Clustering Web Sessions by Sequence Alignment

Weinan Wang Osmar R. Zaïane
University of Alberta
Edmonton, Alberta, Canada
{weinan, zaiane}@cs.ualberta.ca

Abstract

In the context of web mining, clustering could be used to cluster similar click-streams to determine learning behaviours in the case of e-learning, or general site access behaviours in e-commerce. Most of the algorithms presented in the literature to deal with clustering web sessions treat sessions as sets of visited pages within a time period and don't consider the sequence of the click-stream visitation. This has a significant consequence when comparing similarities between web sessions. We propose in this paper a new algorithm based on sequence alignment to measure similarities between web sessions where sessions are chronologically ordered sequences of page accesses.

1. Introduction

Clustering web sessions is the problem of grouping web sessions based on similarity and consists of maximizing the intra-group similarity while minimizing the inter-group similarity. The problem of clustering web sessions is part of a larger work of web usage mining which is the application of data mining techniques to discover usage patterns from Web data typically collected by web servers in large logs [10]. Data mining from web access logs is a process consisting of three consecutive steps: data gathering and pre-processing for filtering and formatting the log entries, pattern discovery which consists of the use of a variety of algorithms such as association rule mining, sequential pattern analysis, clustering and classification on the transformed data in order to discover relevant and potentially useful patterns, and finally, pattern analysis during which the user retrieves and interprets the patterns discovered [11].

Session cluster discovery is an important part of web data mining. In the context of e-learning, our application of interest, the function of clustering can have a myriad uses, such as grouping learners with similar on-line access behaviour, grouping pages with similar access or usage, or grouping similar web sessions to determine different learning behaviours in a given on-line course. Most of these groupings are concerned with categorical data. Learners, pages or sessions are indeed represented by vectors, either feature vectors for learners and pages, or sequences in the case of sessions. Unfortunately, most current clustering algorithms cluster numerical data. Very few are particularly suitable for clustering categorical attributes.

In our study, we are interested in clustering sessions in order to identify significant or dominant learning behaviours in online courses. The ulimate goal is to provide educators with a tool to evaluate not only on-line learners, but also evaluate the course material structure and its effective usage by the learners. In order to cluster sessions, after identifying the sessions in a pre-processing phase, we used clustering algorithms known for their ability to handle categorical data: ROCK [4] an algorithm that acts on a sample of the dataset, CHAMELEON [5], which is based on graph partitioning, and a new algorithm TURN for discrete distributions that we introduced in [2]. All of these algorithms, when used in the past for clustering web sessions, have treated sessions as unordered sets of clicks. The similarity measures used to compare sessions were simply based on intersections between these sets, such as the cosine measure or the Jaccard coefficient. This was also the case for our work in [2] where we also applied the Jaccard coefficient which basically measures the degree of common visited pages in both sessions to be compared. While it is the common practice, it is not an adequate measure since the sequence of events is not taken into account. If page A is visited just before page B, it is different from the statement acknowledging that pages A and B were visited in the same session, disregarding the possible pages visited in between.

In this paper, we introduce a new method for measuring similarities between web sessions that takes into account the sequence of event in a click-stream visitation. This measure also considers similarities between pages visited in a session. This method can be used to cluster web sessions using any clustering algorithm that allows the usage of an arbitrary similarity measure as a distance function for grouping similar data objects. Our preliminary experiments show that

the clusters discovered are more meaningful than those discoved when sets of pages model sessions.

The remainder of the paper is organized as follows: Section 2 presents and underlines shortcomings of some clustering algorithms recently proposed for clustering web sessions. Section 3 describes our similarity measures for comparing pages as well as sequences of pages accesses. We discuss some preliminary experiments in Section 4 using different clustering algorithms for categorical data. Finally, Section 5 concludes our study.

