Clustering Web Logs Using Similarity Upper Approximation with Different Similarity Measures

Rajhans Mishra and Pradeep Kumar

***Abstract*—In this paper we adopted the similarity upper approximation based clustering of web logs using various similarity/distance metrics. The paper shows the viability of our methodology. Web logs capture the information about web sites as well the sequence of the visit. Sequence of visit provides an important insight about the behavior of the user. Rough set, a soft computing technique, deals with vagueness present in data. It captures the indiscernibility at different levels of granularity. The paper has shown the results on *msnbc* data set with different similarity measures along with explanation of results.**

***Index Terms*—Clustering, sequential data, similarity upper approximation.**

I. Introduction

Clustering techniques group objects that are similar to each other with respect to different attributes [1]. Clusters can be crisp or overlapping in nature. Overlapping clusters have some objects common and they are also termed as soft clusters [2]. Clusters without any common element to each other are termed as crisp or hard clusters [2]. Clustering is widely used in the field of data mining.

Web logs contain information of web page visit of users. It provides the traversing pattern of users. Traversing pattern of users provide information about their behavior. Users can be grouped on the basis of their web usage pattern.

Web user visit can also be viewed as the sequence of web page visit. Sequence mining deals with finding the useful patterns from the sequence data sets and understanding their behavior [3]. Analysis of sequential data has become increasingly important in various application areas like biological sequences [4], intrusion detection [5] and web access logs.

Web logs contain the information regarding sequence of web pages in which they are visited by the user. All the web pages that are visited by the user in sequence are of different interest for the user. Thus during the clustering of web user sequence of the visited web pages should also be considered besides the web page.

Sequential data captures the order of occurrence of objects. The order of occurrence has an important aspect of the sequential data. Classical clustering algorithms (as K-Means) are not able to perform well in case of sequential data as it is difficult to find pair wise similarity between sequences. [6].

Any user can be a part of more than one cluster as its traversing behavior fluctuates from one area to other depending upon the need of the user.

A student registering himself for a conference will go through the conference web site and will be a member of group called academician. On the same time he will be traversing web sites for booking the flights/hotels and will be put in tourism cluster as he may have to visit places to attend conference/seminar. To address the above issue rough sets can utilized to generate the soft clusters.

Sequence based clustering has been performed to find the groups on the basis of sequential information available in the data. Sungjune et al. [3] have suggested a general sequence-based clustering method by proposing new sequence representation schemes in association with Markov models.

Lingras [7],[8] has focused on rough set clustering for web mining. Unsupervised classification using rough sets along with genetic algorithms has been used to represent clusters as interval sets. Clusters have been formed by the evolution of rough set genomes. A rough set genome has been constructed to contain the objects that have to be classified. A genome has been represented as a string which indicates its membership. Upper and lower approximation of genome has been calculated to find rough clusters.

De & Krishna [9] have considered clustering of web transactions using rough sets and presented a rough approximation based clustering of web transactions from web access logs. Kumar et al. [10] have proposed a hierarchical clustering algorithm using similarity upper approximation.

Kumar et al. [11] have proposed a clustering algorithm based on similarity upper approximation using S^3M similarity measure. We have implemented the algorithm proposed by Kumar et al. [11] using Jaccard [12], Dice [13], Levenshtein [14] and S^3M [15] similarity metrics on *msnbc* data set.

The current paper deals with finding the outlier (data point which is not grouped in any cluster) and clusters from the web logs preserving the sequential information present in the web logs.

Rest of the paper is organized as follows. Section II describes Rough Sets in brief. Section III discusses the methodology adopted in this paper followed by Experimental Results and Discussion in Section IV. Conclusion has been discussed in Section V.

II. Rough Sets

Rough set theory has been introduced by Pawlak [16], deals with uncertainty and vagueness present in data. Rough set is a soft computing technique theory became popular among scientists around the world due to its application in the field of artificial intelligence and cognitive sciences [16]. The

Manuscript received on May 1, 2012; revised May 23, 2012.

