensionality of the stock set [21]. In [21], the intuitive meaning of the dimension is the change rate of the average degree of the threshold network. Studies have shown that large fluctuations in the market (such as during a financial crisis) correspond to a significant reduction in dimensions [21].

In previous studies, different clustering methods have been used to detect the cluster structure of the correlation matrix, such as the hierarchical clustering method based on the minimum spanning tree used in [22], and in [23], the author applied $k-means$ and *FCM* and *SOM* algorithms in the financial market. In [24], the authors reviewed some time series-based clustering algorithms used in financial markets. These clustering methods have been used in portfolios or analysis of fund styles and so on [22–29]. Due to the characteristics of the financial time series, the cluster structure in the correlation coefficient matrix is not static, as shown in [24]. Thus, the dynamics of the cluster structure needs to be studied.

In previous studies, factor models are often used to characterize financial data [30–32]. In the literature [32], based on random matrix theory, the correlation coefficient matrices of the factor models are studied. In our study, we will use the factor model to generate a correlation coefficient matrix with a clear cluster structure. Then,

☆ Fully documented templates are available in the elsarticle package on CTAN.
E-mail address: niechunxiao2009@163.com

http://dx.doi.org/10.1016/j.chaos.2017.05.039
0960-0779/

in order to study the dynamics of the cluster structure, we will use the correlation dimension as a detection tool. We first construct a toy model based on the model proposed in [32] and generate some correlation coefficient matrices with some clear clusters. Then we can study the relationship between cluster structure and dimension. Secondly, we will use clustering algorithm to detect clusters in the correlation coefficient matrix in real market data. The results are then compared to model-based results. In this paper, since the method in [21] is used to calculate the correlation dimension, we already know that there is a relationship between the dimension and the average degree of the threshold network. In order to make different methods can be combined, here, a time series clustering algorithm based on the threshold network is used [33]. In this algorithm, the threshold network is constructed, and the community detection algorithm is used to detect the clusters in the network, which are corresponding to clusters in the original matrix [33]. At present, a large number of community detection algorithm has been proposed [34–36], this paper will use the classic algorithm proposed by Newman [37].

2. Data and methods

2.1. Data

This paper will use the data generated by the model and the real market data. We use the closing price data for the S&P 100 index constituents from 2005/1/3 to 2015/4/24. Missing stocks were removed, with a total of 93 stocks selected. The model we use will be described in Section 2.2, with specific data details described in the Results section.

Here, we assume that there are N time series, each time series corresponds to a node, in this way, the stock i (or a simulated time series) corresponds to node v_i, so that an N*N matrix corresponds to a network $W = (V, T)$, where $V = \{v_i, i = 1 \dots N\}$ is the node set, T is the adjacency matrix.

In our study, we will use the correlation coefficient matrix of stock returns, the correlation coefficient between stocks i and j as shown in Eq. (1), where $R_i(t) = log(P_i(t+1)) - log(P_i(t))$, and $\{P_i(t)\}$ is the price series of stock i. On the other hand, we need to compute the correlation dimension, and the stock distance proposed in [22] is used, as shown in Eq. (2).

$$\rho_{ij} = \frac{\langle R_i(t)R_j(t)\rangle - \langle R_i(t)\rangle\langle R_j(t)\rangle}{(\langle R_i(t)^2 - \langle R_i(t)\rangle^2\rangle\langle R_j(t)^2 - \langle R_j(t)\rangle^2\rangle)^{1/2}} \tag{1}$$

$$d_{ij} = [2 * (1 - \rho_{ij})]^{1/2} \tag{2}$$

2.2. Toy model

Firstly, based on the model in [32], we give a toy model. The toy model can generate a correlation matrix with clear cluster structure. In [32], a general model is constructed, as shown in Eq. (3). In Eq. (3), there are N variables and K factors. The $\gamma(i, j)(j \neq 0)$ is the weight used to describe the effect of the j factor $f_t(j)$ on the variable $x_t(i)$. The $\gamma(i, 0)$ is also a positive constant that describes the random perturbation term. In the model, the relationship between the factor $f_t(i)$ and $f_t(j)$ is $< f_t(i) f_t(j) >= \delta_{ij}$. The $\epsilon_t(i)$ is the standard Gaussian noise term, and $< \epsilon_t(i)\epsilon_t(j) >= \delta_{ij}$. In addition, the relationship between $< f_t(i) >$ and $\epsilon_t(i)$ is $< f_t(i)\epsilon_t(j) >= 0$.

