Analysis of Dendrogram Tree for Identifying and Visualizing Trends in Multi-attribute Transactional Data

D.Radha Rani[1], A.Vini Bharati[2], P.Lakshmi Durga Madhuri[3], M.Phaneendra Babu[4], A.Sravani[5]

Department Of Computer Science and Engineering, KLCE, Vaddeswaram, Guntur Dt., Andhra Pradesh, India

***Abstract* –Most of the data collected by organizations and firms contains multi-attribute and temporal data. Identifying temporal relationships (e.g., trends) in data constitutes an important problem that is relevant in many business and academic settings. Data mining techniques are used to discover patterns in such data. Temporal data can take many forms, most commonly being general transactional (multi)attribute-value data, for which time series or sequence analysis methods are not particularly well suited. In this paper we present the clustering algorithm with performance and implementation of dataset based on distances in miles between US cities.**

Key Words-Temporal Data Mining, Clustering, Data mining, data visualization, trend analysis

I. INTRODUCTION

Real life transactional data often poses challenges such as very large size, high dimensionality. Consider the problem of technology forecasting for a firm. Technologies possess many features that change over time and understanding how a technology evolves requires trend analysis of multiple attributes at once. Similar issues arise in the trend analysis of consumer purchasing behavior and many other business intelligence applications. An important usage of time sequences is to discover temporal patterns. . In this paper, we present C-TREND, *Cluster-based Temporal Representation of EveNt Data*, a new method for discovering and visualizing trends and temporal patterns in transactional attribute-value data that builds upon standard data mining clustering techniques. The discovery process usually starts with a user specified skeleton, called an event structure, which consists of a number of variables representing events and temporal constraints among these variables; the goal of the discovery is to find temporal patterns. Domain experts have the ability to adjust parameters and clustering mechanisms to fine-tune trend graphs. The C-TREND implementation is scalable. The time required to adjust trend parameters is quite low even for larger data sets, which provides for real-time visualization capabilities. Furthermore, the proposed temporal clustering analysis technique is applicable in many different data analysis contexts and can provide insights for analysts performing historical analyses and generating forecasts.

II. RELATED WORK

Temporal data mining approaches depend on the nature of event sequence being studied. The most common form of temporal data mining –time series analysis [1] [2] [3] – is used to mine a sequence of continuous real-valued elements and is often regression based. Another common technique is sequence analysis [4][5]– used when the sequence is composed of a series of nominal symbols [5]. There are two main statistical techniques-Regression, Classification. Regression is the oldest and most well-known statistical technique that the data mining community utilizes. Basically, regression takes a numerical dataset and develops a mathematical formula that fits the data. Classification technique is capable of processing a wider variety of data than regression and is growing in popularity. We'll also find output that is much easier to interpret. Instead of the complicated mathematical formula given by the regression technique we'll receive a decision tree that requires a series of binary decisions. This system uses clusters identified in multiple time periods and identifies trends based on similarities between them. It is a clustering approach for discovering temporal patterns, which builds on temporal clustering methods and complements existing temporal mining methods. In this project we use DENDROGRAM Data structure for storing and Extracting cluster solutions generated by hierarchical clustering algorithms. Calculations are made using Tree data structures.

Advantages of Proposed System:

Efficiency is considerably increased in terms of both time complexity and space complexity. It provides the end user with the ability to generate graphs from data and adjust the graph parameters dynamically. Visual data exploration is the process of presenting data in some visual form and allowing the human to interact with the data to create insightfulness representations. It typically follows the overview, zoom and filter, and details on demand. Interaction techniques provide the user with the ability to dynamically change visual representations and can empower the user's perception of information. Interactive filtering involves dynamically partitioning a data set into segments and focusing on interesting subsets properties. Interactive zooming is a common technique that provides the user with variable display of data at different levels of analysis. It builds on temporal data mining techniques and develops a tool that provides the user with the ability to interact with temporal cluster graph data visualization. Temporal cluster graphs use hierarchical and graph-based techniques [7][8] to explore temporal data and providing interactive filtering and zooming capabilities for visualization.

Dendrogram tree construction and extracting cluster solutions

In this module hierarchical clustering techniques are used for clustering the datasets.

Procedure for Hierarchical Clustering: Given a set of N items to be clustered, and an NxN distance (or similarity) matrix, the basic process of hierarchical clustering is this: Start by assigning each item to its own cluster, so that if we have N items, we now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster. Compute distances (similarities) between the new cluster and each of the old clusters. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. Step 3 can be done in different ways, which is what distinguishes single-link from complete-link and average-link clustering. In single-link clustering we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster. In complete-link clustering (also called the diameter or maximum method), we consider the distance between one cluster and another cluster to be equal to the longest distance from any member of one cluster to any member of the other cluster. In average-link clustering, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster. This kind of hierarchical clustering is called agglomerative because it merges clusters iteratively.

