An integrated model for next page access prediction

F. Khalil*

Department of Mathematics and Computing,
University of Southern Queensland,
Toowoomba 4350, Australia
E-mail: faten410@gmail.com
*Corresponding author

J. Li

School of Computer and Information Science,
University of South Australia,
Mason Lakes 5095, Australia
E-mail: Jiuyong.Li@unisa.edu.au

H. Wang

Department of Mathematics and Computing,
University of Southern Queensland,
Toowoomba 4350, Australia
E-mail: wang@usq.edu.au

Abstract: Accurate next web page prediction benefits many applications, e-business in particular. The most widely used techniques for this purpose are Markov Model, association rules and clustering. However, each of these techniques has its own limitations, especially when it comes to accuracy and space complexity. This paper presents an improved prediction accuracy and state space complexity by using novel approaches that combine clustering, association rules and Markov Models. The three techniques are integrated together to maximise their strengths. The integration model has been shown to achieve better prediction accuracy than individual and other integrated models.

Keywords: web page prediction; Markov Model; association rules; clustering.

Reference to this paper should be made as follows: Khalil, F., Li, J. and Wang, H. (2009) 'An integrated model for next page access prediction', *Int. J. Knowledge and Web Intelligence*, Vol. 1, Nos. 1/2, pp.48–80.

Biographical notes: Faten Khalil received her PhD Degree from the University of Southern Queensland, Australia in 2008. She has been conducting research in web page access prediction since 2005.

Jiuyong Li is an Associate Professor at the University of South Australia. He worked at the University of Southern Queensland as a Lecturer

and then Senior Lecturer. He received his PhD Degree in Computer Science from Griffith University, Australia. His main research interests are in data mining, privacy preservation and biomedical informatics. His research has been supported by two Australian Research Council Discovery grants. He has published more than 50 referred journal and conference papers, some of which are in high impact publication venues in computer science. He has been a main organiser of Australasian Data Mining Conference since 2006.

Hua Wang is an Associate Professor in the University of Southern Queensland. He was awarded a PhD Degree in Computer Science from the University of Southern Queensland in 2004. He has been active in the areas of information systems management, distributed database management systems, access control and data mining. He has participated in research projects on mobile electronic system, Web service, and role-based access control for Electronic service system, and has already published over 80 research papers.

1 Introduction

The immense volume of online information covering almost all types of applications makes the web susceptible to a wide range of information discovery and retrieval tools. Web applications today, e-commerce in particular, are driven to provide a more personalised experience for their users. Therefore, it is extremely important to form some kind of interaction with web users and always be one step ahead of them when it comes to predicting next accessed pages. For instance, knowing the user's browsing history on the site grants us valuable information as to which one of the most frequently accessed pages will be accessed next. Also, it provides us with extra information like the type of user we are dealing with and the user's preferences as well. Some widely used data mining methods to achieve the goal are association rule mining, clustering and Markov classification.

Association rule mining is a major pattern discovery technique (Mobasher et al., 2000; Agrawal and Srikant, 1994). The patterns are discovered based on previous history. The original goal of association rule mining is to solve market basket problem. For a data set containing shopping transactions, association rules summarise relationships illustrated by the following example. Customers who buy bread and milk will most likely buy eggs, or, bread and milk $\rightarrow$ eggs. The main limitation of association rule mining is that many rules are generated, which result in contradictory predictions for a user session.

Markov Models are very commonly used in the identification of patterns based on the sequence of previously accessed pages (Bouras and Konidaris, 2004; Chen et al., 2002; Deshpande and Karypis, 2004; Eirinaki et al., 2005; Jespersen et al., 2003; Sarwar et al., 2001; Zhu et al., 2002a, 2002b). They are the natural candidates for sequential pattern discovery for link prediction due to their suitability to modelling sequential processes. The Markov Model process calculates the probability of the page the user will visit next after visiting a sequence of web pages in the same session. Markov Model implementations have been hindered due to the fact that low order Markov Models do not use enough history and

therefore, lack accuracy, whereas, high order Markov Models incur high state space complexity.

Clustering groups user sessions into clusters based on similarity between common activities (Adami et al., 2003; Cadez et al., 2003; Strehl et al., 2000). It aims at dividing web sessions into groups where the distance between clusters is maximised while the distance between sessions within the same cluster is minimised. The clustering methods partition data objects into a number of homogeneous groups based on their similarity. Clustering methods do not classify user sessions directly, but will help build better classification models if data objects are properly clustered.

All three methods have been widely used for web page access prediction. However, the limitations associated with them hinder their improvements when it comes to web page access prediction and state space complexity. The main purpose behind implementing such tools for web page access prediction is to achieve reliable prediction accuracy while keeping the model complexity to a minimum. So far, the individual implementation of these tools fails to achieve such results.

This paper aims at improving the web page access prediction accuracy while keeping the model complexity small by integrating Markov Models, association rules and clustering. The paper presents a novel approach to build a combined model. Web pages are clustered into consistent groups before combined Markov and association model is built on each individual group. Markov Model is kept at a low order to maintain a low state complexity. Association rules are used for cases when Markov Model could not make decisive prediction. This paper extends our previous work (Khalil et al., 2006, 2007).

The rest of the paper is organised as follows: Section 2 of the paper covers related work in the area. Section 3 introduces Markov Model, association rules and clustering techniques and discusses their limitations. Section 4 examines the proposed model and explains the integration algorithm. Section 5 provides proficient concept experiments. Finally, Section 6 concludes our work.

2 Related work

A number of researchers attempted to improve the web page access prediction accuracy or coverage by combining different recommendation frameworks. For instance, many papers proposed the use of association rule mining or Markov Model for next page prediction. However, none of the papers have addressed the use of a combination of both methodologies (Khalil et al., 2006). However, many papers combined clustering with association rules (Lai and Yang, 2000; Liu et al., 2001). Lai and Yang (2000) have introduced a customised marketing on the web approach using a combination of clustering and association rule mining techniques. They proved through experimentations that implementing association rules on clusters achieves better results than on non-clustered data for customising the customers' marketing preferences. Liu et al. (2001) have introduced Mining Association Rules using Clustering (MARC) that helps reduce the I/O overhead associated with large databases by making only one pass over the database when learning association rules. Although the authors prove through experimentation that MARC can learn association rules more efficiently, their algorithm does not improve the accuracy of the association rules learned.

