Article can be accessed online at http://www.publishingindia.com

AN EFFICIENT ALGORITHM DISTANCE CALCULATION OF PAGE SEQUENCES USING DYNAMIC PROGRAMMING

Saurabh Dhyani*, Ghanshyam Singh Thakur**

**Department of Computer Application, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India. Email: saurabhdhyani29@gmail.com*
***Department of Computer Application, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India. Email: ghanshyamthakur@gmail.com*

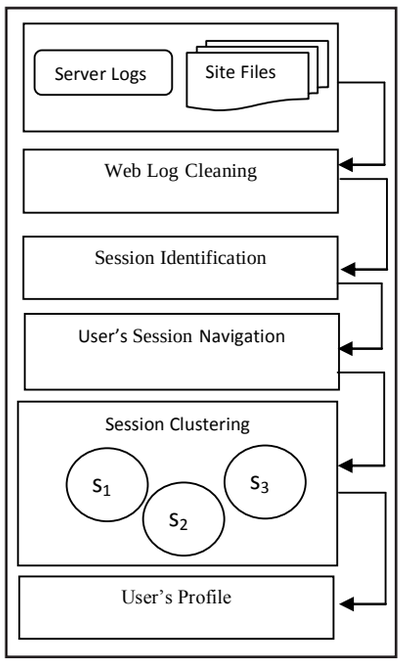
Abstract *Today web data is rapidly growing, but the information residing in the web includes inconsistent information because it is having different types of information, moreover the data are heterogeneous. Due to heterogeneity of data it is a critical task to extract relevant information from the web. Web uses mining technique; extracts the relevant information from huge amount of data available in the web logs format that enclose intrinsic information regarding web pages accessed. Because of this large amount of web log data, it is better to deal with small set of data at a time, instead of handling with complete data. Now we need to find the distance between two user sessions, using some distance similarity function which can accomplish this kind of tasks. Clustering of users tends to establish groups of users exhibiting similar browsing patterns. In this paper we propose an efficient algorithm for calculating the similarity between two user sessions based on sequence alignment that uses one of the dynamic programming techniques that is Longest Common Subsequences*

Keywords: *Clustering, Longest Common Subsequence, Web Logs, Web Usage Mining*

INTRODUCTION

The problem of web session clustering is grouping based on similarity and consists of minimising the inter-group similarity and of maximising the intra-group similarity. The problem of clustering web sessions is a part of major work of web usage mining. Web usage mining is the application of data mining techniques to discover interesting patterns from log data collected by the servers. Data mining from web access logs is a process consisting of four consecutive steps: data collection from various resources, data pre-processing for removing noise in the log entries, pattern discovery which is used to find pattern from web logs, and pattern analysis, to mine frequent data pattern from the all data patterns, which consists lots of algorithm such as clustering, classification and association rule mining on the transformed data in order to mine frequent and useful patterns. Discovery of session cluster is a key part of web data mining. For the log entries that take long time and are also performed by similar types of users, it is likely that the users visited the site more than once. Therefore, it is important to divide the log entries of the same users to sessions (Raju & Satyanarayana, 1996). The term session is defined as a stream of mouse clicks where by a user is trying to perform a specific task (Khasawneh & Chan, 2006).

Fig. 1: Basic Framework for Web Session Clustering



One of the methods for session identification is through timeout. This method is carried out using the assumption

that if the time between two page requests is exceeded than a certain predefined period of time (timeout), then the user is starting a new session at that point. The common used timeout is 30 minutes as a default.

This means that if a time between two page requests from the same user is more than 30 minutes; then two sessions will be identified.

There are some problems in identifying the sessions from log files. Because of proxy servers, the browser's own caching leads to missing parts, when multiple users use the same computer, or when a user uses multiple computers or browsers (Devedzic, 2006).

We can use many types for our application of interest, i.e. the method of clustering, such as grouping users with similar online access behaviour, grouping of web pages with similar usage, or grouping of same types of web sessions to decide different user's behaviours in a given online session (Li, Xue, & Malin, 2012). Most of these groupings are applied with categorical data. But in terms of web log data, it is typical to make cluster. Using session (Timestamp and Page) we can make clusters of users and web pages.