Single Linkage Clustering: The algorithm is an agglomerative scheme that erases rows and columns in the proximity matrix as old clusters are merged into new ones. The N*N proximity matrix is D = [d(i,j)]. The clusterings are assigned sequence numbers 0, 1... (n-1) and L (k) is the level of the k^{th} clustering. A cluster with sequence number m is denoted (m) and the proximity between clusters (r) and (s) is denoted d [(r),(s)].

The algorithm is composed of the following steps

1. Begin with the disjoint clustering having level L (0) =0 and sequence number m=0.
2. Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)] where the minimum is over all pairs of clusters in the current clustering.
3. Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)]
4. Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r, s) and old cluster (k) is defined in this way: d [(k), (r, s)] = min d[(k), (r)], d [(k),(s)].
5. If all objects are in one cluster, stop. Else, go to step 2.

The following pages trace a hierarchical clustering of distances in miles between U.S. Cities. The method of clustering is single-link.

Table 1
Input distance (dissimilarity) matrix

	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NY	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

The nearest pair of cities is BOS and NY, at distance 206. These are merged into a single cluster called "BOS/NY" in the first step. Then we compute the distance from this new compound object to all other objects. In single link clustering the rule is that the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object. So the distance from "BOS/NY" to DC is chosen to be 233, which is the distance from NY to DC. Similarly, the distance from "BOS/NY" to DEN is chosen to be 1771.

Table 2
After merging BOS with NY

	BOS/NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS/ NY	0	223	1308	802	2815	2934	2786	1771
DC	223	0	1075	671	2684	2799	2631	1616
MIA	1308	1075	0	1329	3273	3053	2687	2037
CHI	802	671	1329	0	2013	2142	2054	996
SEA	2815	2684	3273	2013	0	808	1131	1307
SF	2934	2799	3053	2142	808	0	379	1235
LA	2786	2631	2687	2054	1131	379	0	1059
DEN	1771	1616	2037	996	1307	1235	1059	0

The nearest pair of objects is BOS/NY and DC, at distance 223. These are merged into a single cluster called "BOS/NY/DC". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix (as shown in table – 2).

Table 3
After merging DC with BOS-NY

	BOS/NY/DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY/DC	0	1075	671	2684	2799	2631	1616
MIA	1075	0	1329	3273	3053	2687	2037
CHI	671	1329	0	2013	2142	2054	996
SEA	2684	3273	2013	0	808	1131	1307
SF	2799	3053	2142	808	0	379	1235
LA	2631	2687	2054	1131	379	0	1059
DEN	1616	2037	996	1307	1235	1059	0

Now, the nearest pair of objects is SF and LA, at distance 379. These are merged into a single cluster called "SF/LA". Then we compute the distance from this new cluster to all other objects, to get a new distance matrix.(as shown in the Table – 3).

Table 4
After merging SF with LA

	BOS/ NY/DC	MIA	CHI	SEA	SF/LA	DEN
BOS/ NY/DC	0	1075	671	2684	2631	1616
MIA	1075	0	1329	3273	2687	2037
CHI	671	1329	0	2013	2054	996
SEA	2684	3273	2013	0	808	1307
SF/LA	2631	2687	2054	808	0	1059
DEN	1616	2037	996	1307	1059	0

Now, the nearest pair of objects is CHI and BOS/NY/DC, at distance 671. These are merged into a single cluster called "BOS/NY/DC/CHI". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix.

Now, the nearest pair of objects is SEA and SF/LA, at distance 808. These are merged into a single

cluster called "SF/LA/SEA". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix (as shown in table-5).

Table 5
After merging CHI with BOS/NY/DC

	BOS/NY /DC/CHI	**MIA**	**SEA**	**SF/LA**	**DEN**
BOS/NY /DC/CHI	0	1075	2013	2054	996
MIA	1075	0	3273	2687	2037
SEA	2013	3273	0	808	1307
SF/LA	2054	2687	808	0	1059
DEN	996	2037	1307	1059	0

Table 6
After merging SEA with SF/LA

	BOS/NY /DC/CHI	**MIA**	**SF/LA/SEA**	**DEN**
BOS/NY /DC/CHI	0	1075	2013	996
MIA	1075	0	2687	2037
SF/LA/SEA	2054	2687	0	1059
DEN	996	2037	1059	0

Now, the nearest pair of objects is DEN and BOS/NY/DC/CHI, at distance 996. These are merged into a single cluster called "BOS/NY/DC/CHI/DEN". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix (as shown in table-6).

