Feature Matrices: A Model for Efficient and Anonymous Web Usage Mining

Cyrus Shahabi, Farnoush Banaei-Kashani, Jabed Faruque, and Adil Faisal

Department of Computer Science, Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089-2561, USA
[shahabi,banaeika,faruque,faisal]@usc.edu

Abstract. Recent growth of startup companies in the area of Web Usage Mining is a strong indication of the effectiveness of this data in understanding user behaviors. However, the approach taken by industry towards Web Usage Mining is off-line and hence intrusive, static, and cannot differentiate between various roles a single user might play. Towards this end, several researchers studied probabilistic and distance-based models to summarize the collected data and maintain only the important features for analysis. The proposed models are either not flexible to trade-off accuracy for performance per application requirements, or not adaptable in real-time due to high complexity of updating the model. In this paper, we propose a new model, the FM model, which is flexible, tunable, adaptable, and can be used for both anonymous and online analysis. Also, we introduce a novel similarity measure for accurate comparison among FM models of navigation paths or cluster of paths. We conducted several experiments to evaluate and verify the FM model.

1 Introduction

Understanding and modeling user behaviors by analyzing users' interactions with digital environments, such as Web-sites, is a significant topic that has resulted in vast recent commercial interests. Commercial products such as $Personify^{TM}$ [HREF1], $WebSideStory^{TM}$ [HREF2], and $WebTrends^{TM}$ [HREF3], and acquired companies such as $Matchlogic^{TM}$, $Trivida^{TM}$, and $DataSage^{TM}$ are all witnesses of such interests. In addition, several researchers in various industrial and academic research centers are focusing on this topic [1][3][4][5][7]. A nice survey is provided by Srivastava et al. [8]. Meaningful interpretation of the users' digital behavior is necessary in the disparate fields of e-commerce, distance education, online entertainment and management for capturing individual and collective profiles of customers, learners and employees; for targeting customized/personalized commercials or information, and for evaluating the information architecture of the site by detecting the bottlenecks in the information space.

Understanding user behavior from Web usage data, a.k.a Web Usage Mining (WUM), involves the following three steps: 1) collection and preparation of usage data, 2) pattern discovery from usage data, and 3) personalization or

recommendation based on discovered patterns. One typical but naive approach taken by industry is to first build a static profile for each user based on his/her past navigations during the first and second steps. Subsequently, whenever the user re-visits the site, recommend or personalize contents based on the user profile. There are several drawbacks with this approach. First, for this approach to work, the user identity should be detected using some intrusive technique such as through cookies. Second, the profile cannot distinguish between different roles the user might play during various navigations. For example, buying rock music CDs as a gift for a friend versus purchasing classical music CDs for oneself. Third, if multiple users share the same computer/client, the profile becomes a mixture of several (possibly conflicting) tastes. For example, some of us have experienced *amazon.com* suggesting to us those books that are favorites of our significant others. Finally, the profile becomes static and cannot capture changes in the user taste.

To address these problems, we proposed statistical approaches where first a profile is built for a collection of users with similar navigation paths [10]. Then, recommendations/personalizations are mapped to these profiles through some learning mechanism (e.g., by an expert or via utilization of training data). Finally, the navigation of a new user is compared to different profiles and the recommendations/personalizations will be based on either the closest profile (hard classification) or some weighted aggregate of several profiles (soft classification). Therefore, the reaction to a user is not based on his/her previous actions but based on his/her recent activities. Although this approach, henceforth denoted as *"anonymous WUM"*, eliminates the problems with the naive approach, it imposes new challenges to the design of all the three steps of WUM.

In this paper, we propose a novel model, *Feature Matrices* (FM), that not only addresses the new design challenges of anonymous mining, but also can be used for better conventional WUM. For anonymous WUM, during the first step every single action of a new user should be accurately tracked or else the system cannot react to the user in a timely manner. However, due to the large volume of navigation data generated per site, even if this data can be logged accurately online, it cannot be analyzed real-time unless some aspects of the data be dropped out. Hence, in order for a model to be used real-time, it should be flexible to capture less or more features of the logged data in real-time depending on the volume of navigation. During the second step, user clusters must be generated out of several navigation paths. A model is needed to conceptualize each user navigation path and/or a cluster of user paths. The model should be tunable so that one can trade performance for accuracy or vice-versa. Also, in order for the model to be adaptive, it should be possible to update it incrementally as a new path become available. Finally, the third step deals with personalizing the content or recommending new contents towards the user preferences by comparing a new user *partial* navigation path to the model. This process should be performed efficiently enough before the user leaves the site.

