ISSN-2249 5460

INTERNATIONAL JOURNAL OF MATHEMATICAL SCIENCES, TECHNOLOGY AND HUMANITIES

www.internationalejournals.com

Available online at ww.internationaleJournals.com

International eJournals

International Journal of Mathematical Sciences, Technology and Humanities 74 (2012) 832-839

Different Representations Ensemble with Temporal Data Clustering Via Weighted Clustering

1.Vamsi Krishna Jayanti and 2.Vasantha Kumar

1.Department of Information Technology,2. Department of Computer Science and Engineering

Avanthi Institute OF Technology,Makavanipalem Narsipatnam

vamsycareer@gmail.com,cumaar@gmail.com,

Abstract— Temporal data clustering provides underpinning techniques for discovering the intrinsic structure and condensing information over temporal data. In this paper, we present a temporal data clustering framework via a weighted clustering ensemble of multiple partitions produced by initial clustering analysis on different temporal data representations. In our approach, we propose a novel weighted consensus function guided by clustering validation criteria to reconcile initial partitions to candidate consensus partitions from different perspectives, and then, introduce an agreement function to further reconcile those candidate consensus partitions to a final partition. As a result, the proposed weighted clustering ensemble algorithm provides an effective enabling technique for the joint use of different representations, which cuts the information loss in a single representation and exploits various information sources underlying temporal data. In addition, our approach tends to capture the intrinsic structure of a data set, e.g., the number of clusters. Our approach has been evaluated with benchmark time series, motion trajectory, and time-series data stream clustering tasks. Simulation results demonstrate that our approach yields favorite results fora variety of temporal data clustering tasks. As our weighted cluster ensemble algorithm can combine any input partitions to generate a clustering ensemble, we also investigate its limitation by formal analysis and empirical studies.

Index Terms—Temporal data clustering, clustering ensemble, different representations, weighted consensus function, model selection

I. Introduction

Temporal clustering analysis provides an effective way to discover the intrinsic structure and condense information over temporal data by exploring dynamic regularities underlying temporal data in an unsupervised learning way. Its ultimate objective is to partition an unlabeled temporal data set into clusters so that sequences grouped in the same cluster are coherent. In general, there are two core problems in clustering analysis, i.e., model selection and grouping. The former seeks a solution that uncovers the number of intrinsic clusters underlying a temporal data set, while the latter demands a proper grouping rule that groups coherent sequences together to form a cluster matching an underlying distribution.Clusteringanalysis is an extremely difficult unsupervised learning task.

In particular, recent empirical studies reveal that temporal data clustering poses a real challenge in temporal data mining due to the high dimensionality and complex temporal correlation. In the context of the data dependency treatment, we classify existing temporal data clustering algorithms as three categories: temporal-proximity-based, model-based, and representation-based clustering algorithms.

Temporal-proximity-based and model-based clustering algorithms directly work on temporal data. Therefore, temporal correlation is dealt with directly during clustering analysis by means of temporal similarity measures e.g., dynamic time warping, or dynamic models e.g., hidden Markov model. In contrast, a representation-based algorithm converts temporal data clustering into static data clustering via a parsimonious representation that tends to capture the data dependency. Based on a temporal data representation of fixed yet lower dimensionality, any existing clustering algorithm is applicable to temporal data clustering, which is efficient in computation. Various temporal data representations have been proposed from different perspectives. To our knowledge, there is no universal representation that perfectly characterizes all kinds of temporal data; one single representation tends to encode only those features well presented in its own representation space and inevitably incurs useful information loss. Furthermore, it is difficult to select a representation to present a given temporal data set properly without prior knowledge and a careful analysis.

Our approach consists of initial clustering analysis on different representations to produce multiple partitions and clustering ensemble construction to produce a final partition by combining those partitions achieved in initial clustering analysis. While initial clustering analysis can be done by any existing clustering algorithms, we propose a novel weighted clustering ensemble algorithm of a two- stage reconciliation process. In our proposed algorithm, a weighting consensus function reconciles input partitions to candidate consensus partitions according to various clustering validation criteria. Then, an agreement function further reconciles those candidate consensus partitions to yield a final partition.