Other papers combined clustering with Markov Model (Cadez et al., 2003; Zhu et al., 2002a; Lu et al., 2005). However, none of the papers proved to improve prediction accuracy and state space complexity. Cadez et al. (2003) partitioned site users using a model-based clustering approach where they implemented first order Markov Model using the Expectation-Maximisation algorithm. They also developed a visualisation tool called WebCANVAS based on their model. Markov Model was used for clustering rather than for prediction purposes. Zhu et al. (2002a) construct Markov Models from log files and use co-citation and coupling similarities for measuring the conceptual relationships between web pages. CitationCluster algorithm is then introduced to cluster conceptually related pages. The authors then combine Markov Model based link prediction to the conceptual hierarchy into a prototype called ONE to assist users' navigation. The authors implement a hierarchical clustering technique that could lead to running time complexity with large web log files. Lu et al. (2005) were able to generate Significant Usage Patterns (SUP) from clusters of abstracted web sessions. Clustering was applied based on a two-phase abstraction technique. First, session similarity is computed using Needleman-Wunsch alignment algorithm and sessions are clustered according to their similarities. Second, a concept-based abstraction approach is used for further abstraction and a first order Markov Model is built for each cluster of sessions. SUPs are the paths that are generated from first order Markov Model with each cluster of user sessions.

Kim et al. (2004) combine most prediction models (Markov Model, sequential association rules, association rules and clustering) in order to improve the prediction recall. The proposed model proves to outperform classical web usage mining techniques. However, the new model depends on many factors, like the existence of a website link structure and the support and confidence thresholds. These factors affect the order of the applied models and the performance of the new model.

Although web page prediction performance was improved by previous work, the improvement was marginal because they used one model, first order Markov Model, for their recommendations (Khalil et al., 2008). Another integration method was introduced by Kim et al. (2004). Their objective is to trade off recall and precision for multiple page predictions. They use an integration model to improve the trade off, but the trade off improvement is reduced when the number of predicted pages is small. Our work proves to outperform previous works in terms of web page prediction accuracy and state space complexity using a combination of clustering, Markov Model and association rule mining techniques.

3 Existing methods and their limitations

3.1 Markov Model

Markov Models (MMs) are commonly used in the identification of the next page to be accessed by the website user based on the sequence of previously accessed pages (Bouras and Konidaris, 2004; Chen et al., 2002; Deshpande and Karypis, 2004; Eirinaki et al., 2005; Jespersen et al., 2003; Sarwar et al., 2001; Zhu et al., 2002a, 2002b). They have been proposed as the underlying modelling techniques for web prefetching applications (Pons, 2006), to minimise system latency or to improve web server efficiency (Mathur and Apte, 2007).

Let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of pages in a website. Let W be a user session including a sequence of pages visited by the user in a visit. Assuming that the user has visited n pages, then $\text{Prob}(p_i|W)$ is the probability that the user visits pages p_i next. Page p_{n+1} the user will visit next is estimated by:

$$\begin{aligned} P_{n+1} &= \text{argmax}_{p \in P}\{\text{Prob}(P_i = p \mid W)\} \\ &= \text{argmax}_{p \in P}\{\text{Prob}(P_i = p \mid p_n, p_{n-1}, \ldots, p_1)\}. \end{aligned} \tag{1}$$

This probability, $\text{Prob}(p_i|W)$, is estimated by using all W sequences of all users in history (or training data), denoted by W. Naturally, the longer n and the larger W, the more accurate $\text{Prob}(p_i|W)$. However, it is infeasible to have very long n and large W and this leads to unnecessary complexity. Therefore, a more feasible probability is estimated by assuming that the sequence of the web pages visited by users follows a Markov process. The Markov process imposed a limit on the number of previously accessed pages l, where $l \ll n$.

The equation becomes:

$$P_{n+1} = \text{argmax}_{p \in P}\{\text{Prob}(P_{n+1} = p \mid p_n, p_{n-1}, \ldots, p_{n-(l-1)})\}. \tag{2}$$

The resulting model of this equation is called the lth-order Markov Model.

Let S_j^l be a state with l as the number of preceding pages denoting the Markov Model order and j as the number of unique pages in a website.

$$S_j^l = \langle p_{n-(l-1)}, p_{n-(l-2)}, \ldots, p_n \rangle.$$

Using the maximum likelihood principle (Duda et al., 2000), the conditional probability of $P(p_i \mid S_j^l)$ is estimated as follows from a history (training) data set.

$$P(p_i|S_j^l) = \frac{\text{frequency}(\langle S_j^l, p_i \rangle)}{\text{frequency}(S_j^l)}. \tag{3}$$

The fundamental assumption of predictions based on Markov Models is that the next state is dependent on the previous l states. The longer the l is, the more accurate the predictions are. However, longer l causes the following two problems: The coverage of the model is limited and leaves many states uncovered; and the complexity of the model becomes unmanageable. Therefore, the following are three modified Markov Models for Predicting web page access (Deshpande and Karypis, 2004):

1 *All lth Markov Model*: For each test instance, the highest order Markov Model that covers the instance is used to predict the instance (Pitkow and Pirolli, 1999).

2 *Frequency pruned Markov Model*: States with low frequency are removed. The removal of these states affects the accuracy of a Markov Model. However, the number of states of the pruned Markov Model will be significantly reduced.

3 *Accuracy pruned Markov Model*: States with low predictive accuracy can be eliminated. One way to estimate the predictive accuracy using conditional probability is called confidence pruning. Another way to estimate the predictive accuracy is to count (estimated) errors involved, called error pruning.

The evaluation of the pruning process has shown that up to 90% of the states can be pruned leading to less state space complexity and increased coverage but accuracy remains unchanged.

State space complexity of Markov Model is a major issue when implementing Markov Model. Higher orders lead to more states but they usually result in better prediction accuracy since they look at previous browsing history. Another difficulty that rises when constructing Markov Models for prediction purposes is choosing the Markov Model order. Although higher order Markov Models are needed to achieve better prediction accuracy, they are associated with higher state space complexity.