The FM model is in fact a generalization of the Vector model proposed by Yan et al. [9]. It is flexible to allow striking a compromise between accuracy

and efficiency, given the specific requirements of an application. Meanwhile, the FM model can be updated both off-line and, incrementally, online so that it can adapt to both short-term and long-term changes in user behaviors. The FM model benefits from two types of flexibility. First, it can be tuned to capture both spatial and temporal features of Web usage data. In addition, it is an open model so that new features can be incorporated as necessary by an application domain. Second, it has the concept of *order,* similar to that of the Markov model [1], which can be increased to capture more data about the various features of navigations.

Another contribution of this paper is proposing a novel similarity measure, $PPED$, for comparing the FM models of partial paths with clusters of paths. With anonymous WUM, it is critical to accurately compare a partial navigation path of a new user to the cluster representatives. Moreover, we investigate a dynamic clustering approach used to make the system adaptable to short-term changes in user behaviors. Although the dynamic clustering can be executed real-time and incrementally, its accuracy is only 10% worse than that of K-Means. It is important to note that we are not proposing to replace periodical re-clustering with dynamic clustering. Instead we are advocating a hybrid approach. A thorough experimental study is conducted that evaluates and verifies the FM model.

The remainder of this paper is organized as follows. In Section 2, we formally define the FM model. Section 3 explains our novel similarity measure. In Section 4, we discuss our dynamic clustering technique. The results of our experiments are included in Section 5. Finally, Section 6 concludes the paper.

2 The Feature Matrices Model

Here, we present a novel model to represent both sessions and clusters in the context of WUM. We denote this model as the *Feature Matrices (FM)* model. With FM, features are indicators of the information embedded in sessions. In order to quantify the features, we consider universal set of segments in a concept space as basis for the session space. Thus, features of a session are modeled and captured in terms of features of its building segments. This conceptualization is analogous to the definition of basis for a vector space, i.e. "a set of linearly independent vectors that construct the vector space". Therefore, the FM model allows analyzing sessions by analyzing features of their corresponding segments.

For the remainder of this section, we explain and analyze the FM model. First, we define our terminology. Next, basics of the FM model are explained: the features captured from user interactions, and the main data structure used to present these features. Subsequently, we discuss how to extract the session FM model and the cluster FM model, separately. Finally, we analyze complexity and completeness of the model.

2.1 Terminology

Web-site A Web-site can be modeled as a finite set of static and/or dynamic Web pages.
Concept Space (Concept) Each Web-site, depending on its application, provides information about one or more concepts. For example, *amazon.com* includes concepts such as *Books*, *Music*, *Video*, etc. The Web pages within a Web-site can be categorized based on the concept(s) to which they belong. A *concept space* or simply *concept* in a Web-site is defined as the set of Web pages that contain information about a certain concept. Note that contents of a Web page may address more than one concept, therefore concept spaces of a Web-site are not necessarily disjoint sets.
Path A *path* P in a Web-site is a finite or infinite sequence of pages:

$$x_1 \rightarrow x_2 \rightarrow ... \rightarrow x_i \rightarrow ... \rightarrow x_s$$

where x_i is a page belonging to the Web-site. Pages visited in a path are not necessarily distinct.
Path Feature (Feature) Any spatial or temporal attribute of a path is termed a *path feature* or *feature*. Number of times a page has been accessed, time spent on viewing a page, and spatial position of a page in the path are examples of features.
Session The path traversed by a user while navigating a concept space is considered a *session*. Whenever a navigation leaves a concept space (by entering a page that is not a member of the current concept), the session is considered to be terminated. Since each page may belong to more than one concept, several *sessions* from different concepts may be embedded in a single *path*. Also, several sessions from the same concept may happen along a path, while user leaves and then re-enters the concept. For analysis, we compare sessions from the same concept space with each other. Distinction between the "session" and the "path" notions makes the comparison more efficacious. To identify the user behavior, we can analyze all the sessions embedded in his/her navigation path, or proiritize the concepts and perform the analysis on the sessions belonging to the higher priority concept(s). Moreover, among the sessions belonging to the same concept space, we can restrict our analysis to the longer session(s), to decrease complexity of the analysis based on the application specifications. In any case, the result of the analysis on different sessions of the same path can be integrated to provide the final result. For example, in a recommendation system, the recommendation can be generated based on various user preferences detected by analyzing different sessions of the user's navigation path. Thus, hereafter we assume all sessions belong to the same concept. Similar analysis can be applied to sessions in any concept space.
Session Space The set of all possible sessions in a concept space is termed *session space*.
Path Segment (Segment) A *path segment* or simply *segment* E is an n-tuple of pages: $(x_1, x_2, ..., x_i, ..., x_n)$. We denote the value n, as the *order* of the segment E ($n \geq 1$). Note that there is a one-to-one correspondence between tuples and