2. Related work on clustering web sessions

Most of the studies in the area of web usage mining are very new, and the topic of clustering web sessions has recently become popular in the field of real application of clustering techniques. Shahabi et al. [9] introduced the idea of Path Feature Space to represent all the navigation paths. Similarity between each two paths in the Path Feature Space is measured by the definition of Path Angle which is actually based on the Cosine similarity between two vectors. In this work, k-means cluster method is utilized to cluster user navigation patterns. Fu et al. [3] cluster users based on clustering web sessions. Their work employed attribute oriented induction to transfer the web session data into a space of generalized sessions, then apply the clustering algorithm BIRCH [12] to this generalized session space. Their method scaled well over increasing large data. However, problems of BIRCH include that it needs the setting of a similarity threshold and it is sensitive to the order of data input. The paper does not discuss in detail how they measure the closeness between sessions and how they set the similarity threshold which are very important for clustering. Mobasher et al. [8] used clustering on a web log using the Cosine coefficient and a threshold of 0.5. No detail is mentioned of the actual clustering algorithm used as the paper is principally on Association Rule mining. One recent paper which bears some similarity to our work is by Banerjee and Ghosh [1]. This paper introduced a new method for measuring similarity between web sessions: The longest common sub-sequences between two sessions is first found through dynamic programming, then the similarity between two sessions is defined through their relative time spent on the longest common sub-sequences. Applying this similarity definition, the authors built an abstract similarity graph for the set of sessions to be clustered, then the graph partition method was applied to "cut" the abstract graph into clusters. Our method has a similar basic idea on measuring session similarity, but we consider each session as a sequence and borrow the idea of sequence alignment in bioinformatics to measure similarity between sequences of page accesses. However, we look into more detail of each web page by first defining a similarity between each two pages, then instead of simply finding the longest common subsequence, our method utilizes dynamic programming to find the "*Best Matching*" between two session sequences.

3. Similarity Measures for Web Sessions

The first question needed to be answered in clustering web sessions is how to measure the similarity between two web sessions. A web session is naturally a stream of hyper link clicks. Most of the previous related works apply either Euclidean distance for vector or set similarity measures, Cosine or Jaccard Coefficient. Shortcomings for doing this is obvious: (1) the transferred space could be of very high dimension; (2) The original click stream is naturally a click sequence which cannot be fully represented by a vector or a set of URLs where the order of clicks is not considered; (3) Euclidean distance has been proven in practice not suitable for measuring similarity in categorical vector space.

Here we propose to consider the original session data as a set of sequences, and apply sequence similarity measure to measure similarity between sessions. Sequence alignment actually is not a new topic; there exist several algorithms for solving sequence alignment problems [6]. Our method for measuring similarity between session sequences borrows the basic ideas from these algorithms. However, most sequence alignment algorithms for DNA sequencing consider very long sequences consisting of a limited vocabulary. In our case, the sequences are relatively short (hundreds of clicks at most per session) but the vocabulary is very large (in the order of thousands of different pages). The tradeoff between memory efficiency and computational efficiency in protein sequence alignment is obviously different.

There exist two steps in our definition of session similarity. First we need to define similarity between two web pages because each session includes several web pages; the second step is to define session similarity using page similarity as an inner function.

3.1. Similarity Between Web Pages

If we do not consider the content of pages but simply the paths leading to a web page (or script), we notice that there exist similarities between many different web pages. One example is like the following two URLs:
URL#1: http://www.cs.ualberta.ca/labs/database/current.html
URL#2: http://www.cs.ualberta.ca/labs/database/publications.html
Similarity between these two URLs is obvious: They are very similar pages with a similar "topic" about the research work in the Database group of the University of Alberta. In the second example, the similarity between the two URLs is simply the fact that both pages come from the same server:
URL#1: http://www.cs.ualberta.ca/labs/database/current.html
URL#3: http://www.cs.ualberta.ca/theses/

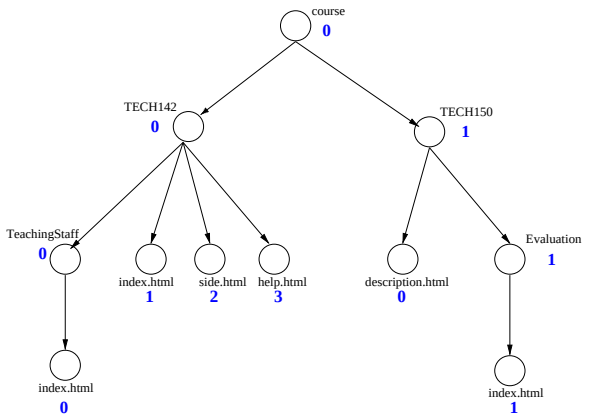

Figure 1. Labeling a web site tree structure

We feel that there is some similarity between URL#1 and URL#3, but the similarity is of course not as strong as the similarity between URL#1 and URL#2 in the previous example. We need a systematic method to give a numerical measure for the similarity between two URLs.