A number of clustering approaches have been proposed in the literature. For example develop a hierarchical clustering algorithm ROCK that employs link and not distance when merging clusters. This method naturally extends to non-numeric similarity measures that are relevant in situations where domain expert table is the only source of knowledge. Xiao, Zhang, & Li (2001) present an approach for measuring similarity of interests among web user from previous access behaviors. They also introduced a multilevel scheme for clustering large number of web users. Zheng, Zha, Yang, Lin, & Zheng (2013) propose a framework to discover different user search goals for a query by clustering the proposed feedback sessions. Feedback sessions are constructed from user click-through logs and can efficiently reflect the information needs of users and also propose a novel approach to generate pseudo-documents to better represent the feedback sessions for clustering. Finally, we propose a new criterion "Classified Average Precision (CAP)" to evaluate the performance of inferring user search goals. Murata & Sai to (2006) describe a method for clarifying users' interests based on an analysis of the site-keyword graph. The method is for extracting sub graphs representing users' main interests a site-keyword graph which is generated from augmented web audience measurement data (Web log data).

Related Work

Spiliopoulou & Faulstich (1998) presented a Web Utilization Miner (WUM) for the discovery of interesting navigation patterns. A specific research topic in Web Usage Mining is clustering of navigation patterns. Shahabi, Zarkesh, Adibi, & Shah (1997) introduced the idea of Path Feature Space to represent all the navigation paths. Path angle is used to measure the similarity between each two paths. It is based on the cosine similarity between two vectors. Clustering is done using k-means method. Fu, Sandhu, & Shih (1999) clustered web sessions using BIRCH algorithm. Their method scaled well over increasing large data set. Poornalatha & Prakash (2011) proposed a technique to measure similarity between any two users session based on sequence alignment technique using the dynamic programming method. Ortega & Aguillo (2010) introduced different characterized distribution of number of hits and time spent by web sessions. It also expects to find if there are significant differences between the length and the duration of a session with regard to the point of access–search engine, link or root. Web usage mining was used to analyse web sessions that were identified from the webometrics.info web site. They show that both distribution of length and duration follow an exponential decay. Significant differences between the different origins of the visits were also found, being the search engines' users those who spent most time and did more clicks in their sessions. We conclude that a good SEO policy would be justified, because search engines are the principal intermediaries to this web site. Jyoti, Sharma, Goel, & Gulati (2009) propose a novel approach for preprocessing wherein rough set clustering is applied to form the clusters of sessions. These sessions could later be used to form the knowledge base of rules on the basis of which the next page to be accessed could be perfected. Wang & Zaiane (2002) propose a new algorithm based on sequence alignment to measure similarities between web sessions where sessions are chronologically ordered sequences of page accesses using dynamic programming.

Our Contribution

Web Sessions Similarity Measurement

First of all the query to be arise in our mind in terms of clustering web sessions is how to determine the similarity between two web sessions. A web session is a stream of clicks on the hyperlinks. Lots of past works applied various kinds of distance similarity functions like Euclidean distance Jacquard or Cosine Coefficient etc. Different kinds of problem may occur.

(i) Euclidean distance has not been appropriate for measuring the distance between categorical data.

(ii) In terms of transfer rate, it may vary.

(iii) The clickstream is a click of sequence which cannot be completely represented by a vector or a set of URLs where the clicks order is not considered (Poornalatha & Prakash (2011).

Here we introduced a new session data with a set of sequences (Clickstream), and applied the methods based on sequence

similarity, for measuring similarity between sessions. Alignment of sequence is actually not a new area. Lot of work has been done for solving sequence alignment problems (Zheng, Zha, Yang, Lin, & Zheng (2013). Our algorithm for measuring similarity between session sequences makes use of the basic ideas from these algorithms. However, a large amount of sequence alignment algorithms for DNA sequencing, it is mostly larger sequences including of a limited terms. In our case, the sequences are relatively smaller (limited click per session) but the terms are large (hundreds of different kinds of pages according to order). The exchange between memory efficiency and computational efficiency in web data sequence alignment is obviously different. There are two parts in our explanation for similarity of session. Primary part is need to define similarity between two pages because each session contained numerous pages and the secondary part is to describe similarity of session using page similarity.