sequences of pages; i.e. $(x_1, x_2, ..., x_i, ..., x_n) \equiv x_1 \rightarrow x_2 \rightarrow ... \rightarrow x_i \rightarrow ... \rightarrow x_n$. We use tuple representation because it simplifies our discussion. Any subsequence of pages in a path can be considered as a segment of the path. For example, the path $x_1 \rightarrow x_3 \rightarrow x_2 \rightarrow x_5 \rightarrow x_2$ contains several segments such as 1st order segment (x_1), 2nd order segment (x_3, x_2), and 4th order segment (x_3, x_2, x_5, x_2). We exploit the notion of segment as the building block of sessions in order to model their features.

Universal Set of Segments $\varepsilon_C^{(n)}$, universal set of order-n segments, is the set of all possible n-tuple segments in the concept space C. Hereafter, since we focus on analysis within a single concept, we drop the subscript C from the notation.

Cluster A *cluster* is defined as a set of similar sessions. The similarity is measured quantitatively based on an appropriate similarity measure (see Section 3).

2.2 Basics

Features We characterize sessions through the following features:

- *Hit (H):* Hit is a spatial feature that reflects which pages are visited during a session. The FM model captures H by recording the number of times each *segment* is encountered in a traversal of the session. Reader may consider H as a generalization of the conventional "hit-count" notion. Hit-count counts number of hits per *page*, which is a segment of order 1.
- *Sequence (S):* Sequence is an approximation for the relative location of pages traversed in a session. As compared to H, it is a spatial feature that reflects the location of visits instead of the frequency of visits. With the FM model, S is captured by recording relative location of each segment in the sequence of segments that construct the session. If a segment has been repeatedly visited in a session, S is approximated by aggregating the relative positions of all occurrences. Thus, S does not capture the exact sequence of segments. Exact sequences can be captured through higher orders of H.
- *View Time (T):* View time captures the time spent on each segment while traversing a session. As opposed to H and S, T is a temporal feature.

Features of each session are captured in terms of features of the segments within the session. We may apply various orders of universal sets as basis to capture different features. Throughout our discussion, we have used $\varepsilon^{(1)}$ for T, and $\varepsilon^{(2)}$ for H and S, unless otherwise stated. Therefore, we extract the feature T for single-page segments, x_i, and features H and S for ordered page-pair segments (x_i, x_j). In Section 2.5, we will explain how using higher order bases results in more complete characterization of the session by the FM model in expense of higher complexity.

The FM model is an open model. It is capable of capturing any other meaningful session features in addition to those mentioned above. The same data structure can be employed to capture the new features. This is another option with which completeness of the FM model can be enhanced. However, our

experiments demonstrate that the combination of our proposed features is comprehensive enough to detect the similarities and dissimilarities among sessions appropriately.

Data Structure Suppose $\varepsilon^{(n)}$ is the basis to capture a feature F for session U, we deploy an n-dimensional *feature matrix*, $M^F_{r^n}$, to record the F feature values for all order-n segments of U. n-dimensional matrix M_{r^n} is a generalization of 2-dimensional square matrix M_{r*r}. Each dimension of M_{r^n} has r rows, where r is the cardinality of the concept space. For example, $M_{4\times4\times4}$ that is a cube with 4 rows in each of its 3 dimensions, is a feature matrix for a 4-page concept space with ε^3 as the basis. Dimensions of the matrix are assumed to be in a predefined order. The value of F for each order-n segment $(x_\alpha, x_\beta, ..., x_\omega)$ is recorded in element $a_{\alpha\beta...\omega}$ of $M^F_{r^n}$. To simplify the understanding of this structure, reader may assume that rows in all dimensions of the matrix are indexed by a unique order of the concept space pages; then the feature value for the order-n segment $(x_\alpha, x_\beta, ..., x_\omega)$ is located at the intersection of row x_α on the 1st dimension, row x_β on the 2nd dimension, ... , and row x_ω on the n-th dimension of the feature matrix. Note that M_{r^n} covers all order-n segment members of $\varepsilon^{(n)}$. for instance, in a 100-page concept space with $\varepsilon^{(2)}$ as the basis, M_{100^2} has 10000 elements. On the other hand, number of segments existing in a session usually is in the order of tens. Therefore, M_{r^n} is usually a sparse matrix. The elements for which there is no corresponding segment in the session are set to zero.

