Binarizing Change for Fast Trend Similarity Based Clustering of Time Series Data

Ibrahim K.A. Abughali(✉) and Sonajharia Minz

School of Computer and System Science,
Jawaharlal Nehru University, New Delhi, India
{barhoom.gali,sona.minz}@gmail.com

Abstract. It is observed that traditional clustering methods do not necessarily perform well on time-series data because of the temporal relationships in the observed values over a period of time. Another issue with time series is that databases contain bulk amount of data in terms of dimension and size. Clustering algorithms based on traditional measures of dissimilarity find trade-offs between efficiency and accuracy. In addition, time series analysis should be more concerned with the patterns in change and the points of change rather than the values of change. In this paper a new representation technique and similarity measure have been proposed for agglomerative hierarchical clustering.

Keywords: Time series representation · Similarity search · Clustering

1 Introduction

Today Time Series data management has become an interesting research topic for the data miners. Particularly, the clustering of time series has attracted the interest.

Clustering is the process of finding natural groups, called clusters, the grouping should maximize inter-cluster variance while minimizing intra-cluster variance [1], most of the clustering techniques can be two major categories, Partition-based clustering and Hierarchical Clustering [2]. Many of the traditional clustering algorithms use Euclidean distance or Pearson's correlation coefficient to measure the proximity between the data points. However, in case of time-series data these parameters involve the individual magnitudes at each time point therefore the traditional algorithms perform poorly with time-series expressions data, to overcome these limitations the proposed work aims to represent the variations in the measurements of the time-series for fast implementation of an efficient agglomerative nesting algorithm, the focus of this work is on fast whole sequence similarity search in the changes in respect to time rather than the values in the time series data.

The rest of the paper is organized as follows: Sect. 2 presents a brief review of related work. Sections 3 and 4 demonstrates the basic concept and presents the analysis of the proposed algorithm respectively. In Sects. 5 and 6 experimental and the conclusions and some future directions.

M. Kryszkiewicz et al. (Eds.): PReMI 2015, LNCS 9124, pp. 257–267, 2015.
DOI: 10.1007/978-3-319-19941-2_25

2 Related Work

Many clustering algorithms have been proposed such as k-means, DBSCAN, STING, p-cluster and COD [4–6]. One of the recently proposed algorithms is VCD algorithm [3] to analyze the trends of expressions based on their variation over time, using cosine similarity measure with two user inputs, it has been enhanced later in EVCD algorithm [2] for same purpose with one single user input and provides results in several levels which allows the user to select the most appropriate level by using different parameters such as the silhouette coefficient, number of clusters and clusters density. Both algorithms Enhanced Variation Co-expression Detection (EVCD) and (VCD) algorithms [2, 3] inferred that the cosine similarity measure was the most appropriate similarity measure for clustering the time varying microarray data.

3 Concepts and Definition

In order to determine the variation patterns in the time series based on the changes in the values observed at fixed time points binarization of the change has been proposed. Some related definitions are presented in this section.

3.1 Variation Vector

Given a sequence of n + 1 measurements observed at time periods t_0, t_1, $t_2 \ldots t_n$ to denote a univariate time series, say, $Y = \langle y_0, y_1, y_2 \ldots y_n \rangle \in \mathbb{R}^{n+1}$. A variation vector $Y_v \in \mathbb{R}^n$ of Y is a sequence of the differences denoted by, $Y_v = \langle d_1, d_2 \ldots d_n \rangle$, where $d_i = y_i - y_{i-1}$, for $1 \leq i \leq n$. The increase in the measurement $(y_i \geq y_{i-1})$ and its magnitude is represented by the difference $d_i \geq 0$. Similarly, the decrease $(y_i < y_{i-1})$ is computed as $d_i < 0$.

The trend is the tendency of a continuous process that is measured during a fixed time interval. The trend analysis may traditionally be carried out by plotting a trend curve or a trend line and by monitoring the increase (decrease) in the values. Thus trend analyses involve observation of the tendencies of the values by way of analyzing the changes that occur in terms of the quantum of the change and/or the nature of the changes. The pattern of increase or decrease in the values of the measurements may play a significant role in the trend analyses. Variation vectors quantify the difference in measurements at two consecutive time periods say t_i and t_{i+1} in terms. The directions of change, increase or decrease, may be captured by the positive or negative sign of the magnitude of difference d_i respectively. Therefore, a binary representation of the direction of change is suitable for computational efficiency. Binarization of the change for any time-series has been proposed by a direction vector. Further, the trend similarity based on the distance metric of the n-dimensional binary vectors has been defined.