Web Page Similarity Based on URL

In this measurement it simply check path of the web page and its content. Many page similarity tools exist to check whether two given URLs are same or not. Let us take an example for URLs.

URL-1: http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

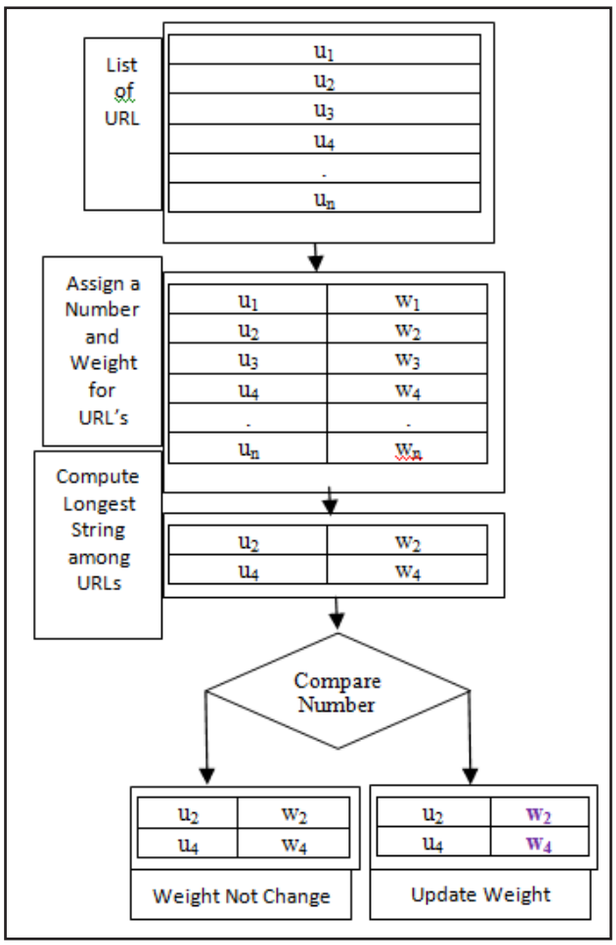
URL-2:http://ita.ee.lbl.gov/html/traces.html.

URL-3: http://archive.ics.uci.edu/ml/datasets.html.

Let us analyse the similarity between the above URLs. Take first two URLs; there is very similarity between them, however on considering first and third or second and third URLs, there is no strong similarity among them (Zheng, Zha, Yang, Lin, & Zheng (2013). Lots of research is going on that topic that is how to measure similarity between web pages (Azimpour-Kivi & Azmi (2011). Assignment of token provided by each URL can be measured for similarity. Compare each token assigned by each URLs. If two tokens are similar then we have to increase the counter accordingly, level by level.

When user visits the website then we can measure the similarity between those pages according to user interest. In this part occurrence of the visit of ith webpage (W_i) and the time spent on this page can be described through the following equation. The Occurrence (W_i) is basically the number of times the webpage (W_i) has been visited in a session. TimeSpent (W_i) is the difference between accurate time of requesting a page (W_i) and requesting a time for next webpage from log data. Now we are discussing the time used on the last page of a session and average time used on another pages of the session.