To map a session to its equivalent FM model, the appropriate feature matrices are extracted for features of the session. The entire set of feature matrices generated for a session constitutes its FM model:

$$U^{fm} = \left\{ M^{F_1}_{r^{n_1}}, M^{F_2}_{r^{n_2}}, ..., M^{F_m}_{r^{n_m}} \right\}$$

If $n = \max(n_1, n_2, ..., n_m)$ then U^{fm} is an order-n FM model.

In subsequent sections, we explain how values of different features are derived for each segment from the original session, and how they are aggregated to construct the cluster model.

2.3 Session Model

Here, we explain how values of different features are extracted from a session to form the feature matrices of its FM model. Recall that we record features of a session in terms of features of its segments. Thus, it suffices if we explain how to extract various features for a sample segment E:

- For Hit (H), we count the number of times E has occurred in the session ($H \geq 0$). Segments may partially overlap. As far as there is at least one non-overlapping page in two segments, the segments are assumed to be distinct. For example, the session $x_1 \rightarrow x_2 \rightarrow x_2 \rightarrow x_2 \rightarrow x_1$, has a total of 4 order-2 segments, including 1 occurrence of (x_1, x_2), 2 occurrences of (x_2, x_2), and 1 occurrence of (x_2, x_1).

- For Sequence (S), we find the relative positions of every occurrence of E and record their arithmetic mean as the value of S for E ($S > 0$). To find the relative positions of segments, we number them sequentially in order of appearance in the session. For example, in the session $x_1 \rightarrow^1 x_2 \rightarrow^2 x_2 \rightarrow^3 x_2 \rightarrow^4 x_1$, S value for the segments (x_1, x_2), (x_2, x_2), and (x_2, x_1) are 1, 2.5 $\left(= \frac{2+3}{2}\right)$, and 4, respectively.
- For View Time (T), we add up the time spent on each occurrence of E in the session ($T \geq 0$).

2.4 Cluster Model

With *clustering*, user sessions are grouped into a set of clusters based on similarity of their features. To cluster sessions, since the FM model is a distance-based model, we need a similarity measure to quantify the similarity between sessions, and a clustering algorithm to construct the clusters. Moreover, we need a scalable model for the cluster. Nowadays, any popular Web-site is visited by a huge number of users. In such a scale, we may employ any similarity measure and clustering algorithm to group the sessions (or better to say session models) into clusters, but mere grouping the sessions is not sufficient. If a cluster is naively modeled as a set of session models, any analysis on a cluster will be dependent on the number of sessions in the cluster which is not a scalable solution. Therefore, we are desperately in need of a cluster model that can be used for analysis independent of the number of cluster members. In this section, we describe our cluster model. Subsequently, in Section 3, we introduce an accurate similarity measure for the purpose of clustering, and finally, in Section 4, we propose a variation to conventional clustering algorithms to make them real-time adaptable to varying behaviors.

With our approach of modeling a cluster, we aggregate feature values of all clustered sessions into corresponding feature values of a virtual session. This virtual session is considered as a representative of all the sessions in the cluster, or equally as the model of the cluster. Consequently, the complexity of any analysis on a cluster will become independent of the cluster cardinality.

Suppose we have mapped all the sessions belonging to a cluster into their equivalent session models. In order to aggregate the features of the sessions into the corresponding features of the cluster model, it is sufficient to aggregate features for each basis segment. Assume we denote the value of a feature F for any segment E in the basis by $F(E)$. We apply a simple aggregation function, namely *arithmetic averaging*, to $F(E)$ values in all sessions of a cluster to find the aggregated value of $F(E)$ for the cluster model. Thus, if M^F is the feature matrix for feature F of the cluster model, and M_i^F is the feature matrix for feature F of the i-th session in the cluster, each element of M^F is computed by aggregating corresponding elements of all M_i^F matrices. This procedure is repeated for every feature of the FM model. The final result of the aggregation is a set of aggregated feature matrices that constitute the FM model of the cluster:

$$C^{fm} = \{M^{F_1}, M^{F_2}, ..., M^{F_n}\}$$

Therefore, the FM model can uniquely model both sessions and clusters.

As mentioned before, the aggregation function we use for all features is the simple arithmetic averaging function. In matrix notation, the aggregated feature matrix for every feature F of the cluster model C^{fm} is computed as follows:

$$M^F = \frac{1}{N} \sum_{i=1}^{N} M_i^F$$

where N is the cardinality of the cluster C. The same aggregation function can be applied incrementally, when cluster model has already been created and we want to update it as soon as a new session, U_j, joins the cluster:

$$M^F \leftarrow \frac{1}{N+1} \left(N \times M^F + M_j^F\right)$$

This property is termed *dynamic clustering*. In Section 4, we leverage on this property to modify the conventional clustering algorithms to become real-time and adaptive.