3.2 Direction Vector

For a variation vector, $Y_v = \langle v_1, v_2, \ldots, v_n \rangle \in \mathbb{R}^n$, a direction vector $Y_d \in \{0, 1\}^n$ is defined as $Y_d = \langle b_1, b_2, \ldots, b_n \rangle$, where,

$$\mathrm{b_i} = \begin{cases} 0 \text{ if } \mathrm{v_i} \geq 0 \\ 1 \text{ if } \mathrm{v_i} < 0 \end{cases}. \tag{1}$$

Example 1: Consider two time series $\mathrm{T_1} = \langle 3,7,2,0,4,5,9,7,2 \rangle$ and $\mathrm{T_2} = \langle 10,15,11,5,19,25,27,24,13 \rangle$. The corresponding variation vectors are, $\mathrm{V_1} = \langle 4,-5,-2,4,1,4,-2,-5 \rangle$ and $\mathrm{V_2} = \langle 5,-4,-6,14,6,2,-3,-11 \rangle$. The direction vectors of $\mathrm{T_1}$ and $\mathrm{T_2}$ are $\mathrm{D_1} = \langle 0,1,1,0,0,0,1,1 \rangle$ and $\mathrm{D_1} = \langle 0,1,1,0,0,0,1,1 \rangle$ respectively.

3.3 Trend Similarity

Let two time series $\mathrm{X} = \langle \mathrm{x_0, x_1, x_2, \ldots, x_n} \rangle$ and $\mathrm{Y} = \langle \mathrm{y_0, y_1, y_2, \ldots, y_n} \rangle$ be measured at the time $\mathrm{t_0}$, $\mathrm{t_1}$,…,$\mathrm{t_n}$. Let $\mathrm{X_v} = \langle \mathrm{v_1, v_2, \ldots, v_n} \rangle$ and $\mathrm{Y_v} = \langle \mathrm{u_1, u_2, \ldots, u_n} \rangle$ be the corresponding variation vectors and $\mathrm{X_d} = \langle \mathrm{l_1, l_2, \ldots, l_n} \rangle$ and $\mathrm{Y_d} = \langle \mathrm{s_1, s_2, \ldots\ldots s_n} \rangle$ be the corresponding direction vectors. Then X and Y are said to be similar in trend if and only if $\mathrm{l_i = s_i}$ for $1 \leq \mathrm{i} \leq \mathrm{n}$.

Both direction vectors $\mathrm{X_d}$ and $\mathrm{Y_d}$ are n-bit binary vectors. For each i if $\mathrm{x_i} \geq \mathrm{x_{i-1}}$ in series X i.e. $\mathrm{v_i} \geq 0$ then $\mathrm{l_i} = 0$ and $\mathrm{l_i} = 1$ for vice versa. In case of the time series Y the bit value of s_i would depict the increase if the value at $\mathrm{t_i}$ from the values at $\mathrm{t_{i-i}}$ as $\mathrm{u_i} \geq 0$ and correspondingly, $\mathrm{s_i} = 0$, and vice versa. If for each i, $\mathrm{l_i = s_i}$ then Y is said to be trend similar to X. It may be noted that for the definition of similarity the magnitude of difference in the two time-series has not been considered. However, only the concept of direction of change i.e. increase or decrease, has been considered. The information in the direction vector may be utilized to determine the degree of similarity.

Example 2: Consider the direction vectors $\mathrm{D_1}$ and $\mathrm{D_2}$ in the above example corresponding the two time-series $\mathrm{T_1}$ and $\mathrm{T_2}$ each of length 9. The magnitude of the differences are represented by the variation vectors $\mathrm{V_1}$ and $\mathrm{V_2}$. It may be noted that for each i, $1 \leq \mathrm{i} \leq 8$, $\mathrm{V_{1i}} \neq \mathrm{V_{2i}}$. However, $\mathrm{D_1}$ and $\mathrm{D_2}$ are bit-wise equal, i.e. $\mathrm{D_{1i} = D_{2i}}$, for $1 \leq \mathrm{i} \leq 8$, therefore, the two series $\mathrm{T_1}$ and $\mathrm{T_2}$ are observed to be similar in trend.

