Stock market co-movement assessment using a three-phase clustering method

Saeed Aghabozorgi *, Ying Wah Teh

Department of Information System, Faculty of Computer Science & Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Keywords:
Stock market
Co-movement
Data mining
Clustering
Time-series

ABSTRACT

An automatic stock market categorization system would be invaluable to individual investors and financial experts, providing them with the opportunity to predict the stock price changes of a company with respect to other companies. In recent years, clustering all companies in the stock markets based on their similarities in the shape of the stock market has increasingly become a common scheme. However, existing approaches are impractical because the stock price data are high-dimensional data and the changes in the stock price usually occur with shift, which makes the categorization more complex. Moreover, no stock market categorization method that can cluster companies down to the sub-cluster level, which are very meaningful to end users, has been developed. Therefore, in this paper, a novel three-phase clustering model is proposed to categorize companies based on the similarity in the shape of their stock markets. First, low-resolution time series data are used to approximately categorize companies. Then, in the second phase, pre-clustered companies are split into some pure sub-clusters. Finally, sub-clusters are merged in the third phase. The accuracy of the proposed method is evaluated using various published data sets in different domains. We show that this approach has good performance in efficiency and effectiveness compared to existing conventional clustering algorithms.

1. Introduction

There are many works related to stock market analysis including clustering (Durante and Foscolo, 2013; Nanda et al., 2010) and prediction (Qu et al., 2012; Svalina et al., 2013; Zarandia et al., 2009). Clustering is a data mining technique in which similar data are automatically placed into related groups without advanced knowledge of the group definitions. Clustering of companies in the stock market is very useful for managers, investors, and policy makers. It can be performed based on several factors, such as the size of the companies, their annual profit, and the industry category. For example, Nanda et al. (2010) used the returns of the stock for variable time intervals along with the validation ratios to cluster the companies listed in Bombay Stock Exchange (BSE). However, these features usually change over time; thus, they are improper for categorization purposes. Industry-based categorization is also not preferable due to evidence that financial analysts are biased by industry categorization (Krüger et al., 2012). Conversely, the closing prices of stocks related to each company are stored as time series data. Companies can be categorized by the clustering of their stock price time series. Clustering companies based on the time series of their stock price is particularly advantageous in co-movement assessment. Identifying homogeneous groups of stocks where the movement in one market affects stock prices in another market is called co-movement. The literature shows that the movement of a stock market in a country is affected by the movement of other stocks in that country or in other regions (Antoniou, 2003; Collins and Biekpe, 2003; Masih and Masih, 2001). As a result, numerous studies have been performed on the recognition of co-movements between different countries (Graham and Nikkinen, 2011; Norden and Weber, 2009; Rua and Nunes, 2009). For example, examining the co-movement of the stock markets of Taiwan and Hong Kong (Liao and Chou, 2013) or co-movement of Asia–Pacific with European and US stock market returns (Loh, 2013). However, most of these studies consider the co-movement of the stock market between different regions or countries but not among different industries or companies in a stock market, such as the Malaysian stock market. Assessment of the stock market co-movement between companies in a stock market can be very helpful for predicting the stock price, based on the similarity of a company to other companies in the same cluster. That is, given two stock market time series, A and B, we can show that whenever the price of A drops, the price of B will also drop, and vice versa. Therefore, clustering of time series related to the stock price of companies can answer the following questions:
1. Can a company category be determined given a historical record of its stock price?
2. How are the movements of a specific company in stock markets across the various companies?
3. How can the stock price be predicted based on its similarity to other co-movement companies?

Researchers have shown that clustering using the best-known conventional techniques, such as partitional and hierarchical algorithms, generate clusters with acceptable structural quality and consistency, and are partially efficient in terms of execution time and accuracy for static data (Jain et al., 1999). However, classic machine learning and data mining algorithms do not work well for time series data due to their unique structure (Lin et al., 2004). In effect, the high dimensionality, very high feature correlation, and (typically) the large amount of noise that characterize time series data present difficult challenges for time series clustering (Keogh and Kasetty, 2003; Lin et al., 2004). Accordingly, massive research efforts have been made to present an efficient approach for time series clustering. However, focusing on the scalability of these methods to deal with time series data has come at the expense of losing the usability and effectiveness of clustering (Ratanamahatana, 2005). For example, Euclidean distance (ED) is adopted as a distance metric on most of the existing works due to its high efficiency, whereas it is not sufficiently accurate because it is only suitable for calculating the similarity in time (i.e., similarity in each time step) (Bagnall and Janacek, 2005; Ratanamahatana et al., 2005).

Fig. 1 shows the hierarchy clustering of several companies based on the similarity in time. Focusing on the inter-temporal co-movement of the daily closing prices in one month for two stocks, company B and company C may seem very different. In time series clustering analysis, these two stocks may be surmised to belong in two different clusters or that no linkage exists between them. However, a precise look at the stock price of these two stocks shows that they are co-moving but with some shifts or daily lead-lag relationships. Hence, the clustering results of these two stocks based on the similarity in time may fail to provide insightful information.

In this paper, a new 3-Phase Time series Clustering model (3PTC) is proposed, which can construct the clusters based on similarity in shape. This model facilitates the accurate clustering of time series data sets and is designed specifically for very large time series data sets. In the first phase of the model, data are pre-processed, transformed into a low dimensional space, and grouped approximately. Then, the pre-clustered time series are refined in the second phase using an accurate clustering method, and are represented by some prototypes. Finally, in the third phase, the prototypes are merged to construct the ultimate clusters. To evaluate the accuracy of the proposed model, the 3PTC is tested extensively using published time series data sets from diverse domains.

