CSIT (November 2014) 2(3):181–190
DOI 10.1007/s40012-014-0056-2

ORIGINAL RESEARCH

Finding representative time sequence for trend analysis

Arpit Baheti · Durga Toshniwal

Received: 25 April 2014 / Accepted: 17 October 2014 / Published online: 4 November 2014

Abstract Trend analysis is important in many applications such as in business, weather, medical, etc. because it imparts knowledge about what has taken place in the past and what will take place in time to come. Trend analysis in the time series is the practice of collecting and attempting to spot patterns. Due to vast amount of data is present in such application and most of them are in the form of time series, we introduce an algorithm, which generates a representative sequence for a group of time series those are clustered by any clustering technique. This algorithm hierarchically merges the time series present in a cluster based on their similarity and gives a Representative Time Sequence which will be further used for trend analysis. We also present verification and validation of the representative time sequence formed by our algorithm on rainfall time series data set.

Keywords Time series · Merging · Trends · Representative Time Sequence

1 Introduction

Trend analysis is a form of comparative analysis that is often employed to identify current and future movements in various applications. In business, trend is the general direction in which the market is headed, an aspect of technical analysis that tries to predict the future movements of stock based on past data. In weather, air temperature and precipitation are principle elements, and examination of their behavior is important for understanding of climate variability. Trend analysis is based on the past behavior that gives information about what will happen in the future.

In most of the real world applications such as stock market, weather records, customer buying pattern, growth pattern of diseases, etc. data is present in time series form. Time series represents or traces the values taken by a variable over a time period, such as a month or year. These applications generate huge amount of data even in a short period of time [1] and if we are applying trend analysis on that data, it will take lots of time. Time series data generated by these applications, found to have some properties like they exhibit similar behavior for a specific time period in case of stock market, customer buying pattern, etc. or similar behavior for a region in case of weather data. Hence we can utilize the fact of having similarity present in time sequences, for trend analysis.

Why would we analyze each sequence if we can put similar time sequence into a cluster and then form an approximate representation for that cluster? Using this we only lose a very less amount of information and can save lot of time.

There is lots of work that has been performed for trend analysis over time-series data in past by [2–5]. All these research works on trend analysis of time-series data are performed with-out forming any representation, which cost lots of time in analyzing each sequence and deriving results for that. Hence we introduce Representative Time Sequence (RTS). RTS is a time sequence, which is used to represent the cluster formed using clustering technique. Using the RTS, we can perform many kind of regression test, trend detection test on such time-series data and analyze it to predict the future behavior.

In this paper, we introduce an algorithm for finding the representative time sequence for the clusters of sequences having similar behavior. Our algorithm uses the concept of

A. Baheti (✉) · D. Toshniwal
Department of Computer Science & Engineering, Indian Institute of Technology, Roorkee, India
e-mail: arpitbaheti7@gmail.com

Agglomerative hierarchical clustering, which we will describe later in this paper. We verify the representative sequence formed by the algorithm described, by calculating its similarity with all the original sequences through which it is formed. The representative sequence is used for trend analysis. As We form RTS by merging all the sequences of cluster, we need to validate it on the the constraint that, it must follow the same behavior (i.e. trend) as all or most of the sequences present in the cluster behave. So for validation purpose, we use Sen's median slope estimation method.

2 Related work

In the task of making representative series very little work has been done in past. For making a representation of time series there are many methods exist like discrete Fourier transformation (DFT) [6], discrete wavelet transformation (DWT) [7], singular value decomposition (SVD) [8]. But all these methods are used to reduce the dimensionality on the data so that large time series data reduce to small. We can't use these dimensionality reductions methods for finding the representation, because if we do so we can't apply trend test on each attribute as they might get eliminated.

To merge the time sequence, a merging algorithm is proposed by [9]. In this algorithm author describe an *"influence term"*, which is associated with every time series, to show its significance in the resulting time series. The advantage of this algorithm is that user can defined the influence term for time series or it might use the property of time series domain to define the influence term. But this advantage might not help when user not known which sequence has to be participated more in result (e.g. for a large data set) or time series domain does not give enough knowledge to define influence term (e.g. time series data set having random value for each data point).

Rest of the paper is organized as, Sect. 3 will contain background detail for finding representative time sequence algorithm and detail of data set used in this paper, in Sect. 4 we will be describing the algorithm for finding representative sequence, which is followed by Sect. 5 in which, we will perform experiments on RTS formed by our algorithm and to verify and validate methodology, we will be showing verification, cross verification and validation test on real time data set and Sect. 6 will conclude the paper and gives the directions for future work.

3 Data-set and background

In this section, we will give brief detail on data set used in this work, we will explain agglomerative hierarchical clustering [1]. Our proposed algorithm follows the hierarchical procedure as agglomerative hierarchical clustering does, but for making RTS. Then we will discuss about the similarity measure technique for time series data and the Sen's Median slope Estimation method, which we will be used to verify and validate the Representative Time Series (RTS), respectively.