The following metric to measure the distance between two n-dimensional binary vectors has been considered in this work. Let $\beta = \{0,1\}$ and $I_n = \{0,1,2\ldots n\}$ then the binary function $d_{binary} : \beta \times \beta \rightarrow \beta$. For $b_1, b_2 \in \beta$,

$$d_{binary}(b_1, b_2) = \begin{cases} 0 & \textit{if } b_1 = b_2 \\ 1 & \textit{otherwise} \end{cases} \tag{2}$$

Then the distance function between a pair of n-dimensional binary vectors is $d_n : \beta^n \times \beta^n \rightarrow I_n$ Consider two n-dimensional binary vectors say $D_1, D_2 \in \beta^n$.

$$d_n(D_1, D_2) = \sum\nolimits_{j=1}^{n} d_{binary}\left(b_{1j}, b_{2j}\right) \tag{3}$$

Let $d_n(D_1, D_2) = k$. Then k = 0 if $\sum_{i=1}^{n} d_{binary}(b_{1i}, b_{2i}) = 0$ and k = n if $\sum_{i=1}^{n} d_{binary}(b_{1i}, b_{2i}) = n$. Therefore $0 \le k \le n$

Example 3: Consider the following two sequences as time series, $T_1 = \langle 3, 7, 2, 0, 4, 5, 9, 7, 2 \rangle$ and $T_3 = \langle 45, 80, 22, 10, 40, 63, 45, 90, 10 \rangle$, then variation vectors V_1 and V_3 of T_1 and T_3 are, $V_1 = \langle 4, -5, -2, 4, 1, 4, -2, -5 \rangle$ and $V_3 = \langle 35, -58, -12, 30, 23, -18, 45, -80 \rangle$, the direction vectors D_1 and D_3 are $D_1 = \langle 0, 1, 1, 0, 0, 0, 1, 1 \rangle$ and $D_3 = \langle 0, 1, 1, 0, 0, 1, 0, 1 \rangle$.

For $D_1, D_3 \in B^8$, the dissimilarity between D_1 and D_3 may be computed using the distance function d_8,

$$d_8(D_1, D_3) = 2 \tag{4}$$

where,

$$d_{binary}(b_{1i}, b_{2i}) = 1 \quad \text{for } i \in \{6, 7\} \tag{5}$$

and

$$d_{binary}(b_{1i}, b_{2i}) = 0 \quad \text{for } i \in \{1, 2, 3, 4, 5, 8\} \tag{6}$$

To allow difference in trends at the certain bits out of the n-bits, the concept of trend dissimilarity of degree-k has been considered where k ≤ n may be the number of bits at which the two n-dimensional direction vectors encounter bit-mismatch.

3.4 Trend Dissimilarity of Degree K

Given two n-dimensional time series T_i and T_j, and their respective direction vectors D_i and D_j, T_i and T_j are said to have dissimilarity of degree k, if $d_n(D_i, D_j) = k$, for $1 \le k \le n$.

The clusters at level-0 may contain identical objects. Consider any two arbitrary objects $\boldsymbol{x}$ and $\boldsymbol{y}$, and the Euclidian distance function d, the traditional measure of dissimilarity. Then $d(\boldsymbol{x}, \boldsymbol{y}) = 0$, i.e. $\sqrt{\sum (x_i, y_i)^2} = 0$ if the two objects are identical. Therefore, the objects $\boldsymbol{x}$ and $\boldsymbol{y}$ must be grouped in the same cluster at level-0, say ith cluster denoted by, $C_{0,i}$. Let $C_{i,j}$ denote cluster-id j at level-i. Then the m clusters at level-0 are $C_{0,1}$, $C_{0,2}$, $C_{0,3}$,…,$C_{0,m}$. Let a measure of dissimilarity at 1 bit represented by distance metric d_1 be associated to the clusters at level-1, dissimilarity at 2 bits represented by d_2 and so on. Then any two arbitrary objects $\boldsymbol{x}, \boldsymbol{y}$ may be in the same cluster at level-1, $C_{1,j}$, only if, $0 < d(x, y) \le d_1$. In this section the concept of Trend Cluster of level-k using the dissimilarity of degree-k is defined.

