A review of alignment-based similarity measures for web usage mining

Vinh-Trung Luu · Germain Forestier · Jonathan Weber · Paul Bourgeois · Fahima Djelil · Pierre-Alain Muller

Abstract In order to understand web-based application user behavior, web usage mining applies unsupervised learning techniques to discover hidden patterns from web data that captures user browsing on web sites. For this purpose, web session clustering has been among the most popular approaches to group users with similar browsing patterns that reflect their common interest. An adequate web session clustering implementation significantly depends on the measure that is used to evaluate the similarity of sessions. An efficient approach to evaluate session similarity is sequence alignment, which is known as the task of determining the similarity of elements between sequences. In this paper, we review and compare sequence alignment-based measures for web sessions, and also discuss sequence similarity measures that are not alignment-based. This review also provides a perspective of sequence similarity measures that manipulate web sessions in usage clustering process.

Keywords web mining · sequence alignment · clustering · sequence similarity

1 Introduction

In recent years, web usage mining has received a growing interest motivated by the unprecedented increase of web traffic (El Azab et al., 2017; Dhandi and Chakrawarti, 2016; Liu et al., 2017). The goal of web usage mining is to identify patterns from browsing data, which are seen as user interaction with web sites that reflect user interests (Raphaeli et al., 2017; Wang et al., 2016; Neelima and Rodda, 2016) in order to adapt web sites to user interests (Gauch et al., 2007), create recommender systems (Lopes and Roy, 2015; Luu et al., 2016a), personalize information (Malik and Fyfe, 2012; Wagh and Patil, 2017), etc. Web session clustering is particularly useful as the obtained clusters can later be targeted by specific actions for business purposes. In this context, a variety of relevant techniques has been proposed to cluster web sessions.

An efficient session clustering does not only rely on the selected algorithm but often on the choice of a similarity measure. Therefore, one of the challenges to create meaningful clusters is the definition of similarity measures which are used to compare web session sequences. In order to be consistent in unsupervised learning context like clustering, the measure has to provide a gradual evaluation of how similar two web sessions are. Consequently, a strong interest has been given to

IRIMAS, Université de Haute-Alsace
12 rue des frères Lumière
68093 Mulhouse FRANCE
Tel.: +33(0)3 89 33 69 63
E-mail: firstname.lastname@uha.fr

sequence alignment techniques (Li and Homer, 2010; Rosenberg, 2009) which allow to examine the similarity of session sequences.

Alignment is a well-known approach which writes one sequence on top of the other. It inserts necessary spaces to equalize sequences' lengths and then vertically aligns one-to-one opposite elements. This technique of sequence comparison has been widely used in the bioinformatics domain. For web usage mining, several alignment-based techniques selected from bioinformatics have been used to evaluate web session similarity. For similarity measure, the distance score which reflects the dissimilarity between sequences should be minimized to appropriately align sequences. In other words, sequences should be aligned to maximize their identical elements. On the contrary, dissimilarity measures seek the highest score which is obtained by mismatches or gaps to decide which alignment is the best to compare the sequences. Subsequently, sessions are clustered according to their high intra-group similarity and separated by their low inter-group similarity (Chaofeng, 2009; Anandhi and Ahmed, 2017).

Despite the long investigation of web session alignment implementation, recent researches have indicated considerable progress in scalability, correctness and robustness to noise when aligning sessions. In this paper, we review alignment-based measures that are applicable or have been proposed for web session similarity evaluation, including their advantages and disadvantages. We propose five categories of these measures: (1) Considering sequences with equal lengths, (2) Considering sequences with different lengths, ignoring their element order, (3) Considering sequences with different lengths and their element order, ignoring element succession, (4) Considering sequences with different lengths, their element order and local matching, and (5) Considering sequences with different lengths, their element order, local and global matching. Each category is composed of a summary table and corresponding description which provides more details on each measure. After describing measure categorization, we also present other approaches which are not alignment-based for comparison purposes. We continue to discuss challenges (*i.e.* in reference to the limitation of sequence alignment application) which the measures still face for session similarity evaluation.

The rest of the paper is organized as follows: in Section 2, we introduce the concept of sequence alignment, scoring and its corresponding formalization. The alignment-based methods are specifically categorized into five main groups by their features in detail in Section 3. Section 4 presents other approaches that are not alignment-based but substantial in web session similarity evaluation. Following, we discuss about how distinctive web sessions and biological sequences are when applying similarity evaluation, computational complexity of particular measures, and clustering hierarchical strategy and external validation in Section 5. Finally, Section 6 concludes the paper and reveals our upcoming research.

2 Sequence, alignment and score

In this section, we introduce the notations that will be used afterward to present and compare sequence alignment approaches.

2.1 Sequence

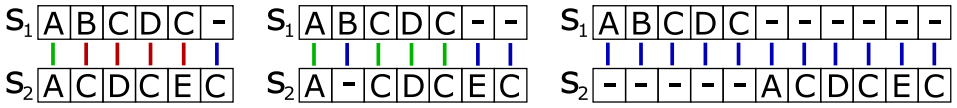
Let Σ be a finite alphabet set that consists of symbols or characters. Let Σ^+ be the set of all strings over the alphabet Σ. Any possible string $s \in \Sigma^+$ formed by characters drawn from Σ is defined as $s[1 \dots n] = s[1], s[2] \dots s[n]$ where $s[i] \in \Sigma$, $1 \leq i \leq n$, and $n = |s|$ denotes the length of s. Given $\Sigma = \{\texttt{A,B,C,D,E}\}$, a set of k strings $S = \{s_1, s_2, \dots, s_k\}$ possibly contains $s_1 = \texttt{AB}, s_2 = \texttt{ABCD}, s_3 = \texttt{BDCA}, \dots, s_k = \texttt{ACDEB}$.

Web Access Sequences (WAS) (Yang et al., 2017; Ting et al., 2009) are ordered sequences of pages accessed in a session. They can be extracted either from log files or recorded from low-level events, in an online or offline manner. If we consider each page-visit to be an alphabetical character, then each WAS can be treated as a string in the example above, representing the ordered web pages accessed by the users. Accordingly, a sequence such as $s_i =$ `ACDEB` shows that a user entered the website by page `A`, then accessed `C`, `D`, `E`, and ended up with page `B`. Consequently, S is a set of all WASs which appeared in a website.

A *subsequence* is a common sequence-related notion, s_i is a subsequence of s_j if ones can find s_i by removing none or some characters from s_j. For instance, $s_i =$ `ADB` is a subsequence of $s_j =$ `ACDEB` (and s_j is called the *supersequence* of s_i).

2.2 Alignment

The sequence alignment over a set $S = \{s_1, s_2, \ldots, s_k\}$ is another set $S_a = \{s_1^a, s_2^a, \ldots, s_k^a\}$ of identical length sequences where each s_i^a can be built from s_i through gap "-" insertions, $1 \leq i \leq k$. S_a is called an *aligned* sequence set (Della Vedova, 2000). Given a set of two or more sequences $S = \{s_1, s_2, \ldots, s_k\}$, an alignment of such set can be defined as a matrix $\mathbf{A}$ of k-rows where the i-th row is sequence s_i^a, each cell holds an element of Σ or "-"and there is no column of the matrix which holds only gaps. Under an alignment $\mathbf{A}$ of two sequences s_1 and s_2, given n the length of aligned sequences, positions of elements $s_1^a[i]$ and $s_2^a[i]$ are vertically opposite, $1 \leq i \leq n$. Figure 1 illustrates three possible alignments of two sequences.



(a) First alignment (b) Second alignment (c) Third alignment

Fig. 1: Three among possible alignments of two sequences (green for matches, red for substitutions and blue for indels).

If $s_1^a[i]$ is identical to $s_2^a[i]$, a *match* appears. Otherwise, if $s_1^a[i]$ and $s_2^a[i]$ are different and neither of them is a gap, it is a *mismatch*. Alternatively, a *gap* occurs. A *mismatch* can be considered as a substitution of $s_1^a[i]$ with $s_2^a[i]$, or vice versa. Differently, if $s_1^a[i]$ is "-"and $s_2^a[i]$ is not, a *gap* is an insertion of $s_2^a[i]$ at the position of $s_1^a[i]$ or a deletion of $s_2^a[i]$. These edit operations, called *indels*, are applied to two sequences in order to convert one to another. A pairwise alignment is an alignment involving two sequences. When three or more sequences are involved, the alignment is then called multiple pairwise alignment (Madeira et al., 2019; Omega et al., 2015).