3.1 Data-set

We used 100 years long, rainfall time-series data of 28 states (624 districts) of India. The data set used in this study is obtained from the Indian Meteorological Department (IMD), Pune [10]. Time-series considered in the experiment is of equal length and due to real world data used, it might contain outlier, noise, etc. In this paper, we will discuss results and experiments of 5 states Maharashtra, Madhya Pradesh, Orissa, Punjab and West Bengal (includes 136 districts).

While defining similarity measure for time series data there are many difficulties are faced [11] and to remove them at some extent we need to normalize the time sequence [1]. Z-score normalization technique [12] can be used to assure that all values of the input, time-series $T = \{t_1, t_2, t_3, \ldots, t_n\}$ (n = no. of time-series) are transformed into the series whose mean $\mu(T)$ is approximately 0 while standard deviation $\sigma(T)$ is in the range close to 1. Using eq. (1) input time-series T is replaced by the normalizes series T', where

$$t'_i = \frac{t_i - \mu(T)}{\sigma(T)} \quad \text{for i} = 1 \text{ to } n \tag{1}$$

3.2 Agglomerative hierarchical clustering

Agglomerative Nesting (AGNES), an agglomerative hierarchical clustering method [13]. It start with assumption that data contains that many cluster as many data points are present in it and place each data point into a separate cluster, then it start merging these data point step by step using some criteria [14]. AGNES terminate when single cluster is remain to be merge. In this way AGNES create a binary tree structure known as dendrogram (a simplified model in which data that are "close" have been grouped into a hierarchical tree).

3.3 Similarity/distance measure

There are many similarity measures which can be used for time series data dynamic time warping (DTW) [15], triangle distance measure (TDM) [16], Spear-man rank correlation coefficient, Pearson correlation coefficient, Euclidean [16]. In this paper, we will use TDM and DTW for verification and cross verification respectively. Similarity measures give an

$n \times n$ matrix as output; in which each cell represent the distance between the two time series.

TDM is used to verify the results produce by our algorithm and to generate the RTS. TDM considers each time-series as a vector in n-dimensional space. Let r_i be a time-series object of n-dimension, $r_i = \{r_{i1}, r_{i2}, r_{i3}, \ldots, r_{in}\}$. The standardized time-series object $\widehat{r}_i = \{\widehat{r}_{i1}, \widehat{r}_{i2}, \widehat{r}_{i3}, \ldots, \widehat{r}_{in}\}$, where

$$\widehat{r}_{ij} = \frac{r_{ij}}{\left(\sum_{k=1}^{t} r_{ik}^2\right)^{1/2}} \tag{2}$$

The TDM between r_i and r_j is defined by eq. (3)

$$d(r_i, r_j) = \frac{\sum_{k=1}^{t} r_{ik} r_{jk}}{\left(\sum_{k=1}^{t} r_{ik}^2\right)^{1/2}\left(\sum_{k=1}^{t} r_{jk}^2\right)^{1/2}} = 1 - \sum_{k=1}^{t} \widehat{r}_{ik}\widehat{r}_{jk} \tag{3}$$

TDM is the cosine of the triangle between two vectors, so the value lies from 0 to 2 [15]. The value is 0, if two vectors having similar direction and overlapping, which shows that two time-series vectors are almost similar to each other. On the other hand, if two time-series are opposite in direction, but overlapping, the value is 2, it shows the two most different time-series vectors.

DTW is a classical option available for calculating the similarity between the two time-series [15]. DTW is important because it doesn't require the same length time-series objects. DTW gives similarities in walking patterns, because if one time-series vary in time and speed with another, yet we get accurate results. It optimally align (or 'warping') two time sequences $r_i = \{r_1, r_2, r_3, \ldots, r_n\}$ and $s_i = \{s_1, s_2, s_3, \ldots, s_n\}$ of length n and m respectively, so that the difference between them is minimized. Using dynamic programming efficiently, this difference can be obtained between two time-series, with the matrix D which is initialized to $D_{0,0} = 0$ and all other cells, $D_{i,j} = \infty$, recursively applying

$$D_{i,j} = d(r_i, s_i) + min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\} \tag{4}$$

Where $i = 1$ to n and $j = 1$ to m, we get an n x m matrix, where the distance between the two points, r_i and s_i are calculated using the Euclidean distance function [1] and the distance between the sequence r and s is value of the last cell $D_{m,n}$.

3.4 Sen's median slope estimator

Sen's Median Slope Estimator test is commonly used with Mann-Kendall (MK) test [17] to detect the trend present in series. This test will used in this paper to measure magnitude of the trend to validate the RTS.