This paper contributes to the existing literature by proposing a new model of time series clustering which (1) is more accurate than conventional approaches, (2) is scalable (on large datasets) due to the use of multi-resolution time series in different levels of clustering, and (3) can overcome the limitations of comparative clustering algorithms in finding the clusters of similar time series in shape. This salient feature is very advantageous for co-movement assessment in stock markets.

The rest of this paper is organized as follows. In Section 3, the related works are described. The proposed model is explained in Section 4. In Section 5, the algorithm is applied on diverse time series data sets, and the experimental results are reported. In Section 6, conclusions and future perspectives are drawn.

2. Literature review

A current research issue in the finance literature is co-movements of the world’s national financial market indexes (Antonakakis et al., 2013; Chen and Wu, 2013; Chow et al., 2011; Graham et al., 2012; Liao et al., 2011; Madaleno and Pinho, 2012; Wahal and Yavuz, 2013). In recent years, many researchers have investigated the co-movement in stock markets by different data mining approaches. For instance, Liao and Chou (2013) employed association rules and clustering algorithm to investigate the co-movement in the Taiwan and China (Hong Kong) stock markets. They categorized the stock indexes into thirty clusters to perceive the behavior of stock index associations. In another study (Dutt and Mihov, 2013), the authors employed monthly stock indices to construct pairwise correlations of returns. They explained these correlations with risk-adjusted differences in industrial structure across 58 countries. The result of their study indicates that countries with similar industries exhibit higher stock market co-movements. Graham and Kiviat (2013) investigated the short term and long term co-movement of MENA (Middle East and North Africa) region stock markets with the U.S. stock market and the regional co-movement among these markets. In this work, the wavelet squared coherency method was employed to examine the co-movement of stock markets. Likewise, using the wavelet coherence approach, Loh (2013) investigated the co-movement of 13 Asia-Pacific stock market returns with stock market of European and US stock market returns. Durante and Foscolo (2013) proposed an index to measure the contagion effects between a group of markets. This index was used by the authors to derive a...
dissimilarity matrix for different markets. Then, applying crisp and fuzzy clustering methods on the dissimilarity matrix, they created a clustering of European stock market indices to assess their behavior in the recent years.

One of the approaches for detecting the co-movement is clustering approaches. To make the clusters, similar time series instances are grouped by a conventional clustering algorithm. This process is known as “whole sequence matching” in which entire lengths of time series are considered during distance calculation. However, time series are usually converted into a proper space for financial systems. The perceptually important points (PIPs) posed by Chung et al. (2001) is one of the first motivations for a group of works on representation methods. The concept of PIPs was introduced for financial time series analysis to hold the important points and delete others. Later, Fu et al. (2007) used the concept of PIPs to propose a new representation where the important points of time series are collected by measuring the distance between the time series points and their trend. In this representation, the point that has the largest distance is chosen. In another work (Bao, 2007), the authors suggest a generalized model in financial time series using turning points in financial data. Turning points are the important points presented in the study of Fu et al. (2007). In a study (Lkhagva et al., 2006), the authors consider the importance of accuracy in financial systems and propose the Extended Symbolic Aggregate Approximation (ESAX), which is a symbolic representation customized for financial time series. Later, in another article (Liu and Shao, 2009), the authors present a similarity measure based on SAX for financial time series. They focus on the shortage of SAX representation, that is, consideration for the dynamic information about trends is lacking. However, as all these works suggest, the authors attempt to address the problem of overlooking data only by representing time series with some constraints regardless of clustering algorithm.

Clustering of time series in terms of adopted algorithm is classified broadly into partitioning, hierarchical, model-based, density-based, and grid-based clustering algorithms. In hierarchical clustering of time series, the nested hierarchy of similar groups is generated based on a pair-wise distance matrix of time series (Vlachos et al., 2003). Hierarchical clustering has significant visualization power (Keogh and Pazzani, 1998) and does not require the number of clusters as an initial parameter. This characteristic leads to the use of hierarchical time series clustering to a great extent (Hirano and Tsumoto, 2005; Oates et al., 2000). However, hierarchical clustering cannot deal effectively with large time series data sets (Wang et al., 2006) due to its quadratic computational complexity. In contrast, partitional clustering approaches, such as k-Means and k-Medoids, are very fast compared to hierarchical clustering (Bradley et al., 1998), and this characteristic has made them very suitable for time series clustering. Therefore, they have been used in many works either in a “crispy” manner (Bagnall and Janacek, 2005; Beringer and Hullermeier, 2006; Guo et al., 2008; Hautamaki et al., 2008; Lin et al., 2004; Ratanamahatana and Nienattrakul, 2006) or “fuzzy” manner (Aghabozorgi et al., 2011; Alon and Sclaroff, 2003; Nasibova and Pekerb, 2011; Tran and Wagner, 2002). However, they have either the problem of constructing an effective prototype (Nienattrakul and Ratanamahatana, 2007) or determining the number of initial clusters (Antunes and Oliveira, 2001; Wang et al., 2006) in time series clustering. Model-based clustering assumes a model for each cluster, and finds the best fit of data to that model. A few articles use model-based clustering of time series data (Bagnall and Janacek, 2005; Bicego et al., 2003; Biernacki et al., 2000; Corduas and Piccolo, 2008; Hu et al., 2006; Ramoni et al., 2000); however, model-based clustering needs to set parameters, and it is based on user’s assumptions that may be false and result in inaccurate clusters. Moreover, it has a slow processing time (especially neural networks) on large data sets (Andreopoulos et al., 2009). The quota of density-based clustering is only the model proposed by Chandrakala and Chandra (2008) because of its relatively high complexity; to the best of our knowledge, no work in the literature has applied grid-based approaches for clustering of time series.

