Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 450–457

Procedia Computer Science

The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)

Alignment Based Similarity distance Measure for Better Web Sessions Clustering

Poornalatha G[a,*], Prakash Raghavendra[b]

[a,b]*Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore, 575025, India*

Abstract

The evolution of the internet along with the popularity of the web has attracted a great attention among the researchers to web usage mining. Given that, there is an exponential growth in terms of amount of data available in the web that may not give the required information immediately; web usage mining extracts the useful information from the huge amount of data available in the web logs that contain information regarding web pages accessed. Due to this huge amount of data, it is better to handle small group of data at a time, instead of dealing with entire data together. In order to cluster the data, similarity measure is essential to obtain the distance between any two user sessions. The objective of this paper is to propose a technique, to measure the similarity between any two user sessions based on sequence alignment technique that uses the dynamic programming method.

Keywords: web usage mining; clustering; k-means; dynamic programming

1. Introduction

Given that, there is an exponential growth in terms of amount of data available in all fields, that may not give the required information immediately, how does the end user quickly find what s/he is looking for - a mission in which direction the present day researchers are interested to work. Due to the wide popularity of the web in recent years and also the enormous amount of data available in the access log of the web server, web usage mining has become significant research topic in recent times. From the technical point of view, web usage mining is the application of data mining techniques to usage logs of large data repositories maintained by web servers [1].

A number of clustering approaches have been proposed in the literature. For example, Facca et al. [2]

* Corresponding author.
E-mail address: poornalathag@yahoo.com.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.
doi:10.1016/j.procs.2011.07.058

presented a survey of the developments in the area of web usage mining, where the view points on various techniques like association rules, clustering, sequence patterns etc. are given. Xu et al. [3] focused on various clustering algorithms in general and reviewed variety of approaches appearing in the literature like, hierarchical clustering, squared error based clustering (Vector Quantization), neural network based clustering, graph theory based clustering etc.

Krol et al. [4], investigated on internet system user behavior using cluster analysis. Here sessions are represented as vectors where each dimension represents a web page and stores the value of user interest in each page of a session. The sessions are clustered using Hard C-Means algorithm. Fu et al. [5] proposed a generalization based clustering method which employs the attribute-oriented induction method to reduce the large dimensionality of data. Shi [6] considered the web pages visited by users and time spent at each of the web page to reveal the interests of web users while surfing. These approaches consider sessions as a vector of same length but in general each user' session may not be of equal length.

Various approaches are also discussed in the literature about the different types of distance measures used for clustering web user sessions. Hay et al. [7] illustrated a new method for mining navigation patterns using a sequence alignment method. This method partitions navigation patterns according to the order in which web pages are requested and handles the problem of clustering sequences of different lengths but the distance is measured based on the number of insertion/deletion/reordering operations and also the results are compared with a method based on euclidean distance measure which does not consider the sequence information.

Khasawneh et al. [8] introduced a multidimensional session comparison method using dynamic programming. Though more than one dimension is considered for comparison between pair of sessions, page list is the primary dimension for comparison. Chaofeng et al. [9] introduced a method for measuring the similarities between web pages that takes into account viewing time of the visited web page along with the URL and based on this measured the similarities of web sessions using sequence alignment in computational biology but only 500 web sessions are considered corresponding to six subjects for their experiment. Mojica et al. [10] cluster web pages using a distance based algorithm by modifying gravitational algorithm which is similar to the basic k-means algorithm. Yilmaz et al. [11] used ontology and sequence information for extracting behavior patterns from web navigation logs that tries to merge web usage mining with web content mining. Xu et al. [12] used cosine similarity as a distance measure that requires sessions to be of same length.

Our earlier work [13] proposed an effective modified K-means algorithm to cluster the web sessions based on VLVD function which takes care of the issue of variable length sessions. But, the order of page visits was not considered. For example, the sessions' p1, p2, p3 and p1, p3, p2 were considered as same. Whether the page p3 is visited before p2 or after p2 is not given importance and both sessions were considered as exactly similar based on the pages accessed irrespective of the order in which these pages are accessed. But, for certain applications like web page prediction, caching and pre-fetching, retaining the order information become very essential. Based on the navigation pattern of usage, web page prediction/caching/pre-fetching can be done and hence the present work tries to deal with the order in which web pages are accessed by user.