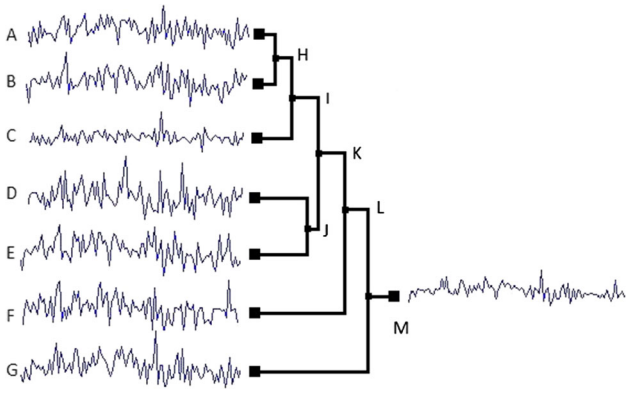

Fig. 1 Dendrogram for Representative Time Sequence

Table 1 Cluster 1 information for Maharashtra State

State:Maharashtra (MH)	
Cluster identifier	Name of districts
MH_C1	AHEMADANAGAR
	KOLHAPUR
	NASIK
	SINDHUDURG

The MK test suggests the presence of a trend in the series, but its magnitude shows the trend nature, i.e. whether the trend is increasing or decreasing. To estimate the trend nature Sen's median slope estimator test is use, which is not affected by outliers[18]. For N pairs of data, slope estimate as Eq. (5):

$$Q_i = \frac{(x_j - x_k)}{j - k} \quad \text{for i} = 1 \text{ to n} \tag{5}$$

Here x_j and x_k are annual values in year j and k of a series, respectively, where $j > k$. Median of these N values of Q_i is Sen's median estimator of slope. Where

$$N = \frac{n(n-1)}{2} \tag{6}$$

4 Finding representative time sequence

Growing time-series data depict the trends present in the observed value over time, and hence, it is important to capture valuable information that users may wishes to analyze and understand. Due to huge time-series data generated from many applications described in introduction section, trend analysis becomes an important and challenging problem.