The contributions of this paper are summarized asfollows: First, we develop a practical temporal data clustering model by different representations via clustering ensemble learning to overcome the fundamental weakness in the representation-based temporal data clustering analy- sis. Next, we propose a novel

weighted clustering ensemble algorithm, which not only provides an enabling technique to support our model but also can be used to combine any input partitions. Formal analysis has also been done. Finally, we demonstrate the effectiveness and the efficiency of our model for a variety of temporal data clustering tasks as well as its easy-to-use nature as all internal parameters are fixed in our simulations. In the rest of the paper, Section 2 describes the motivation and our model, and Section 3 presents our weighted clustering ensemble algorithm along with algorithm analysis. Section 4 reports simulation results on a variety of temporal data clustering tasks. Section 5 discusses issues relevant to our approach, and the last section draws conclusions.

II. Representations of Data Clustering

II.I. Document Clustering

Document clustering is closely related to concept of data clustering. It is a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering. It breaks down huge linear results into manageable sets. It is an automatic grouping of text documents into clusters where documents of the same cluster are more similar than the documents in different clusters.

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision or recall in information retrieval systems and as an efficient way of finding the nearest neighbors of a document. More recently, clustering has been proposed for use in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query. Document clustering has also been used to automatically generate hierarchical clusters of documents.

There are two main classes of clustering algorithms

1. Hierarchical clustering and
2. Partitional clustering algorithms

II.II Hierarchical Clustering

Cluster Analysis, also called data segmentation, has a variety of goals. All relate to grouping or segmenting a collection of objects into subsets or clusters , such that those within each cluster are more closely related to one another than objects assigned to different clusters. Central to all of the goals of cluster analysis is the notion of degree of similarity (or dissimilarity) between the individual objects being clustered .

In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object. There are two basic approaches to generating a hierarchical clustering:

a) **Agglomerative**: Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.
b) **Divisive**: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

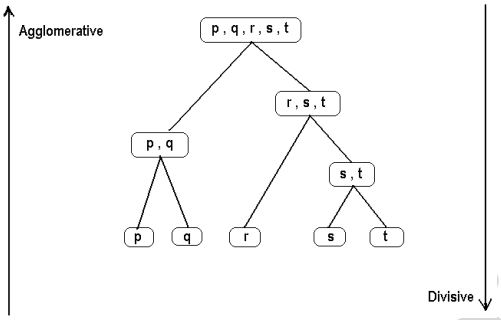

Figure 1.1.2 Hierarchical Clustering

II.III Partitional Clustering

Partitional clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

The partitional clustering is achieved by three methods.

1. K-Means Clustering

2. Fuzzy C-Means Clustering

3. QT Clustering

II.IV K-Means clustering

The *K*-Means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster - that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.

Simple K-Means Clustering Algorithm

- Choose the number of clusters, *k*.
- Randomly generate *k* clusters and determine the cluster centers, or directly generate *k* random points as cluster centers.
- Assign each point to the nearest cluster center, where "nearest" is defined with respect to one of the distance measures discussed above.

- Re-compute the new cluster centers.
- Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

III. Dynamic Document Clustering

An approach for dynamic document clustering based on structured MARDL technique is our objective. At first the documents are clustered in Static method using Bisecting K-means algorithm. For clustering of documents in bisecting K-Means, all documents should be preprocessed in the initial stage. The preprocessing stage includes stop word removal process and stemming process. In stop word removal process, words having negative influence like adverbs, conjunctions are removed and in stemming process root word will find out by removing prefixes and suffixes of the word.

After the preprocessing process, the documents should grouped into desired number of clusters. To make desired number of clusters, bisecting K-Means clustering method is used. In this method, each document is assigning a weight by term frequency and inverse document frequency method using cosine similarity measure. After assigning weight to each document, the documents are first separated into clusters using k-Means method. After clustering of documents using K-means method the largest cluster will split and forms two sub clusters and this step would be repeated for many times until clusters formed are with high similarity.