2.5 Analysis of the Model

WUM involves three categories of tasks: constructing clusters of sessions (clustering), comparing sessions with clusters, and integrating sessions into clusters. Regardless of the model employed for analysis and the algorithm used for clustering, complexity of constructing the clusters is dependent on N, which is the number of sessions to be clustered. This is true simply because during clustering each session should be analyzed at least once to detect how it relates to other sessions. The FM cluster model is defined so that it reduces the time complexity of the other two tasks. If the complexity of comparing a session with a cluster and integrating it into the cluster is independent of the cluster cardinality, user classification and cluster updating can be fulfilled in real-time.

The price we have to pay to achieve lower space and time complexity is to sacrifice *completeness*[1]. If the cluster model is merely the set of member sessions stored in their complete form, although the model is *complete* in representing the member sessions, it does not scale. On the other hand, if we aggregate member sessions to construct the cluster model, the model will lose its capability to represent its members with perfect accuracy. The more extensive aggregation is applied, the less complete the cluster model. The FM model is flexible in balancing this trade-off based on the specific application requirements[2].

[1] A model is more complete if it is a better approximation for the real session/cluster.

[2] A formal proof for uniqueness of the FM model for a sesson/cluster is included in [11].

FM Complexity versus the Vector and Markov Models Let $FM^{(n)}$ be an FM model of the order n (see Table 1 for the definitions of terms), where $n = \max(n_1, n_2, \ldots, n_m)$. In the worst case, $FM^{(n)}$ comprises m n-dimensional matrices M_{r^n}, one for each of the model features. Thus, *space* cost of $FM^{(n)}$ is $O(mr^n)$. *Time* complexity for user classification is $O(mL)$ and for updating a cluster by assigning a new session to the cluster is $O(mr^n)$. Therefore, space and time complexity of $FM^{(n)}$ model are both independent of M.

Parameter	Definition
F_i	i-th feature captured in FM
n_i	Order of the basis used to capture F_i
m	Number of features captured in FM
n	$\max(n_1, n_2, \ldots, n_m)$
r	Cardinality of the concept space
L	Average length of sessions
M	Average cardinality of clusters

Table 1. Parameters

From $O(mr^n)$, complexity increases exponentially with n, which is the order of the FM model. Based on Property 1, as the order n increases, the FM model becomes more complete in describing its corresponding session or cluster:

Property 1. If $p_1 > p_2$ then $FM^{(p_1)}$ is more complete than $FM^{(p_2)}$

Thus, added complexity is the price for a more accurate model. An appropriate order should be selected based on the accuracy requirements of the specific application. Formal proof for Property 1 is included in [11].

The other crucial parameter in $O(mr^n)$ is m, the number of features captured by the FM model. Features are attributes of the sessions, used as the basis for comparison. The relative importance of these attributes in comparing the sessions is absolutely application-dependent. The FM model is an open model in a sense that its structure allows incorporating new features as the need arises for different applications. Performing comparisons based on more features result in more accurate clustering, though again the complexity is increased.

Now let us compare the performance of FM with two other conventional models, namely the Vector model and the Markov model. The Vector model can be considered as one special case of the FM model. As used in [9], the Vector model is equivalent to an $FM^{(1)}$ model with H as the only captured feature. Thus, the Vector model scales as $O(r)$, but as discussed above, since it is an order-1 FM model, it performs poorly in capturing information about sessions. Our experiments illustrate that an $FM^{(2)}$ model with S and H as its features outperforms the Vector model in accuracy (see Section 5). The other model, typically employed in dependency-based approaches, is the "Markov" model. Although whether or not Web navigation is a Markovian behavior has been the

subject of much controversy [2], the Markov model has demonstrated acceptable performance in the context of WUM [1]. The transition matrix of an order-n Markov model is extractable from H feature matrix of an $FM^{(n+1)}$ model. Thus, the FM model at least captures the same amount of information as with an equivalent Markov model. They also benefit from the same time complexity of $O(L)$ for dynamic user classification. However, the Markov model cannot be updated in real-time because the complexity of updating a cluster is dependent on the cardinality of the cluster. Moreover, the Markov model is not an *open* model, as described for FM because it is defined to capture order and hit.

3 Similarity Measure

A *similarity measure* is a metric that quantifies the notion of "similarity". To capture behaviors of the Web-site users, user sessions are to be grouped into clusters, such that each cluster is composed of "similar" sessions. Similarity is an application-dependent concept and in a distance-based model such as FM, a domain expert should encode a specific definition of similarity into a pseudo-distance metric that allows the evaluation of the similarity among the modeled objects. With the FM model, these distance metrics, termed *similarity measures*, are used to impose order of similarity upon user sessions. Sorting user sessions based on the similarity is the basis for clustering the users. Some similarity measures are defined to be indicator of dissimilarity instead of similarity. For the purpose of clustering, both approaches are applicable.