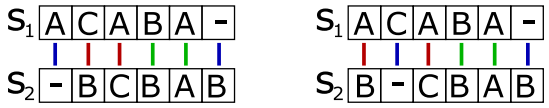
2.3 Score

A distance between sequences is defined as a reflection of the amount of work required to make them identical, as mentioned previously. In general, this distance can be represented by a *cost*. The cost for each substitution or indel can be defined as a function d: $(\mid \Sigma \cup \{-\} \mid) \times (\mid \Sigma \cup \{-\} \mid) \rightarrow \mathbb{R}$. Given $a, b \in \Sigma \cup \{-\}$,

$$d(\mathtt{a},\mathtt{a}) = 0 \tag{1}$$
$$d(\mathtt{a},\mathtt{b}) = d(\mathtt{b},\mathtt{a}) \tag{2}$$

Depending on the defined measure, cost of a substitution or indel may be various. The cost has an impact on the *best alignment* selection since the final alignment cost is the sum of all *pair-cost* through sequence pairs (Pramanik and Setua, 2017; Bose and van der Aalst, 2012). For example, there are 1 match, 1 indel and 4 substitutions in the first alignment in Figure 1, the second one has 4 matches and 3 indels, the last one has 11 indels. If costs of 0 for matches, 1 for indels and 2 for substitutions are applied, the final score of the first, second and last alignments will be $(1 \times 0) + (1 \times 1) + (4 \times 2) = \mathbf{9}, (4 \times 0) + (3 \times 1) = \mathbf{3}$ and $11 \times 1 = \mathbf{11}$, respectively. However, if $0, 2$ and 1 are correspondingly assigned to matches', indels' and substitutions' costs, both the first and second alignments in Figure 1 will have a cost of **6**, and the last alignment will have a score of **22**. Alternatively, in case both substitutions and indels have a cost of 1 and matches have a cost of 0, the maximum cost of **11** results from the third alignment, while for the first and second one it will be **5** and **3**. Consequently, if the best alignment is evaluated based on cost, the choice of the cost metric can have a significant impact on the outcome. Subsequently, the cost metric can influence clustering result, for example two compared sequences may be in the same cluster using the last cost metric and in different clusters by applying the first one.

It is also possible to set different weights for edit operators, namely the costs of match, indel or substitution, for specific pair of symbols so that they are not all alike through the whole alignment. For example, indels between a and "-" can be 1, but it is 2 between b and "-" , and 3 between c and "-" . Besides, substitution cost of a and b, and b and c is 2, which is different from 1 of a and c. Additionally, match score of a may be different from match score of b, for example 1 and 0, respectively. Applying this weighted cost metric to the two sequence pairs in Figure 2 will correspondingly lead to the total alignment costs of $1+2+1+0+1+2 = \mathbf{7}$ and $2+3+1+0+1+2 = \mathbf{9}$. Likewise, these distinct outcomes, due to weighted cost metric, have a significant impact on sequence similarity evaluation, and consequently on clustering result.



(a) First alignment (b) Second alignment

Fig. 2: Two sequence alignments with the same number of matches, substitutions and indels.

In the following section, we address some specific features that a similarity measure should have to figure out hidden groups of users having similar browsing patterns. These features are the abilities to work with variable length sequences, to take element order into consideration, and to count not only global but local matches of sequences (Daily, 2016).

3 Approaches

Among various comparison techniques, many previous studies have focused on sequence alignment algorithms to evaluate the similarity of sessions. Accordingly, pairwise alignment is commonly used

to compare this kind of sequences. Defining a similarity measure for web sessions is very important as it is generally the fundamental of the process of clustering.

As introduced previously, we present five categories of measures in this section as following: each category is composed of a summary table and corresponding description which provides more details on their characteristics. The summary table includes columns such as "Measure feature" for category name, corresponding measures of the category are shown in "Measure name" by their names, and "Measure description" briefly show their characteristics.

3.1 Considering sequences with equal lengths

Table 1: Measures that consider sequences with equal lengths.

Measure feature	**Measure name**	**Measure description**
Considering sequences with equal lengths	Euclidean	Calculates pairwise distance between sequences based on square root of coordinates, and produces a symmetric distance matrix correspondingly.
	Manhattan	Known to be similar to Euclidean distance, this measure figures out dissimilarity between arrays of numerical values like coordinates rather than between sequences of symbols.
	Hamming	Seeks the number of indexes of sequences where symbols are not vertically one-to-one identical. The measure disregards insertions and deletions that make sequences identical, for example, "K***a******r******o******l***in" and "K***e******r******s******t***in" have a distance of 3.
	Cosine	Page visits in session are numbered by unique values and Cosine distance is utilized to compute angular distance between two sessions as they are vectors.

The length of the shortest path connecting two points is commonly referred to when distance is mentioned. Accordingly, in this category, there are distances such as Euclidean, given by the Pythagorean theorem, computes square root of point's coordinates. Alternatively, the Manhattan distance is based on the sum of absolute coordinate differences of high dimensional vectors. Figure 3 shows these two distances. However, the distance between a pair of objects could possibly be determined by other information as their properties, besides their numeric location or coordinates in space. Additionally, Mandal and Azad (2014) presented the calculation of distance between two sessions using Cosine measure. The cosine distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as follows :

$$Cosine(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|} = 1 - \frac{\sum_{i=1}^{n} \mathbf{a}_i \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{a}_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{b}_i^2}} \tag{3}$$

Where $\mathbf{a}_i$ and $\mathbf{b}_i$ are elements of $\mathbf{a}$ and $\mathbf{b}$, respectively. However, this requires sessions of the exact same length, which is more suitable to vectors than web sessions. Furthermore, this approach also

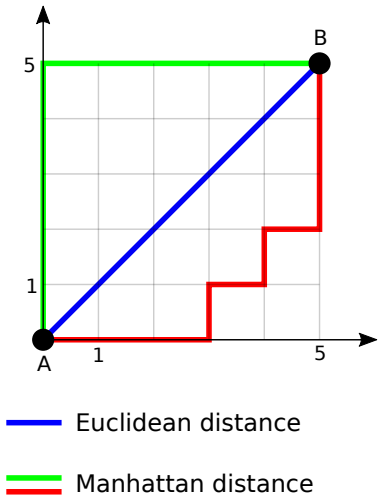
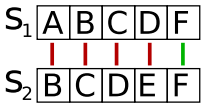


Fig. 3: Example of Euclidean and Manhattan distances between two points A and B. The Euclidean distance is $\sqrt{5^2 + 5^2} = 7.07$ tile's length while Manhattan distance is $5 + 5 = 10$ tile's length.



$$Hamming(\texttt{ABCDF}, \texttt{BCDEF}) = 4$$

Fig. 4: Hamming distance is computed by counting the number of dissimilar symbol pairs when aligning two equal length sequences.

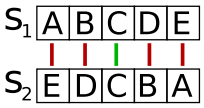
ignores regions of local similarity of session pages. Hamming or Association (Breitinger and Baier, 2012) denotes the dissimilarity by using XOR on corresponding elements between two strings, as illustrated in Figure 4. In other words, this method counts the number of substitutions needed to transform one string to another, both having the same length. These measures works on distance between pairs of objects as Euclidean, Manhattan or Cosine, thus sequences are also required to be of equal length like vectors, and only indels allowed to make the sequences identical, these sequences are considered to be binary vectors. Figure 4 is used as a representation of mentioned alignment methods in this category. These distances, as introduced in Table 1, are obviously inappropriate to session similarity measurement since the sequences have variable length.

3.2 Considering sequences with different lengths, ignoring their element order

Statistical distances, as described by Deza and Deza (2013), are also used to find similarity between sequences. Jaccard, as one of them, returns the value of intersection size divided by union size as the similarity coefficient of two sequences of symbols. In other words, the correspondence and diversity are compared to obtain the distance between two symbol sets. A simple example describing how Jaccard works is presented in Figure 5. For example, Vorontsov et al. (2013) applied Jaccard as a metric of positional weight matrices similarity. Poornalatha and Raghavendra (2011b) also proposed VLVD (Variable Length Vector Distance) that handles web sessions regardless of the length difference. However, both approaches quantitatively define the correspondence between sequences through simple count of common elements that are contained in sequences. They may be suitable

Table 2: Measures that consider sequences with different lengths, ignoring their element order.

Measure feature	Measure name	Measure description
Considering sequences with different lengths, ignoring their element order	Jaccard	Jaccard distance measures the overlap degree among two sets of objects. $Jaccard(S_1, S_2) = \frac{(S_1 \cap S_2)}{(S_1 \cup S_2)}$, given S_1 and S_2 are two sequences.
	VLVD	Stands for Variable Length Vector Distance. $VLVD(S_1, S_2) = \frac{l_1+l_2-2(S_1 \cap S_2)}{l_1+l_2}$, given S_1 and S_2 are two sequences of length l_1 and l_2, respectively.



$$Jaccard(\texttt{ABCDE}, \texttt{EDCBA}) = \frac{(\texttt{ABCDE} \cap \texttt{EDCBA})}{(\texttt{ABCDE} \cup \texttt{EDCBA})} = 1$$

Fig. 5: Jaccard index of the sequences pair equals 1, hence the corresponding Jaccard distance is 0.

for transactional data as in Bouguessa (2011) which computes the homogeneity of sequence pair by the frequency of common element occurrence. Nonetheless, these methods do not take into account element order which is a key feature to differentiate sessions in web usage context.

Therefore, the feature that these measures have in common, as presented in Table 2, is the ignorance of the order of elements. As they regard the common page visits as sequence similarity indication, they could not address different user behaviors that are indicated by page visit order. In other words, to understand user interest through their browsing pattern, visiting page `A` then `B` and visiting page `B` then `A` should not be considered the same behavior. An example of the application of the Jaccard distance is given in Figure 5.