Fig. 2: Flow Diagram for Similarity Checking



$$Occur(Wi) = \frac{Occurence(Wi)}{\sum Occurence(W)} \quad (1)$$

The length of the webpage is affected by the time which is required to visit those web pages. Equation (2) shows the time which is spent on web page.

$$\text{SpentTime(Wi)} = \text{SpentTime(Wi)} \Big/ \text{Size(Wi)} \quad (2)$$

Similarity between Sessions

Equation (3) shows the interest of visitor. This equation uses the harmonic mean of occurrence and spent time.

$$\text{SpentTime(Wi)} = \frac{2}{1/Occurence(Wi) + 1/TimeSpent(Wi)} \quad (3)$$

For measuring the similarity of the interestingness of two visitors from ith web page and jth pages, (Wi) and (Wj) are the symbols of that page. Equation (4) shows the similarity between two web pages, where (Wi) and (Wj) are two web pages belonging to different sessions.

$$SI(Wi,Wj) = \frac{MAX(Interest(Wi), Interest(Wj)}{MAX(Interest(Wi), Interest(Wj)} \quad (4)$$

Web Page Similarity among All Sessions

It is the fundamental concept of measuring similarity of sessions. First of all take each session as a sequence of web page visits, and use dynamic programming techniques (Longest Common Subsequences) to find the good matching between two sub sequences.

In this procedure, web similarity method already discussed in the earlier part serves as a page matching. The actual similarity measurement between two subsequences is based on their length of the subsequence matching. If two subsequences are given then to find their length we take an example of dynamic programming. Using dynamic programming can solve the common pattern mining problem. Longest Common Subsequence is one of the applications of dynamic programming (Aini, Rashid, Abdullahl, Zawawi, Talibl, & Ali (2006). A subsequence of a given sequence is one in which zero or more elements are left out. For example X=(P1,P2,P3,P2,P4,P1,P2) then S1=(P1), S2=(P1,P2,P2,P2), S3=(P1, P1) are valid subsequences and S4=(P1,P2,P2, P1) is invalid subsequence. Two given subsequences, X and Y can be said to common subsequences if Z is a subsequence of both X and Y, for example, if X=(P1,P2,P3,P2,P4,P1,P2), Y=(P2,P4,P3P1,P2,P1) then CS1=(P1,P1), CS2=(P1, P2 , P1) and CS3=(P2,P3,P2,P1) are common subsequences out of which CS3 is the longest common subsequence. Using this technique we can solve page matching problem.

Sequence Alignment Method (SAM)

The term alignment means relationship between two sequences. The Sequence Alignment Method (SAM) is used to measure non-Euclidean distance which is affected by order sequence of elements in various kinds of research domain. Basically this method is being developed for aligning protein and DNA sequences. The first question arising in our minds is why to compare sequences. We can explain this in various ways. This comparison is done for finding whether two or more proteins are evolutionarily related to each other, or to find structurally or functionally similar regions within proteins (Poornalatha & Prakash, 2011). There are various sequence comparison methods like Dot matrix analysis, Dynamic Programming, Word or k-tuple methods (FASTA and BLAST, n.d.). A sequence is a number of elements arranged one after the other some order. An element in a substring should be continuous whereas, an element in a subsequence embedded in a string can be non-continuous. For example the string "P Q R" is a subsequence, but not a substring of string "T P T Q T R". Basically the similarity between sequences is dependent on the quantity of tasks that has to be done to convert one sequence to another. As a result, the SAM distance measure is represented by a score. The high/ low the score, the more/ less effort it takes to make equal the sequences. Moreover the SAM scores for the following functions during the equalisation process: insert, delete, and record. Insert and delete functions are applied to single or unique elements and the record function is applied to common elements. Common elements are elements, which appear in both compared sequences whereas unique elements appear in one of the two compared sequences (Fu, Sandhu, & Shih, 1999). The scheme of aligning two sequences (Azimpour-Kivi & Azmi, 2011) of maybe different sizes is to write one on top of the other, and break them into small parts by adding spaces in one or the other so that identical subsequences are eventually aligned in a one-to-one correspondence. But spaces are not added in both sequences at the same location. In the end, the sequences end up with the same size (Wang & Zaiane, 2002). A global alignment of two strings A and B is acquired by primary adding to choose spaces, either into or at the last of A and B, and then locating the two resulting strings one above the other so that every character or space in either string is opposite to a unique character or a unique space in the other string. The alignment of two sequences over their full length is referred as global alignment and the detection of local similarities between two sequences is referred as local alignment (Li, Xue, & Malin, 2012). Two strings may not be highly similar in their entirety but may contain regions that are highly similar, this is called as local alignment (Reiners, 2008).