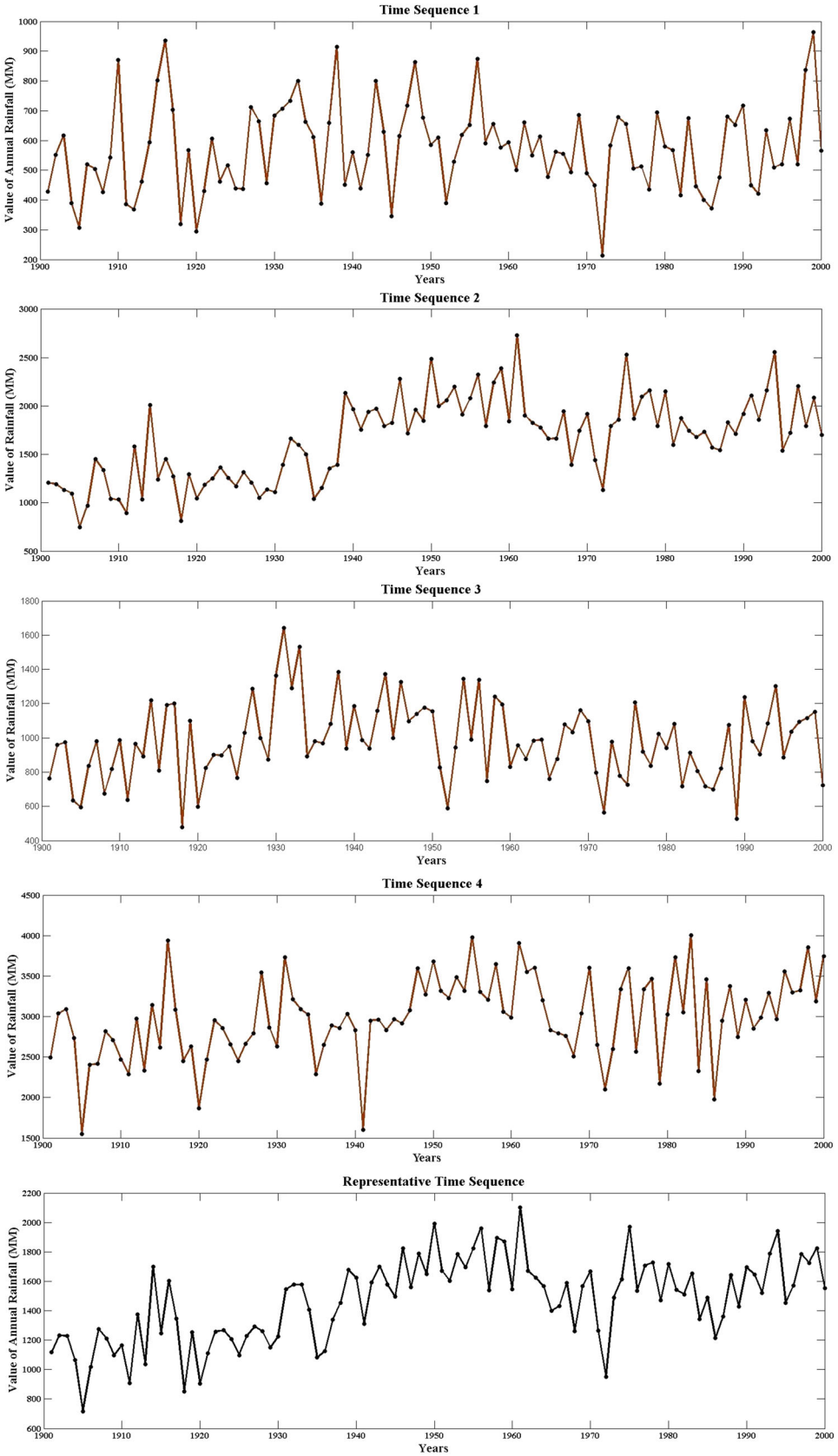

Fig. 2 Representative tme sequence

Table 2 Clustering result

Cluster ID	Districts
State 1: Madhya Prades	
MP_C1	BHIND, DATIA, GWALIOR, MORENA, SHEOPUR, SHIVPURI
MP_C2	BARWANI, BETUL, BHOPAL, DEWAS, DHAR, HARDA, INDORE, JHABUA, KHANDWA, MANDSAUR, NEEMUCH, RATLAM, SEHORE, SHAJAPUR, UJJAIN
MP_C3	BALAGHAT, CHHATARPUR, CHINDWARA, DAMOH, HOSHANGABAD, JABALPUR, KATNI, MANDLA, NARSIMHPUR, PANNA, RAJGARH, REWA, SAGAR, SATNA, SEONI, SHAHDOL, SIDHI, VIDISHA
MP_C4	RAISEN, TIKAMGARH, GUNA, KHARGAON, DINDORI
State 2: Orissa	
OR_C1	SAMBALPUR, DEBAGARH, JAGATSINGHAPUR, RAYAGADA
OR_C2	JHARSUGUDA, KENDUJAR, MAYURBHANJ, BALESHWAR, BHADRAK, JAJPUR, KENDRAPARA, CUTTAK, DHENKANAL, ANGUL, BAUDH, KANDHAMAL, GANJAM, GAJAPATI, NAYAGARH, KORDHA, PURI, KORAPUT
OR_C3	BALANGIRI, KALAHANDI
OR_C4	SUNDARGARH, BARAGARH, NUAPADA, SONAPUR, NABARANGAPUR, MAL-KANGIRI
State 3: Punjab	
PB_C1	LUDHIANA
PB_C2	GURDASPUR, AMRITSAR. HOSHIARPUR, JALANDHAR, NAWANSHAHR, ROOPNAGAR, PATIALA, SANGRUR, MOGA
PB_C3	FIROZPUR, MUKTSAR, BHATINDA
PB_C4	KAPURTHALA, FATEHGARH, FARIDKOT, MANSA
State 4: West-Bengal	
WB_C1	PURULIYA, ALIPUR, DARJEELING
WB_C2	MURSHIDABAD, BIRBHUM, BARDDHAMAN, HUGLI, HAORA, MEDINPUR, BANKURA, KOLKATA, NADIA
WB_C3	DINAJPUR NORTH, DINAJPUR SOUTH, MALDA
WB_C4	JALPAIGURI, BIHAR

As we are not interested in what the exact values each time-series has, we introduce a hierarchical time-series merging algorithm which can be very useful to analyze time-series data because it reduces the overhead of considering all time sequences in analysis. There are many clustering techniques exist for time series data [11] such as k-means, hierarchical, relocation, etc. By applying clustering technique on time series, we will get the clusters, which contain time sequences those are similar to each other, but dissimilar to sequences present in other clusters. By applying clustering technique on time series, we will get the clusters, which contain time sequences those are similar to each other, but dissimilar to sequences present in other clusters.

Clusters containing time sequences, which are used to form representative sequence. We compute distance matrix for the sequences of each cluster using similarity measure, every cell of matrix shows the distance between the two sequences, now the two time sequences having least distance between them are combine to each other and form a new sequences. To combine two sequences we take average of each data point in sequence. Let $R = \{r_1, r_2, r_3, \ldots, r_n\}$ and $S = \{s_1, s_2, s_3, \ldots, s_n\}$ are the two time sequences of length n, then new sequences T is formed using eq. (7).

$$t_i = \frac{(r_i + s_i)}{2} \quad \text{for i} = 1 \text{ to n} \tag{7}$$

Algorithm 1: Finding Representative Time Sequence (RTS)

Input: Clusters form by Hierarchical clustering.
Output: RTS for Each Cluster.

```
repeat
    combine the two closest time sequence by taking
    average for each data point.
    update the distance matrix to reflect distances
    between new sequence and remaining time
    sequences.
until only one sequence remains to process.
return RTS
```