In [6], we introduce a similarity measure for session analysis that does not satisfy an important precondition: "the basis segments used to measure the similarity among sessions must be orthogonal". Here, we define a new similarity measure, $PPED$, particularly defined to alleviate the overestimation problem attributed to pure Euclidean distance measure in the context of WUM [9]. This measure satisfies the mentioned precondition. Before defining the function of this similarity measure, let us explain how the FM model is interpreted by a similarity measure.

With a similarity measure, each feature matrix of FM is considered as a uni-dimensional matrix. To illustrate, assume all rows of an n-dimensional feature matrix are concatenated in a predetermined order of dimensions and rows. The result will be a uni-dimensional ordered list of feature values. This ordered list is considered as a vector of feature values in $R^{(r^n)}$, where r is the cardinality of the concept space. Now suppose we want to measure the quantitative dissimilarity between the two sessions U_1^{fm} and U_2^{fm}, assuming that the certain similarity measure used is an indicator of dissimilarity (analogous procedure applies when the similarity measure expresses similarity instead of dissimilarity). Each session model comprises a series of feature vectors, one for each feature captured by the FM model. For each feature F_i, the similarity measure is applied on the two F_i feature vectors of U_1^{fm} and U_2^{fm} to compute their dissimilarity, D^{F_i}. Since the dissimilarity between U_1^{fm} and U_2^{fm} must be based on all the FM features, the

total dissimilarity is computed as the weighted average of dissimilarities for all features:

$$D^F = \sum_{i=1}^{m} w_i \times D^{F_i} \qquad \left(\sum_{i=1}^{m} w_i = 1 \right) \tag{1}$$

where m is the number of features in the FM model. D^F can be applied in both hard and soft assignment of sessions to clusters. Weight factor w_i is application-dependent and is determined based on the relative importance and efficacy of features as similarity indicators. In Section 5, we report on the results of our experiments in finding the compromised set of weight factors for H and S features.

Here, we explain our new similarity measure. Throughout this discussion, assume $\overrightarrow{A}$ and $\overrightarrow{B}$ are feature vectors equivalent to n-dimensional feature matrices M_1^F and M_2^F, and a_i and b_i are their i-th elements, respectively. Vectors are assumed to have $N = r^n$ elements, where r is the cardinality of the concept space.

Projected Pure Euclidean Distance (PPED) $PPED$ is a variant of Pure Euclidean Distance measure (PED) to alleviate the *overestimation* problem. To illustrate the overestimation problem with PED, suppose a user navigates the session U that belongs to cluster C. It is not necessarily the case that the user traverses every segment as captured by C^{fm}. In fact, in most cases user navigates a path similar to only a subset of the access pattern represented by C^{fm} and not the entire pattern. In evaluating the similarity between U^{fm} and C^{fm}, we should avoid comparing them on that part of the access pattern not covered by U or else their dissimilarity will be overestimated. Overestimation of dissimilarity occasionally results in failure to classify a session to the most appropriate cluster.

Assume $\overrightarrow{A}$ and $\overrightarrow{B}$ are two feature vectors of the same type belonging to a session and a cluster model, respectively. To estimate the dissimilarity between $\overrightarrow{A}$ and $\overrightarrow{B}$, $PPED$ computes pure Euclidean distance between $\overrightarrow{A}$ and the projection of $\overrightarrow{B}$ on those coordinate planes at which $\overrightarrow{A}$ has non-zero components:

$$PPED\left(\overrightarrow{A}, \overrightarrow{B}\right) = \left(\sum_{i=1, a_i \neq 0}^{N} (a_i - b_i)^2 \right)^{\frac{1}{2}} \tag{2}$$

where $PPED \in [0, \infty)$. Note that $PPED$ is not commutative.

Non-zero components of $\overrightarrow{A}$ belong to those segments that exist in the session. Zero values, on the other hand, are related to the remainder of the segments in the basis universal set. By contrasting $\overrightarrow{A}$ with the projected $\overrightarrow{B}$, we compare the session and the cluster based on just the segments that exist in the session and not on the entire basis. Thus, the part of the cluster not covered in the session is excluded from the comparison to avoid overestimation.