The overview of all these studies indicates two approaches in dealing with time series clustering problem: (1) reducing the dimension of time series data using a representation method. This approach is very common in the time series mining community because it reduces the effect of noise, allows fitting the data in memory, handles the small shifts, and has a low complexity of distance calculation. However, representation methods suffer from overlooking of data (Lai et al., 2010). This problem is caused by dimensionality reduction (sometimes reduction to very low resolutions of data), which is carried out to remedy the high complexity of using raw time series. (2) In contrast to similarity in time, clusters generated based on similarity in shape (i.e., the time of occurrence of patterns is unimportant), are very accurate and generate meaningful clusters. For instance, elastic methods (Agrawal et al., 1993; Arif et al., 2004) such as Dynamic Time Warping (DTW) (Chu et al., 2002) can compute the similarity in shape between time series. However, this metric cannot be used for large time series data sets (and is not usually used in existing works) because it is very expensive in terms of time complexity.

3. Research design

The 3PTC includes three phases: (1) pre-clustering, (2) purifying, and (3) merging (see Table 1). In the first phase, data are clustered to reduce the search space. In this phase, time series data are used in a low-resolution mode. In the second phase, time series data are used in their high-resolution mode. In this phase, a new approach is designed to split the pre-clusters into some pure sub-clusters. Finally, in the third phase, sub-clusters are merged using an arbitrary scheme. Each phase is described and explained in detail in the following sub-sections.

3.1. Phase 1: pre-clustering (Approximate clustering)

The key idea of pre-clustering is to apply clustering on the low-resolution time series data rather than on the original (raw) time series data set. The major reason for using dimensionality reduced time series in the pre-clustering phase is the problem of disk I/O constraint, specifically in the large data sets. The generic solution for this problem is to create an approximation of the data (Keogh et al., 2000; Lin et al., 2003), which will fit in the main memory, yet retains the essential features of interest. As a result, the whole data can be loaded in the main memory, and the problem at hand is solved approximately. However, the data set size can be very large such that even with dimensionality reduction, it cannot fit in the memory. The solution for this case is utilizing an incremental clustering where clusters are updated (or expanded) incrementally (Aghabozorgi et al., 2012). The second reason for the pre-clustering is the reduction in the search area for the second phase of the 3PTC.

Many studies have focused on representation or dimensionality reduction of time series (Ding et al., 2008; Lin et al., 2007). Considering all these works, Symbolic Aggregate Approximation (SAX) representation is used in the first phase due to its strength in the representation of time series. SAX is defined based on Piecewise Aggregate Approximation (PAA) (Keogh et al., 2000; Yi and Faloutsos, 2000) where PAA is the approximation of time series using the segmentation approach. However, before discretization, the time series are standardized using z-score (z-Normalization) (Han and Kamber, 2011), which make time series invariant to scale and
offset. In PAA, the mean values of the equal-length segmentation of time series are considered as the approximate value of those parts. Given \( F_i = \{ F_{i1}, \ldots, F_{in} \} \) as a time series, PAA then discretizes time series to \( F \), where \( F = \{ f_1, \ldots, f_w \} \). Therefore, each segment of \( F \) (i.e., \( f_i \)) is a real value that is the mean of all data points in the \( j \)th segment of \( F \). Finally, SAX maps each PAA coefficient to a symbol. SAX was developed by Keogh et al. in 2003, and has been used by more than 50 groups in different data mining investigations (Lin et al., 2007). Considering \( F \) as discretized time series by PAA transformation, \( F \), where \( \tilde{F} = \{ \tilde{f}_1, \ldots, \tilde{f}_w \} \), is defined by mapping \( f_i \) to “a” SAX symbols. “a” is the alphabet size (e.g., for the alphabet={a, b, c, d, e, f}, “a”=6) defined by “breakpoints.” Based on the definition by Keogh, a list of numbers \( B = b_1, \ldots, b_a \) is defined as “breakpoints” to determine the area of each symbol in SAX transformation.

To make the pre-clusters, an appropriate distance measure compatible with SAX is also desirable. In the literature, Euclidean (Lai et al., 2010) and minimum distance (MINDIST) (Lin et al., 2007) measures are used to calculate the similarity between two time series represented by SAX. Lin et al. (2007) introduced MINDIST as a compatible distance metric for SAX. However, MINDIST has been introduced to address the indexing problem in time series domain and is not sufficiently accurate for calculating distance among time series in the clustering problem. The reason is that MINDIST considers the distance between the neighbor symbols as zero, and ignores the maxima and minima points of time series. As a result, in this paper, an extended MINDIST, so-called APXDIST, is defined. In APXDIST, the distance between regions is calculated as the distance between the indicators of each region. The indicator of a region is defined in such a way that the closeness of PAA coefficients (in the region) to the indicator is the highest in that region. For defining the indicator, the arithmetic mean of each area (minimum and maximum) is defined as the indicator of the area as the best estimator of the regions as:

\[
Ind_{i} = \frac{\beta_{1} + \beta_{0}}{2} \quad 0 < i < a
\]

where \( \beta_0 \) is the global minimum, and \( \beta_1 \) is the global maximum. Using the indicators, the approximate distance of two time series is calculated, where the distance between the neighbor symbols is more than zero, and the maxima and minima points of the time series are not ignored. Based on this definition, the APXDIST between each pair of symbolized time series is defined as follows:

\[
\text{dis}_{\text{APXDIST}}(F_i, F_j) = \sqrt{\frac{1}{W} \sum_{n=1}^{W} \text{dis}(\text{Ind}_{i}, \text{Ind}_{j})^2}
\]

To cluster the approximated data, k-Modes (Huang, 1997) is used to divide the n time series into k partitions. k-Modes is based on the k-Means family algorithm, and is used for clustering of categorical data. Considering that time series represented by SAX is categorical data in each segment, and k-Modes work finely with categorical data (Huang, 1998), it is fit for pre-clustering. Moreover, similar to other partitioning algorithms, it has a high speed (especially in large data sets) (Huang, 1998) and provides good quality by choosing the centroids on the low-dimension approximation of data (e.g., SAX), which increases the quality in the partitioning clustering (Ding et al., 2002).