3.3 Considering sequences with different lengths and their element order, ignoring element succession

Referred to as the edit distance, Levenshtein scores the difference between two sequences. Unlike Hamming, Levenshtein distance is able to deal with sequences of different length. Hence, not only substitutions but deletions and insertions of elements are used to transform one sequence into another. Levenshtein guarantees to find the minimal number of edit operators. Each of them changes one symbol into another one. The definition of the Levenshtein distance between two sequences S_1 and S_2 of lengths l_1 and l_2, respectively is given by:

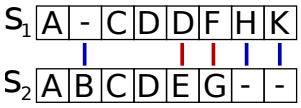
$$Lev_{S_1,S_2}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{S_1,S_2}(i-1,j)+1 \\ lev_{S_1,S_2}(i,j-1)+1 \\ lev_{S_1,S_2}(i-1,j-1)+1_{S_{1_i} \neq S_{2_j}} \end{cases} & \text{otherwise} \end{cases} \quad (4)$$

Table 3: Measures that consider sequences with different lengths and their element order, ignore element succession

Measure feature	Measure name	Measure description
Considering sequences with different lengths and their element order, ignoring element succession	Levenshtein	This distance is characterized as the minimum number of elements' insertions, deletions or substitutions to convert one sequence into another one.
	Algiriyage	An approach applying Levenshtein to calculate similarity of navigational sequences. Moreover, this method considers the time spent on page visits.
	LCS	This measure detects one of the longest subsequence which is lain in both sequences. This subsequence ignores the continuity of matching symbols between these sequences.
	NW	It measures the global similarity between two sequences. When sequences are in similar length, it guarantees to find their best alignment to boost the similarity.
	SAM	This is a non-Euclidean distance. It scores similarity of sequences through insertions and deletions of unique elements and transpositions of common elements to make sequences identical. However, SAM discounts sequence length.
	FOGSAA	Fast Optimal Global Sequence Alignment Algorithm globally explores two sequences by establishing a Branch-and-Bound tree where each path from root to leaf is shown as an achievable path of sequence pair alignment, implemented by greedily growing branches to end up with the best path.
	DTW	Dynamic Time Warping has been recognized as a good alignment method for time-series. Nonetheless, DTW neglects the identical continuing elements in sequences due to its flexible transformation allowance on time series so that their similar shapes can be revealed.

With $1_{S_{1_i} \neq S_{2_j}}$ an indicator function and $lev_{S_1,S_2}(i,j)$ the distance between the i first characters of S_1 and the j first characters of S_2. Examples of Levenshtein distance are shown in Figure 6 and Table 4 to exemplify this category of methods. The alignment considers two sequences of different lengths, it respects their element order but does not reveal the separation of matches in the distance result. In other words, Levenshtein is able to deal with unequal length sequences to count sequence element order but fail to grasp common element succession.

Levenshtein's extension, Damerau-Levenshtein, alternatively permits transposition that reverses the order of sequence elements. Web access similarity measure presented by Algiriyage et al. (2015)



$$Levenshtein(\texttt{ACDDFHK}, \texttt{ABCDEG}) = 5$$

Fig. 6: Levenshtein distance is computed by the minimal number of edit operations to duplicate two sequences.

is one of the approaches using Levenshtein to calculate session similarity (in collaboration with time spent on page by users) but not identifying the succession of common visits in session sequence.

Step	Sequence before step modification	Modification	Result
1	C **O** U R T	Substitution	C ***A*** U R T
2	C A **U** R T	Deletion	C A — R T
3	C A — **R** T	Deletion	C A — — T
4	C A — — T	Insertion	C A — — T ***S***

Table 4: A step by step Levenshtein distance calculation of the sequences "COURT" and "CATS". The distance value is 4 since there are 4 steps.

There have been similarity evaluation algorithms working on different length sequences and recognizing the importance of the order of items. Longest Common Subsequence (LCS) is among them and also another edit distance which seeks one of the LCS that is concatenated by separate but common elements of two sequences. Thus, the LCS length indicates the similarity between sequences. It does not take mismatches and the continuation of common elements in sequences into account, hence only deletions and insertions are allowed to take part in building LCS. Given sequences $S_1 = \texttt{\{ABCBDAB\}}$ and $S_2 = \texttt{\{BDCABA\}}$, there are three LCS between S_1 and S_2 : `{BDAB}`, `{BCBA}` and `{BCAB}` of length 4. Likewise, regardless of the disjointedness when aligning over the entire length of two biological sequences, Needleman and Wunsch (1970) (NW) globally computes the similarity of sequences. The similarity score resulted by this algorithm is optimal because NW is using dynamic programming (DP) to compare sequences. DP is a well-known computational method which repeatedly breaks a complex problem into smaller parts to facilitate its resolution (Howard, 1966). Figure 7 shows an example of the NW alignment. Beginning with a similarity score of 0 at the top left corner of the matrix. Each cell is filled depending of :

$$CellScore = max \begin{cases} TopCellScore - 1 \\ LeftCellScore - 1 \\ \begin{cases} TopLeftCellScore + 1 & \text{if the cell corresponds to a match} \\ TopLeftCellScore - 1 & \text{otherwise} \end{cases} \end{cases} \tag{5}$$

The best alignment is given by the path which ends in the bottom right corner with the highest similarity score. In Figure 7, we can see four matches and two gaps in the best path. The two gaps correspond here to one insertion and one deletion.

Lu et al. (2005) and Yilmaz and Senkul (2010) studied how to generate significant usage patterns using NW. However it ignores element consecution that is essential to evaluate similarity of web sessions. Alternatively, NW can be modified to adapt to $semi-global$ alignment when start and end gaps are disregarded on purpose, in order to find considerable overlaps of sequences. Accordingly,

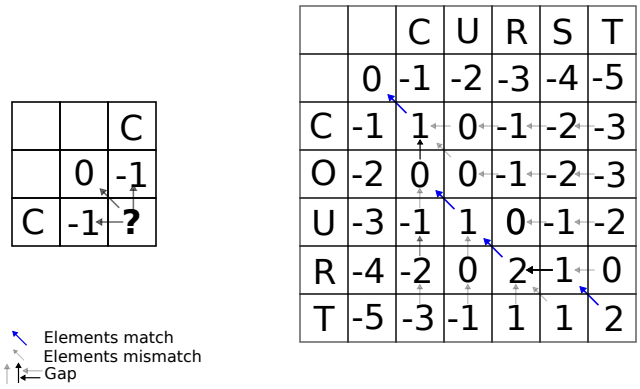


Fig. 7: Pairwise alignment using Needleman-Wunsch of "CURST" and "COURT" sequences.

gaps appearing before the first element and after the last one are not taken into consideration while scoring.

Another suggested measure is SAM (Hay et al., 2004), which operates as a generalized version of the edit distance. SAM calculates indels' cost for unique pages and re-orders cost for common pages that is needed to equalize web session sequence pairs.

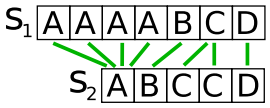
$$SAM(S_1, S_2) = (\omega_d D + \omega_i I) + \eta R \tag{6}$$

where ω_d is the weight of the deletion operation ($\omega_d > 0$), ω_i is the weight of the insertion operation ($\omega_i > 0$), η is the reordering weight ($\eta > 0$), D is the number of deletion operations, I is the number of insertion operations and R is the number of reordering operations.

Repeatedly, like the two previous approaches (NW and LCS), SAM respects the sequential order of elements but not their continuity as it uses open sequences to evaluate the similarity. For certain applications such as web prediction and recommendation, the succession of common pages are essential to differentiate usage patterns.

Different from dynamic programming which is used in NW, but related to dynamic programming back tracking process, FOGSAA (Chakraborty and Bandyopadhyay, 2013b) is a similarity measure which attempts to find the optimal alignment path of two sequences by greedy pairwise alignment. Thereby, a Branch-and-Bound tree with paths from root to leaf is built, and each of these paths serves as a possible alignment of the sequence pair. Branches of paths start from elements of sequences, and expand when the *fitness score* at a chosen node is greater than its sibling, or pruned when the path's score is no longer favorable. Fitness score is calculated by matches, mismatches and gaps scoring on the aligning way. FOGSAA repetitively expands possible paths until no better path is found at each node. Although FOGSAA consistently finishes with an optimal path, for resource saving, the algorithm is supposed to be terminated if the similarity of the sequences is less than a threshold, for instance 30%. Either ended up with the best alignment of two sequences or terminated by exceeding a threshold, the algorithm preserves the obtained similarity score.

A unique method which works differently from others in this category, as it is not a pairwise alignment method and does not allow gaps, is DTW (Petitjean et al., 2014). DTW optimally minimizes the cost function (Nakamura and Kudo, 2011) (*i.e.* distance between pair of data point or event sequences) whereas NW optimally maximizes similarity score. As a result, DTW measures the dissimilarity between sequences. In the context of web usage mining, two sessions with less dissimilarity in page visits are more similar, then DTW can be taken into consideration as an



$$DTW(\texttt{AAAABCD}, \texttt{ABCCD}) = 0$$

Fig. 8: DTW score equals to zero as successive identical elements in sequences are considered to be identical.