In order to measure the similarity between two web pages, we first represent each level of a URL by a token; the token string of the full path of a URL is thus the concatenation of all the representative tokens of each level. This process corresponds to marking the tree structure of a web site as shown in Figure 1.

The web page "/course/TECH142/index.html" in Figure 1, is represented by the token string "**001** ", the webpage "/course/TECH150/description.html" is represented by the token string "**010**". The computation of web page similarity is based on comparing the token string of web pages. Our web similarity computation works in two steps:

Step1: We compare each corresponding token of the two token strings one by one from the beginning, and this process stops at the first pair of tokens which are different.

Step2: We compute the similarity of two web pages by first determining the length of the longest token string among the two. We then give a weight to each level of tokens: the last level of the longest token string is given weight 1, the second to the last is given weight 2, etc. Next, the similarity between two token strings is defined as the sum of the weight of those matching tokens divided by the sum of the total weights. If the two pages are totally different, i.e. no same corresponding token, their similarity is 0.0. If the two pages are exactly the same, their similarity would be 1.0.

3.2. Similarity Between Sessions

Our basic idea of measuring session similarity is to consider each session as a sequence of web page visits, and use dynamic programming techniques to find the best matching between two sequences. In this process, web similarity technique discussed in the previous section serves as a page matching goodness function. The final similarity between the two sequences is based on their matching goodness and the length of the sequences. One difference between our similarity measure and many of the previous works is: we consider a session as a sequence, while many of previous results measure session similarity in either Euclidean space or sets, for example Jaccard Coefficient, such as: $sim(T_1, T_2) = \frac{T_1 \cap T_2}{T_1 \cup T_2}$, is widely used. We argue that a URL sequence can better represent the nature of a session than a set. For example, using Jaccard Coefficient similarity measure there is no difference between the session "**abcd**", "**bcad**" and "**abdc**". Using our session sequence similarity measure, we can see that the three are different, and "**abcd**' is more similar to "**abdc**" than to "**bcad**".

We use a scoring system which helps find the optimal matching between two session sequences. An optimal matching is an alignment with the highest score. The score for the optimal matching is then used to calculate the similarity between two sessions. These are the principles in matching the sequences:

- The session sequences can be shifted right or left to align as many pages as possible. For example, session#1 includes a sequence of URLs **1, 2, 21, 22**, here each web page is represented by its token string as described previously. Suppose session#2 includes a sequence of visiting to URLs **2, 21, 22**. The best matching between the two session sequences can be achieved by shifting session#2:

  $session\#1 : 1\ \ 2\ 21\ 22$
  $session\#2 : -\ 2\ 21\ 22$

  In our program, each identical matching, i.e. a pair of pages with similarity 1.0, is given a positive score 20; Each mis-matching, i.e. a pair of pages with similarity 0.0 or match a page with a gap, is given a penalty score -10. For a pair of pages with similarity α, where $0.0 \leq \alpha \leq 1.0$, the score for their matching is between -10 and 20.

- Gaps are allowed to be inserted into the middle, beginning or end of session sequences. This is helpful for achieving better matching. For example, for the following two sessions, a gap in session#2 helps getting the best matching.

  $session\#1 : 1\ 2\ 21\ \ 22$
  $session\#2 : 1\ 2\ -\ \ 22$

- We do not simply count the number of identical web pages when we are aligning session sequences. Instead, we create a *scoring function* based on web page similarity measure. For each pair of web pages, the scoring function gives a similarity score where higher

score indicates higher similarity between web pages. A pair of identical web pages is only a special case of matching – the *scoring function* return 1.0 which means the two pages are exactly the same.