### 3.2. Phase 2: purifying and summarization

This phase of the 3PTC mainly aims to refine (purify) the pre-clusters and improve their quality. Moreover, the number of time series in the data set (cardinality) is reduced by defining a prototype for each group of very similar time series. It considerably decreases the complexity of 3PTC.

In this phase, a distance matrix is developed between the time series in each pre-cluster. Time series clustering substantially relies on distance measure. Many distance measures have been proposed in the literature, such as ED (Faloutsos et al., 1994), DTW (Sakoe and Chiba, 1971, Sakoe and Chiba, 1978), Edit Distance with Real Penalty (ERP) (Chen and Ng, 2004), Threshold Queries (TQuEST) (Aßfalg et al., 2006), Longest Common Sub-Sequence (LCSS) (Banerjee and Ghosh, 2001; Vlachos et al., 2002), Edit Distance on Real sequence (EDR) (Chen et al., 2005), and Compression-based Dissimilarity Measure (CDM) (Keogh et al., 2007). However, according to the literature, ED and DTW are the most common methods in the time series clustering due to the efficiency of ED and the effectiveness of DTW in similarity measurement. ED is a one-to-one matching measurement used in most of the works (approximately 80%) in the literature (Chan et al., 2003; Keogh et al., 2000). Let \( F_i \) and \( F_j \) be two time series of length \( n \) in a pre-cluster. The ED between \( F_i \) and \( F_j \) is defined mathematically as:

\[
\text{dis}_{\text{ED}}(F_i, F_j) = \sqrt{\sum_{n=1}^{n} (f_i - f_j)^2}
\]

where the square root step can be removed because the square root function is monotonic and returns the same rankings in clustering and classifications (Keogh and Kasety, 2003). Fig. 2 depicts the intuition behind using ED in the second phase, where ED rapidly finds the similarity in time between time series.

The pre-clusters constructed in the first phase are refined. The dilemma is as follows: given a cluster, it should be decomposed into separated sub-clusters as pure as possible.

**Definition 1. Sub-cluster:** a sub-cluster \( SC_{i,j} \) is a set of individual time series that are close to each other (similar in time), created by splitting pre-clusters \( PC_{i} \), and is represented as a single prototype. In this paper, the purity of a cluster (or a pure cluster) is defined as follows:

**Definition 2. Pure cluster:** a cluster is pure if all its members are of the members of a natural cluster (ground truth).
Thus, the sub-clusters (made from a pre-cluster) are desirable, such that most of their members are members of a natural cluster (class). That is, given that the pre-clusters are generated approximately and not precisely, they are often mixed with time series from different classes. Therefore, they should be recognized and separated by searching in the pre-clusters. As a result, the pre-clusters are broken into so-called pure sub-clusters. Of course assigning each time series to a separate cluster (singleton cluster) provides the highest purity. However, the best solution is the purest and smallest number of clusters.

Considering the pre-cluster $PC_i$, a similarity measure between the members of $PC_i$ is calculated and stored in an $m$-by-$m$ similarity matrix, $M_{m \times m}$, where $M_{ij}$ is the similarity measure between time series $F_i$ and time series $F_j$. A developed algorithm, Pure Cluster Search (PCS), is then carried out on the similarity matrix to split the pre-cluster into sub-clusters. That is, for pre-cluster $PC_i$, in which its members are not scattered solidly (low affinity time series), the members are divided into several sub-clusters. The sub-clusters are constructed based on the affinity of the time series in the pre-cluster. The concept of cluster affinity is borrowed from the Cluster Affinity Search Technique (CAST) (Ben-Dor et al., 1999) that is used to find the close objects.

**Definition 3. Cluster affinity:** the affinity of a time series $F_x$ to a sub-cluster $SC$ is defined as follows:

$$a(F_x) = \frac{\sum_{y \in SC} M_{xy}}{|SC|}$$  \hspace{1cm} (4)

where $M_{xy}$ is the similarity between time series $F_x$ and $F_y$, and $|SC|$ is the number of time series that exists in the sub-cluster.

The PCS sequentially constructs sub-clusters (from pre-clusters) with a dynamic affinity threshold. The output of PCS consists of discrete clusters that are separated without predetermining the number of sub-clusters. In this process, each sub-cluster is constructed with a time series and is gradually completed by new time series added to the sub-cluster based on the average similarity (affinity) between unassigned time series (in the pre-cluster) and the current sub-cluster members. Defining a specific threshold value, the cluster accepts high-affinity time series. That is, an affinity threshold, $a$, is specified to determine what is considered significantly similar. This parameter controls the number and sizes of the produced sub-clusters. After forming a sub-cluster, the PCS deletes the low-affinity objects from the sub-clusters. This process of adding to and removing from the sub-clusters is performed consecutively until no more changes occur in the sub-cluster.