3.5 Trend Cluster of Level-K

For $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, a set of n-dimensional time series of cardinality m, and the set of corresponding direction vectors $\Gamma = \{D_1, D_2, \ldots, D_m\}$, a trend cluster of level-k,

$C_{k,j}$ would include all time-series T_i and T_j in the same cluster if $d_n(D_i, D_j) = k$. However, if $d_n(D_i, D_j) \neq k'$ for all $k', 0 \leq k' < k$, then T_i and T_j will be allocated to distinct trend clusters of level-0, level-1, up to level-k′, say $C_{k',i}$ and $C_{k',j}$, but would be grouped in the same trend clusters of level-k, say $C_{k,i.}$

Example 4: Consider time series T_1, T_2 and T_3 as in the Examples 1 and 3. The direction vectors of each is $D_1 = \langle 0,1,1,0,0,0,1,1 \rangle$, and $D_2 = \langle 0,1,1,0,0,0,1,1 \rangle$ $D_3 = \langle 0,1,1,0,0,1,0,1 \rangle$. Consider D_1 and D_2, $d_8(D_1, D_2) = 0$ therefore, T_1 and T_2 must be grouped in the same cluster of level-0. Consider D_1 and D_3, $d_8(D_1, D_3) = 2$. i.e. the series T_1 and T_3 have the trend dissimilarity of degree-2. Therefore, T_1 and T_3 must be grouped in different trend clusters of level-0 and level-1 say $C_{0,1}$ and $C_{0,3}$, and $C_{1,1}$ and $C_{1,3}$ respectively. However, the two must be grouped in the same trend cluster of level-2 say, $C_{2,1}$.

Example 5: Consider the 5-dimensional view of the four gene expressions *a, b, c* and *d*, as shown in Fig. 1. The direction vectors D_a and D_c are identical therefore genes *a* and *c* are trend similar. Even visually the vectors *a* and *c* are the most similar to each other than to the vectors *b* and *d*.

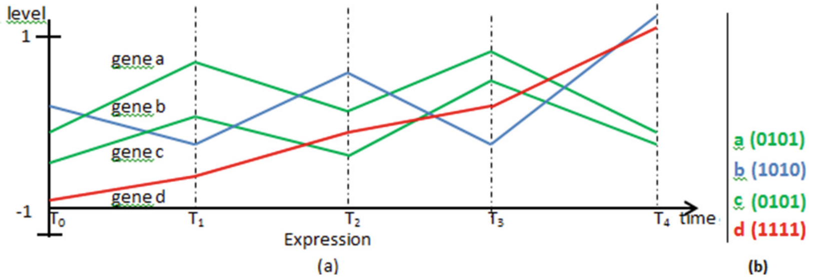

Fig. 1. Trend similarity in gene expressions

An advantage of this approach is the simplicity of representation of the objects of m-dimensional time series database, using only one bit to represent the change in value from time t_i to t_{i+1},

$$b_i = \begin{cases} 0 \; x_{i+1} \geq x_i \\ 1 \; x_{i+1} < x_i \end{cases} ; 0 < i < m - 1 \tag{7}$$

The direction vectors are loss transformation of the original data from which no original values can be retrieved. Thus it is a novel representation from the perspective of security and privacy preservation of the original data.