The problem of finding the optimal matching can typically be solved using dynamic programming [6], and its process can be described by using a matrix as shown in Figure 2. One sequence is placed along the top of the matrix and the other sequence is placed along the left side. There is a gap added to the start of each sequence which indicates the starting point of matching. The process of finding the optimal matching between two sequences is actually finding a optimal path from the top left corner to the bottom right corner of the matrix. Any step in any path can only go right, down or diagonal. Every diagonal move corresponds to matching two web pages. A right move corresponds to the insertion of a gap in the vertical sequence and matches a web page in the horizontal sequence with a gap in the vertical sequence. A down move corresponds to the insertion of a gap in the horizontal sequence and matches a web page in the vertical sequence with a gap in the horizontal sequence.

In solving the optimal matching problem, the dynamic programming algorithm propagates scores from the matching start point (upper-left corner), to the destination point (lower-right corner) of the matrix. The optimal path is then achieved through back propagating from destination point to starting point. In the given example, the optimal path found through back propagating is connected by arrows where the numbers in brackets indicate the step number in back propagating. This optimal path tells the best matching pattern. The score of any element in the matrix is the maximum of the three scores that can be propagated from the element on its left, the element above it and the element above-left. The score that ends up in the lower-right corner is the optimal sequence alignment score [6]. After finding the final score for the optimal session alignment, the final similarity between the two sessions is computed by considering the final optimal score and the length of the two sessions.

We argue that our similarity measure is better than previous set similarity measures, for example Jaccard Coefficient. This is due to two reasons: (1) considering session as sequence of URLs is better than considering session as a set of URLs. As mentioned before, Jaccard Coefficient cannot differentiate session "**abcd**" from "**bcad**" and "**abdc**", here each token "**a**, **b**, **c** and **d** represents a URL. Our method can not only tell the difference, but also precisely measure the cross similarity between each two of them. (2) In measuring the similarity between sessions, our method considers URL similarity. For instance if two sessions have no common URLs, but they actually have similar paths, since web page similarity in our method uses tokens to represent paths, these sessions would still bear some similarity. This result

	–	1	123	126	1	2
–	0(7)	−10	−20	−30	−40	−50
1	−10	20(6)	10	0	−10	−20
12	−20	10(5)	35	25	15	5
123	−30	0	30(4)	50	40	30
124	−40	−10	20	45(3)	55	45
12	−50	−20	10	35	55(2)	45
22	−60	−30	0	25	45	65(1)

Figure 2. Session matching example

better reflects the true connection between the two sessions.

4. Web Sessions Clustering

The session similarity method described in the previous section can be applied to compute the similarity between each pair of sessions, and construct a similarity matrix. Proper clustering algorithms are applied to this similarity matrix to find the session clusters.

An important issue is how to evaluate the quality of clusters in the result. Clustering Validation is a field where attempts have been made to find rules for quantifying the quality of a clustering result [7]. This issue, however, is a difficult one and typically people evaluate clustering results visually or compare to known manually clustered data. Visually inspecting clusters for categorical data such as web session data is hard and hasn't been done. We devised a method to visualise similarity within clusters and dissimilarity between clusters. For this we ordered the resulting clusters according to their descending sizes on two axies of a 3 dimensional graph. Sessions within clusters are also ordered with the same ordering on both axies. The third axis simply represents the level of similarity between sessions. Figure 3 shows an idealistic example with 1000 sessions in 3 clusters. In this idealistic case where the cross similarity between each pair of sessions within a same cluster is 1.0, and cross similarity between each pair of sessions from two different clusters is 0.0, we can see that only the diagonal has some values on the similarity dimension. The presence of the diagonal indicates good clustering while high similarity values outside the diagonal would indicate inadequate clustering.