As mentioned, an affinity threshold, $a$, of pre-cluster $PC_i$ is defined dynamically, which is very important because such would limit our ability to enforce our prejudices, expectations, and assumptions on the size and shape of the sub-clusters. That is, the PCS is proposed as an algorithm that works without predetermining the parameters. The affinity threshold, $a$, is calculated dynamically based on the remaining time series in a pre-cluster, that is, unassigned time series in the pre-cluster $PC_U$, before constructing each new sub-cluster $SC_{new}$, as:

$$a = \frac{\sum_{x,y \in PC_U} M_{xy} \cdot \left(1 - \frac{|M_{xy} - \mu|}{\mu} \right)}{|PC_U|} + \mu$$  \hspace{1cm} (5)

where

$$\mu = \frac{\sum M_{xy}}{|PC|}$$  \hspace{1cm} (6)

is the mean of the similarities of each time series to other time series in a pre-cluster ($\mu$ is initialized one time in the start of algorithm), and $|PC|$ is the number of all time series in the pre-cluster. Eq. (5) calculates a threshold based on the within-variance of affinities in the pre-cluster dynamically before creating each new sub-cluster. This value is used to distinguish the time series with low affinity by placing them into a new sub-cluster. Then, given a set of $n$ time series in sub-cluster $CS_k$, the time series are represented by a prototype time series $R_k = \{r_1, \ldots, r_m, \ldots, r_n\}$, where $r_x$ is defined by:

$$r_x = \frac{\sum f_{r_j}}{n}$$  \hspace{1cm} (7)

where $F_i = \{f_1, \ldots, f_m, \ldots, f_n\}$ is a time series in $CS_k$. As a result, the output of this algorithm consists of some prototypes that represent different sub-clusters. Table 2 shows the pseudo code related to the PCS.

One of the strengths of the 3PTC is its flexibility in making different cluster shapes by choosing arbitrary algorithms for merging. That is, given the sub-clusters and the definition of a similarity measure, many distance-based algorithms (i.e., partitioning, hierarchical, and density-based clustering) can be used for merging the sub-clusters in the third phase. However, instead of applying the clustering algorithm on the entire data, only the prototype of each sub-cluster is contributed in the merging process, which considerably reduces process complexity. Moreover, sometimes the prototypes can increase clustering accuracy. That is, highly unintuitive results may be obtained in other approaches using raw time series because some distance measures are very sensitive to some “distortions” in the raw time series. In contrast, in the 3PTC, the effect of the distortions or outliers on time series is alleviated because each prototype is made by the averaging of time series. The experiment performed in Section 5.3 to show the scalability of 3PTC verifies that PCS can reduce data size, but it cannot significantly reduce its effectiveness.

### 3.3. Phase 3: merging

The output of the second phase comprises a relatively few number of prototypes (compared to the original data set). In the third phase, the distances between the prototypes are calculated to form the final clusters.

In the second phase of the 3PTC, similar time series are grouped using ED based on their similarity in time. However, two time series that are dissimilar in time can be similar in shape. As mentioned, similarity in shape is desirable in the clustering of time series because finding clusters of time series that are similar in shape are very close to ground truth (natural clusters). This fact can be easily proved experimentally by the clustering of time series using ED (which is based on the similarity in time) and DTW (which is based
on the similarity in shape). In contrast to ED that proposes a one-to-one matching, DTW is suggested as a one-to-many metric. DTW is a generalization of ED that solves the local shift problem in the time series to be compared. Local shift problem is a time scale issue that is a characteristic of most time series such as stock time series. Handling local shifts allows similar shapes to correspond to their prototypes. All time series of some clusters constructed from the prototypes of the sub-clusters are compared. Given \( (r_{x1}, r_{x2}, \ldots, r_{xT}) \) is the prototype of \( S_{x} \), and \( r_{y} \) is calculated by Eq. (7). Then, to compute the distance between the prototype of \( S_{x} \) and \( S_{y} \), an \( n \times n \) matrix is constructed for the distance of all pairs as \( M(r_{x}, r_{y}) \), where \( M_{ij} = d_{DTW}(r_{x_i}, r_{y_j}) \) and \( d_{ED}(\cdot) \) is ED. Given \( W = (w_{1}, w_{2}, \ldots, w_{n}) \) as a set of warping paths, \( w_{0} = (r_{x1}, r_{y1}) \), \((r_{x2}, r_{y2}), \ldots, \( (r_{xT}, r_{yT}) \) is a set of points that define a traversal of matrix M, the DTW between two prototypes \( r_{x} \) and \( r_{y} \) is a warping path that minimizes the distance between \( r_{x} \) and \( r_{y} \),

\[
d_{DTW}(r_{x}, r_{y}) = \min \left( \sum_{a=1}^{U} \frac{W_{a}}{U} \right) \tag{8}\]

where \( (r_{x1}, r_{y1}) = (1,1) \) and \( (r_{xT}, r_{yT}) = (n,n) \), and that \( 0 \leq r_{xT-1} - r_{yT} \leq 1 \) and \( 0 \leq r_{yT} - r_{xT-1} \leq 1 \) for all \( T < n \).

Different approaches of clustering algorithm depends both on the type of the desired clusters and on the particular purpose (Warrenliao, 2005). The merging process results in the emergence of some clusters constructed from the prototypes of the sub-clusters. Thus, a mapping activity is carried out to assign the original time series to their corresponding prototypes. All time series of each sub-cluster are assigned to the cluster to which the corresponding prototype is assigned.

### 4. Evaluation method

The Kuala Lumpur Stock Exchange (KLSE) is End-Of-Day (EOD) data related to the stock exchange of Kuala Lumpur, Malaysia. The historical data are provided by www.klseod.com, which are retrieved from public websites and published freely for educational purposes. This data set includes the stock prices of a variety of companies. The stock price of companies is from different markets, such as Main Market, ACE Market, Structure warrants, ETF, and Bound. These companies belong to different categories, such as Property, Plantation, and Mining. Finding groups of similar companies can be used for prediction tasks in the stock market. For this experiment, a set of time series related to one year (2010) of stock market data set is chosen. The number of companies (time series) in this experiment is 870, and each time series consists of 240 points. To evaluate the accuracy of the proposed model, an internal metric is used to compare it to other approaches. However, evaluating extracted clusters (patterns) is difficult in the absence of data labels (Zhang et al., 2006). Therefore, another set of data sets is selected, which have class labels and can be used for evaluating the 3PTC using external indices. This set includes 19 different data sets in various domains and sizes from the UCR Time series Data Mining Archive (Keogh et al., 2011). This set is chosen because it consists of various numbers of clusters and different cluster shapes.
and density, contains noise points, and is used in many articles in the literature as a benchmark.