3.2 Association rules

Association rule discovery on usage data results in finding groups of items or pages that are commonly accessed or purchased together. Association rules are mainly defined by two metrics: support and confidence. Support is defined as the discovery of frequent itemsets and confidence is defined as the discovery of association rules from these frequent itemsets (Agrawal and Srikant, 1994).

Let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of pages in a website. Let W be a user session including a sequence of pages visited by the user in a visit, and D includes a collection of user sessions. Let A be a subsequence of W, and p_i be a page. We say that W supports A if A is a subsequence of W, and W supports $\langle A, p_i \rangle$ if $\langle A, p_i \rangle$ is a subsequence of W. The support for sequence A is the fraction of sessions supporting A in D as follows:

$$\sigma = \sup p(A) = \frac{\{W \in D : A \subseteq W\}}{|D|}. \tag{4}$$

The confidence of the implication is:

$$\alpha = \text{conf}(A) = \frac{\sup p(\langle A, P \rangle)}{\sup p(A)}. \tag{5}$$

When we use the same terminologies of Markov Model, $\sup p(\langle A, p_i \rangle) = \text{prob}(\langle A, p_i \rangle)$ and confidence $(A, p_i) = \text{prob}(p_i \mid A)$. An implication is called an association rule if its support and confidence are not less than some user specified minimum thresholds.

Since a full session in web usage mining context includes many pages, it gets very difficult to find matching rule antecedents. Therefore, association rule algorithms usually use a sliding window w whose size is iteratively decreased until an exact match with the antecedent of a rule is found.

There are four types of sequential association rules presented by Yang et al. (2004):

1 *Subsequence rules*: They represent the sequential association rules where the items are listed in order.

2 *Latest subsequence rules*: They take into consideration the order of the items and most recent items in the set.

3 *Substring rules*: They take into consideration the order and the adjacency of the items.

4 *Latest substring rules*: They take into consideration the order of the items, the most recent items in the set as well as the adjacency of the items.

The main problem associated with association rule mining is the frequent item problem where the items that occur together with a high frequency will also appear together in many of the resulting rules and, thus, resulting in inconsistent predictions. As a consequence, a system cannot give recommendations when the data set is large.

In order to overcome this problem and to produce more concise rules, some improved association rule mining methods have emerged. One type of the improved association rules is the non-redundant association rules [http://portal.acm.org/citation.cfm?id=1017508]. A non-redundant association rule set excludes rules with the same support and confidence as their corresponding simpler form rules. Another type of improved association rules is optimal rules [http://portal.acm.org/citation.cfm?id=1128596.1128759 &coll=&dl=]. An optimal rule set excludes rules with the same or lower confidence (or other interestingness criteria) than their corresponding simpler form rules. They have reduced the number of rules greatly in comparison with association rules, but their rule sets are still large in web log data where the number of pages is big and the length of sessions is long.

3.3 Clustering

The primary motivation behind the use of clustering is to improve the efficiency and scalability of the real-time personalisation tasks (Adami et al., 2003; Cadez et al., 2003; Papadakis and Skoutas, 2005; Rigou et al., 2006; Strehl et al., 2000). Generally speaking, clustering aims at dividing the data set into groups (clusters) where the inter-cluster similarities are minimised while the similarities within each cluster are maximised (Srivastava et al., 2000). Clustering web sessions can be achieved through page clustering or user clustering. Page clustering is performed by grouping pages having similar content. On the other hand, clustering user sessions involves selecting an appropriate data abstraction for a user session and defining the similarity between two sessions (Wang et al., 2004).

Clustering can be model-based or distance-based. With model-based clustering (Zhong and Ghosh, 2003), the model type is often specified a priori and the model structure can be determined by model selection techniques and parameters estimated using maximum likelihood algorithms, e.g., the Expectation Maximisation (EM). Distance-based clustering involves determining a distance measure between pairs of items, and then grouping similar items together into clusters. The most popular distance-based clustering techniques include partitional clustering and hierarchical clustering. A partitional method partitions the items into K groups and is represented by k-means algorithm. A hierarchical method builds a hierarchical set of nested clusterings, with the clustering at the top level containing a single cluster of all items and the clustering at the bottom level containing one cluster for each item. Model-based clustering have been shown to be effective for high dimensional text clustering (Zhong and Ghosh, 2003).

Partitional distance-based clustering is disadvantaged by the large number of proposed different distance measures for clustering purposes and defining a good similarity measure is very much data dependent and often requires expert domain knowledge. However, it displayed its ability to produce more efficient web documents clustering results (Strehl et al., 2000; Gunduz and OZsu, 2003).

Clustering can also be supervised (Eick et al., 2004; Finley and Joachims, 2005), semi-supervised (Basu et al., 2004) and unsupervised (Albanese et al., 2004). The difference between supervised and unsupervised clustering is that with supervised clustering, patterns in the training data are labelled. Unsupervised clustering can be classified as hierarchical or non-hierarchical (Jain et al., 1999). Hierarchical clustering can become computationally complex with large data sets and can be difficult to analyse with the absence of logical hierarchical structure in the data. On the other hand, non-hierarchical clustering is where the samples are divided into a predefined number of clusters according to the distance between the data and specific centers. A common method of non-hierarchical clustering is the k-means algorithm that tends to cluster data into even populations. Numerous papers addressed the partitional non-hierarchical clustering algorithm, k-means, and attempted at improving the algorithm. Xiong et al. (2006) investigate the impact data distributions can have on the performance of k-means clustering. The paper illustrates the relationship between k-means and the true cluster sizes as well as the entropy measure. The authors prove experimentally that:

- k-means results in uniform cluster sizes
- regardless of the Coefficient of Variation (CV) of the true cluster sizes, the CV values of the clustering results range between 0.3 and 1.0
- the entropy measure has the favourite on k-means and can be an unsuitable k-means clustering validation measure.

This work implements k-means algorithm because it is efficient, as opposed to hierarchical methods that usually lack the capability to handle data sets with large number of objects as in web data.

Due to the diversity of clustering applications and the large number of distance measurements and data groupings, there exists a large number of clustering algorithms. Data could be represented by different patterns and could have different types of clusters. Also, despite the variety of clustering approaches, clustering alone is not an appropriate approach for web page prediction (Kim et al., 2004). Another clustering limitation is the ability to evaluate and compare their performance due to the lack of an objective evaluation criteria that is independent of the specific application.