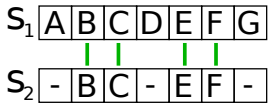
approach of sequence alignment. DTW aligns one individual element from one sequence with many identical and consecutive elements in the other, if they all are alike, which makes the warping path segment vertical or horizontal. In other words, identical and consecutive elements in this case are merged into only one. This is a drawback in web usage mining as a series of duplicate visits is a user behavior that is worth considering. Therefore, DTW is appropriate for time series stretching or compressing but not for mining web sessions. The sequence alignment mechanism of DTW, which returns the distance between sequences, is briefly explained in Figure 8. Main characteristics of measures in this category is shown in Table 3.

3.4 Considering sequences with different lengths, their element order and local matching

Table 5: Measures that consider sequences with different lengths, their element order and local matching

Measure feature	Measure name	Measure description
Considering sequences with different lengths, their element order and local matching	SW	Built on but in contrast with NW, SW optimally searches for local similarity segments of any lengths. It is appropriate for working with sequences of unequal length.
	Sequence Alignment Based Distance Measure (SABDM)	This method takes advantages of SW to seek local identical portions between two sequences and then calculate the ratio of their length over longer sequence length to score the similarity.
	N-gram	This algorithm utilizes sub-sequences of length N to compute similarity score between sequences. Accordingly, the matching of small segments might lead to the correspondence of whole sequences.
	SBS	Similarity Between Sessions is an approach of sequence similarity evaluation through a characteristic sequence that covers all of possible sequence pattern. Thus, two sequences are not considered to be similar if one is not the subset of the other.

Together with NW, SW (Smith and Waterman, 1981) is one of two popular alignment approaches implementing dynamic programming (Maleki et al., 2016; Zahid et al., 2015). Both NW



$SW(\texttt{ABCDEFG}, \texttt{BCEF}) = 4$

Fig. 9: Scoring SW alignment by counting pairwise matches between two sequences.

and SW take into account the similarity between sequences in different alignments (Yan et al., 2013) and have their own strengths and drawbacks (Barton et al., 2015). In contrary to NW, that explores local regions which are of high similarity between protein or nucleotide sequences, SW optimizes the adjustment to maximize the region similarity score. As ones can see in Figure 9, SW locally detects similarity segments of two sequences, or partial similarities (Aruk et al., 2012). If gaps are only allowed at the ends instead of all over sequences to detect if one sequence is partially a substring of the other, it is not SW but a *semi-global* alignment. Stimulated by the basic function of SW and to make it more global, SABDM (Poornalatha and Raghavendra, 2011a) takes advantage of SW score and the length of the longest sequence in web session pair to evaluate their likelihood. As a result, it focuses on local similarity and discounts the unaligned pages even if they have a significant effect on similarity comparison. Given two sequences S_1, S_2 of respective lengths l_1, l_2, the following equation shows how to calculate SABDM.

$$\text{SABDM}_{S_1,S_2} = \frac{max(l_1, l_2) - similarityCount}{max(l_1, l_2)} \tag{7}$$

where *similarityCount* represents the number of matches within SW's alignment method of S_1 and S_2. SABDM's values lie between 0 and 1. A value of 0 indicates that the compared sequences are perfectly similar, whereas a value of 1 corresponds to entirely different sequences. Another distance in this category, N-gram (Kondrak, 2005), focuses on mismatches of length N when comparing two sequences. It is called a *bigram* in case $N = 2$, *trigram* in case $N = 3$, and N is used as a threshold to recognize the similarity of sequences. Namely, if it does not exist any mismatch of length N, then two sequences are evaluated to be similar, and vice versa. For example, (Buscaldi et al., 2012) use N-gram and do not consider element orders in a sequence but only N consecutive symbols. It runs as a probabilistic model to predict the next item based on current N-item set and can be used as a unigram sequence similarity measure. Alternatively, SBS (Similarity Between Sessions) (Anupama and Gowda, 2015) considers the similarity between web sessions by creating subsets of a representative session that carries most achievable patterns of web browsing occurrence. This approach leads to a similarity measure defined through sequence subset occurrence number. Consequently, if one sequence is not the subset of the other, even if it is caused by only one different element, two sequences are supposed to be different.

As the similarity between sessions should be evaluated over their entire length, regional similarity metrics is not appropriate to be merely applied in web usage mining. Providing scoring scheme of match 1, SW with similarity score is illustrated in Figure 9 and can be regarded as the representative method of this category. Features of category measures are outlined in Table 5.

3.5 Considering sequences with different lengths, their element order, local and global matching

In this category, approaches that take both quantitative and qualitative aspects when considering the similarity of web sessions are presented in Table 6. HSAM (Poornalatha and Prakash, 2013) incorporates SAM into SABDM to take advantage of their effectiveness and reduce shortcomings

Table 6: Measures that consider sequences with different lengths, their element order, local and global matching.

Measure feature	Measure name	Measure description
Considering sequences with different lengths, their element order, local and global matching	HSAM	This proposed distance measure is a hybrid of SAM and SABDM. As a result, Hybrid Sequence Alignment Measure can globally and locally deal with uneven length web sessions with no need of altering page order.
	Combination	The combination between global and local alignments based on dynamic programming like NW and SW to perform a comprehensive similarity evaluation.
	Hybrid	Similar to the Combination, Hybrid is a composite of NW and SW but does not equally take global and local pairwise alignment into account in scoring similarity between two sequences.
	S^3M	This takes into account both element existence and their occurrence order when assessing the sequence similarity by a collaboration of featured LCS and Jaccard.

of the two approaches. In this manner, distance between sequences is measured through uncommon elements and number of aligned element with or without gap insertions. Consequently, regional and entire alignment are performed on sequences as shown in the corresponding equation :

$$\text{HSAM}(S_1, S_2) = \frac{NUP + 2 \times |NAP - NDA|}{l_1 + l_2} \tag{8}$$

where :
- S_1 and S_2 are the compared sequences of lengths l_1, l_2 respectively.
- NUP is the Number of Unaligned Pages. It corresponds to the number of pages in the sequences which have no correspondence in the other sequence.
- NAP is the Number of Aligned Pages. This refers to the number of pages' alignments once both sequences are aligned using local alignment method used in SABDM.
- NDA stands for Number of Direct Alignments. These are the number of pages which are already aligned in the original sessions.

The idea of Hybrid (Chordia and Adhiya, 2011; Dimopoulos et al., 2010) and Combination (Luu et al., 2016b) is a blending of NW and SW. Since this merging strategy incorporates global into local similarity algorithm, it makes the output much better in similarity than the use of NW or SW alone, as presented in these approaches (Chordia and Adhiya, 2011; Dimopoulos et al., 2010; Luu et al., 2015). Figure 10 shows global and local alignment score of a sequence pair using NW and SW, respectively. Chordia and Adhiya (2011) described a hybrid similarity measure as:

$$Hybrid(S_1, S_2) = (1 - p) \times SW(S_1, S_2) + p \times NW(S_1, S_2) \tag{9}$$

where $SW(S_1, S_2)$ is the score of the local alignment of sequences S_1 and S_2 (using SW's method), $NW(S_1, S_2)$ is the score of the global alignment of these sequences (using NW's method). p is a

parameter expressing the importance given to each score. Generally, if $l_1 \geq l_2$, given l_1 is the length of S_1 and l_2 the length of S_2, p is defined by :

$$p = \frac{l_2}{l_1} \tag{10}$$

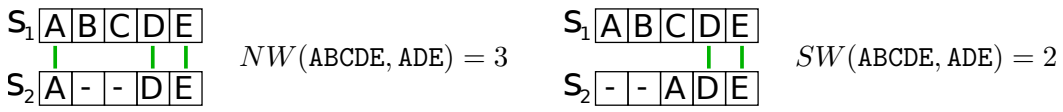
Thus when sequences have the same length, global alignment has more impact on this alignment measure. Conversely, a large difference between sequences' lengths will give more weight to local alignment. Similarly, a measure was developed by Dimopoulos et al. (2010) to "glocally" measure the similarity between two sequences. According to this method, the distance between two sequences is computed by taking global and local alignments and their weights into consideration. These weights are inversely proportional to each other, depending on the difference in length of the sequences. Specifically, local alignment weight is greater if the sequences' lengths are more different. In other words, the more contrasting in sequence lengths, the more local similarity influences, and vice versa. Because it turns to be global for shorter sequence and local for longer sequence, this measure is meaningful in some specific situations, such as similarity detection on dissimilar sequences. Nevertheless, it is not really useful in the evaluation of web access similarity. A local alignment scoring scheme like SW does not take into account the difference of length in the compared sequences. Indeed, the longer this difference is, the more impact it should have on similarity measure. As local alignment's scoring does not take the difference in the sequences' lengths into account, the dissimilarity in users' browsing behaviors is not fully revealed.