Authors are with Information Technology & Systems Department, Indian Institute of Management Lucknow, IIM Lucknow Uttar Pradesh- India (e-mail: rajhans@ iiml.org; pradeepkumar@iiml.ac.in).

basic concept of rough set theory is that every set of the universe, associate some information in the form of data and knowledge.

Objects are clustered on the basis of similarity of information that is associated with the object. The similarity generated in this way forms the basis for rough set theory.

In rough set theory, a set can be approximated by a pair of crisp sets called lower and upper approximations of a set.

Let $T = (U, A)$ is an information set and let $R \subseteq A$ & $X \subseteq U$. X can be approximated using only the information contained in R by constructing the *R-lower* and *R-upper* approximations of X, denoted $\underline{R}X$ and $\overline{R}X$ respectively, where

$\underline{R}X = \{x \mid [x]_R \subseteq X\}$ (Lower approximation of X).

$\overline{R}X = \{x \mid [x]_R \cap X \neq \phi\}$. (Upper Approxima- tion of X)

where U represents the universe.

A set is said to be rough if its boundary region is non-empty, otherwise the set is crisp. Different region and approximations of rough sets has been shown in Fig: 1.

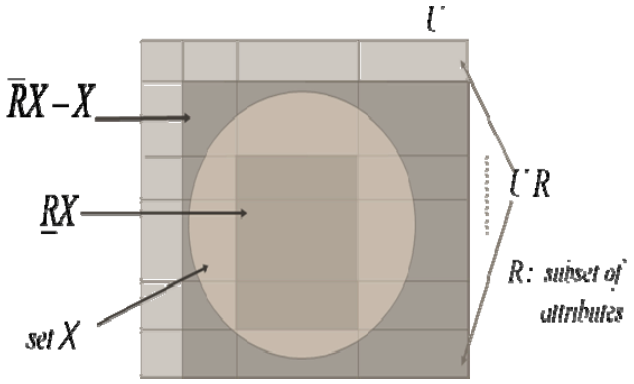

Fig. 1. Lower and Upper Approximation of Set X

A rough set based cluster is defined in a similar to rough set i.e. with a lower and upper approximation. The lower approximation of a rough cluster contains objects that only belong to that cluster. The upper approximation of a rough cluster contains objects in the cluster which are also members of other clusters. Thus making the rough clusters, soft in nature.

III. Methodology

This paper is an extension to the paper, Kumar et al. [11] using Jaccard, Dice, Levenshtein and S^3M similarity measures on *msnbc* data set. Kumar et al. [11] have used rough set upper approximation for generating the clusters. The upper approximation gives the group of sequences on the basis of specified value of similarity measure. In this paper we implemented the algorithm proposed by Kumar et al [11] with different similarity measures like Jaccard, Dice, Levenshtein and S^3M.

The steps in the algorithm [11] are as follows:

1. The similarity value between the sequential data has been calculated using Jaccard, Dice, and Levenshtein or S^3M similarity measures.
2. First upper approximation of sequences has been calculated with a given Similarity Threshold (δ). This step will result in initial clusters.
3. Generated supersets of the clusters formed in the step 2 have been identified and proper subsets are removed. This step results in reduction of number of clusters formed in step 2.
4. Cluster centers of other clusters have been removed from the cluster as clusters formed in step 3 can contain cluster centers of other clusters. Clusters formed are soft in nature till step 4
5. This step deals with constructing hard clusters from the soft clusters found in step 4.The objects, present in more than one clusters have been identified and their similarity values have been compared with respect to each cluster center. The object is assigned to the cluster having highest similarity value between cluster center and object.