4 Proposed model: IMAC

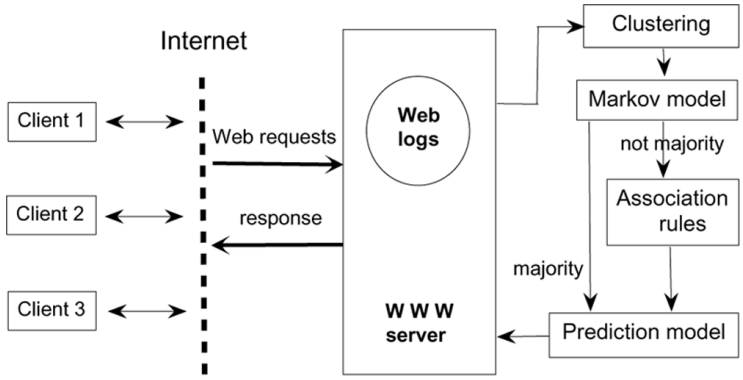
4.1 Motivation for the combined approach

Using Integration of Markov Model, Association rules and Clustering (IMAC), all three methods have been applied to next page prediction but their weaknesses prevent them from achieving best results. For example, low order Markov Models

lack web page prediction accuracy because they do not use enough pages in history and high order Markov Models suffer high state space complexity. Association rules have the problem of large number of rules especially with large web pages. It is also a difficult task to set the minimum support threshold. Clustering methods are unsupervised methods, and normally are not used for classification directly. However, proper clustering groups users sessions with similar browsing history together, and this facilitates classification.

Our integration method follows the following intuition: various types of customers (clients) visit a website, and different types of customers (clients) need different models to simulate their behaviours. We try to use short recent visit history to predict their intention if possible. We only explore long visit history when the recent visit could not bring reliable prediction. We propose to use clustering techniques to cluster the data sets so that homogenous sessions are grouped together. As a result, a more accurate Markov Model is built based on each group rather than the whole data sets. The proposed Markov Model is a low order all lth Markov Model so that the state space complexity is kept to a minimum. The accuracy of low order Markov Models is normally not satisfactory. Therefore, we use association rule mining to make prediction when long history click stream is necessary. Figure 1 depicts the architecture of the integration model (IMAC).

Figure 1 IMAC model architecture



4.2 Algorithm

Integration of Markov Model, Association rules and Clustering (IMAC) involves combining the three web usage mining prediction models clustering, Markov Model and association rules together. In order to reduce the number of transactions used, the web sessions are first divided into categories according to feature selection measures. The web sessions categories are then clustered using k-means clustering algorithm and Cosine distance measure. Each data set is grouped into a different number of clusters. The integration model then computes Markov Model prediction on the resulting clusters. Association rules are used to make prediction when Markov Model could not make decisive decision and long history information is needed.

4.2.1 Algorithm training process

The training process occurs offline. It involves preparing the data and creating the models used for prediction. The IMAC training process is noted in Algorithm 1. The training process of the IMAC algorithm is explained in details as follows:

Algorithm 1 Training Process of IMAC Model

```
1: Combine functionally related pages according to services
   requested
2: Cluster user sessions into k-clusters
3: Build l-Markov model for each cluster
4: for each Markov model state where the majority is not
   clear do
5:    Collect all sessions satisfying the state
6:    Construct association rules
7:    Store the association rules with the state
8: end for
```

Clustering is the first step. A number of data transformations are required before useful clusters can be found. The two major steps are feature selection and feature categorisation. Then, we explain possible similarity metrics for clustering and how to choose the best clustering model. Last, we will explain how to combine Markov Model with association rules.

4.2.1.1 Feature selection

The first step of the training process is feature selection. Since the improved web personalisation is subject to proper preprocessing of the usage data (Eirinaki et al., 2004), it is very important to group data according to some features before applying clustering techniques. This will reduce the state space complexity and will make the clustering task simpler. However, failing to appropriately select the features would result in wrong clusters regardless of the type of clustering algorithm that is used. Wang et al. (2004) presented different feature selections and metrics that form the base of E-commerce customer groupings for clustering purposes. They examined features like services requested, navigation pattern and resource usage. The result of their experimentations proved that all features yield similar results and thus, grouping customers according to one of the features selected should do the job. For our purposes, we will group the pages, and not users, according to services requested since it is applicable to our log data and is simple to implement. Grouping pages according to services requested yields best results if it is carried out according to functionality (Wang et al., 2004). This could be done either by removing the suffix of visited pages or the prefix. In our case, we cannot merge according to suffix because, for example, pages with suffix index.html could mean any default page like OWOW/sec4/index.html or OWOW/sec9/index.html or ozone/index.html. Therefore, merging will be according to a prefix. Since not all websites have a specific structure where we can go up the hierarchy to a suitable level, we had to come up with a suitable automatic method that can merge similar pages automatically. For data set D1 log file, the chosen prefix will be delimited by slash, dot or space. For example, consider the following set of pages:

```
cie/metadata.txt.html cie/index.html
cie/summer95

cie/summer95/articles WhatsHot.html
OER/RFA waisicons/text.xbm

waisicons/eye2.xbm
```

This would lead to the following categories: cie, WhatsHot, OER, and waisicons. Note that the pages are grouped according to their functionality. A program runs and examines each record. It only keeps the delimited and unique word. A manual examination of the results also takes place to further reduce the number of categories by combining similar pages. This was possible because it was carried out on the reduced number of categories as it appears in Table 13, and not the whole data sets.

4.2.1.2 Categorisation

Categorisation or labelling is important for either supervised clustering or classification purposes. Classification methods aim at finding common categories among a set of transactions and mapping the transactions to the predefined categories. Clustering methods, on the other hand, aim at identifying a finite set of categories to describe the data set. The difference between classification and clustering is that in clustering it is not known in advance which categories will be used. In our model, we rely on clustering techniques since the categories are not predefined and they are extracted from the actual data sets.