4 Fast Trend Similarity-Based Clustering (FTSC) Algorithm

FTSC algorithm starts with generating the variation vectors, second is binarization of the variation vector, and third is direction vectors indicate similarity in trend in the time series thus forming the trend clusters of level-0 in the hierarchy of clusters. The higher level clusters may result from merging the closest clusters in the previous level starting by smaller clusters, each cluster is represented by a direction vector as medoid of the cluster. The distance between clusters is computed by the distance between the medoids of the two clusters.

```
Input:    T, Time series database of n series;
     m+1, Dimension of each time series;
     m, maximum level of clustering;
Output: Hierarchical clusters up to level m;
Initialization:
$V = \emptyset$;                    // set of Variation vectors.
$D = \emptyset$;                    // set of Direction vectors.
$i = 0$ ;
$C_{i,j} = \emptyset,\ 1 \le j \le n$;         // i: cluster-level, j: cluster-id.
Begin
 I.   Variation Vector Generation
      $\forall t \in T$, $generate\ V(t)$;
      $V = V \cup \{V(t)\}$;
 II.  Binalization: Direction Vector Generation
                        $\forall X \in V, generate\ b(X)$;
      $D = D \cup \{b(X)\}$;
III.  Level-0 Cluster Generation
      $\forall D_t, D_l \in D$ if $d_m(D_t, D_l) = 0$
                     Then $C_{0,j} = merge(D_t, D_l)$ for $1 \le j \le n$;
 IV.  Higher Level Cluster  Generation
      For i = 1 to m
              For (j= 1 to maximum-cluster at level i-1)
      and
                    (l= 1 to maximum-cluster at level i-1)
           For $\forall C_{i-1,j}$, $C_{i-1,l}$ clusters of level i-1,
              If $d_m(C_{i-1,j}, C_{i-1,l}) \le i$ then construct a cluster
           of level-i
                 $C_{i,s} =$ merge  $(C_{i-1,j}, C_{i-1,l})$
End.
```

The FTSC algorithm is a nonparametric algorithm and it does not require any prior information related to data or number of clusters.

The asymptotic time complexity of the algorithm is quadratic on the product of the dimension of the time series and number of clusters level-i, $n_i < n$, therefore the complexity of the algorithm is O($(mn)^2$). However, due to the binarization of the variation in the time series, the comparisons of the m bits and distance computation may be implemented using fast bit operators.

5 Experiments and Results

5.1 Data Sets

The experiments have been carried out to perform clustering on two microarray data sets and two financial data sets. Table 1 describes the data sets.

Table 1. Data set

Data set	Repository	Type	No of rows	No of dim
1	NCBI	Microarray/Affymetrix	12488	8
2	NCBI	Microarray /Drosophila genome	3456	8
3	PWT	Financial/exchange rates and PPPs over GDP	29	61
4	NSE	Financial/(NSE) India	1555	9

5.2 System Configuration

Windows 8 enterprise © 2012, 64-bit, with processor intel® core (TM) i7 CPU, U 640@1.20 GHz an. Dot Net platform has been used to implementation.

5.3 Design of Experiments

The experiments have been designed to assess the performance of FTSC algorithm in terms the efficiency and accuracy. Efficiency is mainly observed in terms of execution time. The accuracy of the algorithm is considered to be the consistency in cluster allocation to a time series irrespective of the number of re-executed, cluster allocation to multiple copies of the time series data, and the order of input of the time series to the algorithm. Second experiment compares both algorithms FTSC and EVCD.

5.4 Efficiency and Accuracy of FTSC

The first experiment has been designed to examine the speed of Fast Trend Similarity Clustering algorithm to cluster the four data sets. The experiment of running the program implementing the algorithm repeated five times, the average running time to yield the hierarchical clusters for each of the four data sets Affymetrix, Drosophila genome, Exchange Rates and PPPs over GDP and NSE with execution time 00:00:02.66, 00:00:01.72, 00:00:10.11 and 00:00:01.34 respectivly.

The outcomes of running the FTSC algorithm on Affymetrix are presented in Tables 2, 3 and 4. In Table 2 the 7-bit direction vector of gene Id 11251 is 0000001 which is in cluster $C_{0,0}$ while the two genes 11152 and 12182 in serial 7 and 8 have identical direction vectors 0000101. Therefore, $C_{0,3}$ includes two genes. The total clusters of level-0 is 115.