Algorithm 1 explains the complete procedure for producing RTS. Fig. 1 show how Algorithm 1 works, which is

Table 3 Similarity of representative time sequence with original sequence of cluster

Original Time [t] Sequence of Cluster	State 1: Madhya Pradesh				State 2: Orissa			
	MH_C1	MH_C2	MH_C3	MH_C4	OR_C1	OR_C2	OR_C3	OR_C4
T.S. 1	0.43	0.42	0.43	0.56	0.35	0.50	0.12	0.48
T.S. 2	0.45	0.34	0.35	0.47	0.39	0.40	0.23	0.38
T.S. 3	0.28	0.45	0.44	0.51	0.23	0.48		0.51
T.S. 4	0.47	0.40	0.36	0.37	0.55	0.38		0.61
T.S. 5	0.43	0.45	0.36	0.55		0.50		0.62
T.S. 6	0.33	0.54	0.52			0.61		0.45
T.S. 7		0.35	0.43			0.62		
T.S. 8		0.49	0.49			0.45		
T.S. 9		0.49	0.31			0.60		
T.S. 10		0.48	0.60			0.63		
T.S. 11		0.45	0.41			0.57		
T.S. 12		0.49	0.42			0.49		
T.S. 13		0.44	0.39			0.39		
T.S. 14		0.61	0.43			0.52		
T.S. 15		0.54	0.35			0.39		
T.S. 16			0.24			0.24		
T.S. 17			0.15			0.19		
T.S. 18			0.38			0.39		

Table 4 Similarity of representative time sequence with original sequence of cluster

Original tme sequence of cluster	State 3: Punjab				State 4: West-Bengal			
	PB_C1	PB_C2	PB_C3	PB_C4	WB_C1	WB_C2	WB_C3	WB_C4
T.S. 1	0.00	0.65	0.25	0.12	0.34	0.59	0.29	0.12
T.S. 2		0.64	0.39	0.28	0.20	0.58	0.35	0.09
T.S. 3		0.64	0.37	0.16	0.84	0.56	0.30	
T.S. 4		0.24		0.11		0.70		
T.S. 5		0.30				0.42		
T.S. 6		0.12				0.61		
T.S. 7		0.28				0.47		
T.S. 8		0.16				0.38		
T.S. 9		0.11				0.51		

quite similar as Hierarchical clustering algorithm, because every time two most similar time sequences get merged and similarity matrix is updated for new time sequence with the original time sequences those are left to process. Algorithm terminates when single sequence is left to process, which is known as the RTS of the cluster on which algorithm operates. Algorithm 1 operates for each cluster formed in time-series data.

We use clustering results in our algorithm and form dendrogram for each cluster to combine the time-series. All series of a cluster are merged in hierarchical fashion. We have shown some results for RTS, which is for Cluster 1 (MH_C1) of Maharashtra state having 4 time sequences in it. Table 1 shows the cluster 1 information for Maharashtra state. Figure 2 represents time sequences of MH_C1 and its RTS, which graphically demonstrates how well RTS, follows the other sequences of cluster.

5 Experiment and results

We introduce an algorithm for finding the RTS for the clusters time series data. We verify RTS formed by proposed algorithm, via calculating its similarity with all the original sequences through which that RTS is formed. The representative sequence will used for trend analysis and

because we form it by merging all the sequences of cluster, we need to validate it on constraint that, it must follow the same behavior (i.e. trend) as all or most of the sequences present in the cluster have and to do this we will use Sen's median slope estimation method.

5.1 Clustering result

To perform verification and validation over RTS, we need to form clusters of time-series data, to form clusters of time sequences we use group average hierarchical clustering. In this paper, we are showing results for 100 years long rainfall time-series data of 108 districts of 4 states Madhya Pradesh, Orissa, Punjab and West-Bengal respectively. Table 2 show the clustering result for Madhya Pradesh, Orissa, Punjab, West-Bengal respectively.

5.2 Verification

To verify that the RTS formed by our algorithm, we measure the similarity of RTS with all the original time sequences of that cluster and to measure the similarity between RTS and original sequences, TDM similarity measure is used, because it produces output in the range from 0 to 2. Value near to zero represents the closeness of RTS to original sequences of cluster.

We show the results of similarity measure in Tables 3 and 4 for rainfall time-series of 4 states Madhya Pradesh, Orissa, Punjab and West-Bengal respectively. Columns in Tables 3 and 4 represents the RTS of cluster and rows represent the original time sequences of the cluster corresponding to the column number. Result for similarity measure shows almost all values near to 0, which depicts that the representative series is well formed for each cluster and can be useful to represent each cluster using RTS.

5.3 Cross verification

To Cross Verifiy that the RTS found for the a particular cluster best fits for only that cluster, we have calculated similarity measure for RTS of one cluster to the original time sequences of other clusters.