The desired number of cluster forms as a result of bisecting K-Means clustering. After the clustering of given documents into desired number of clusters, the dynamic algorithm will be used to assign a new document into a appropriate cluster. The overall process is explained in the diagram below.

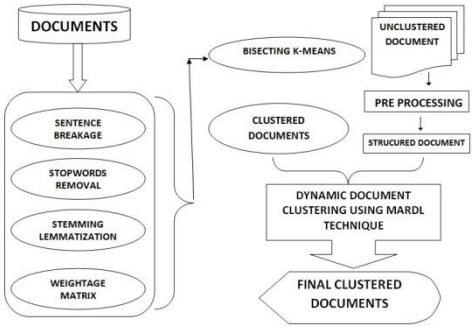

Fig 3.1.1 Overall process of Proposed work.

In dynamic clustering the new documents are preprocessed and then new documents are assigned to cluster one by one in recursive steps. The new documents are assigned to a cluster dynamically in run time without the need of re-clustering. As a result, the final clustering is obtained which yields good performance over static Bisecting K-means algorithm.

III.II Description of Proposed Dynamic Clustering Algorithm

The dynamic algorithm starts with set of clusters which is obtained from result of bisecting K-Means clustering. Then the dynamic clustering algorithm takes samples from each cluster. The samples are chosen randomly from the set of documents in each cluster. The number of samples taken should be one third of number of documents present in each cluster. The samples taken should be unique and should not any replication of documents in samples.

The new documents are preprocessed first which includes stop word removal process and stemming process. The new documents are stemmed using stemming algorithm. After preprocessing of new document, the new document is compared with samples and we calculate the influence of new document in each cluster termed as frequency value. The frequency value should be divided by number of samples taken from each cluster to make uniform frequency value of new document with each samples.

Then the dynamic algorithm assigns the new document to the cluster with high frequency value if the frequency value is within the threshold value. The threshold value is maintained for clustering process to make a document to form into a new cluster or assigning a document to appropriate cluster. If all the clusters results in frequency value less than threshold value, then the new document forms as a separate cluster.

A value is calculated through a series of experiments on all worst, average, best case inputs and it is termed as Threshold value (T_{max}).For a newly arrived document, its frequency value falls less than the Threshold value (T_{max}) and so it forms a separate cluster. Thus ensuring that no document goes without clustering even it doesn't patches with any of the existing clusters.

IV.DISCUSSION

The use of different temporal data representations in our approach plays an important role in cutting information loss during representation extraction, a fundamental weakness of the representation-based temporal data clustering. Conceptually, temporal data representations acquired from different domains, e.g., temporal versus frequency, and on different scales, e.g., local versus global as well as fine versus coarse, tend to be complementary. In our simulations reported in this paper, we simply use four temporal data representations of a complementary nature to demonstrate our idea in cutting information loss. Although our work is concerning the representation-based clustering, we have addressed little on the representation-related issues per se including the development of novel temporal data representations and the selection of representations to establish a synergy to produce appropriate partitions for clustering ensemble. We anticipate that our approach would be improved once those representation-related problems are tackled effectively.

The cost function derived in suggests that the performance of a clustering ensemble depends on both quality of input partitions and a clustering ensemble scheme. First, initial clustering analysis is a key factor responsible for the performance. According to the first term of in Section 3.3, the good performance demands the property that the variance of input partitions is small and the optimal "mean" is close to the intrinsic "mean," i.e., the ground truth partition. Hence, appropriate clustering algorithms need to be chosen to match the nature of a given problem to produce input partitions of such a property, apart from the use of different representations. When domain knowledge is available, it can be integrated via appropriate clustering algorithms during initial cluster- ing analysis. Moreover, the structural information under- lying a given data set may be exploited, e.g., via manifold clustering [40], to

produce input partitions reflecting its intrinsic structure. As long as an initial clustering analysis returns input partitions encoding domain knowledge and characterizing the intrinsic structural information, the "abstract" similarity (i.e., whether or not two entities are in the same cluster) used in our weighted clustering ensemble will inherit them during combination of input partitions. In addition, the weighting scheme in our algorithm also allows any other useful criteria and domain knowledge to be integrated. All discussed above pave a new way to improve our approach.