Since $PPED$ can compare sessions with different lengths, it is an attractive measure for real-time clustering where only a portion of a session is available at

any given time (see Section 4). $PPED$ also helps in reducing the time complexity of the similarity measurement. According to Equation 2, the time complexity of $PPED$ is $O(mL)$ (refer to Table 1 for the definitions of the terms). In Section 5, we report on the superiority of $PPED$ performance as compared to two classical similarity measures, i.e. PED and cosine of the angle formed by the feature vectors (Vector Angle or VA).

4 Dynamic Clustering

As discussed in Section 2.5, since the FM model of a cluster is independent of the cluster cardinality, any cluster manipulation with FM has a reasonably low complexity. Leveraging on this property, we can apply the FM model in real-time applications.

One benefit of this property is that FM clusters can be updated dynamically and in real-time. Note that in most common cluster representations, complexity of adding a new session to a cluster is dependent on the cardinality of the cluster. Therefore, practically in large scale systems, they are not capable of updating the clusters dynamically. By exploiting *dynamic clustering*, the WUM system can adapt itself to changes in users' behaviors in real-time. New clusters can be generated dynamically and existing clusters adapt themselves to the changes in users' tendencies. Delay-sensitive environments such as stock market, are among those applications for which this property is most advantageous. Figure 1 depicts a simple procedure to perform dynamic clustering when a new session is captured.

```
1. Find the distance/similarity between the session and every cluster available in the
current cluster set using  any reasonable similarity measure;

  // All similarity measures discussed in this paper are applicable. These similarity
  // measures are defined based on the data structure of the FM model

2. If there is no cluster closer than TDC to the session
       create a new cluster and use the FM model of the new session as the cluster model;
   else
       update the closest cluster to the session by joining the session to that cluster;

  // TDC is a threshold value specific to Dynamic Clustering. If the distance between the
  // new session and every existing cluster is more than TDC, then it is reasonable to
  // create a new cluster because a new user ehavior has been discovered
```

Fig. 1. An algorithm for *dynamic clustering*

Periodical re-clustering is the typical approach in updating the clusters. This approach results in high accuracy, but it cannot be performed in real-time. According to our experiments to compare the accuracy of the dynamic clustering with that of a periodical re-clustering (see Section 5), dynamic clustering shows

lower accuracy in updating the cluster set. In fact, with dynamic clustering, we are trading accuracy for adaptability. Thus, dynamic clustering should not be used instead of classical clustering algorithms, but a hybrid solution is appreciated. That is, the cluster set should be updated in longer periods through periodical re-clustering to avoid divergence of the cluster set from the trends of the real user behaviors. Meanwhile, dynamic clustering can be applied in real-time to adapt the clusters and the cluster set to short-term behavioral changes.

5 Performance Evaluation

We conducted several experiments to: 1) compare the efficacy of the path features in characterizing user sessions, 2) study the accuracy of our similarity measure in detecting the similarity among user sessions, 3) compare the performance of the FM model with that of the traditional Vector model, and 4) investigate the accuracy of the dynamic clustering. Here, we summarize the results of these experiments. The detailed description of the results, and also our experimental methodology is included in [11].

5.1 Efficacy of the Path Features

A set of experiments was conducted to study the relative efficacy of the path features H and S in detecting similarities between user sessions. In Equation 1, the weight factor w_i indicates relative importance of the path feature F_i in computing the aggregated similarity measure. The higher weights are assigned to the features that are more effective in capturing the similarities. Our experiments were intended to find the compromised set of weight factors w_S (weight factor for S) and w_H that results in the optimum accuracy in capturing the similarities.

The experiment results show that regardless of the weight factors applied, the accuracy is always above 94%. Thus, both features (Hit and Sequence) are equally successful in identifying the spatial similarities. Depending on distinguishability of the dataset, the optimum accuracy is achieved by employing a compromised combination of the similarities detected in *Hit* and *Sequence*. In brief, when similarity among users of the same cluster decreases, it is more important to track which pages they visit (*Hit*) rather than where in the session they visit each page (*Sequence*).

5.2 Accuracy of the Similarity Measures

In Section 3, we introduced $PPED$ as an accurate similarity measure. Here, we compare accuracy of $PPED$ with two classical similarity measures, i.e. PED and cosine (VA).

The experiment results demonstrate that for real data, which assumes low distinguishability, $PPED$ outperforms VA with a wide margin. The results also show that since $PPED$ can measure the similarity between a user and a cluster based on user characteristics rather than cluster characteristics, overestimation

of the distance between the session and its intended cluster is avoided by disregarding unnecessary cluster characteristics in distance estimation. Therefore, $PPED$ can achieve up to 30% improvement in accuracy as compared to PED.