In Tables 5, 6, 7, 8 we shows the results for cross verification of RTS for rainfall time-series of Maharashtra

Table 5 Cross verification for cluster 1

MH_C1 Time Seqeunce	RTS of clusters			
	RTS 1	RTS 2	RTS 3	RTS 4
1	**0.78**	0.88	8.88	40.58
2	**0.39**	1.43	10.50	43.96
3	**0.76**	0.90	8.96	40.74
4	**0.74**	0.93	9.03	40.89

Bold column corresponds to their RTS number, which shows that the RTS of particular cluster number, represent that cluster very well and not suitable for any other cluster

Table 6 Cross verification for cluster 2

MH_C2 Time Seqeunce	RTS of clusters			
	RTS 1	RTS 2	RTS 3	RTS 4
1	6.03	**0.40**	1.99	23.04
2	4.91	**0.16**	2.72	25.39
3	3.21	**0.00**	4.30	29.86
4	4.81	**0.14**	2.79	25.62
5	4.81	**0.14**	2.80	25.62
6	4.19	**0.05**	3.31	27.13
7	6.76	**0.60**	1.60	21.67

Bold column corresponds to their RTS number, which shows that the RTS of particular cluster number, represent that cluster very well and not suitable for any other cluster

Table 7 Cross verification for cluster 3

MH_C3 Time Seqeunce	RTS of clusters			
	RTS 1	RTS 2	RTS 3	RTS 4
1	17.75	5.71	**0.12**	9.26
2	14.64	4.02	**0.00**	11.75
3	15.06	4.24	**0.00**	11.39
4	19.55	6.76	**0.31**	8.03
5	15.25	4.34	**0.00**	11.22
6	10.19	1.88	**0.45**	16.51
7	22.71	8.66	**0.81**	6.20
8	13.49	3.42	**0.04**	12.84
9	17.95	5.83	**0.14**	9.11
10	21.60	7.98	**0.61**	6.80
11	20.12	7.09	**0.38**	7.67

Bold column corresponds to their RTS number, which shows that the RTS of particular cluster number, represent that cluster very well and not suitable for any other cluster

Table 8 Cross verification for cluster 4

MH_C4 Time Seqeunce	RTS of clusters			
	RTS 1	RTS 2	RTS 3	RTS 4
1	41.31	21.20	6.56	**0.69**
2	35.28	16.95	4.30	**1.73**
3	40.83	20.86	6.37	**0.75**
4	38.03	18.87	5.30	**1.18**
5	37.19	18.29	4.99	**1.34**
6	52.79	29.63	11.56	**0.00**

Bold column corresponds to their RTS number, which shows that the RTS of particular cluster number, represent that cluster very well and not suitable for any other cluster