As Combination (Luu et al., 2015) considers NW and SW equally, it comes up with the empirical improvement of Hybrid :

$$Combination(S_1, S_2) = \frac{NW(S_1, S_2)}{max(l_1, l_2)} + \frac{SW(S_1, S_2)}{2 \times max(l_1, l_2)} \tag{11}$$

With l_1,l_2 the respective lengths of sequences S_1, S_2. Namely, the scoring of local and global similarity in Combination is not affected by the difference of sessions' lengths as in Hybrid. Unlike biological sequences with comparative lengths, sessions have a variation of lengths appearing frequently through variable user's behaviors. This is valuable for e-commerce in order to target different customers' groups.

Proposed in (Mishra et al., 2014), Sequence and Set Similarity Measure (S^3M) integrates the element content of sequences into their order information. Jaccard metric is adopted here to find the ratio of common elements over unique elements of the two session sequences which indicates the similarity of them. The approach also exploits the proportion of LCS length to longer sequence length in order to reflect the similarity of element order across two sequences. These two aspects are merged afterward, by the summation of coefficients of each aspect, and this summation ranges from 0 to 1.



(a) (b)

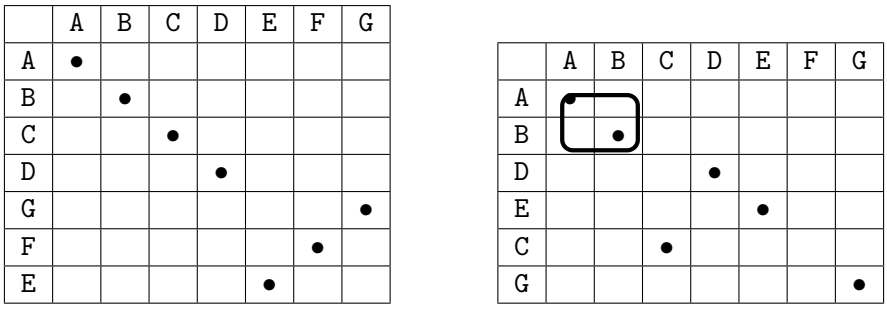
Fig. 10: The difference between NW similarity and SW similarity, applying the same scoring scheme.

4 Other approaches

There also exist other approaches that are not categorized in the preceding sections, but are possibly applicable in web usage mining. These are presented in the following.

Dot-matrix (DM) (Sonnhammer and Durbin, 1995) treats sequence pair as two axes of a dot-matrix plot so that their similar regions can be revealed. Nevertheless, DM is not an alignment method as it just visually compares and shows similarity regions, but ignores gaps or considers them to be mismatches between sequences. Although DM does not score the comparison, it provides a better view of possible alignments that can be made through pair of sequences than other alignment methods. Yet another unique feature of DM is the ability of *inverse- matching* exposure, for instance `ABCD` in one sequence and `DCBA` in the other, which is useful in collaboration with edit-distance methods that allow transposition, like Damerau-Levenshtein or SAM. It is also a useful extension for other edit distances that disallow transposition, such as Levenshtein or LCS. However, it can be improved to be a formal alignment by adding gaps (corresponding to horizontal and vertical gaps in dot-matrix) to link similar segments and then applying some scoring schemes. Alternatively, the distinct level of dot color shades may be adopted to facilitate the degrees of matching between elements. In addition, a combination of dot plots and n-gram (Maetschke et al., 2010) may help to avoid common noise randomly caused by cross-matches of long sequences. In order to print countable matching dots, this kind of combination needs a definition of window size n and a ratio of matches over window size as threshold.

Figure 11 shows how two sequences can be aligned using dot matrix. In Figure 11a, the corresponding symbol matches between `ABCDEFG` and `ABCDGFE` is shown as 4 dots in the main diagonal. The inverse matching portion between the two sequences, which is `GFE` and `EFG`, is displayed by 3 other dots. In Figure 11b, there are gaps and mismatches between two sequences `ABCDEFG` and `ABDECG` but their distinction is not presented in the dot plot. Furthermore, a window size of 2 is illustrated by the solid square working as a criteria of similarity in order to filter noise such as the matches of `C` or even `G` in two sequences. In other words, `AB` and `DE` are recognized as similarity segments between sequences.



(a) (b)

Fig. 11: In dot matrix method, each sequence is put as an axis of a grid. Subsequently, dots are positioned in cells to represent matching portions of sequences. Visual diagonal lines formed by the dots are used to track the expanse of matches.

Hidden Markov Model (HMM) is a probabilistic sequence model. HMM assigns labels to units of an input sequence in order to compute the probability distribution towards label sequences, then ends up with the best label sequence selection (Eddy, 2004). HMM contains interconnected states, transitions among them, and probability values of each state and transition describing their

frequency distributions. Accordingly, a given sequence can be represented by a Markovian path (*i.e.*, next state in path only depends on the current state) through series from initial to terminal hidden states and their transitions. In order to align a given set of sequences to a template, HMM finds probabilities of states such as symbols, insertions, deletions and transitions by training a set of unaligned sequences. This training creates a template or family of possible alignments, called *profile*, which is used to characterize the alignment and search for more similar sequences by computing the similarity score between them and the profile. This score, which is sequence probability, is calculated by the multiplication of sequence corresponding state and transition probabilities along the path. From the initial model of training sequences, there may be iterative update models while adding other sequences to compare. If a specific profile is defined at the first place, its feature parameters such as transition number or frequency of states can be estimated and deployed to find new sequences with maximum likelihood as a family member. However, this model has a main drawback. If the training set is highly similar and not large enough, the initial model will be biased or under-fitting. Therefore, alignment characteristics such as global or local type are required to feature in the training set. Nevertheless, HMM profiles have a formal probabilistic basis which provides the true residue frequency at any given position in alignment. It does not use or depend on gap penalty or scoring scheme, but it aligns multiple sequences as effective as any other alignment methods.

Figure 12 illustrates HMM mechanism of multiple sequences' alignment. Each of illustrated nodes has its associated probability that summation of matching probabilities in each Mx, Ix, Dx and transitions initiate from them is equal to 1.0. As a result, each given sequence from Figure 12a is used as an input of Figure 12b to generate the corresponding sequence with its likelihood of being a member of the trained family. For instance, if hidden state sequence of the trained model is `ABCDEF`, a sequence such as `ABCDEF` from Figure 12a will go straight and through the matching states from M1 until reaching the END state. Nonetheless, another sequence from Figure 12a consists of a deletion at node 6 like *ABCDEFG* will transition from M5 to D6 and ends up at the END state. Alternatively `A_CDEF` from the sequence set consists of an insertion between nodes 1 and 2, its corresponding path goes to I2 after M1 and before reaching M2. This insertion may be iterative depending on the necessary insertion number for aligning to the hidden state sequence of model. An alignment of the HMM profile to any observed sequence derives the most probable state sequence with a probability. Namely, the probability of each sequence from Figure 12a is computed by multiplying the probabilities from initial to end states, and the multiplication result is supposed to be the sequences' similarity score. As in Figure 12c, the probability of `A_CDEF` along the path is calculated as $0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.2 \times 1.0 = 0.00625$.

Pandi et al. (2011) proposed a novel sequences' comparison method through common elements in sequences' pairs, considering correlation, distance and steps between them. Accordingly, from original pair of sequences, there are two corresponding extracted sequences which hold common elements. Other elements which are uncommon of original sequences are converted to distance between common elements. Considering the two featured sequences to be the referred and the referring ones, the shortest distances between connected elements in the referred sequence are recorded. In order to build a Hasse diagram, it is necessary to explore the referring sequence in association with the referred sequence. More specifically, each element of referring sequence is formed as nodes, and connections between these nodes that satisfy the corresponding connections of referred sequence will be added as edge to the Hasse diagram. This adding process is iterative and finishes when a valid Hasse diagram is built. In the next step, edges' weights are computed based on their distance and step length in the Hasse diagram. The sum of edges' weights is the similarity score between the referring and the referred sequences. Finally, the similarity between two original