Moreover, a syntactic data set is used to experiment 3PTC with large data sets. Control Chart Time series (CC) is a synthetic data set proposed by Alcock and Manolopoulos (1999). This data set has six classes that are produced by Eq. (9).

\[
f(x) = \begin{cases} 
  m + rs, & \text{if } c(o) = \text{Normal} \\
  m + rs + \sin(\frac{2\pi}{k(t)}), & \text{if } c(o) = \text{Cyclic} \\
  m + rs + gt, & \text{if } c(o) = \text{Increasing} \\
  m + rs - gt, & \text{if } c(o) = \text{Decreasing} \\
  m + rs + kx, & \text{if } c(o) = \text{Upward} \\
  m + rs - kx, & \text{if } c(o) = \text{Downward}
\end{cases}
\]

where \( m = 30 \) and \( s = 2 \). \( r, x, g \) are uniform random values in the range of \([-3, 3]\), \([7.5, 20]\), and \([0.2, 0.5]\), respectively. \( k(t) \) is equal to \( 1 \) for \( t > a \), where \( a \) is a uniform random value in the range of \([20, 40]\). Different cardinalities of the CC data set are used to evaluate the scalability of 3PTC. Up to 12,000 time series are generated to show the experiment results. Fig. 4 depicts three sample time series of each class.

The most commonly used external indices in the time series clustering domain are used for evaluating the accuracy of 3PTC (i.e., Rand Index, Adjusted Rand Index, Entropy, Purity, Jacard, F-measure, FM, CSM, and MNI) (refer to (Amigó et al., 2009; Wu et al., 2009) for definitions and comparison). The conclusions are drawn based on the average value of the indices to avoid biased evaluation. Moreover, the average quality of 100 runs is calculated to prevent the bias of random initialization (De Gregorio and Maria Iacus, 2010; Lin et al., 2004; Petitjean et al., 2011; Vlachos et al., 2003). Although the focus of this study is on improving the accuracy of 3PTC, the scalability of the proposed model is also calculated to theoretically prove its feasibility.

5. Experimental results

5.1. Stock market categorization

This experiment aims to cluster the stock time series to find the clusters of companies that are similar in shape of their stock prices. At first, the time series are normalized using Z-normalized to provide fair conditions. The compression ratio = 4 is then used to symbolize the time series. The time series of this data set are not clustered in advance; hence class labels cannot be used to verify the final clustering results. As a result, the accuracy of 3PTC is evaluated using internal criteria, which are very realistic compared to external criteria, because class labels are unavailable in most of the real-world problems.

Moreover, considering the unsupervised nature of clustering and the unavailability of the number of clusters, the number of clusters should be determined by the user. Essentially, the right number of clusters depends on the distribution shape and scale in the data set, as well as the clustering resolution required by the user (Han and Kamber, 2011). For this data set, the number of clusters is set to about \( \sqrt{N}/2 \), where \( n \) is the number of time series in the data set (Han and Kamber, 2011); approximately \( \sqrt{N} \) time series are found in each cluster. Subsequently, the 3PTC is applied to cluster the data set. Fig. 5 shows a sample of the constructed clusters related to the stock exchange time series.

The clustering results show that the 3PTC can reveal the outliers in the data. For example, the second cluster includes only one time series considered as an outlier company or missing data. Second, the 3PTC discloses the co-movement of different companies with similar stock prices. However, the generated clusters are not very clear and useful because some local shifts usually exist in the stock time series. Nevertheless, the 3PTC has the ability to generate the dendrogram of time series using the prototypes of the second phase, providing the experts with insights into different markets. That is, different patterns of the stock market are revealed using the dendrogram of prototypes, which can also be utilized in classification tasks.

In addition, similar groups of the time series (co-movements) in the lower levels of the dendrogram can be used as a tool for prediction. This capability comes from the strength of 3PTC in finding similar time series in shape. Fig. 6 shows a sample hierarchy of clustering companies based on the similarity of their stock market in shape. This kind of dendrograms can be used to predict the stock market based on similar companies in a cluster.

Fig. 7 shows the handling of local shifts in the time series of stock price related co-moving companies. For example, two companies (AFFIN and AFG) with a very similar shape (not in time) are shown in Fig. 7, in which the changes in AFG are very similar to AFFIN, but with a time shift.

Although the results provide clusters of companies that may be interpretable for financial system experts, using some validity criteria to show the superiority of 3PTC is necessary. Hence, conventional k-Medoids algorithm is adopted to prove that the proposed model is more efficient than conventional algorithms for clustering whole time series. Different amounts of records (cardinalities)
from the data set are collected to show how the 3PTC works in terms of accuracy.

As mentioned, clustering is an unsupervised learning technique; however, no predefined classes are found to compare the clustering results of different clustering algorithms. The most common measure for internal criteria, Sum of Squared Error (SSE) (Han and Kamber, 2011), is used for evaluating such clusters in terms of accuracy. For each time series, the error is the distance to the prototype of the nearest cluster. The following formula is used to calculate SSE:

\[
\text{SSE} = \sum_{j=1}^{K} \sum_{i \in C_j} (\text{dist}(F_i, R_j))^2
\]

where \(F_i\) is a time series in cluster \(C_j\), and \(R_j\) is the prototype of cluster \(C_j\). In this equation, \(\text{dist}()\) is the DTW distance, which has been shown to be very useful for making natural clusters. The result illustrated in Fig. 8 is related to the SSE average of 10 times run of the 3PTC and k-Medoids with different SAX resolutions (different compression ratios).