The categorisation process follows the feature selection implementation and each category is represented by a selected feature. This gives rise to allocating web pages in each session to the appropriate category. The distance between clusters depends on the number of visited pages in each category. Therefore, a weight is allocated to each page based on the number of times it was visited. Consider a data set D containing N number of sessions. Let W be a user session including a sequence of pages visited by the user in a visit. $D = \{W_1, \ldots, W_N\}$. Let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of unique pages in a website. For each page in a session, if the page is visited, a weight factor w is added to p_i representing the number of times the page was visited. The outcome is session $W_i = (p_1^i, w_1^i, \ldots, p_m^i, w_j^i)$ where each W is composed now of the number of times each unique page is visited as it appears in Table 2. If the page is not visited, it is assigned a zero. According to the weight distribution, The set of pages P is divided into a number of categories C_i where $C_i = \{p_1, p_2, \ldots, p_n\}$. This results in less number of pages since $C_i \subset P$ and $n < m$. The outcome is a new session S_i where $S_i = \{(c_1^i, w_1^i), \ldots, (c_L^i, w_j^i)\}$ as it shows in Table 3. D_s is the data set containing N number of sessions S_N.

Combining the similar web pages into categories C_i, makes all sessions of equal length. According to Casale (2005), sessions of equal length give better similarity measures results. As an example, consider the following three sessions apparent in Table 1. Before categorisation, preprocessing of web sessions takes place and each page is assigned a number: Zero if the page is not visited at all, one if the page is visited once, two if twice and so on, as it appears in Table 2. When performing

categorisation, we find out that we have two categories where page 1 and page 2 belong to category I and page 3 belongs to category II. The web sessions become as it appears in Table 3. Thus, using categorisation, the three initial web sessions ended up being of equal length. also, the length of the categorised sessions is shorter because the number of categories is usually smaller than the number of pages.

Table 1 Example: initial web sessions

W1	1, 2, 3, 1, 3
W2	1, 2, 1
W3	3, 1, 3

Table 2 Allocating weight to pages: the number of time each page appeared in a web session

Page	*1, 2, 3*
W1	2, 1, 2
W2	2, 1, 0
W3	1, 0, 2

Table 3 Web sessions after categorisation: assigning pages 1 and 2 to category I and page 3 to category II

Category	*I*	*II*
S1	3	2
S2	3	0
S3	1	2

4.2.1.3 Similarity metrics and quality measures for clustering

A common clustering algorithm is k-means clustering algorithm. It is distance-based, unsupervised and partitional. K-means clustering algorithm is the simplest and most commonly used clustering algorithm, especially with large data sets (Jain et al., 1999). It involves:

1 user sessions are described in vectors with the same length

2 chose a number of clusters (k)

3 initialise k cluster centres randomly

4 assign sessions to the closest cluster.

The k-means clustering algorithm repeatedly performs the following until convergence is achieved:

1 calculate the mean vector for all sessions in each cluster

2 reassign a session to the cluster whose centre is closest to the session.

Because the initial clusters are created randomly, k-means runs different times each time it starts from a different point giving different results. The different clustering solutions are compared using the sum of distances within clusters. The clustering solution with the least sum of distances is preferred. Therefore, k-means clustering depends greatly on the number of clusters (k), the number of runs and the distance measure used. The output is a number of clusters with a number of sessions in each cluster.

The distance measured between sessions in each cluster plays a vital role in forming the clusters. Due to different units of measure in different dimensions, the Euclidean distance measure may not be an adequate measure of closeness even though it is commonly used. It is important to mention that other non-Euclidean distance measures have been proposed (Strehl et al., 2000) and can be useful for the same purpose. In this paper, we examine five distance measures: Euclidean and Squared Euclidean, City Block, Cosine, Pearson Correlation and Hamming; and we choose the most appropriate one.

Euclidean: This is the most straightforward and the most commonly chosen type of distance. It forms the actual geometric distance in the multidimensional space. It is computed as follows:

$$\text{Euclidean}(x, y) = \sqrt{\sum (x_i - y_i)^2}. \tag{6}$$

If greater weight needs to be assigned on items that are further apart, Squared Euclidean distance is used instead and it is computed as follows:

$$\text{Squared Euclidean}(x, y) = \sum (x_i - y_i)^2. \tag{7}$$

City Block: Also known as Manhattan distance, is another common distance measure and it yields results that are similar to the Euclidean distance results. It is only different in that it lessens the outliers effect. It is simply computed by finding the average difference between dimensions:

$$\text{City Block}(x, y) = \sum |x_i - y_i|. \tag{8}$$

Hamming: For real valued vectors, the Hamming distance is equivalent to the City Block distance. It is commonly used to compare binary vectors because of its simplicity. The Hamming distance measures the number of substitutions required to change one string into the other. It can be performed with an exclusive OR function, XOR. It is defined as follows:

$$\text{Hamming}(x, y) = \sum |x_i - y_i|. \tag{9}$$

The hamming distance measure is unsuitable for our data sets because it calculates the percentage of bits that differ disregarding the bits that are similar. Also, data items have to be converted to binary data. This means that the weights we placed on the pages to specify the number of their occurrences will be eliminated.

Cosine: It determines similarity by the cosine of the angle between two vectors (Strehl et al., 2000). Cosine distance measure is the most popular measure for

text documents since the similarity does not depend on the length and it allows documents with the same composition but different totals to be treated identically. The Cosine distance is given by:

$$\text{Cosine}(x, y) = \frac{\sum(x_i y_i)}{\sqrt{\sum(x_i)^2 \sum(y_i)^2}}. \tag{10}$$

Pearson Correlation: It is mostly used in collaborative filtering to predict a feature from a highly similar mentor group of objects whose features are known (Strehl et al., 2000). It is defined as follows:

$$\text{Correlation}(x, y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}. \tag{11}$$

K-means computes centroid clusters differently for different k-means supported distance measures. Therefore, a normalisation step was necessary for Cosine and Correlation distance measures for comparison purposes. The points in each cluster, whose mean forms the centroid of the cluster, are normalised to unit Euclidean length. According to Strehl et al. (2000) and Halkidi et al. (2003), Cosine distance measure which is a direct application of the extended Jaccard coefficient, yields better clustering results than Pearson Correlation and the Euclidean distance measures. Because different distance measures have been applied for different purposes, there is no apparent one clustering validation measure we can rely on to test our clusters in terms of their proximity. The importance of the validation measure is significant in order to form the most appropriate clusters to be used in conjunction with Markov Model. The most common clustering validation technique is entropy (Strehl et al., 2000; Xiong et al., 2006; Wang et al., 2004). Entropy is defined as follows:

$$\Lambda^{(E)}(C_x) = \sum \frac{n_x^{(h)}}{n_x} \log\left(\frac{n_x^{(h)}}{n_x}\right). \tag{12}$$