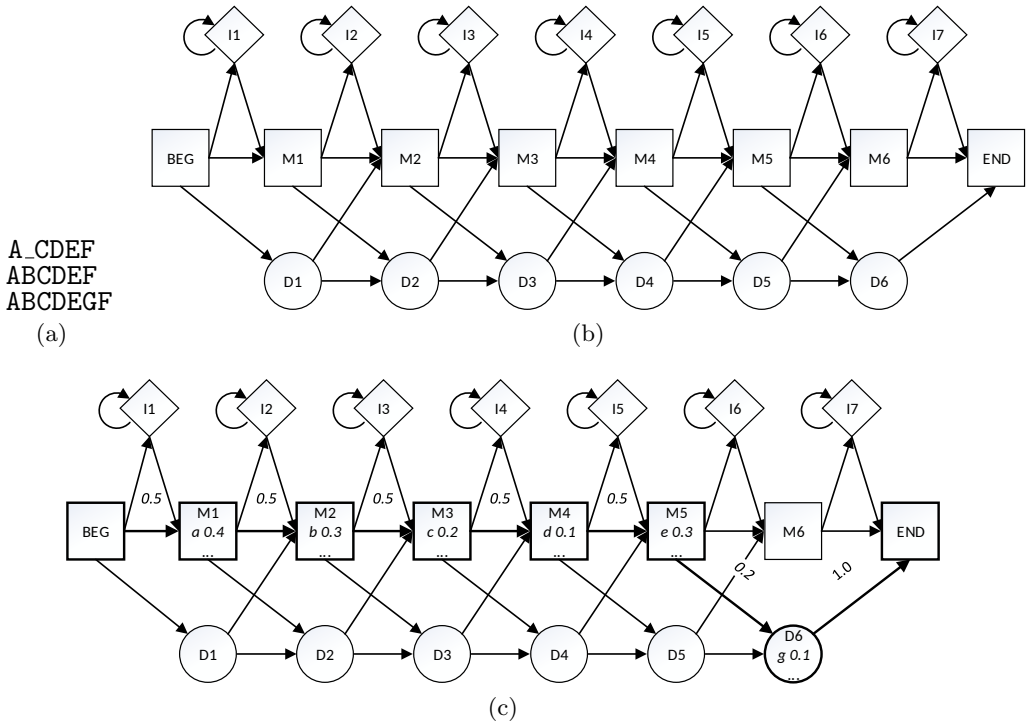


Fig. 12: Sequence set of symbols (a) are aligned using HMM (b)(c) which is a trained state machine consists of node types: Mx represents matches in index x, Dx represents deletions in index x, Ix represents insertions in index x, and arrows represent transitions among them.

sequences is the average of the two alternative similarity summations when one sequence takes the referred role, the other takes the referring role, and vice versa.

Figure 13 describes a simple example of how this approach works. The two original sequences as in Figure 13a, S_1 = `AFBEGCEFDYZ` and S_2 = `IAIBJJDNCM` are converted to featured sequences of S'_1 = (`1A2B3C3D3`) and S'_2 = (`2A2B3D2C2`) by keeping common elements and changing unique elements to numeric values of distance. For instance, distance between `A` and `B` is 2 in both sequences. At the beginning of featured sequence S'_1, it takes 1 from open parenthesis to the first element `A`. In order to illustrate the case where featured sequences S'_1 and S'_2 are referring and referred sequences respectively, we extract the distance information of S'_2. Since the elements of S'_2 are unique, distances between them are the shortest ones. For instance, the shortest distance between "`(`" and `B` is 4, the shortest distance of `B` and `D` is 3, and so forth. S'_1 is used to build the Hasse diagram in Figure 13c by adding edges to featured sequence elements in Figure 13b as diagram nodes. As ones can see, referring to the coherence of connections in S'_2, there is no connection between `C` and `D` although it exists in S'_1. Furthermore, the added edges are of shortest distances since featured sequence S'_2 are similar to S'_1 with unique elements. Consequently, given the weight formula in (Pandi et al., 2011), where weight of an edge, for example between "`(`" and `A`, weighted as $\frac{1}{1(1+|2-1|)}$, total weight of referring sequences can be computed as follows: $S(S_1, S_2) = \frac{1}{1(1+|2-1|)} + \frac{1}{1(1+|1-1|)} + \frac{1}{1(1+|3-5|)} + \frac{1}{2(1+|6-3|)} + \frac{1}{1(1+(3-4))} = 2.46$ as the similarity of referring to referred sequence. The next steps, as previously mentioned, are computing the alternative similarity and final average similarity to indicate the correspondence of the two original sequences.

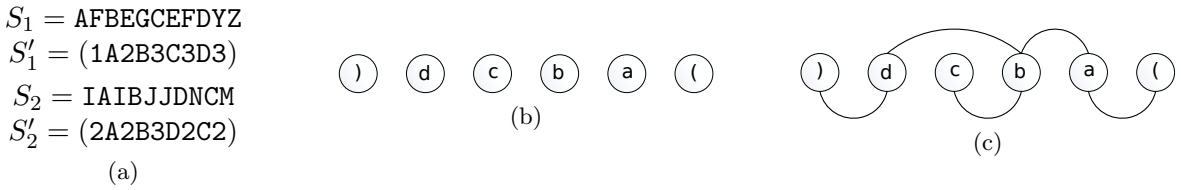


Fig. 13: Input sequences S_1 and S_2 in (a) are parsed into nodes as in (b) and used to build Hasse diagram as in (c).

5 Discussion

5.1 Bioinformatics approaches and web usage features

Click streams generated by user interaction with web pages often contains more information than the click order. A similarity or dissimilarity measure would make sense if it can make good use of this additional information. For instance, if the significance of each web page is identified based on their weights in web owner's business, the extended scoring schemes defined previously in Section 2 can be considered and applied. For this purpose, our suggestion is possibly similar to substitution biology matrices Point Accepted Mutation (PAM) (Delmestri and Cristianini, 2010) and BLOcks SUbstitution Matrix (BLOSUM) (Tong, 2013) to seek for the homogeneity of sequences. These matrices show the dependence of substitution cost upon involved amino acid sequences, which were implemented by some NW and SW applications (Luu et al., 2015; Poornalatha and Raghavendra, 2011a).

However, by reducing unnecessary features, many algorithms that are originally dedicated to bioinformatics have been adapted to web usage mining (Wang et al., 2015). For instance, from the web usage mining point of view, number of gaps in session sequence alignment (no matter a gap of length k or k isolated gaps) may not make sense as the way they do in bioinformatics. Thus, the *constant, linear* or *affine gap penalty* may be useless for web usage mining. Additionally, progressive and iterative alignment as used in T-COFFEE (Di Tommaso et al., 2011; Taly et al., 2011) and MUltiple Sequence Comparison by Log-Expectation (MUSCLE) (Edgar, 2004; Edgar et al., 2005), respectively, are not necessary because there is no need of session compatible arrangement like protein sequences to study their evolutionary relationship. Alternatively, greedy approach such as ClustalW (Hung and Weng, 2016) was not maintained as an optimal alignment (Bucka-Lassen et al., 1999) due to its heuristic and slowness, that caused by "once a gap always a gap" when aligning a sequence with, not another sequence, but an alignment. Besides, alphabet size of biological sequences is limited to 20 (amino acids) or 4 (DNA, RNA) that is certainly not enough to index all pages of most web sites. They all make biological methodologies and tools inappropriate to web patterns, although these algorithms can be modified or inspire web usage mining approaches.

Alternatively, web usage pattern discovery can benefit from other information from web sessions such as the time spent on pages or the visit frequency of pages. Considering these information in session similarity assessment facilitates the similarity of user behaviors, not only for page visits correspondence. Azimpour-Kivi and Azmi (2011) recommended to integrate the time spent and the visit frequency of pages into SAM as another dimension of non-Euclidean algorithm. Banerjee and Ghosh (2001) introduced a web session similarity evaluation method considering the time users spent on pages. Nonetheless, only time spent on pages reside in the LCS which is the intersection of the two sessions should be taken into consideration. As mentioned previously, this approach has the limitation of catching the succession of the common elements. In a similar, Li and Lu (2007)

regards page viewing time of users as one of the factors used in similarity measurement between sessions. Gündüz and Özsu (2003) merges normalized time spent on page into the alignment scoring component together with sessions' length and pages' order. However, according to Pinkham (2010), the approach has one drawback : the time spent on web pages can be influenced by external reasons. In order to partially improve this kind of method, Li (2009) proposed to weight the interaction time between web server and client before finishing page load, and page size as it affects page loading time. Yet the difference between activity duration and the duration from start time to end time on page is another disadvantage of all these approaches, since there is no distinction of time spent on page by scrolling, highlighting, hovering, zooming, etc. and interruption time caused by other irrelevant activities. Another approach of Chakraborty and Bandyopadhyay (2013a) takes this problem into consideration through the fraction of time spent on page and page size, as well as the portion of that fraction in sum of page fraction of a session. However, the explanation of this portion function is inappropriate since it does not reveal the idle time of a web page while displayed for a long time.

One more web feature referred to in Chakraborty and Bandyopadhyay (2013a) is page similarity assessment through URL structural similarity. However, since URLs nowadays tend to be shortened by random representation string rather than any fixed structure form, the correspondence in URL structure should not be widely used to evaluate the similarity of pages. Such shortcomings can be found in other similar approaches, for instance Wang and Zaïane (2002) and Shi (2009). Not to mention, web user interest is not necessarily revealed by URL structure, for example `car.html` and `bike.html` may have exactly the same prefix in their URLs but each page can be accessed by a distinct group of interest. There are also other information that should be considered for user clustering such as meta tags, `robots.txt` file or titles but they are not in the scope of this review.