Fig. 8 shows that different cardinalities of the data set are chosen for clustering by the 3PTC. The quality of 3PTC is competitive with k-Medoids (using RAW time series). The result reveals that the SSE of 3PTC is less than the conventional clustering algorithms that use SAX representation. Moreover, the quality of 3PTC is high compared to raw time series (not dimensionality reduced data). The superiority of the 3PTC is due to the use of similarity in shape, whereas conventional clustering algorithms use similarity in time. Although, no significant difference is found in the quality of 3PTC and k-Medoids (Raw) in the small cardinalities of the data set, the quality trend clearly shows that the difference in SSE also increases as the cardinality is increased, indicating that the 3PTC works well with high cardinality data sets. The effect of using DTW shows its impact on higher cardinality, whereas the effect of using raw time series decreases as the data set (cardinality) size grows.

### 5.2. Comparing 3PTC to conventional approaches

Two approaches in dealing with time series clustering are available. The first approach reduces the dimension of the time series data using a representation method. The second is the work that uses elastic methods for calculating the similarity between time series. Therefore, we first investigate the first approach.

Based on the literature review, most of the studies on clustering of time series use a representation method to reduce the dimension of time series before performing clustering. Therefore, in the first experiment, the raw time series are represented by SAX, and the 3PTC is compared in front of dimensionally reduced time series. However, the raw time series are represented using three different compression ratios (compressionratio = [4,6,8]) because the SAX accuracy depends on the compression ratio value. As a result, the mean of three accuracies for each data set is calculated as the average accuracy of k-Medoids. Likewise, the same resolutions are used for the 3PTC because the 3PTC can also accept different resolutions in the first phase. The average of accuracies is reported as the quality of data clustering using the 3PTC (with k-Medoids scheme), as depicted in Fig. 9.

Fig. 9 shows that the accuracy of 3PTC is much better than conventional k-Medoids for all data sets. The result reveals that a better quality of clustering is obtainable without reducing the dimension of time series using the prototypes of very similar time series. This result proves the researcher’s claim that the 3PTC model can outperform the conventional algorithms, which use dimensionality reduction approaches that may overlook the data. Experiments for other algorithms, such as hierarchical algorithms and EM, also show the same results. However, these results cannot be shown due to space limitation.

The second approach is the use of elastic distance metrics for calculating the distance between time series. In this section, the results (of 3PTC) are compared to the second approach. First, k-Medoids algorithm, which has been proven to be effective in time series clustering domain, is chosen [30–34]. DTW is used as a distance metric to calculate the distance between the time series data. Fig. 10 illustrates the quality of the 3PTC approach in front of the quality of k-Medoids on raw time series.

Using raw time series and DTW [i.e., k-Medoids (Raw-DTW)], the 3PTC is still superior to k-Medoids (Raw-DTW) on several data sets because of the mechanism of using a prototype of similar time series instead of the original time series. DTW is very sensitive to outliers because all points have to be matched. Using prototypes of the raw time series mitigates the effect of the noises and outliers in the time series data sets, thereby decreasing the DTW sensitivity. In addition, using raw time series and DTW [i.e., k-Medoids (Raw-DTW)] is not a fair comparison because it is not practically feasible.

![Fig. 4. Three sample time series of each class of CC.](image-url)
in the real world due to its very high complexity. That is, it needs $N(N - 1)/2$ distance calculation to calculate the confusion matrix (needed for clustering), where $N$ is the number of time series. Moreover, considering the length of the time series in the data set as $d$, the number of running the instructions to calculate a distance measure for a pair of time series is $d^2$. As a result, the complexity of only the distance matrix (not the whole clustering process) is equal to $N(N - 1)d^2/2$, which is very high. For example, in the Wafer data set, given $N = 1000$ and $d = 152$, the number of instruction executions is $11,540,448,000$. However, the same process using 3PTC is approximately $177,084,440$ (with SAX4 and reduction factor = 90%; see Section 5.3 for more details). As a result, the 3PTC is superior to $k$-Medoids (Raw-DTW).

Some experiments are carried out on the large synthetic data set to further confirm the effectiveness of 3PTC. The 3PTC (with different schemes of merging) is investigated in front of conventional $k$-Medoids and hierarchical approach (with single and average linkages). The average quality of 3PTC with different schemes is depicted in Fig. 11.

The results imply that the quality of 3PTC is superior to other algorithms. However, the fluctuation of 3PTC in the CC data set is slightly more than the other algorithms (except for hierarchical algorithm using single linkage). This difference is the effect of, first, showing the result by averaging the accuracy per different parameter values of SAX, and second, using prototypes of data instead of the original time series (i.e., the error of the second phase in purifying the clusters). Moreover, for most of the cardinalities of the data set, the minimum quality of 3PTC is still more than the other algorithms. Moreover, the trend shows better quality as the cardinality is increased. Therefore, using very low dimensional time series for clustering the large time series data sets is unnecessary; instead, the clustering can be applied on smaller sets of high dimensional time series through prototyping. That is, approximated sub-clusters have a less destructive effect on the accuracy of the final clusters compared to the use of the approximated time series.