The main idea behind adapting various distance/similarity measures in step 1 is to study the impact of various similarity/distance measures while using similarity upper approximation for generating clusters. The Dice and Jaccard measure capture only the content information while Levenshtein measure capture the sequence information.S^3M measure is the combination of Jaccard and Levenshtein thus captures the sequential as well as content information.

IV. Experimental Results & Discussion

Experiments were performed on a PC having an Intel Core 2 Duo Processor (1.83 GHz) with 2 GB RAM. The proposed algorithm [11] has been implemented in JAVA on the Windows XP platform.

We have used msnbc dataset and used sequences of length six for the purpose of the experiment. The dataset is present (http://archive.ics.uci.edu/m1/datasets/MSNBC.com+Anonymous+Web+Data) and has seventeen categories that are "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports". We have converted the categories in to numbers starting from 1 to 17 respectively for the purpose of experiment.

Jaccard, Dice, Levenshtein and S^3M similarity measures have been used with the proposed algorithm [1] for finding the clusters and outliers in the *msnbc* dataset. The combination of number of outliers and number of clusters has been used for validating the results of algorithm with different similarity measures. Too many and too less clusters in the output are extreme solutions and are not desirable. The number of clusters which lies in between the extreme case with less number of outliers is most suitable and appropriate output. The output with too many outliers implies a lot of objects have been left ungrouped and hence not desirable. These outliers can be generated by user while traversing a web site and performing back and forth click.

The algorithm is a hierarchical clustering algorithm which groups the objects in steps to form the clusters. The results are shown in tables Table I-Table II. The experiments have been performed for δ=0.1 on *msnbc* web navigation dataset. The experiments have been performed on different size of

data sets which are of sizes100, 200, 300, 400, 500, 1000, 2000, and 3000 with different similarity measures. In these datasets many user sequences have been repeated. These user sequences have chosen arbitrarily.

The table shows both the number of clusters as well as outliers for different size of data set with various similarity measures. For S^3M, experiment were conducted using different value of "p" ranging from p=0.1 to p=0.9.The "p" value in S^3M measure capture the weight of sequential information. The higher the value of "p" more sequential aspect is considered, similarly less the value of "p" less sequential aspect is considered. In the table we have quoted the result for p=0.5 (i.e. equal weight is given to sequential aspect as well as content aspect during the clustering) and p=0.8 (more weight has been given to sequential aspect than content aspect).Results suggest that S^3M perform better than other similarity measures The number of outliers discovered and number of clusters formed using S^3M is reasonable. The results also suggest that while clustering the web user, both aspects i.e. sequential aspect and content aspect, should be considered.

TABLE I: NUMBER OF OUTLIERS AND CLUSTERS WITH DIFFERENT SIMILARITY MEASURES IN DATASET OF DIFFERENT SIZES

	Data Set-100		Data Set-200		Data Set-300		Data Set-400	
For δ=0.1	*Outliers*	*Clusters*	*Outliers*	*Clusters*	*Outliers*	*Clusters*	*Outliers*	*Clusters*
Jaccard	44	27	0	1	14	37	47	89
Dice	44	27	0	1	14	37	47	89
Levenshtein	0	1	0	1	0	1	0	1
S^3M(p=0.5)	23	31	**2**	**22**	**0**	**8**	55	164
S^3M(p=0.8)	**2**	**10**	115	42	231	39	**2**	**21**

TABLE II: NUMBER OF OUTLIERS AND CLUSTERS WITH DIFFERENT SIMILARITY MEASURES IN DATASET OF DIFFERENT SIZES)

	Data Set-500		Data Set-1000		Data Set-2000		Data Set-3000	
For δ=0.1	*Outliers*	*Clusters*	*Outliers*	*Clusters*	*Outliers*	*Clusters*	*Outliers*	*Clusters*
Jaccard	465	23	126	290	375	713	2845	94
Dice	465	23	126	290	375	713	2845	94
Levenshtein	0	1	0	1	0	1	0	1
S^3M(p=0.5)	249	117	241	228	780	411	**33**	**165**
S^3M(p=0.8)	**48**	**50**	**14**	**80**	**114**	**266**	168	375

Note: For S^3M bold values are considered as they indicate better result.