Entropy measures the purity of the clusters with respect to the given class labels. For our data sets, entropy is measured by calculating the probability that a page in a cluster x belongs to category n_x. Entropy tends to favour small clusters. If the cluster has all its pages belonging to one category, the entropy will be 0. The entropy measure increases as the categories become more varied. The overall entropy of the whole clustering solution is measured as the weighted sum of entropy measures of all clusters within the clustering solution. Xiong et al. (2006), proved through experimentations that the entropy evaluation does not confirm with the k-means true clusters and its results could be misleading. In our distance measures evaluations, we run entropy evaluation measures, we calculate the mean of the distances and we plot clusters figures on the clusters obtained using different distance measures. As a result, Clustering the resulting sessions S_N is implemented using k-means clustering algorithm according to the Cosine distance between the sessions. Consider two sessions Sa and Sb. The Cosine distance between Sa and Sb is given by:

$$\text{distCosine}(Sa, Sb) = \frac{\sum(Sa_i Sb_i)}{\sqrt{\sum(Sa_i)^2}\sqrt{\sum(Sb_i)^2}}. \tag{13}$$

Table 4 has 4 sessions with 4 pages each. If we are to form two clusters with two sessions each, we have to measure the distances between the sessions.

Table 4 Example: four sessions

S1	3, 0, 5, 1
S2	2, 0, 5, 0
S3	0, 5, 0, 4
S4	0, 3, 0, 3

Table 5 reveals the distances calculated using equation (13): Clusters are formed according to the least distances between sessions, or the closest distances between sessions. Therefore, $\{S1, S2\}$ will form a cluster and $\{S3, S4\}$ will form another cluster.

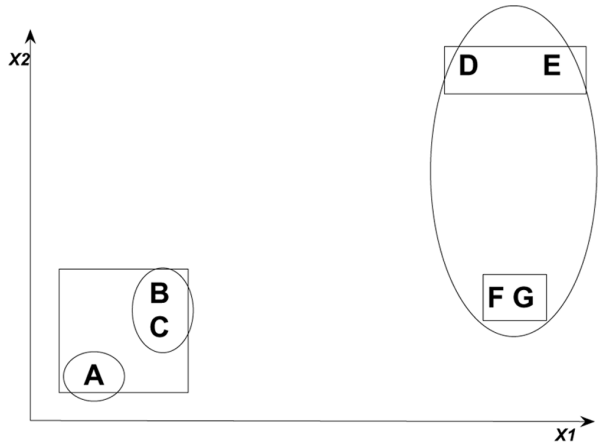
Table 5 Distances between sessions calculated using Cosine distance measure

distCosine$(S1, S2)$	0.019
distCosine$(S1, S3)$	0.89
distCosine$(S2, S3)$	1.0
distCosine$(S1, S4)$	0.88
distCosine$(S3, S4)$	0.06

4.2.1.4 Number of clusters k

The second step in the training process of the IMAC prediction model is to determine the number of clusters (k) for k-means clustering algorithm. Correctly assigning the number of clusters (k) before running the k-means algorithm, creates a major problem because better clusters could be achieved using a different number of clusters. Determining an optimal (k) is not an easy task. Therefore, a number of variations to k-means clustering emerged. The most common variant is ISODATA (Ball and Hall, 1965). The ISODATA algorithm adds further refinements to the k-means algorithm because it allows for different number of clusters while the k-means algorithm assumes that the number of clusters is known a priori. The ISODATA algorithm is a continuation of the k-means algorithm. It employs the splitting and merging of clusters. The clusters are merged if the centers of two clusters are closer than a certain threshold. The clusters are split into two different clusters if the cluster standard deviation exceeds a predefined value. Using ISODATA, it is possible to obtain the optimal partition starting from any arbitrary initial partition. Figure 2 is based on Figure 14 in Jain et al. (1999). Figure 2 reveals seven patterns. We start with patterns A, B, and C as the initial centroids, then we end up with the partition $\{\{A\}, \{B, C\}, \{D, E, F, G\}\}$, using k-means clustering algorithm, shown by ellipses. If ISODATA is given this partition as the initial partition, it will first merge the clusters $\{A\}$ and $\{B, C\}$ into one cluster because the distance between their centroids is smaller than a predefined threshold. It will then split the cluster $\{D, E, F, G\}$ into two clusters $\{D, E\}$ and $\{F, G\}$ because the distance between them is larger than a predefined value. The optimal three clusters are represented by rectangles in Figure 2.

Figure 2 ISODATA (rectangles) improves the k-means clusters (ellipses)



The running time of the ISODATA algorithm is the same as the running time of the k-means algorithm, $O(wki)$ where w is the number of sessions, k is the number of clusters, and i is the number of iterations. Since k and i are normally small in size, the running time of the algorithm has linear time complexity in terms of the size of the data set. The space complexity of both k-means and ISODATA algorithms is $O(k + w)$.

4.2.1.5 Integrating Markov Models and association rules

Before applying Markov Model algorithm to each of the predefined clusters, it is important to return the processed data to its uncategorised and expanded format. Web session categorisation serves as an aid in forming better clusters. Markov Model has to be implemented using the initial web sessions W and not categories in order to preserve the sequential property of web sessions. Markov Model analysis were carried out on each cluster using frequency pruned l-order Markov Model. We first build an all lth order Markov Model for sessions in each cluster separately. We then, prune the Markov Model results in each cluster according to the frequency pruned model requirements. To continue with the training process, if the Markov Model prediction results in no state or a state that does not belong to the majority class, association rule mining is used instead. Association rules are built based on window size 4, 90% confidence threshold and 4% minimum support. In this dissertation, a variant to the Apriori algorithm (AprioriAll) (Agrawal and Srikant, 1996) is used. The main difference between Apriori and AprioriAll algorithm is the fact that AprioriAll algorithm takes the sequence of the patterns into consideration. This is very essential when mining web sessions because the web pages are accessed in a particular order. The AprioriAll algorithm uses litemsets instead of the large itemsets generated by the Apriori algorithm. The main difference is that the support count is incremented only once per web session.