$$x_t(i) = \sum_{j=1}^{K} \gamma(i, j) f_t(j) + \epsilon_t(i)\gamma(i, 0), i = 1 \dots N \tag{3}$$

Next, based on Eq. (3), we present a simplified model, as shown in Eq. (4). In this model, the number of factors in each time series is 1, but different clusters correspond to different factors, and there is correlation between these factors. In the following, we describe the steps of the model in detail. We assume that there are m clusters and all the factors are standard normal random variables. So that m factors need to be generated, and the correlation coefficient matrix between these m factors is denoted as $G = [G(i, j)]$, where the $G(i, j)$ is the correlation coefficient between f_t^i and f_t^j. In the model, the number of variables in different clusters can be different. Assuming that the number of variables in cluster q is n_q, the total number of nodes generated by the model is $N = \sum_{j=1}^{m} n_j$. We generate the random numbers $\{\gamma^q(i)\}$ as the coefficient of the factor f_t^q, and $\{\gamma^q(i)\}$ satisfies the uniform distribution on $[u_1^q, u_2^q]$. In our model, $\epsilon_t^q(i)$ is also a Gaussian random term with zero-mean, but the standard deviation σ^q cannot be 1. The values of σ^q for different clusters may differ from each other. In addition, in the model, $< \epsilon_t^{q_1}(i), f_t^{q_2} >= 0$, and when $q_1 \neq q_2$, $< \epsilon_t^{q_1}(i), \epsilon_t^{q_2}(j) >= 0$.

Thus, in the model here, the following parameters need to be set, respectively, the correlation coefficient matrix G, the number set $N_{cluster} = \{n_q, q = 1 \dots m\}$, a set of intervals $I = \{[u_1^q, u_2^q], q = 1 \dots m\}$, and a set of standard deviation values $S = \{\sigma^q, q = 1 \dots m\}$.

$$x_t^q(i) = \gamma^q(i) f_t^q + \epsilon_t^q(i), q = 1 \dots m, i = 1 \dots n_q \tag{4}$$

First, we need to generate the time series corresponding to these m factors. Here, we use Cholesky decomposition, as shown in the literature [38]. The main steps to generate model data are as follows.

Step.1 Based on the correlation coefficient matrix G and Cholesky decomposition, G can be decomposed into $G = G_1 G_1^t$, where t means matrix transpose.

Step.2 By the following Eq. (5), m time series are generated, where E_ϵ is the $m \times 1$ matrix, and the elements of E_ϵ are unrelated standard normal random variables. The element $\Phi(q, 1)$ of the random vector Φ corresponds to the factor f_t^q. We can verify $E[\Phi\Phi^t] = G_1 E[E_\epsilon E_\epsilon^t] G_1^t = G_1 G_1^t = G$.

$$\Phi = G_1 E_\epsilon \tag{5}$$

Step.3 Based on the parameters in sets $N_{cluster}$, I and S, we generate N time series using Eq. (4) and $\{f_t^q, q = 1 \dots m\}$.

Formally, the model we use here is similar to the one-factor model, but here, the different clusters correspond to different factors, and there is a correlation between the different factors. In addition, we use the random perturbation term to control the clarity of the cluster. This allows us to study the relationship between the clarity of the cluster structure and the correlation dimension. In contrast to the proposed model in [32], there is a correlation between the factors of the proposed toy model, and the number of factors is the number of clusters.

2.3. Algorithm for detecting clusters

In this paper, we use the method in [33] to detect clusters in the correlation coefficient matrix. The core idea of this method is to detect the communities in the threshold network by the community detection algorithm, and then the communities correspond to the clusters. Note that in this method, the variable $x_t(i)$ is assigned to the node v_i in the threshold network, assuming a total of N time series, then a node set $V = \{v_i, i = 1 \dots N\}$ is given. In the original paper [33], the threshold network thus generated is called ϵ-nearest neighbor network ($\epsilon - NN$). The main steps of the method are as follows.

Step.1 A threshold r is set and the distance matrix D is converted to a threshold network $W_r = \{V, D_r\}$, where $D_r(i, j) = 1$, if $D(i, j) \leq r$, otherwise $D_r(i, j) = 0$. In addition, in order to avoid the existence of self-loop in the network, we define $D(i, i) = 0$.