5.2 Computational complexity

All approaches considering only sequences of similar length or ignoring element order (Section 3.1 and Section 3.2) have a linear time complexity of $\mathcal{O}(l)$ with l the length of the sequences. Almost all methods using sequence alignment discussed in this paper (Section 3.3, Section 3.4 and Section 3.5) have a time complexity of $\mathcal{O}(l_1 l_2)$ for two sequences S_1 and S_2 of lengths l_1 and l_2 respectively. More generally, without loss of generality, assuming that $l_1 \geq l_2$, the time complexity can be said to be $\mathcal{O}(l_1^2)$. Most implementations use dynamic programming (Gonnet and Benner, 1996) which refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. When computing the similarity between each pair of sequences of a dataset, the complexity becomes $\mathcal{O}(N^2 l^2)$ with N the number of sequences and l the length of the sequences. In order to mitigate the quadratic time complexity of pairwise alignment, some approximation methods have been proposed. For example, Chakraborty and Bandyopadhyay (2013b) introduced FOGSAA that allows a time gain of 25-40% compared to NW without drastically reducing the quality of the alignment. Another strategy consists in using lower bounds which provide an estimation of the dissimilarity greater than or equal to the exact distance. For example, DTW is well known to have multiple lower bounds that can be used to avoid having to compute the entire cost matrix (Tan et al., 2018). These techniques are generally used directly in the application of data mining algorithms to avoid having to compute the full alignment for all pairs of sequences. While important progress has been made in this area, using alignment based techniques for very large amount of sequences is still challenging and is an important area of research in the big data community.

5.3 External validation

Web usage mining detects hidden user groups (*i.e.* unsupervised learning) sharing similar behavior of navigation. For that reason, the last but not least challenge is the clustering results' validation. There have been specific cluster characteristics to count when quantifying cluster quality such as compactness or separation, which are carried out by internal and external validation. In order to eliminate the bias toward synthetic data in internal validation, external validation should be performed on ground truth to prove that a proposed measure can make sense in real web usage situation. To conduct experiments on ground truth, not any available datasets but those containing appropriately classified data to the web business have to be provided.

Besides k-means which was used by Chitraa and Thanamni (2012), Si et al. (2012) or Hung et al. (2013), a trendy clustering strategy is hierarchical clustering. After the distance or similarity matrix is produced by adopted sequence alignment algorithm, the selection of linkage criteria has a considerable impact on how clusters are created. Single-linkage mostly takes advantage on elements that have the compact among them and are separated with others adequately. On the contrary, complete-linkage is sensitive to outliers due to its non-local merge criterion. Another popular criteria, average-linkage, neutrally computes distance between two clusters as the average of distances between elements in one cluster and the other, thus it is generally slower than the previous two. In internal validation, an appropriate clustering linkage strategy can be selected in order to adapt to features of generated synthetic dataset, and consequently the clustering result primarily depends on sequence alignment efficiency. However, in external validation, data characteristic and how clusters should be formed from ground truth may be not as obvious as from generated synthetic dataset, thus the decision of the proper kind of linkage among possible criterion is complicated. External validation indices that were used to evaluate cluster quality such as Rand or Jaccard in Milligan and Cooper (1986) are not able to make a distinction between the influence of sequence alignment algorithm and linkage criteria.

In the situation that there is no appropriate labeled data available for external benchmarking, internal benchmarking may be used. Liu et al. (2010) analyzed two kinds of benchmarking and concluded the advantage of internal benchmarking in multiple validation aspects compared to some external benchmarkings. On the other hand, Rendón et al. (2011) revealed that a superior accuracy can be achieved in their scenario by internal instead of external benchmarking.

6 Conclusion

Sequence alignment methods for web sessions have recently gained significant interest in web usage clustering applications. In this paper, we introduced sequence alignment mechanisms and proposed a categorization of a large array of alignment-based methods which are relevant and have been used in various approaches of web session similarity examination. We pointed out their advantages and drawbacks, common and distinct features through categorizing. We presented the different impacts of methods in the same category when working on sequences like web sessions. In addition, some significant non-alignment approaches which are applicable for session similarity evaluation, for instance dot matrix or Hidden Markov Model, were mentioned for further investigations of their utilization in session sequence similarity analysis. We also discussed remaining challenges like web usage characteristics which make web sessions different from biological sequences in similarity evaluation and computational complexity of certain measures. We mentioned hierarchical strategy for clustering, and ground truth for clustering validation so that a significant improvement may be carried on to enhance sequence alignment implementation. Our literature review would be beneficial

in algorithm comparison and selection to perform web session alignment and clustering, and other circumstances of unsupervised learning. Our next study would be a comparison of experimental results of discussed methods to identify how they should be used and their legitimate effect in clustering process, since a method that seems advantageous does not inevitably produce good results.

Acknowledgements The authors would like to thanks the Beampulse company for providing datasets to perform experiments. They also like to thanks VIET and Campus France for funding this research.

References

Algiriyage N, Jayasena S, Dias G (2015) Web user profiling using hierarchical clustering with improved similarity measure. In: Moratuwa Engineering Research Conference (MERCon), 2015, IEEE, pp 295–300

Anandhi D, Ahmed MI (2017) Prediction of user's type and navigation pattern using clustering and classification algorithms. Cluster Computing pp 1–10

Anupama D, Gowda SD (2015) Clustering of web user sessions to maintain occurrence of sequence in navigation pattern. Procedia Computer Science 58:558–564

Aruk T, Ustek D, Kursun O (2012) A comparative analysis of smith-waterman based partial alignment. In: Computers and Communications (ISCC), 2012 IEEE Symposium on, IEEE, pp 000250–000252

Azimpour-Kivi M, Azmi R (2011) A webpage similarity measure for web sessions clustering using sequence alignment. In: Artificial Intelligence and Signal Processing (AISP), 2011 International Symposium on, IEEE, pp 20–24

Banerjee A, Ghosh J (2001) Clickstream clustering using weighted longest common subsequences. In: Proceedings of the web mining workshop at the 1st SIAM conference on data mining, Citeseer, vol 143, p 144

Barton C, Flouri T, Iliopoulos CS, Pissis SP (2015) Global and local sequence alignment with a bounded number of gaps. Theoretical Computer Science 582:1–16

Bose RJC, van der Aalst WM (2012) Process diagnostics using trace alignment: opportunities, issues, and challenges. Information Systems 37(2):117–141

Bouguessa M (2011) A practical approach for clustering transaction data. In: Machine Learning and Data Mining in Pattern Recognition, Springer, pp 265–279

Breitinger F, Baier H (2012) A fuzzy hashing approach based on random sequences and hamming distance. In: Proceedings of the conference on digital forensics, security and law, Association of Digital Forensics, Security and Law, p 89

Bucka-Lassen K, Caprani O, Hein J (1999) Combining many multiple alignments in one improved alignment. Bioinformatics (Oxford, England) 15(2):122–130

Buscaldi D, Tournier R, Aussenac-Gilles N, Mothe J (2012) Irit: Textual similarity combining conceptual similarity with an n-gram comparison method. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, Association for Computational Linguistics, pp 552–556

Chakraborty A, Bandyopadhyay S (2013a) Clustering of web sessions by fogsaa. In: Intelligent Computational Systems (RAICS), 2013 IEEE Recent Advances in, IEEE, pp 282–287

Chakraborty A, Bandyopadhyay S (2013b) Fogsaa: Fast optimal global sequence alignment algorithm. Scientific reports 3

Chaofeng L (2009) Research on web session clustering. Journal of Software 4(5):460–468
Chitraa V, Thanamni AS (2012) An enhanced clustering technique for web usage mining. In: International Journal of Engineering Research and Technology, ESRSA Publications, vol 1
Chordia BS, Adhiya KP (2011) Grouping web access sequences using sequence alignment method. Indian Journal of Computer Science and Engineering (IJCSE) 2(3):308–314
Daily J (2016) Parasail: Simd c library for global, semi-global, and local pairwise sequence alignments. BMC bioinformatics 17(1):81
Della Vedova G (2000) Multiple sequence alignment and phylogenetic reconstruction: Theory and methods in biological data analysis. PhD thesis, Citeseer
Delmestri A, Cristianini N (2010) String similarity measures and pam-like matrices for cognate identification. UOB-ISLTR2010
Deza MM, Deza E (2013) Distances and similarities in data analysis. In: Encyclopedia of distances, Springer, pp 291–305
Dhandi M, Chakrawarti RK (2016) A comprehensive study of web usage mining. In: Colossal Data Analysis and Networking (CDAN), Symposium on, IEEE, pp 1–5
Di Tommaso P, Moretti S, Xenarios I, Orobitg M, Montanyola A, Chang JM, Taly JF, Notredame C (2011) T-coffee: a web server for the multiple sequence alignment of protein and rna sequences using structural information and homology extension. Nucleic acids research 39(suppl_2):W13–W17
Dimopoulos C, Makris C, Panagis Y, Theodoridis E, Tsakalidis A (2010) A web page usage prediction scheme using sequence indexing and clustering techniques. Data & Knowledge Engineering 69(4):371–382
Eddy SR (2004) What is a hidden markov model? Nature biotechnology 22(10):1315–1316
Edgar RC (2004) Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research 32(5):1792–1797
Edgar RC, Edgar RC, Edgar RC, USCLE M (2005) Muscle user guide. Tech. rep., Technical Report. 2004 Available at http://www. drive5. com/muscle/docs. htm
El Azab A, Mahmood MA, El-Aziz A (2017) Effectiveness of web usage mining techniques in business application. The Dark Web: Breakthroughs in Research and Practice: Breakthroughs in Research and Practice p 227
Gauch S, Speretta M, Chandramouli A, Micarelli A (2007) User profiles for personalized information access. In: The Adaptive Web, Springer, pp 54–89
Gonnet GH, Benner SA (1996) Probabilistic ancestral sequences and multiple alignments. In: Scandinavian Workshop on Algorithm Theory, Springer, pp 380–391
Gündüz Ş, Özsu MT (2003) A web page prediction model based on click-stream tree representation of user behavior. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 535–540
Hay B, Wets G, Vanhoof K (2004) Mining navigation patterns using a sequence alignment method. Knowledge and Information Systems 6(2):150–163
Howard RA (1966) Dynamic programming. Management Science 12(5):317–348
Hung JH, Weng Z (2016) Sequence alignment and homology search with blast and clustalw. Cold Spring Harbor Protocols 2016(11):pdb–prot093088
Hung YS, Chen KLB, Yang CT, Deng GF (2013) Web usage mining for analysing elder self-care behavior patterns. Expert Systems with Applications 40(2):775–783
Kondrak G (2005) N-gram similarity and distance. In: International symposium on string processing and information retrieval, Springer, pp 115–126
Li C (2009) Research on web session clustering. Journal of Software 4(5):460–468