Our testing session set used in our experiments has 1000 randomly selected sessions from a real e-learning sys-

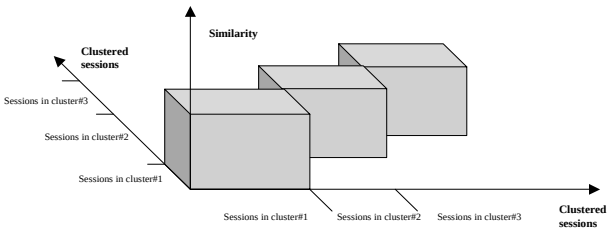

Figure 3. Session clustering visualization

tem web log. Both Jaccard similarity and our Dynamic-Programming-Based similarity methods were used to provide similarity matrices for the given session set. ROCK[4], CHAMELEON[5] and TURN[2] were then applied on the similarity matrices to each produce clustering result. From the clustering results, we found that ROCK tends to find bigger clusters with lower average similarity. CHAMELEON and TURN can find clusters with high internal cross similarity. The difference between the two is that TURN can identify outliers while CHAMELEON cannot. Rare sessions dissimilar to most other sessions are identified by TURN, while CHAMELEON forces them to belong to a given cluster. Using the Jaccard Coefficient as a similarity measure for sessions tends to give more clusters than our Dynamic-Programming-Based similarity measure. In general when evaluated manually, the cluster quality between clusters using the Dynamic-Programming-Based similarity measure was better than when using the Jaccard Coefficient similarity measure. The clusters were simply more meaningful, which is an expected result since we took in consideration the sequence of clicks in a session. However, we do not currently have the means to compute quantitatively this cluster quality, and it would be very difficult to manually evaluate and compare the quality of the clusters resulting from the different similarity measures when the dataset is very large. Nevertheless, our method scales well with the size of the dataset to cluster, and we are confident, given our preliminary tests with the 1000 session set, that the web session clustering with sequence alignment would always yield more significant results than the commonly used approximation of sessions with sets.

5. Conclusions

Session clustering is an important task in web mining in order to group similar sessions and identify trends of web user access behaviour. This is useful not only in e-commerce for user profiling, but also in e-learning for online learner evaluation. Accurate clustering of web sessions depends on good similarity measures between sessions. In this paper we introduce a new similarity measure based on sequence alignment using dynamic-programming. This measure also considers the notion of similarity between pages. In our experiments, we compared the clustering characteristics of three algorithms on the session similarity measures: Jaccard Coefficient and Dynamic Programming Based measure. Among the three algorithms, we determined that TURN was the winner based on our 3D graph for the visualisation of cluster "goodness". Our sequence alignment approach produced more meaningful clusters than the commonly used Jaccard coefficient. However, we do not have a quantitative measure to ascertain the righteousness of sequence alignment in session clustering with certitude. This can be achieved by testing the clustering on labelled data, where the exact cluster to wich a session should belong is known a-priori and hidden from the algorithm. Precise measure of the quality of clustering can be computed by comparing the results with the known cluster labels.

References

[1] A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Proc. of Workshop on Web Mining in First International SIAM Conference on Data Mining*, pages 33–40, Chicago, April 2001.

[2] A. Foss, W. Wang, and O. R. Zaïane. A non-parametric approach to web log analysis. In *Proc. of Workshop on Web Mining in First International SIAM Conference on Data Mining*, pages 41–50, Chicago, April 2001.

[3] Y. Fu, K. Sandhu, and M.-Y. Shih. Clustering of web users based on access patterns. *WEBKDD* workshop, 1999.

[4] S. Guha, R. Rastogi, and K. Shim. ROCK: a robust clustering algorithm for categorical attributes. In *ICDE*, 1999.

[5] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68–75, August 1999.

[6] K.Charter, J.Schaeffer, and D.Szafron. Sequence alignment using fastlsa. In *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS'2000)*, pages 239–245, 2000.

[7] M. V. M. Halkidi, Y. Batistakis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, December 2001.

[8] B. Mobasher, R. Cooly, and J. Srivastava. Automatic personalization based on web usage mining. TR99-010, Department of Computer Science, Depaul University, 1999.

[9] C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. In *workshop on Research Issues in Data Engineering*, England, 1997.

[10] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations*, Jan 2000.

[11] O. R. Zaïane and J. Luo. Towards evaluating learners' behaviour in a web-based distance learning environment. In *Proc. of IEEE Intl. Conf. on Advanced Learning Technologies*, pages 357–360, Madison, WI, August 2001.

[12] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD*, pages 103–114, June 1996.