The goal of this paper is to present an algorithm for finding out the distance between any two web sessions by taking sequence alignment as a similarity measure. The sequence information of navigation order of pages viewed by user is retained and the sessions need not be of same length. Majority of the researchers does not consider the unequal length of sessions effectively and also retaining the information of sequence in which pages are accessed in a session is useful for web page prediction, caching, pre-fetching etc. The web sessions are available from server logs that have the information of users' web page access. Section 2 discusses about the sequence alignment method in general. The section 3 discusses the proposed method; section 4 gives results with discussion, followed by conclusion in section 5.

2. Sequence Alignment Method (SAM)

The sequence alignment method is used for aligning DNA and protein sequences. Each DNA sequence contains a sequence of amino acids. It is possible to determine whether significant homology exists between the proteins by finding similarities in the amino acid sequences of two proteins. Short sequences can be aligned manually but, the problems in general/reality require the alignment of lengthy sequences that cannot be aligned only by human effort. Therefore, computational approaches to sequence alignment are essential that fall into two categories: global alignments and local alignments. One sequence is transformed into the other in global alignment, whereas, in local alignment, all locations of similarity between the two sequences are returned [14].

Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similarity within their larger sequence context. The Smith-Waterman algorithm [15] is a general local alignment method and both global as well as local alignment methods are based on dynamic programming. Global alignments, which attempt to align every residue in every sequence, are most useful when the sequences in the query set are similar and of roughly equal size. The Needleman-Wunsch algorithm [16] is a general global alignment technique.

In general, a session consists of sequence of web pages accessed by user. Therefore, the problem of computing the similarity between web sessions is converted to find the best matching between two web page access sessions which consist of a sequence of web pages. Thus the techniques used in DNA/protein alignment, can be employed to find the alignments between any two user's web access sessions and based on these alignments the distance between pair of web page session may be obtained. Further, clustering of sessions may be formed based on this distance.

3. Proposed Method

The algorithm proposed in this paper is based on the concept of Smith-Waterman local alignment technique. The algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981 for identification of common molecular sub sequences. The similarity measure used in their work allows for arbitrary length deletions and insertions. The proposed method uses the Smith-Waterman local alignment technique for computing the number of alignments between two web sessions and the distance between two sessions is given by the equation given in step 7 of the proposed algorithm given in section 3.1.

3.1. Proposed Algorithm

When any two sessions are considered, individual pair wise comparisons between two sessions are required in order to get the distance between them. Here, two pages are said to be a match, if they are same. The mismatch results, if the pages are different, indicating that either insertion/deletion/substitution operation is required so as to make both sessions exactly similar to each other. It is required to assign suitable scores for match and mismatch. Since, the intension here is to find the number of pair wise alignments between two sessions, the insertion/deletion/substitution operations is considered as same (mismatch). The score is assigned for both match and mismatch suitably say, match=2, mismatch=-1. By assigning score for match and mismatch, the comparison of each pair of pages is weighted into a matrix called score matrix. The distance between two web user sessions is computed based on the number of alignments obtained for pages in two sessions that are compared. The major steps of the proposed algorithm that finds the distance between any two user's web sessions is given below:

Algorithm: Sequence Alignment Based Distance Measure (SABDM)

Input: sessions s1=(pr1,pr2,…,prm), s2=(pc1,pc2,…,pcn), *match*=2, *mismatch*=-1 , *similarity-count*=0
Output: distance d between s1 and s2

Method:

1. Construct score matrix of size (m+1, n+1) and initialize as follows:
   score (i, 0) = mismatch where, 0<i<=m+1
   score (0, i) = mismatch where 0<i<=n+1
   if pi=pj, score (i, j) = match
   if pi! =pj, score (i, j) = mismatch
2. Compute distance matrix Dist and pointer matrix Pointer of size (m+1, n+1) each
   Pointer (0, i) = 0 where, 0<=i<=m+1
   Pointer (i, 0) = 0 where, 0<=i<=n+1
   Dist (0, i) =Dist (0, i-1) +mismatch for 0<i<=m+1
   Dist (i, 0) =Dist (i-1, 0) +mismatch for 0<i<=n+1
3. Dist (i, j) = max (0,
   Dist (i-1, j) +mismatch,
   Dist (i, j-1) +mismatch,
   Dist (i-1, j-1) +score (i, j))
   The value 0 is included to ignore possible negative alignment score. The second and third terms handle an extension of alignment by inserting a gap for either insertion/deletion/substitution operation. Finally the fourth term considers an extension of the alignment by extending the two sequences of sessions compared by one page each. Store the pointer value as either top/left/left-top/combination of top, left and left-top in Pointer (i, j) depending upon Dist (i, j) where, 0<i<=m+1, 0<i<=n+1
4. Trace the distance matrix back, by finding the position of cell with maximum value, check for match or mismatch from score matrix. Use the Pointer matrix to move to the next location. Whenever match is found increment the similarity-count
5. Repeat the tracing process till a cell with value zero is encountered in Dist matrix
6. If more than one cell in Dist matrix contains the same maximum value, repeat the steps 3 to 5.
7. Find the normalized distance between s1 and s2 as given below:
   distance = (max (m, n)-similarity-count)/max (m, n)

3.2. Time and Space Complexity

The time complexity of the initialization is O(m+n) because of initialization of 0th row and 0th column. In filling the distance matrix, each cell of the matrix is traversed, constant number of operations are performed in each cell and therefore, the time complexity for this part is O(mn). Similarly to fill score and pointer matrix time complexity is O(mn). In the trace back, cell with the largest value is found which is done by traversing the entire matrix and thus the time complexity is O(mn) for this part. Thus the total time complexity is given by equation 1.

$$3O(m+n)+3O(mn)+O(mn) = O(mn) \quad (1)$$

Since the algorithm fills three matrices of size (m, n) and stores at most n positions for the trace back, the total space complexity of the algorithm is given by equation 2.

$$3O(mn)+O(n) = O(mn) \quad (2)$$

3.3. *Illustration through an example*

Consider a simple example to find the distance between two sessions' s1 and s2 by applying the

sequence alignment based distance measure (SABDM) algorithm to understand the proposed technique of finding distance between any two web user sessions.

Let s1= (p1, p2, p3, p4, p5) and s2 = (p1, p3, p4, p2, p5) be two sessions to be aligned so as to compute the distance between them. Let m and n be the length of session s1 and s2 respectively. In this example the length of both s1 and s2 are same. Therefore, m =5 and n=5.

The score matrix is constructed as given in table 1 with size (m+1, n+1). The score values considered for match and mismatch are 2 and -1 respectively because, matches should be rewarded and mismatches should be penalized. Initialize the first row and first column of the score matrix with the value -1 as per the requirements to use the concept of the dynamic programming and also fill rest of the cells with the value of either match or mismatch according to the comparison made for pair of pages considered at a time. i.e., enter value of match in the cells of score matrix whenever s1(i) is equal to s2(j) and set the cell value as mismatch if s1(i) is not equal to s2(j) for all i from 1 to m+1 and for all j from 1 to n+1.

Construct a distance matrix with size (m+1, n+1). Compute the distance matrix values as shown in table 2 based on equations given in equation 3.

$$\begin{aligned} &Dist(0, i) = Dist(0, i-1) + mismatch \quad for\ 0 < i <= m+1 \\ &Dist(i, 0) = Dist(i-1, 0) + mismatch \quad for\ 0 < i <= n+1 \\ &Dist(i, j) = max\{0, Dist(i-1, j) + mismatch, Dist(i, j-1) + mismatch, Dist(i-1, j-1) + score(i, j)\} \\ &for\ 0 < i <= m+1,\ 0 < j <= n+1 \end{aligned} \tag{3}$$

Construct a pointer matrix of size (m+1, n+1) to store the positions of cells from which a value is obtained in distance matrix so that, these pointers can be used to trace back the distance matrix at later stage. The table 3 shows the pointer matrix for the example considered. Here value 1 indicates link to top cell, value 2 indicates link to left cell and value 3 indicates link to left-top cell.