Li C, Lu Y (2007) Similarity measurement of web sessions by sequence alignment. In: Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on, IEEE, pp 716–720

Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. Briefings in bioinformatics 11(5):473–483

Liu Y, Li Z, Xiong H, Gao X, Wu J (2010) Understanding of internal clustering validation measures. In: International Conference on Data Mining, IEEE, pp 911–916

Liu Z, Wang Y, Dontcheva M, Hoffman M, Walker S, Wilson A (2017) Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. IEEE Transactions on Visualization and Computer Graphics 23(1):321–330

Lopes P, Roy B (2015) Dynamic recommendation system using web usage mining for e-commerce users. Procedia Computer Science 45:60–69

Lu L, Dunham M, Meng Y (2005) Discovery of significant usage patterns from clusters of clickstream data. In: Proc. of WebKDD, Citeseer, pp 21–24

Luu VT, Forestier G, Fondement F, Muller PA (2015) Web site audience segmentation using hybrid alignment techniques. In: Trends and Applications in Knowledge Discovery and Data Mining, Springer, pp 29–40

Luu VT, Forestier G, Ripken M, Fondement F, Muller PA (2016a) Web usage prediction and recommendation using web session clustering. In: Digital Information Management (ICDIM), 2016 Eleventh International Conference on, IEEE, pp 107–113

Luu VT, Ripken M, Forestier G, Fondement F, Muller PA (2016b) Using glocal event alignment for comparing sequences of significantly different lengths. In: Machine Learning and Data Mining in Pattern Recognition, Springer, pp 58–72

Madeira F, Lee J, Buso N, Gur T, Madhusoodanan N, Basutkar P, Tivey A, Potter SC, Finn RD, Lopez R, et al. (2019) The embl-ebi search and sequence analysis tools apis in 2019

Maetschke SR, Kassahn KS, Dunn JA, Han SP, Curley EZ, Stacey KJ, Ragan MA (2010) A visual framework for sequence analysis using n-grams and spectral rearrangement. Bioinformatics 26(6):737–744

Maleki S, Musuvathi M, Mytkowicz T (2016) Efficient parallelization using rank convergence in dynamic programming algorithms. Communications of the ACM 59(10):85–92

Malik ZK, Fyfe C (2012) Review of web personalization. Journal of Emerging Technologies in Web Intelligence 4(3):285–296

Mandal OP, Azad HK (2014) Web access prediction model using clustering and artificial neural network. In: International Journal of Engineering Research and Technology, ESRSA Publications, vol 3

Milligan GW, Cooper MC (1986) A study of the comparability of external criteria for hierarchical cluster analysis. Multivariate Behavioral Research 21(4):441–458

Mishra R, Kumar P, Bhasker B (2014) An alternative approach for clustering web user sessions considering sequential information. Intelligent Data Analysis 18(2):137–156

Nakamura A, Kudo M (2011) Packing alignment: alignment for sequences of various length events. In: Advances in Knowledge Discovery and Data Mining, Springer, pp 234–245

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology 48(3):443–453

Neelima G, Rodda S (2016) Predicting user behavior through sessions using the web log mining. In: Advances in Human Machine Interaction (HMI), 2016 International Conference on, IEEE, pp 1–5

Omega C, Kalign L, MAFFT L, MView L (2015) Multiple sequence alignment

Pandi M, Kashefi O, Minaei B, et al. (2011) A novel similarity measure for sequence data. Journal of Information Processing Systems 7(3):413–424

Petitjean F, Forestier G, Webb G, Nicholson AE, Chen Y, Keogh E, et al. (2014) Dynamic time warping averaging of time series allows faster and more accurate classification. In: International Conference on Data Mining, IEEE, pp 470–479

Pinkham J (2010) Method of tracking & targeting internet payloads based on time spent actively viewing. US Patent App. 12/393,546

Poornalatha G, Prakash SR (2013) Web sessions clustering using hybrid sequence alignment measure (hsam). Social network analysis and mining 3(2):257–268

Poornalatha G, Raghavendra P (2011a) Alignment based similarity distance measure for better web sessions clustering. Procedia Computer Science 5:450–457

Poornalatha G, Raghavendra PS (2011b) Web user session clustering using modified k-means algorithm. In: Advances in Computing and Communications, Springer, pp 243–252

Pramanik S, Setua S (2017) An opposition based differential evolution to solve multiple sequence alignment. In: International Conference on Computational Intelligence, Communications, and Business Analytics, Springer, pp 440–450

Raphaeli O, Goldstein A, Fink L (2017) Analyzing online consumer behavior in mobile and pc devices: A novel web usage mining approach. Electronic Commerce Research and Applications 26:1–12

Rendón E, Abundez I, Arizmendi A, Quiroz E (2011) Internal versus external cluster validation indexes. International Journal of computers and communications 5(1):27–34

Rosenberg MS (2009) Sequence alignment: methods, models, concepts, and strategies. Univ of California Press

Shi P (2009) An efficient approach for clustering web access patterns from web logs. International Journal of Advanced Science and Technology 5(1):354–362

Si J, Li Q, Qian T, Deng X (2012) Discovering k web user groups with specific aspect interests. In: Machine Learning and Data Mining in Pattern Recognition, Springer, pp 321–335

Smith TF, Waterman MS (1981) Identification of common molecular subsequences. Journal of molecular biology 147(1):195–197

Sonnhammer EL, Durbin R (1995) A dot-matrix program with dynamic threshold control suited for genomic dna and protein sequence analysis. Gene 167(1):GC1–GC10

Taly JF, Magis C, Bussotti G, Chang JM, Di Tommaso P, Erb I, Espinosa-Carrasco J, Kemena C, Notredame C (2011) Using the t-coffee package to build multiple sequence alignments of protein, rna, dna sequences and 3d structures. nature protocols 6(11):1669

Tan CW, Herrmann M, Forestier G, Webb GI, Petitjean F (2018) Efficient search of the best warping window for dynamic time warping. In: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, pp 225–233

Ting IH, Clark L, Kimble C (2009) Identifying web navigation behaviour and patterns automatically from clickstream data. International Journal of Web Engineering and Technology 5(4):398–426

Tong JC (2013) Blocks substitution matrix (blosum). In: Encyclopedia of Systems Biology, Springer, pp 152–152

Vorontsov IE, Kulakovskiy IV, Makeev VJ (2013) Jaccard index based similarity measure to compare transcription factor binding site models. Algorithms for Molecular Biology 8(1):1

Wagh R, Patil J (2017) Enhanced web personalization for improved browsing experience. Advances in Computational Sciences and Technology 10(6):1953–1968

Wang G, Zhang X, Tang S, Zheng H, Zhao BY (2016) Unsupervised clickstream clustering for user behavior analysis. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, pp 225–236

Wang W, Zaïane OR (2002) Clustering web sessions by sequence alignment. In: Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on, IEEE, pp 394–398

Wang XD, Liu JX, Xu Y, Zhang J (2015) A survey of multiple sequence alignment techniques. In: International Conference on Intelligent Computing, Springer, pp 529–538

Yan R, Xu D, Yang J, Walker S, Zhang Y (2013) A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. Scientific reports 3

Yang J, Huang H, Jin X (2017) Mining web access sequence with improved apriori algorithm. In: Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on, IEEE, vol 1, pp 780–784

Yilmaz H, Senkul P (2010) Using ontology and sequence information for extracting behavior patterns from web navigation logs. In: Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, IEEE, pp 549–556

Zahid SK, Hasan L, Khan AA, Ullah S (2015) A novel structure of the smith-waterman algorithm for efficient sequence alignment. In: International Conference on Digital Information, Networking, and Wireless Communications (DINWC), IEEE, pp 6–9