Table 9 Trends on representative time sequence with original sequence of cluster

Clusters	Segments	Time sequences of cluster															Counts			RTS	AVG
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	P	N	L		
Cluster 1	1	P	P	P	P	P	P	P	P	P	P						10	0	0	P	P
	2	N	N	N	P	N	P	P	P	P	N						**5**	**5**	0	**P**	**P**
	3	N	N	N	N	P	P	P	P	P	N						**5**	**5**	0	**P**	**P**
	4	N	N	N	N	N	P	N	N	P	N						**2**	**8**	0	**N**	**P**
Cluster 2	1	P	P	P	P	P	N	N	P	N	N	N	N				6	6	0	P	P
	2	N	P	N	N	P	P	P	P	N	N	P	P				**7**	**5**	0	**P**	**N**
	3	N	N	N	P	N	N	N	N	N	N	N	P				2	10	0	N	N
	4	N	N	P	P	N	P	P	N	P	N	P	N				6	6	0	N	P
Cluster 3	1	L	P	P	P	P	N	P	P	P	N	P	P	P	P	P	12	2	1	P	P
	2	P	P	N	P	P	P	P	P	N	N	N	P	P	P	P	11	4	0	P	P
	3	N	N	P	P	N	P	P	N	P	N	P	N	N	N	N	**6**	**9**	0	**N**	**P**
	4	N	N	N	N	N	P	N	N	P	P	N	N	P	N	N	4	11	0	N	N
Cluster 4	1	P	P	P	P	P	P	P	N	P	P	P	P	P	P	P	14	1	0	P	P
	2	P	P	P	P	P	N	P	N	P	P	P	N	P	P	P	12	3	0	P	P
	3	N	N	N	N	N	P	N	P	N	N	N	N	N	P	P	4	11	0	N	N
	4	N	N	N	N	N	N	N	N	N	N	P	P	P	P	P	5	10	0	N	N
Cluster 5	1	P	P	P	P	P	N	P	P	N	P	P	P	P	N	P	12	3	0	P	P
	2	P	P	P	P	P	P	N	P	P	P	P	P	P	P	P	14	1	0	P	P
	3	N	N	N	P	N	P	N	N	N	N	P	N	N	P	P	**5**	**10**	0	**N**	**P**
	4	P	N	N	N	N	P	N	N	N	N	N	P	P	N	N	3	12	0	N	N
Cluster 6	1	P	P	P	P	P	P	P	P	P	P	P	N	P	P	P	14	1	0	P	P
	2	P	P	P	P	P	P	P	P	N	N	N	N	N	N	P	9	6	0	P	P
	3	N	N	N	P	P	P	P	P	N	N	N	P	N	N	N	6	9	0	N	N
	4	N	P	N	P	P	N	N	N	P	N	P	N	P	N	P	**7**	**8**	0	**P**	**N**
Cluster 7	1	P	P	P	P	P	P	P	N	N	P	P	P	P	P	P	13	2	0	P	P
	2	N	P	P	P	N	P	P	N	P	P	P	P	N	P	P	12	4	0	P	P
	3	N	N	N	N	P	N	N	N	P	P	P	N	N	P	P	6	9	0	N	N
	4	P	N	N	N	N	P	N	N	P	P	N	P	P	P	P	8	7	0	N	N
Cluster 8	1	P	P	P	P	P	P	P	N	P	P	N	N	N	P	N	10	5	0	P	P
	2	P	P	P	P	P	P	N	P	P	P	P	P	P	P	P	14	1	0	P	P
	3	P	P	P	N	N	N	P	N	P	N	P	P	P	N	P	6	9	0	N	N
	4	P	N	P	P	N	N	N	N	N	P	N	P	N	N	N	5	10	0	N	N
Cluster 9	1	P	P	N	N	P	P	P	P	P	P	P	P	P	P	P	13	2	0	P	P
	2	P	N	N	N	P	P	P	P	P	N	P	N	P	P	P	10	5	0	P	P
	3	P	P	N	N	N	P	N	P	N	N	P	N	N	P	N	**6**	**9**	0	**N**	**P**
	4	P	P	P	N	P	N	P	N	N	P	P	P	P	N	P	**10**	**5**	0	**N**	**N**
Cluster 10	1	P	P	P	P	P	P	P	N	P	P	P	N	P	P	P	13	2	0	P	P
	2	P	P	P	N	N	P	P	P	N	P	P	P	P	P	P	12	3	0	P	P
	3	P	P	P	N	N	N	N	N	N	N	N	N	P	P	P	**6**	**9**	0	**N**	**P**
	4	N	N	N	N	N	N	N	N	N	P	N	N	N	N	N	1	14	0	N	N
Cluster 11	1	P	P	N	P	P	P	P	P	P	P	P	P	P	P	P	14	1	0	P	P
	2	P	P	P	P	P	P	P	P	P	P	N	P	P	P	P	14	1	0	P	P
	3	N	N	P	P	P	N	N	N	P	P	N	N	N	P	P	7	8	0	N	N
	4	N	P	N	P	N	N	N	P	N	N	P	N	N	P	P	6	9	0	N	N

Table 9 continued

Clusters	Segments	Time sequences of cluster															Counts			RTS	AVG
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	P	N	L		
Cluster 12	1	P	P	P	P	P	P	P	N	P	P	P	P	P	P	N	13	2	0	P	P
	2	P	N	P	P	P	N	P	P	P	N	N	P	P	P	P	**11**	**4**	0	**P**	**N**
	3	P	N	N	N	N	P	P	N	N	N	N	N	N	N	N	3	12	0	N	N
	4	N	N	N	P	N	N	N	P	N	N	P	N	P	N	P	5	10	0	N	N

Bold letters are to show the differences between RTS and AVG algorithm

Table 10 Comparison between hierarchical merging vs simple average method

Summary			RTS formed using hierarchical merging	RTS formed by taking simple average	Possible reasons
Positive	Negative	Plateu			
5	5	0	P	P	HP Sequences present.
2	8	0	N	P	HP Sequence present.
7	5	0	P	N	HN Sequence present.
6	9	0	N	P	HP Sequence present.
5	10	0	N	P	HP Sequence present.
7	8	0	P	N	HN Sequence present.
6	9	0	N	P	HP Sequence present.
10	5	0	N	N	HN Sequence present.
6	9	0	N	P	HP Sequence present.
11	4	0	P	N	HN Sequence present.

state divided in to four clusters. In these table, cells contains similarity of RTS of cluster (e.g. for cluster 1 RTS is known as RTS 1 and so on) to the original time sequences of the cluster. DTW is used as similarity measure to calculate the similarity.

The results derived by this experiment clearly shows that the RTS of particular cluster represent that cluster very well and not suitable for any other cluster.

5.4 Validation

The RTS will have used further to analyze the trends in series. To validate that the RTS of a cluster and original sequences of that cluster give approximately the same results in trend, we perform validation test over RTS.