Table 2. Direction vectors, clusters of level-0 of AffyMetrix data

S.no.	GENE ID	Direction vector	Cluster no.
1	11251	0000001	0
2	6599	0000010	1
:	:	:	:
6	11278	0000011	2
7	11152	0000101	3
8	12182	0000101	3
:	:	:	:
13	8001	0001001	6
:	:	:	:
16	11668	0001001	6
:	:	:	:
12487	10226	1111011	114
12488	10461	1111011	114

Table 3. Level-3 cluster formation

Cluster id	Medoid	$C_{2,*}$	$C_{2,*}$	$C_{2,*}$	$C_{2,*}$	$C_{2,*}$	Cluster density
$C_{3,0}$	0	4	7	14	29	58	486
$C_{3,1}$	20	26	52	81	–	–	689
$C_{3,2}$	54	48	41	–	–	–	73
$C_{3,3}$	87	93	97	105	–	–	1657
$C_{3,4}$	107	77	–	–	–	–	103
$C_{3,5}$	9	70	–	–	–	–	48
:	:	:	:	:	:	:	:

Table 4. Level-4 cluster formation

Cluster id	Medoid	$C_{3,*}$	$C_{3,*}$	$C_{3,*}$	$C_{3,*}$	Cluster density
$C_{4,0}$	0	9	16	31	60	581
$C_{4,1}$	20	54	83	–	–	764
$C_{4,2}$	87	99	107	–	–	1880
:	:	:	:	:	:	:

Tables 3 and 4 present the clusters of level-2 and level-3 respectively. In the two tables the rows display all the clusters $C_{i,j}$, i denoting the cluster level and j the cluster ID. The cluster medoid has been presented in the second column by the identifier of the direction vector representing the cluster of level-0. In Table 3, the 3rd, 4th, 5th, 6th and 7th column display the clusters of level-2 that are merged to form the cluster of level-3. Thus the cluster id $C_{3,0}$ represented by the medoid 0 is formed by merging the clusters of level-2 represented by the medoids 4, 7, 14, 29 and 58 yielding the cluster with a total of 486 genes. The cluster $C_{3,1}$ is the outcome of merging three clusters of level-2 that are represented by the medoids 26, 52 and 81 to the cluster represented by medoid 20 at level-3 having a total of 689 genes. To obtain the clusters $C_{3,6}$ to $C_{3,15}$ no other clusters of level-2 were merged to the ones represented by the respective medoids indicated in column two. The blank '−' entries in the table indicate no clusters of level-2. Therefore, the row pertaining to the cluster $C_{3,6}$ with medoid 16 indicates that no cluster of level-2 satisfied the criterion for the merge operation although the total number of genes in the cluster $C_{3,6}$ is 2, where number of clusters of level-3 are 16.

The clusters from $C_{3,6}$ to $C_{3,15}$ in level-3 have not changed from the previous level with the same medoids and densities.

Similarly the Table 4 exhibits the details of the clusters of level-4. From both Tables 3 and 4 it may be observed that the cluster $C_{4,0}$ with medoid 0 has been formed by merging the clusters $C_{3,0}$, $C_{3,5}$, $C_{3,6}$, $C_{3,7}$ and $C_{3,11}$ referred to by the medoids 0, 9, 16, 31 and 60 respectively. It may also be observed that the density of $C_{4,0}$ is the sum of the densities of the $C_{3,0}$, $C_{3,5}$, $C_{3,6}$, $C_{3,7}$ and $C_{3,11}$. Similarly the cluster $C_{4,2}$ is formed by merging $C_{3,13}$, and $C_{3,4}$, to $C_{3,3}$ resulting in the density 1880.

As the FTSC algorithm is an agglomerative clustering algorithm yielding a hierarchical clustering of levels 0–7 for Affymetrix data. The cluster at the highest level $C_{6,0}$, represented by the medoid 0 includes all the 12488 genes (Figs. 2 and 3).

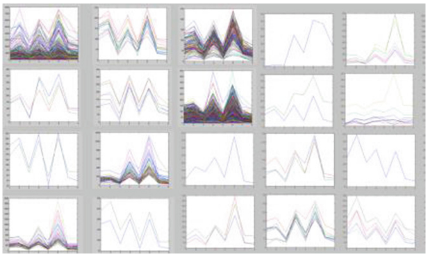

Fig. 2. Random clusters plot for DS 1 level 0

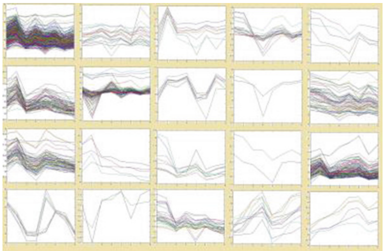

Fig. 3. Random clusters plot for DS 2 level 0

In order to estimate the efficiency, accuracy and sensitivity to order of data inputs, all the rows of the Affymetrix data set were duplicated four times and randomly shuffled. Therefore, the algorithm was executed with a total of 4 × 12488 = 49952 rows with 8 dimensions. The output of the program was a hierarchical clustering with levels