As a generic technique, our weighted clustering ensemble algorithm is applicable to combination of any input partitions in its own right regardless of temporal data clustering. Therefore, we would link our algorithm to the most relevant work and highlight the essential difference between them.

The Cluster Ensemble algorithm presents three heuristic consensus functions to combine multiple partitions. In their algorithm , three consensus functions are applied to produce three candidate consensus partitions, respectively, and then, the NMI criterion is employed to find out a final partition by selecting the one of the maximum NMI value from candidate consensus partitions. Although there is a two-stage reconciliation process in both their algorithm and ours, the following characteristics distinguish ours from theirs. First, ours uses only a uniform weighted consensus function that allows various clustering validation criteria for weight generation. Various clustering validation criteria are used to produce multiple candidate consensus partitions (in this paper, we use only three criteria). Then, we use an agreement function to generate a final partition by combining all candidate consensus partitions other than selection.

V.CONCLUSIONS AND FUTURE SCOPE

A novel architecture and algorithm for dynamic clustering , which allows clustering of new document into existing cluster or forming into a new cluster without the need of re-clustering of the documents has been implemented. This project concludes that it achieves comparable quality to static Bisecting K-Means algorithm which provides significant speedup and also good performance in terms of purity, inter-cluster similarity and intra cluster similarity . The importance of this contribution provides clustering of documents in run time.

For future work, we plan to extend dynamic algorithm to give more performance than present algorithm in terms of speed, purity, inter-cluster similarity and intra cluster similarity which will give more sophisticated clustering. We are also investigating the possibility of making the clustering algorithm more global by allowing centroids to cross neighborhoods through higher levels; i.e., clusters at lower level neighborhoods should be a function of higher level centroid. We believe that this will create an opportunity for better global clustering solutions but on the expense of computational complexity.

VI .REFERENCES

[1] J. Kleinberg, "An Impossible Theorem for Clustering," Advances in Neural Information Processing Systems, vol. 15, 2002.

[2] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Study," Knowledge and Data Discovery, vol. 6, pp. 102-111, 2002.

[3] A. Jain, M. Murthy, and P. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, pp. 264-323, 1999.

[4] R. Xu and D. Wunsch, II, "Survey of Clustering Algorithms," IEEE Trans. Neural Networks, vol. 16, no. 3, pp. 645-678, May 2005.

[5] P. Smyth, "Probabilistic Model-Based Clustering of Multivariate and Sequential Data," Proc. Int'l Workshop Artificial Intelligence and Statistics, pp. 299-304, 1999.

[6] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," PhD thesis, Dept. of Computer Science, Univ. of California, Berkeley, 2002.

[7] Y. Xiong and D. Yeung, "Mixtures of ARMA Models for Model- Based Time Series Clustering," Proc. IEEE Int'l Conf. Data Mining, pp. 717-720, 2002.

[8] N. Dimitova and F. Golshani, "Motion Recovery for Video Content Classification," ACM Trans. Information Systems, vol. 13, pp. 408-439, 1995.

[9] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proc. ACM SIGMOD, pp. 419-429, 1994.

[10] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrota, "Locally Adaptive Dimensionality Reduction for Indexing Large Scale Time Series Databases," Proc. ACM SIGMOD, pp. 151-162, 2001.

[11] F. Bashir, "MotionSearch: Object Motion Trajectory-Based Video Database System—Index, Retrieval, Classification and Recogni- tion," PhD thesis, Dept. of Electrical Eng., Univ. of Illinois, Chicago, 2005.

[12] E. Keogh and M. Pazzani, "A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases," Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, pp. 122-133, 2001.

[13] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," J. Machine Learning Research, vol. 3, pp. 583-617, 2002.

[14] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data," Machine Learning, vol. 52, pp. 91-118