Proposed Method

In this paper we are proposing an algorithm which is based on Needleman/Wunsch techniques. Using Dynamic Programming (Longest Common Subsequence Technique) we are solving the sequence similarity problem. There are several sequence comparisons methods like Dot matrix analysis, Dynamic Programming, Word or k-tuple methods (FASTA and BLAST, n.d.). We are using a dynamic programming to find the sequence comparison.

Proposed Algorithm

Let us take two sessions and we have to find out the distance between them. If two pages are same then it is said to be match. If two pages are different then it is said to be mismatch. Since, the addition here is to discover the number of pair wise alignments between two web sessions, the addition/ deletion/ replacement operations is considered as mismatch. The score is assigned for both match and mismatch suitably say, match=1, mismatch=0.

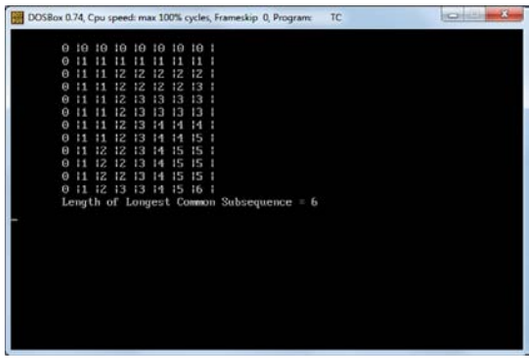
Fig. 3: Code for Longest Common Subsequences between Two Web Sequences

```
int n = strlen(a);
int m = strlen(b);
int l[12][13];
int i,j;
clrscr();

for(i=n;i>=1;i--)
a[i] = a[i-1];
for(i=m;i>=1;i--)
b[i] = b[i-1];

printf("\n\t");
for(i=0;i<=n;i++)
{
for(j=0;j<=m;j++)
{
if(i==0 || j==0)
l[i][j]=0;
else if(a[i] == b[j] )
l[i][j] = l[i-1][j-1] + 1;
else
l[i][j] = max(l[i][j-1],l[i-1][j]);
printf("%d |",l[i][j]);
}
```

Fig. 4: Output for Longest Common Subsequences



By assigning score for match and mismatch, the comparison of each pair of pages is weighted into a matrix called score matrix. The computation of the two user session distance is based on number of alignments getting for pages in two web session that are compared.

Algorithm

Distance Calculation from Sequence Alignment (DCSA)

Terms using DCSA:

Score Matrix (SM)

Sequence (Seq)

Longest Common Subsequences (LCS)

Row (r)

Column (c)

Similarity (S)

Input:

Seq-I and Seq-II

Output:

All Similar Sequences

---

1. Construct SM of size r+1 and c+1 .
2. Set Si,j = 1 if the remains at location i of seq.(I) is the same as the remains at location j of seq.(II) (matching score). Otherwise Si,j = 0 (mismatching score) GP = 0 (gap penalty).
3. Item initialise the first r and first c of matrix to zero.
4. Calculate OSi using the formula.

$$OSi,j = \begin{Bmatrix} \left\{ OSi-1, j + Si, j(\frac{Match}{Mismatch}), \right. \\ |OSi, j-1 + GP(Seq.[1]GAP), \\ OSi-1, j + GP(Seq.[1]GAP) \end{Bmatrix}$$

5. After filling all the values in matrix, the last cell in the matrix shows the overall distance between given two sequences.
6. Use LCS approach for two sessions and implement in c language.
7. Apply trace back technique to find the position of cell with maximum value, for checking the match value or mismatch value from score matrix. Using the pointer find its neighbor cell and compare it.
8. Repeat the process of tracing till a cell with value zero is encountered

Complexity of Our Algorithm

Time Complexity

Time complexity of the initialization of first row and first column is taken as O (r+c). In filling the value, each cell of the matrix is navigated, constant number of tasks are performed in every cell and therefore, the time complexity for this part is O(rc). 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 (rc) for this part. Thus the total time complexity is O (rc).