As we are having time sequences of 100 data points, we divide each sequence and RTS of the cluster into 4 segments of 25 data points in each. Then using Sen's median slope estimator, the magnitude of the trend line of each segment is measured. We assign a letter 'P', 'N' and 'L' for positive, negative and plateau trend respectively. Table 9 shows the result, which indicates that the pattern formed by RTS of each cluster, approximately follows the pattern formed by original time-series present in respective cluster.

To validate, we used 172 districts of India. In Table 9, first column is cluster number, second column show the segment number of the time sequence, third (big column contain 15 sub-column for original time sequences of cluster)and fifth column represent the trend result of original and RTS respectively for each segment. Forth column contains the number of positive (P), negative (N) and plateau (L) trend occurred in each segment by original time sequences of cluster.

5.5 Comparision of RTS

In Table 9 we compare the result of merging algorithm with average algorithm to form RTS. In average algorithm, RTS is formed by taking the average of all sequences simultaneously. To highlight the results we use bold fonts where difference occur in merging algorithm and average algorithm. There are 8 times in which average algorithm gives wrong answer while merging algorithm gives 2 times out of 48 chances.

In Table 10, we highlight the difference between the results for RTS formed using Hierarchical merging algorithm and RTS formed by taking simple average. In that table we also include the possible reason behind the result derived. In cluster of time sequences there are many type of combination can be found out. We explain some of combination which might changes the results for two algorithm used.

- Highly positive (HP): sequence containing high positive values.
- Highly negative (HN): sequence containing high negative values.
- Low positive (LP): sequence containing low positive values.
- Low negative (LN): sequence containing low negative values.

6 Conclusion and future work

Trend analysis is most common subject in time series domain and time series data set is often very large, due to which trend analysis task becomes time consuming. To reduce that time at some extend, in this paper, we introduced an algorithm, which uses clusters of time series as input and generate representative time series for each cluster. In this way lots of time sequences similar to each other not consider in trend analysis again and again. We also demonstrate the empirically quality of RTS by calculating its similarity with original sequences and also derive the trend results for RTS and original time sequences of clusters.

Using RTS, we can perform many kind of regression test, trend detection test on it and analyze it to predict the future behavior.

In future, we might try to apply our algorithm on some other time series dataset. Concept of RTS is not only covers the trend analysis but can also be very useful in other analysis performed on time series data such as finding regular pattern, forecasting, etc.

References

1. Ratanamahatana CA, Lin J, Gunopulos D, Keogh E, Vlachos M, Das G (2010) Mining time series data. In: Maimon O, Rokach L (eds) Data mining and knowledge discovery handbook. Springer, New York, pp 1049–1077
2. Oguntunde PG, Abiodun BJ, Lischeid G (2011) Rainfall trends in Nigeria, 19012000. J Hydrol 411(3):207–218
3. Babar SF, Ramesh H (2013) Analysis of south west monsoon rainfall trend using statistical techniques over Nethravathi basin. IJATCE 2.1:130–136
4. Jain SK, Kumar V (2012) Trend analysis of rainfall and temperature data for India. Curr Sci India 102(1):37–49
5. Longobardi A, Villani P (2010) Trend analysis of annual and seasonal rainfall time series in the Mediterranean area. Int J Climatol 30(10):1538–1546
6. Agrawal R, Faloutsos C, Swami A (1993) Efficient similarity search in sequence databases. Springer, Berlin, pp 69–84
7. Chan K-P, Fu AWC (1999) Efficient time series matching by wavelets. In: Proceedings of the 15th IEEE international conference on data engineering, pp 126–133
8. Wu D, Singh A, Agrawal D, Abbadi AEI, Smith TR (1996) Efficient retrieval for browsing large image databases. In: Proceedings of the fifth international conference on Information and knowledge management. ACM, pp 11–18
9. Keogh EJ, Pazzani MJ (1998) An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. KDD 98:239–243
10. For data-set: http://www.imdpune.gov.in/ (2014)
11. Warren Liao T (2005) Clustering of time series data—a survey. Pattern Recognit 38(11):1857–1874
12. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J Finance 23(4):589–609
13. Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis, vol 344. Wiley, Chichester, pp 199–250
14. Pang-Ning T, Steinbach M, Kumar V (2006) Introduction to data mining, 3rd edn. Pearson Education, New Delhi, pp 487–554
15. Keogh E, Ratanamahatana CA (2005) Exact indexing of dynamic time warping. Knowl Inf Syst 7:358–386
16. Zhang, X, Wu J, Yang X, Ou, H, Lv T (2009) A novel pattern extraction method for time series classication. Optimization and Engineering 10, vol. 2, pp. 253-271
17. Mann HB (1945) Non-parametric test against trend. Econometrika 13:245–259
18. Yue S, Paul P, George C (2002) Power of the MannKendall and Spearman's rho tests for detecting monotonic trends in hydrological series. J Hydrol 259(1):254–271