The majority class includes states with high probabilities where probability differences between two pages are significant. On the other hand, the minority class includes all other cases. In particular, the minority class includes:

1 States with high probabilities where probability differences between two pages are below a confidence threshold (ϕ_c).

2 States where test data does not match any of the Markov Model outcomes. This is due to the states pruning associated with the frequency pruned l-order Markov Model implemented.

A Markov Model state is retained only if the probability difference between the most probable state and the second probable state is above (ϕ_c) (Deshpande and Karypis, 2004). An important issue here is defining the majority class and identifying whether the new state belongs to the majority or the minority class. The confidence threshold is calculated as follows:

$$\phi_c = \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{14}$$

where $z_{\alpha/2}$ is the upper $\alpha/2$ percentage point of the standard normal distribution, and n is the frequency of the Markov state. Equation (14) stresses the fact that states with high frequency would lead to smaller confidence threshold. That means that even if the difference between the two most probable pages is small, the state with higher probability will be chosen in the case of high frequency of the state occurrence. The smaller confidence threshold results in larger majority class. The effect of the confidence threshold value and, therefore, the majority class size on the prediction accuracy depends on the actual data set. To determine the optimal value of $z_{\alpha/2}$ and, as a result, the value of the confidence factor ϕ_c, we conducted an experiment using data set D1. The increase of the minority class or, in other words, the increase in the confidence factor is affected by the decrease of $z_{\alpha/2}$. During the training process, if the Markov Model probability belongs to the minority class, association rule probability is calculated and stored with the state. This concludes the training phase of the IMAC algorithm. Next, we examine the IMAC prediction phase in details.

4.2.2 Algorithm prediction process

The prediction or test phase takes place online. The IMAC prediction process is noted in Algorithm 2.

Algorithm 2 Prediction Process of IMAC Model

```
for each coming session do
    Find its closest cluster
    Use corresponding Markov model to make prediction
    if the predictions are made by states that do not
    belong to a majority class
        Use association rules to make a revised
        prediction
    end if
end for
```

The first step in the prediction process is to examine each coming session and identify the cluster the new session belongs to before applying Markov Model

prediction techniques on that particular cluster. Each new web session the user accesses is examined and the appropriate cluster the new test item session belongs to is identified. Let i_t be a new test session where $i_t \subset I$. Web sessions W are divided into k groups or clusters. The new session i_t has probability $\text{prob}(x_i = k)$ of belonging to cluster k where $\sum_k \text{prob}(x_i = k) = 1$ and x_i indicates the cluster membership of the new session i_t. The actual cluster k that the session i_t belongs to depends on the minimum distance of i_t to the mean values of K cluster centroids using the Cosine distance measure where k refers to the subscript of the components of the vectors i and μ.

$$\text{distCosine}(i_t, \mu) = \frac{\sum_{k=1}^{K}(i_t \mu)}{\sqrt{\sum_{k=1}^{K}(i_t)^2}\sqrt{\sum_{k=1}^{K}(\mu)^2}}. \tag{15}$$

Markov Model prediction is carried out on the particular cluster the new session belongs to. If the Markov Model prediction fails the majority class test mentioned above, association rules are used for prediction. The Markov Model prediction accuracy is calculated by dividing the number of tests that result in a value $\neq 0$ to the total number of tests. Prediction accuracy results were achieved using the maximum likelihood based on conditional probabilities. All predictions in the test data that did not exist in the training data sets were assumed incorrect and were given a zero value.

4.3 IMAC example

Consider Table 6 that depicts web sessions after preprocessing, feature selection and categorisation were performed.

Table 6 Example: five user sessions

S1	A,F,I,J,E,C,D,H,N,I,J,G,D,H,N,C,I,J,G
S2	F,D,H,N,I,J,E,A,C,D,H,N,I,J,G
S3	E,C,A,C,F,I,A,C,G,A,D,H,M,G,J
S4	F,D,H,I,J,E,H,F,I,J,E,D,H,M
S5	G,E,A,C,F,D,H,M,I,C,A,C,G

Performing clustering analysis on the data set using k-means clustering algorithm and Cosine distance measure where the number of clusters $k = 2$ results in the two clusters shown in Tables 7 and 8.

Table 7 First cluster where number of clusters= 2 and using Cosine distance measure

S1	A,F,I,J,E,C,D,H,N,I,J,G,D,H,N,C,I,J,G
S2	F,D,H,N,I,J,E,A,C,D,H,N,I,J,G
S4	F,D,H,I,J,E,H,F,I,J,E,D,H,M

Table 8 Second cluster where number of clusters = 2 and using Cosine distance measure

S3	E,C,A,C,F,I,A,C,G,A,D,H,M,G,J
S5	G,E,A,C,F,D,H,M,I,C,A,C,G

Consider the following test data state I → J →?. Applying the 2nd order Markov Model to the above training user sessions we notice that the state $\langle I, J\rangle$ belongs to cluster 1 and it appeared 7 times as follows:

$$P_{n+1} = \text{argmax}\{P(E|J, I)\} = \text{argmax}\{E \to 0.57\}$$
$$P_{n+1} = \text{argmax}\{P(G|J, I)\} = \text{argmax}\{G \to 0.43\}.$$

This information alone does not provide us with correct prediction of the next page to be accessed by the user as we have high probabilities for both pages, G and E. Although the result does not conclude with a tie, neither G nor E belong to the majority class. The difference between the two pages (0.14), is not higher than the confidence threshold (in this case 0.2745). In order to find out which page would lead to the most accurate prediction, we have to look at previous pages in history. This is where we use subsequence association rules as it appears in Table 9.

Table 9 Looking at user sessions history

A, F,	$\langle I, J\rangle$	E
C, D, H, N,	$\langle I, J\rangle$	G
D, H, N, C,	$\langle I, J\rangle$	G
F, D, H, N,	$\langle I, J\rangle$	E
A, C, D, H, N,	$\langle I, J\rangle$	G
F, D, H,	$\langle I, J\rangle$	E
H, F,	$\langle I, J\rangle$	E

Tables 10 and 11 summarise results of applying subsequence association rules to the training data. Table 10 shows that F → E has the highest confidence of 100%, while Table 11 shows that C → G has the highest confidence of 100%.