V. CONCLUSION

The paper has implemented the algorithm proposed by Kumar et al. [11] with four similarity measures that are Jaccard, Dice, Levenshtein and S^3M.The algorithm [11] uses rough set upper approximation to find out the clusters of users and outliers. The results are better with S^3M measure as it captures the sequence and content both. Jaccard and Dice only measure the similarity based on content and Levenshtein measures the similarity based on sequence. S^3M measure the similarity based on content and sequence hence produces better result.

REFERENCES

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

[2] A. Joshi & R. Krishnapuram, "Robust Fuzzy Clustering Methods to Support Web Mining,".*Proceeding of the work shop on Data Mining and Knowledge Discovery, SIGMOD*, 1998, pp. 1-15.

[3] P.Sungjune, N.C. Suresh and B. K. Jeong, "Sequence-based clustering for Web usage mining:A new experimental framework and ANN-enhanced K-means algorithm," *Data & Knowledge Engineering,* vol 65, pp. 512-543, 2008.

[4] H.C. Wang, H.C. Kuo, H.H.Chen, Y.Y. Hsiao and W.C.Tsai, "KSPF: using gene sequence patterns and data mining," *Expert Systems with Applications* , vol 28, pp. 537–545, 2005.

[5] P.Kumar, M.V. Rao, P. Krishna,R.S.Bapi and A. Laha, "Intrusion Detection System Using Sequence and Set Preserving Metric, " *Proceedings of Intelligence and Security Informatics*,Atlanta, 2005, pp. 498-504, .

[6] G. Navarro, "A guided tour to approximate string matching". *ACM Computing Surveys* , vol 33, no.1, pp. 31–88, 2001.

[7] P. J. Lingras, "Unsupervised rough set classi-cation using Gas," *Journal of Intelligent Information Systems* , vol.16, pp. 215-228, 2001.

[8] P. J. Lingras, "Rough Set Clustering for Web Mining,". *Rough Set Clust Proceedings of IEEE International Conference on Fuzzy Systems.* Honolulu, 2002, pp. 1039 – 1044.

[9] S.K. De and P.R. Krishna, "Clustering web transactions using rough approximation," *Fuzzy Sets and Systems* , vol. 148, pp. 131–138, 2008.

[10] P.Kumar, P.R. Krishna, R.S. Bapi and S.K. De, "Rough clustering of sequential data," *Data & Knowledge Engineering* , vol 63, pp. 183–199, 2007.

[11] P. Kumar, P.R. Krishna, R. S. Bapi and S.K. De, "Clustering using Similarity Upper Approximation," *IEEE International Conference on Fuzzy Systems*, Vancouver,2006, pp. 893-844.

[12] P. Berkhin, "Survey of clustering data mining techniques," Technical report, Accrue Software, San Jose, CA, 2002.

[13] A.M. Qamar, E. Gaussier, J.P. Chevallet & J. H. Lim, "Similarity Learning for Nearest Neighbor Classification," *Eighth IEEE International Conference on Data Mining*, Pisa, 2008, pp. 983-988.

[14] T.A. Runkler and J.C. Bezdek, "Automatic Keyword Extraction with Relational Clustering and Levenshtein Distances," *In the proceedings of ninth IEEE International Conference on Fuzzy Systems*, San Antonio 2000, pp. 636-640.

[15] P. Kumar, R.S. Bapi and P.R. Krishna, "SeqPAM: A Sequence Clustering Algorithm for Web Personalization," *International Journal of Data Warehousing & Mining* , vol 3, no. 1, pp. 29-53, 2007.

[16] Z.Pawlak, "Rough sets," *International Journal of Computer and Information Sciences* , vol. 2, pp. 341–346, 1982.