Table 1. Score matrix

	j	p1	p3	p4	p2	p5
i	0	-1	-1	-1	-1	-1
p1	-1	2	-1	-1	-1	-1
p2	-1	-1	-1	-1	2	-1
p3	-1	-1	2	-1	-1	-1
p4	-1	-1	-1	2	-1	-1
p5	-1	-1	-1	-1	-1	2

Table 2. Distance matrix

	j	p1	p3	p4	p2	p5
i	0	-1	-2	-3	-4	-5
p1	-1	2	1	0	0	0
p2	-2	1	1	0	2	1
p3	-3	0	3	2	1	1
p4	-4	0	2	5	4	3
P5	-5	0	1	4	4	6

Table 3. Pointer matrix

	j	p1	p3	P4	p2	p5
i	0	0	0	0	0	0
p1	0	3	2	2	0	0
p2	0	1	3	23	3	2
p3	0	1	3	2	12	3
p4	0	0	1	3	2	2
p5	0	0	1	-1	3	3

Trace the distance matrix back, by finding the position of cell with maximum value. In this example the maximum value is 6 present at the cell with position (5, 5) in the distance matrix. Check for match or mismatch by referring the score matrix. Increment the similarity Count value by 1 if match is found. Move to the next location according to the links stored in the pointer matrix and repeat the process till the value 0 is obtained in distance matrix or (0,0) cell is reached or first row/first column of distance matrix is reached. Finally, compute the normalized distance value between s1 and s2 by using equation 4 which gives the distance as 0.2.

$$Distance = (max (m, n) – similarity Count) / max (m, n) \quad (4)$$

If the distance value always lie in the range of 0 and 1. The value 1 indicates that the two sequences are entirely different and the value 0 indicates that the two sequences are exactly similar to each other. As the distance becomes closer to 0, it means that, the sequences are closer and as the sequence becomes different the distance value tends to be closer to 1 than 0. Since the distance value 1 indicates that the two sessions are completely different, such session can be considered as outlier and need not be included to the cluster while performing the clustering process. Thus the proposed SABDM distance measure can also be useful in finding the outliers, if any, when forming the clusters.

4. Results and discussions

The proposed algorithm is implemented in java and the results obtained are compared with the Needleman-Wunsch (NW) global sequence alignment method and the SAM method proposed in [7]. The table 4 illustrates the outcome of the proposed method and also gives the comparison with global alignment and SAM methods.

Compare to NW method the proposed method finds more alignments between sequences because it tries to find regions of similarities within sequences instead of aligning every residue of sequence. Therefore the distance value obtained will be lesser than NW method. For example, if we look at the case of second row of table 4, NW method aligns only page p1. If both sessions' s1 and s2 are observed, it is clear that after visiting page p1, user visits pages p6 and p7 in both sessions. Only difference is that, in s1, pages p2 and p3 are visited before going to p6, p7 whereas in s2, immediately after visiting p1, user accesses p6 and p7. The reason could be there may some correlation between pages p1, p6 and p7. This information is not captured by NW method, but the proposed algorithm takes care of these kinds of issues also. Therefore, the number of alignments found is more in the SABDM method compare to NW method.

Since the SAM method finds the distance between pair of sessions based on the number of insertion, deletion and reordering operations, the distance obtained will be more compare to the proposed SABDM method. Thus the table 4 illustrates the goodness of the proposed technique compare to the NW technique of sequence alignment as well as the SAM method.

From the table 4, it is very clear that, the proposed method yields good measure of distance values compare to others. As the number of alignments found will be more in the proposed SABDM method, the

distance between two sessions will also become less whenever more correlations are present between pair of sessions considered compare to other two methods. Thus by retaining the information of navigation order along with considering the issue of unequal length of sessions, the proposed SABDM method finds the distance between two sessions effectively.

Table 4. Comparison of results

No.	Session s1	Session s2	Distance between session s1 and s2		
			NW method	SAM method	SABDM
1	p1,p2,p3,p4,p5	p1,p3,p4,p2,p5	0.6	2	0.2
2	p1,p2,p3,p6,p7,p8	p1,p6,p7,p4,p5,p9	0.83	6	0.67
3	p1,p2,p3,p4	p1,p5,p4	0.5	5	0.5
4	p1,p3,p4,p2	p1,p2,p3	0.74	3	0.5
5	p1,p2,p5,p6,p3,p1,p2,p5,p4, p8,p7,p2	p1,p2,p7,p7,p5,p4,p1,p2,p6, p5,p8,p7p3	0.53	19	0.38
6	p1,p3,p4,p2	p1,p3,p2	0.25	3	0.25