5.3. Time complexity

The overall computational complexity of 3PTC depends on the amount of time required to construct the first phase clusters, refine the clusters in the second phase, and perform the merging of the prototypes in the last phase. Suppose that the number of time series is $N$ and the size (length) of time series is $d$, which changes into $r$ after the dimensionality reduction process, where $r < d$. The complexity of dimensionality reduction using SAX is $O(Nd)$ and the time complexity of $k$-Modes is $O(k_{PC}N_r)$, where $k_{PC}$ is the number of clusters and $I$ is the number of iterations converging. Therefore, the complexity of the first phase is linear in the number of time series ($N$) because $I$ is not large and most of the changes happened in the first iterations running on dimensionality reduced time series. The complexity of PCS (second phase), in the best case, is the same as the CAST algorithm (Ben-Dor et al., 1999), which is slightly more than $O(N \log N)$, and in the worst case, is $O(N^2)$. Considering the complexity of ED as $O(d)$, the complexity of PCS is $O(N^2 k_{PC}^2 d)$. Given that there are $k_{PC}$ such partitions, the overall complexity becomes $O(N^2 k_{PC}^2 d)$. The following experiment verifies that PCS can reduce the size of data, but does not significantly reduce its effectiveness. At first, a parameter is defined as reduction rate. Given $k_{CS}$ as the number of sub-clusters generated by PCS, the reduction rate of the second phase is then defined as:
where $R_{rate}$ is the reduction rate and $k_{SC}/n$ is the ratio of the size of the sub-clusters to the size of the data set. Moreover, the purity of the sub-clusters is calculated based on the number of items in the same sub-cluster that belong to the same class (ground truth) (Van Rijsbergen, 1979). To find a good trade-off between the reduction ratio ($R_{rate}$) and the purity of sub-clusters ($P_{SC}$), the objective function is defined as the quality–gain ratio:

$$QGR = \frac{P_{SC} + R_{rate}}{2}$$

(12)
Fig. 12 shows the QGR and the proportion of the purity and reduction rate for different data sets. The result reveals that a good trade-off between reduction rate and purity is obtained without parameter settings (approximately 80%). Therefore, in PCS algorithm, the affinity threshold is adjusted dynamically regardless of the characteristics of time series or size of data, which is of great importance. That is, PCS can reduce the data size without decreasing its effectiveness.

If the final clustering scheme is a hierarchical clustering (e.g., average linkage) with its complexity of $O(n^2 \log(n)$) where $n$ is the number of input objects, and $O(d^2)$ for DTW between each pair of the time series, the total computation of merging (third phase) is then $O(n^2 \log(n)d^2)$.

**Overall complexity:** Substituting $k_{SC}$ (i.e., number of prototypes) in the equation of the third phase, the overall time required for three phases is

$$O(MTC) = O\left(\frac{k_{SC}^2 \log(k_{SC})}{k_{PC}} d^2 + \frac{N^2}{k_{PC}} d + k_{PC} Nr + Nd\right) \quad (13)$$

Given the complexity of 3PTC (using the hierarchical scheme) as $O(MTC)$, and conventional hierarchical algorithm in the order of $O(N^2 \log(N)d^2)$ as $O(Hier)$, for comparing the 3PTC to hierarchical algorithm, the ratio of the complexity of 3PTC on hierarchical algorithm is calculated as:
From a technical viewpoint, the results show that unlike other approaches, the use of 3PTC to cluster time series based on similarity in shape does not require calculating the exact distances among all time series in a data set. Instead, accurate clusters can be obtained using prototypes of similar time series. It has important implications on similar works in different domains requiring an accurate clustering of time series, particularly in large time series data sets. For example, the proposed method can be used to find patients with similar characteristics based on the pattern of their blood pressure, or heterogeneous consumers based on the pattern of their electricity consumption.

Our future work will initially include the extension of our analysis in detecting changes in clusters for a long period. This research will reveal changes in co-movement among stock prices of individual companies in the Malaysian stock market for a long period. Second, we will analyze the impact of different industries from neighboring Asian countries on changes in clusters conducted from internal stock markets. This work can be considered as the integration of internal and international co-movements, which has important implications to international investors.

6. Conclusion

The paper presented a new approach on the co-movement of the stock market, by using a three-phase method: (1) pre-clustering of time series; (2) purifying and summarization; and (3) merging. This study addressed a couple of challenges such as inaccuracy and overlooking of time series caused by dimensionality reduction, large time series clustering, and clustering based on similarity in shape.

First, by using different evaluation methods, 3PTC was proven to outperform other conventional clustering algorithms in terms of accuracy when experimenting on various datasets. This result implies that the accuracy of 3PTC is superior to those of other algorithms.

Second, the proposed method does not need to use an extremely low-dimension time series for clustering large datasets when it is applied on a large time series. Instead, clustering can be applied to smaller sets of high-dimensional time series by prototyping. Moreover, the time complexity of the proposed model was computed, and the execution time of the model was found acceptable.

Finally, a conclusion was drawn that clustering could be applied on a time-series data set to generate a hierarchy of meaningful and accurate clusters using 3PTC. That is, the proposed model can calculate similarities in shape among time series, which is advantageous in co-movement assessment. Therefore, the use of 3PTC to assess the comovement of stocks was demonstrated to be a feasible alternative to existing approaches for handling local shifts in a time series of stock prices related to co-moving companies.

The most important implication of co-movement assessment is asset allocation and risk management in the stock market. From an economic viewpoint, the proposed method can uncover dependencies among companies because it reveals co-movement of companies across local stocks in the Malaysian stock market. This process provides insightful information that increases the power of investors and fund managers in investing for higher returns in stocks based on other co-moving companies. Our results show the advantage of the proposed method wherein the analysis allows better prediction and understanding of the co-movement of companies even with local shifts. For example, 3PTC can detect changes in AFG which are similar to those in AFFIN. The AFFIN and AFG companies were shown to be co-moving with a time shift. Therefore, AFG stock price can be predicted based on fluctuations in AFFIN stock.

Acknowledgements

This paper is supported by BK026-2013 University of Malaya Research Grant.

References