Space Complexity

The algorithm fills score matrices of size (r, c) and stores at most n positions for the trace back, the space complexity of the algorithm is given O(rc).

Example of Web Sequences

In this example, we are taking two sequences.

Sequence 1- P_1 P_2 P_2 P_3 P_3 P_4 P_2 P_1 P_3 P_3 P_2.

Sequence 2- P_1 P_1 P_2 P_3 P_4 P_1 P_2.

So P = 11 and Q= 7 (the length of sequence #1 and sequence #2, respectively)

where P is the length of sequence 1 and Q is the length of sequence 2. In Longest Common subsequence, to find the common length of subsequence three steps are required.

i. Initialization
ii. Matrix Filling
iii. Back Tracing

First step is to create a matrix with r+1 row and c+ 1 column where r and c is the size of subsequence. Firstly assume that, there is no gap penalty, and the initialization of first row and first column to zero. The second step is the matrix fill step to discover the highest alignment score by starting from left corner in the matrix and find the optimal score $O_{i,j}$ for each position in the matrix.

Fig. 5: Initialization Matrix

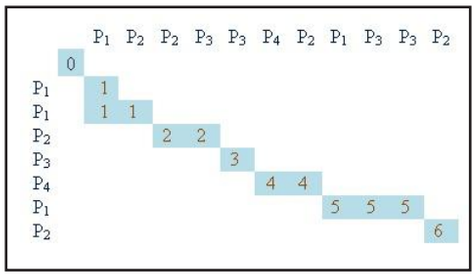
		P_1	P_2	P_2	P_3	P_3	P_4	P_2	P_1	P_3	P_3	P_2
	0	0	0	0	0	0	0	0	0	0	0	0
P_1	0											
P_1	0											
P_2	0											
P_3	0											
P_4	0											
P_1	0											
P_2	0											

Fig. 6: Filling Matrix

		P_1	P_2	P_2	P_3	P_3	P_4	P_2	P_1	P_3	P_3	P_2
	0	0	0	0	0	0	0	0	0	0	0	0
P_1	0	1	1	1	1	1	1	1	1	1	1	1
P_1	0	1	1	1	1	1	1	1	2	2	2	2
P_2	0	1	2	2	2	2	2	2	2	2	2	3
P_3	0	1	2	2	3	3	3	3	3	3	3	3
P_4	0	1	2	2	3	3	3	4	4	4	4	4
P_1	0	1	2	2	3	3	3	4	4	5	5	5
P_2	0	1	2	3	3	3	3	4	5	5	5	6

In order to find $O_{i,j}$ for any i,j it is lowest to know the score for the matrix positions to the left, above and diagonal to i, j. In terms of matrix positions, it is necessary to know Oi-1, j, $O_{i,j-1}$ and Oi-1, j-1. For each position, $O_{i,j}$ is defined to be the highest score at position i,j. See equation (5). Applying this kind of information, the score at position 1, 1 in the matrix can be calculated. Since the first element in both sequences is a G, S1, 1 = 1, and by the assumptions stated at the beginning, w = 0.

Fig. 7: Back Tracing Matrix



Then the value at 1, 1 in score matrixes is 1. Since the gap penalty (GP) is 0, the rest of row 1 and column 1 can be filled in with the value 1 to match the element. Then add 1 and repeat this procedure till the end. The final step is back tracing step. After completing the matrix filling step the longest common subsequence is 6. We have also implemented this process through programming languages.

In this steps the last cell or current cell looking for the adjacent cell and search it to the highest value according to element alignment. After searching an alignment, mark that value and repeat that process till it reaches 0.