5. Conclusions

With the explosive growth of the web-based applications, there is significant interest in analyzing the web usage data for the task of understanding the users' web page navigation and apply the outcome knowledge to better serve the needs of user. This paper presents a sequence alignment based distance measure motivated by the basic work of Smith-Waterman local alignment algorithm for identification of common molecular sub sequences. The Smith-Waterman algorithm is suitable for sequences with dissimilar lengths and also tries to find the similarities within regions of a large sequence whereas the Needleman-Wunsch algorithm is preferred for sequence with similar lengths. Since, the web sessions are usually of dissimilar lengths by their inherent nature, the Waterman-Smith algorithm is considered as the basis in the proposed work. The proposed SABDM method finds the distance between user sessions that takes care of the issue of the uneven lengths of sessions and also retains the order of pages visited by the user. Since order of navigation path is considered while clustering, it is further useful for applications like web page prediction, caching and pre-fetching.

As a future work, it is planned to test the impact of this method for clustering user sessions for various number of clusters and also analyze the goodness of the clustering done by using the present work proposed.

References

[1] Nina, S, P., Rahman, M,M., Bhuiyan, M,K,I., & Ahmed,K,E,U. (2009) Pattern discovery of Web Usage Mining. International Conference on Computer Technology and Development.

[2] Facca,F, M., & Lanzi, P, L. (2005) Mining interesting knowledge from web logs : a survey. Journal of Data and Knowledge Engineering 53, Science Direct (pp. 225–41).

[3] Xu, R., & Wunsch,D. (2005) Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3), 645-78.

[4] Krol, D., Scigajlo, M., & Trawinski, B. (2008) Investigation of Internet System User Behavior Using Cluster Analysis. The Seventh International Conference on Machine Learning and Cybermetics, IEEE (3408-12).

[5] Fu, Y., Sandhu, K., & Shih, M. (1999) Clustering of Web Users Based on Access Patterns. KDD workshop on Web Mining, San Diego.

[6] Shi, P. (2009) An Efficient Approach for Clustering Web Access Patterns from Web Logs. International Journal of Advanced Science and Technology, 5.

[7] Hay, B., Wets, G., & Vanhoof, K. (2004) Mining Navigation Patterns Using a Sequence Alignment Method. Journal of Knowledge and Information Systems, Springer-Verlag (150-63).

[8] Khasawneh, N., & Chan, C. (2008) Multidimensional Sessions Comparison Method Using Dynamic Programming. IEEE (581-5).

[9] Chaofeng, L., & Yansheng, L. (2007) Similarity Measurement of Web Sessions Based on Sequence Alignment. Wuhan University Journal of Natural Sciencesm, 12(5), 814-8.

[10] Mojica, J, A., Rojas, D, A., Gomez, J., & Gonzalez, F. (2005) Page Clustering Using Distance Based Algorihm. The Third Latin American Web congress (LA-WEB'05), IEEE.

[11] Yilmaz, H., & Senkul, P. (2010) Using Onyology and Sequence Information for Extracting Behavior Patterns from Web Navigation Logs. IEEE International Conference on Data Mining Workshop 549-56.

[12] Xu, J., & Liu, H. (2010) Web User Clustering Analysis based on KMeans Algorithm. International conference on Information, Networking and Automation (ICINA), IEEE (V26-9).

[13] Poornalatha, G., & Raghavendra, P, S. (2011) Web User Session Clustering Using Modified K-means Algorithm. First Internatinal Conference on Advances in Computing and Communications (ACC – 2011), CCIS(191),Springer-Verlag (243-52).

[14] Brudno, M., Malde, S., Poloakov, A., Do, C, B., Courancne, O., Dubchak, I., & Batzogiou, S. (2003) Glocal alignment: finding rearrangements during alignent. Journal of Bioinformatics (19),.i54-63.

[15] Smith, T, F., & Waterman, M, S., (1981) Identification of Common Molecular Subsequences. Journal of Mol. Biology (147), 195–7.

[16] Needleman,S, B., & Wunsch, c, D. (1970) A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. Journal of Mol. Biology (48), 443-53.