5.3 Performance of the FM Model

We conducted some experiments to compare performances of a sample FM model, namely $FM^{(2)}$ with H and S as its features, with the traditional Vector model, which is considered equivalent to $FM^{(1)}$ with H as its only feature.

Results of this study demonstrate that accuracy of the Vector model decreases as the user sessions become less distinguishable, while the FM model can reasonably maintain its accuracy even with highly indistinguishable datasets. This superiority is because of: 1) incorporating $Sequence$ into the model, and 2) capturing features based on order-2 segments.

5.4 Performance of the Dynamic Clustering

In Section 4, we introduced *dynamic clustering* as an approach to update cluster models in real-time. However, we also mentioned that dynamic clustering trades accuracy for adaptability. We conducted several experiments to study the degradation of the accuracy due to applying the dynamic clustering. For this purpose, we compared dynamic clustering with K-Means.

The experiment results show that as expected accuracy of dynamic clustering is less than accuracy of K-Means but the degradation of the accuracy is tolerable. Thus, the dynamic clustering can be applied to achieve adaptability but it should be complemented by long-term periodical re-clustering. The results also demonstrate that the performance of the dynamic clustering is much better in updating the existing clusters as compared to creating new clusters.

6 Conclusions

We defined a new model, the FM model, which is a generalization of the Vector model to allow for flexible, adaptive, and real-time Web Usage Mining (WUM). We argued that these characteristics are not only useful for off-line and conventional WUM, but also critical for on-line anonymous WUM. We demonstrated how flexibility of FM allows conceptualization of new navigation features as well as trading performance for accuracy by varying the *order*. For FM, we proposed a similarity measure, $PPED$, that can accurately classify partial sessions. This property is essential for real-time and anonymous WUM. We then utilized $PPED$ within a dynamic clustering algorithm to make FM adaptable to short-term changes in user behaviors. Dynamic clustering is possible since unlike the Markov model, incremental updating of the FM model has a low complexity. Finally, we conducted several experiments, which demonstrated the high accuracy of $PPED$ (above 98%), the superiority of FM over the Vector model (by at least 25%) and the tolerable accuracy of dynamic clustering as compared to K-Means (only 10% worse), while being adaptable.

Acknowledgments

This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and ITR-0082826, NASA/JPL contract nr. 961518, DARPA and USAF under agreement nr. F30602-99-1-0524, and unrestricted cash/equipment gifts from NCR, IBM, Intel and SUN.

References

1. Cadez I., Heckerman D., Meek C, Smyth P., and White S.: Visualization of Navigation Patterns on Web-Site Using Model Based Clustering. Technical Report MSR-TR-00-18, Microsoft Research, Microsoft Corporation, Redmond, WA,(2000)
2. Huberman B., Pirolli P., Pitkow J., and Lukos R.: Strong Regularities in World Wide Web Surfing. Science, 280, p.p 95-97 (1997)
3. Mobasher B., Cooley R., and Srivastava J.: Automatic Personalization Based on Web Usage Mining. Special Section of the Communications of ACM on "Personalization Technologies with Data Mining", 43(8):142-151 (2000)
4. Mulvenna M.D., Anand S.S., Bchner A.G.: Personalization on the Net using Web mining: Introduction. CACM 43(8): 122-125 (2000)
5. Perkowitz M., Etzioni O.: Toward Adaptive Web-Sites: Conceptual Framework and Case Study. Artificial Intelligence 118, p.p 245-275 (2000)
6. Shahabi C., Zarkesh A.M., Adibi J., and Shah V.: Knowledge Discovery from Users Web Page Navigation. IEEE RIDE97 Workshop, April (1997)
7. Spiliopoulou M.: Web usage mining for site evaluation: Making a site better fit its users. Special Section of the Communications of ACM on "Personalization Technologies with Data Mining", 43(8):127-134 (2000)
8. Srivastava J., Cooley r., Deshpande M., Tan M.N.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations, Vol. 1, Issue 2 (2000)
9. Yan T.W., Jacobsen M., Garcia-Molina H., Dayal U.: From User Access Patterns to Dynamic Hypertext Linking. Fifth International World Wide Web Conference, Paris, France, (1996)
10. Zarkesh A., Adibi J., Shahabi C., Sadri R., and Shah V.: Analysis and Design of Server Informative WWW-Sites. ACM CIKM '97 (1997)
11. Shahabi C., Banaei-Kashani F., Faruque J., Faisal A.: Feature Matrices: A Model for Efficient and Anonymous Mining of Web Navigations. Technical Report USC-CS-00-736, Computer Science Department, University of Southern California.

Hypertext References

HREF1: *http://www.personify.com*
HREF2: *http://www.websidestory.com*
HREF3: *http://www.webtrends.com*