Now the nearest pair of objects is BOS/NY/DC/CHI/DEN and SF/LA/SEA, at distance 1059.these are merged into a single cluster called BOS/NY/DC/CHI/DEN/SF/LA/SEA (as shown in table-7)

Table 7
After merging DEN with BOS/NY/DC/CHI

	BOS/NY/DC /CHI/DEN	**MIA**	**SF/LA/SEA**
BOS/NY/DC /CHI/DEN	0	1075	1059
MIA	1075	0	2687
SF/LA/SEA	1059	2687	0

Table 8
After merging SF/LA/SEA with BOS/NY/DC/CHI/DEN

	BOS/NY/DC/CHI /DEN/SF/LA/SEA	**MIA**
BOS/NY/DC/CHI /DEN/SF/LA/SEA	0	1075
MIA	1075	0

Finally, we merge the last two clusters at level 1075.

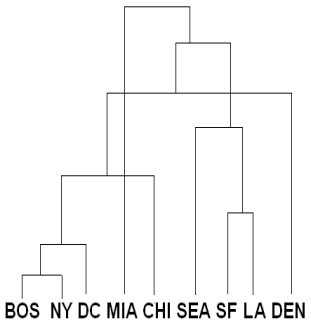

Figure 2.1: Dendrogram Tree

C-trend process

It provides the user with the ability to generate graphs from data and adjust the graph parameters. User logins by providing appropriate user id and password. Later he provides the dataset to observe the trends. In the preprocessing phase, the data set is partitioned based on time periods, and each partition is clustered using one of many traditional

clustering techniques such as a hierarchical approach. Here we have explained only upto dendrogram tree construction. Further steps can be implemented as shown in the diagram. The results of the clustering for each partition are used to generate two data structures: the node list and the edge list. Creating these lists in the preprocessing phase allows for more effective (real-time) visualization updates of the C-TREND output graphs. Based on these data structures, graph entities (nodes and edges) are generated and rendered as a temporal cluster graph in the system output window. In the interactive analysis phase, C-TREND allows the user to modify $\mathbf{K_i}$, α and β on demand in real time and, as a result, updates the view of the temporal cluster graph.

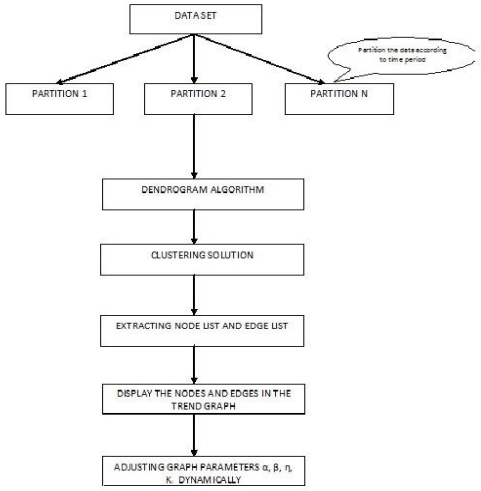

Fig 2.2: C-Trend Process Overview

III. CONCLUSIONS

In this paper we have explained the dendrogram tree construction by collecting the data of US cities. By taking the metric as distances between them. We used hierarchal clustering algorithm for this tree construction. The further implementation of this c-trend process overview (as specified in Fig 2.2) is versatile, and the implementation of the technique as the C-TREND system gives significant data representation power to the user—domain experts have the ability to adjust parameters and clustering mechanisms to fine-tune trend graphs.

ACKNOWLEDGMENT

We are very much obliged to Dr.K.Raja Shekar Principal, Koneru Lakshmaiah College of Engineering for permitting us to carry out our paper work and for providing we all support required. We are greatly indebted to our Prof.S.Venkateswarlu, Head of the Department for giving moral support and also permitting us to do this Paper. We would like to convey my heart full thanks to our guide D.RadhaRani, for her guidance and support in every step of this paper. We convey our sincere thanks to all the faculty and friends who directly or indirectly helped me for the successful completion of our paper.

REFERENCES

[1]. P. Brockwell and R. Davis, Time Series: Theory and Methods. Springer, 2001.

[2]. E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, "Data Mining and Knowledge Discovery, vol. 7, no. 4, pp. 349-371, 2003.

[3]. J. Roddick and M. Spiliopoulou, "A Survey of Temporal Knowledge Discovery Paradigms and Methods," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 4, pp. 750-767, July/Aug. 2002.

[4]. J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 10, pp. 1-17, Oct. 2004.

[5]. M. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," Machine Learning, vol. 42, no. 1-2, pp. 31-60, 2001.

[6]. C.M. Antunes and A.L. Oliveira, "Temporal Data Mining: An Overview," Proc. ACM SIGKDD Workshop Data Mining, pp. 1-13, Aug. 2001.

[7]. C. Chen, Information Visualization and Virtual Environments. Springer, 1999.

[8]. M.C.F. de Oliveira and H. Levkowitz, "From Visual Data Exploration to Visual Data Mining: A Survey," IEEE Trans. Visualization and Computer Graphics, vol. 9, no. 3, pp. 378-394,July-Sept. 2003.