0−7 with same number of clusters at each level as before but the density of each cluster was four time the previous density. i.e. the cluster $C_{4,5}$ with inputs four time the first run was represented by a gene that had direction vector identical to the gene 9 and contained 192 genes. The same phenomenon was observed for all the clusters of each level from level-0 to level-7. Thus the accuracy of the algorithm has been assessed. The average running time of repeated execution of the four times the original data set was 00:00:10.714.

The repeated execution of the program after randomly shuffling the rows yielded the same number of clusters. However, each time the execution time was differed in the 3rd or the 4th decimal point with the mean being 00:00:02.6599 (Figs. 4 and 5).

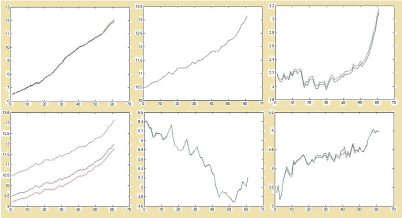

Fig. 4. Random clusters plot for DS 3 level 0

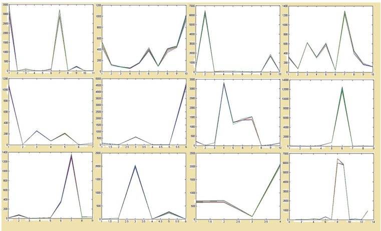

Fig. 5. Random clusters plot for DS 4 level 0

5.5 Comparison of FTSC and EVCD Algorithms

In this experiment the results of EVCD algorithm and FTSC algorithm have been compared. Two real world data sets Affymetrix and Drosophilia data sets as described in Table 1 are used in this experiment to assess the novelness of trend dissimilarity as the changes in the time series are represented by direction vectors. The EVCD algorithm is also a parametric algorithm while FTSC algorithm is not. EVCD algorithm requires one user input as the parameter ε. The experiment has been repeated for three values of ε, i.e. 0.01, 0.05 and 0.1 respectively. As EVCD performs a hierarchical clustering, for ε = 0.01, 10 clusters and 6 singletons were obtained at level 14, while for ε = 0.05, 10 clusters and 6 singletons were obtained at level 2, and finally 11 clusters and 2 Singleton were obtained at level 1 for ε = 0.1.

6 Conclusions

The experiments indicate that although the FTSC algorithm has the complexity O ($(mn)^2$) it is fast in terms of execution time due to the binarizing the change in the time-series. The binary representation in terms of the direction vector affect the distance computation implemented using bit level operators. The binarization also helps in privacy and security of the actual data. The nonparametric characteristic of the algorithm keeps the end user from exercise of parameter tuning. User also does not require any prior knowledge of the data or the clusters. The FTSC algorithm is time efficient and has the potential to yield accurate clusters of time-series data. The scalability of the algorithm in terms of multi-dimensions time-series and dealing with noise shall be investigated in future. To select a better medoid of the cluster of each higher level is also considered as future work.

References

1. Esling, P. Agon, C.: Time-series data mining. ACM Computing Surveys, **45**(1), 5 (2012)
2. Minz, S., Abughali, I.K.A.: Time-varying microarray data sets: co-expression detection. In: IEEE 2011 9th International Conference on ICT & Knowledge Engineering, IEEE Explore 978–1-4577-2162-5/11, pp. 43–46 (2011)
3. Yin, Z.-X., Chiang, J.-H.: Novel algorithm for coexpression detection in time-varying microarray data sets. IEEE/ACM Trans. Comput. Biol. Bioinf. **5**, 120–135 (2008)
4. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Trans. Neural Netw. **16**, 645–678 (2005)
5. G-Means Algorithm (2007). http://www.cs.utexas.edu/users/dml/Software/gmeans.html
6. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. In: Proceedings Nat'l Academy of Sciences USA, vol. 95, issue no. 25, pp. 14863–14868 (1998)