Conclusion and Future Work

With the exponential expansion of the web data, there is important attention in analysing the usage data which reside on the web. To understand the user's needs, first of all we have to understand the web data and its hidden pattern. But the web data is heterogeneous and due to heterogeneity of data it is a challenging task to retrieve relevant information from web. Using web usage mining technique, mine the relevant information from large amount of data available in the web logs format that enclose intrinsic information regarding web pages accessed. In this paper we propose an algorithm, for measuring the similarity between two user sessions based on sequence alignment that uses the Longest Common Subsequence (LCS) method and also implemented LCS approach through programming language. Through this approach we can find common sequence of web pages clicked by the user. Due to this we can easily find out most common pages among all pages visited by user.

We would like to extend our research in the following direction: predict the behaviour of users or visitors who visited web sites, retrieve intrinsic information from users, which will be helpful for social networking services.

References

Aini, N., Rashid, A., Abdullahl, R., Zawawi, A., Talibl, H., & Ali, Z. (2006). *Fast Dynamic Programming Based Sequence Alignment Algorithm*. IEEE/ACM Transaction on Computational Biology and Bioinformatics, 3(4), 408-422.

Azimpour-Kivi, M., & Azmi, R. (2011). *A Webpage Similarity Measure for Web Sessions Clustering Using Sequence Alignment.* IEEE International Symposium on Artificial Intelligence and Signal Processing, (pp. 20-24).

Chordia, B. S., & Adhiya, K. P. (2011). Grouping web access sequences using sequence alignment method. *Indian Journal of Computer Science and Engineering*, 2(3), 208-314.

Devedzic, V. (2006). *Semantic web and education*, Springer, 2006. Fasta & Blast (n.d.). Retrieved from http://en.wikipedia.org/wiki/ FASTA

Fu, Y., Sandhu, K., & Shih, M. Y. (1999). *Clustering of web users based on access patterns*. In proceedings of the KDD workshop of Web Mining.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences- Computer Science and Computational Biology*, Cambridge University Press.

Sharma, J., Goel, A. K., & Gulati, P. (2009). *A Novel Approach for clustering web user sessions using RST*. International Conference on Advances in Computing, Control, and Telecommunication Technologies.

Khasawneh, N., & Chan, C. (2006). *Active user-based and ontology-based web log data preprocessing for web usage mining*. EEE/WIC/ACM International Conference on Web Intelligence.

Li, X., Xue, Y., & Malin, B. (2012). *Detecting Anomalous User Behaviors in Workflow-driven Web Applications*. 31st International Symposium on Reliable Distributed Systems.

Murata, T., & Saito, K. (2006). *Extracting Users' Interests from Web Log Data*. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence.

Ortega, J. L., & Aguillo, I. (2010). Differences between web sessions according to the origin of their visits. *Journal of Informetrics*, 4(3), 331-337.

Poornalatha, G., & Prakash, R. (2011). Alignment Based Similarity distance Measure for Better Web Sessions Clustering. *Procedia Computer Science*, 5, 450-457.

Raju, G. T., & Satyanarayana, P. S. (1996). Knowledge discovery from web usage data: Complete preprocessing methodology. *IJCSNS International Journal of Computer Science and Network Security*, June, 5(8), 473-480.

Reiners, P. (2008). Dynamic Programming and Sequence Alignment. Retrieved from www.ibm.com/developerwork.

Shahabi, C., Zarkesh, A., Adibi, J., & Shah, V. (1997). *Knowledge discovery from users web-page navigation*. workshop on Research Issues in Data Engineering, England.

Spiliopoulou, M., & Faulstich, L. C. (1998). *WUM: A Web Utilization Miner. EDBT Workshop WebDB98*. Springer: Valencia, Spain.

Wang, W., & Zaiane, O. R. (2002). *Clustering Web Sessions by Sequence Alignment*. Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA).

Xiao, J., Zhang, Y., Jia, X., & Li, T. (2001). *Measuring Similarity of Interests for Clustering Web-Users*. Proceedings on 12th Australasian Database Conference.

Zheng, L., Zha, H., Yang, X., Lin, W., & Zheng, Z. (2013). *A New Algorithm for Inferring User Search Goals with Feedback Sessions*. IEEE Transactions on Knowledge and Data Engineering.