Table 10 Confidence of accessing page E using subsequence association rules

A → E	1/2	50%
F → E	4/4	100%
D → E	2/6	33%
H → E	2/7	29%
N → E	1/4	25%

Table 11 Confidence of accessing page G using subsequence association rules

C → G	3/3	100%
D → G	3/6	50%
H → G	3/7	43%
N → G	3/4	75%
A → G	1/2	50%

Using Markov Models, we can determine that the next page to be accessed by the user after accessing the pages I and J could be either E or G. Whereas subsequence association rules take this result a step further by determining that if the user accesses page F before pages I and J, then there is a 100% confidence that the user will access page E next. Whereas, if the user visits page C before visiting pages I and J, then there is a 100% confidence that the user will access page G next.

5 Experimental evaluation

5.1 Data collection and preprocessing

All experiments were conducted on a P4 1.8 GH PC with 1GB of RAM running Windows XP Professional. The algorithms were implemented using MATLAB.

For our experiments, the first step was to gather log files from active web servers. Usually, web log files are the main source of data for any e-commerce or web related session analysis (Spiliopoulou et al., 1999). The logs are an ASCII file with one line per request, with the following information: The host making the request, date and time of request, requested page, HTTP reply code and bytes in the reply. The first log file used is a day's worth of all HTTP requests to the EPA WWW server located at Research Triangle Park, NC. The logs were collected for Wednesday, 30 August 1995. There were 47,748 total requests, 46,014 GET requests, 1622 POST requests, 107 HEAD requests and 6 invalid requests. The second log file is SDSC-HTTP that contains a day's worth of all HTTP requests to the SDCS WWW server located at the San Diego Supercomputer Center in San Diego, California. The logs were collected from 00:00:00 PDT through 23:59:41 PDT on Tuesday, 22 August 1995. There were 28,338 requests and no known losses. The third log file is CTI that contains a random sample of users visiting the CTI website for two weeks in April 2002. There were 115,460 total requests. The fourth log file is Saskatchewan-HTTP which contains one week worth of all HTTP requests to the University of Saskatchewan's WWW server. The log was collected from 1 June, 1995 through 7 June, 1995, a total of seven days. In this one week period there were 44,298 requests. All web server log files can be downloaded from "http://ita.ee.lbl.gov/html/contrib".

Before using the log files data, it was necessary to perform data preprocessing (Zhao et al., 2005; Sarukkai, 2000). We removed erroneous and invalid pages. Those include HTTP error codes 400s, 500s, and HTTP 1.0 errors, as well as, 302 and 304 HTTP errors that involve requests with no server replies. We also eliminated multi-media files such as gif, jpg and script files such as js and cgi.

Next step was to identify user sessions. A session is a sequence of URLs requested by the same user within a reasonable time. The end of a session is determined by a 30 min threshold between two consecutive web page requests. If the number of requests is more than the predefined threshold value, we conclude that the user is not a regular user; it is either a robot activity, a web spider or a programmed web crawler. The sessions of the data sets are of different lengths. They were represented by vectors with the number of occurrence of pages as weights. Table 12 represents the different data sets after preprocessing.

Table 12 Summary of four data sets

	D1	*D2*	*D3*	*D4*
# Requests	47,748	28,338	115,460	44,298
# Sessions	2,520	4,356	13,745	5,673
# Pages	3,730	1,072	683	2,385
# Unique IPs	2,249	3,422	5,446	4,985

Further preprocessing of the web log sessions took place by removing short sessions and only sessions with at least 5 pages were considered. This resulted in further reducing the number of sessions. Finally, sessions were categorised according to feature selection techniques introduced by Wang et al. (2004).

After web session identification, session categorisation took place and the details of the number of categories for each data set are represented in Table 13.

Table 13 Number of categories

	D1	*D2*	*D3*	*D4*
# Sessions	2,520	4,356	13,745	5,673
# Categories	196	154	267	231

The next step before implementing k-means clustering algorithm was to identify the number of clusters used and evaluate the most appropriate distance measure for all four data sets.

5.2 Determining parameters

5.2.1 Choosing similarity metric

Our basic motivation behind using clustering techniques is to group functionally related sessions together based on web services requested in order to improve the Markov Model accuracy. The Markov Model accuracy increases if the web sessions are well clustered due to the fact that more functionally related sessions are grouped together. To help find an appropriate k-means clustering distance measure we can apply to all four data sets, we examine the work presented by Strehl et al. (2000) and Halkidi et al. (2003). In order to back up their findings, we calculate the entropy measures, we perform means analysis and we plot different clusters using different distance measures for data set D1. Table 14 lists entropy measures for only some of the clusters for data set D1 due to space limitation. The table demonstrates that, in general, Cosine and Pearson Correlation yield lower entropy measures and, therefore, they constitute better clusters than the other distance measures.

Figure 3 represents clusters using Euclidean, Hamming, City Block, Pearson Correlation respectively for data set D1 using 7 clusters. Cosine distance measure is illustrated in Figure 5(a). Figure 3 reveals that the order of distance measures from worst to best are Hamming, City Block, Euclidean, Pearson Correlation and Cosine respectively. For instance, the maximum silhouette value in (b) for Hamming distance is around 0.5, whereas, the silhouette value in Figure 5(a) for

Cosine distance ranges between 0.5 and 0.9. The larger silhouette value of the Cosine distance implies that the clusters are separated from neighbouring clusters.

Table 14 Entropy measures for different clusters

Clusters	*2*	*3*	*4*	*5*	*10*	*20*	*40*	*50*
Euclidean	0.42	0.38	0.32	0.58	0.26	0.21	0.23	0.22
City	0.52	0.48	0.50	0.49	0.29	0.27	0.24	0.23
Hamming	0.56	0.49	0.53	0.50	0.36	0.29	0.31	0.34
Cosine	0.36	0.32	0.37	0.43	0.17	0.16	0.22	0.23
Correlation	0.30	0.28	0.30	0.37	0.20	0.19	0.19	0.21

Figure 3 Best clusters achieved using different distance measures: (a) silhouette value of Euclidean distance measure; (b) silhouette value of Hamming distance measure; (c) silhouette value of City Block distance measure and (d) silhouette value of Correlation distance measure (see online version for colours)

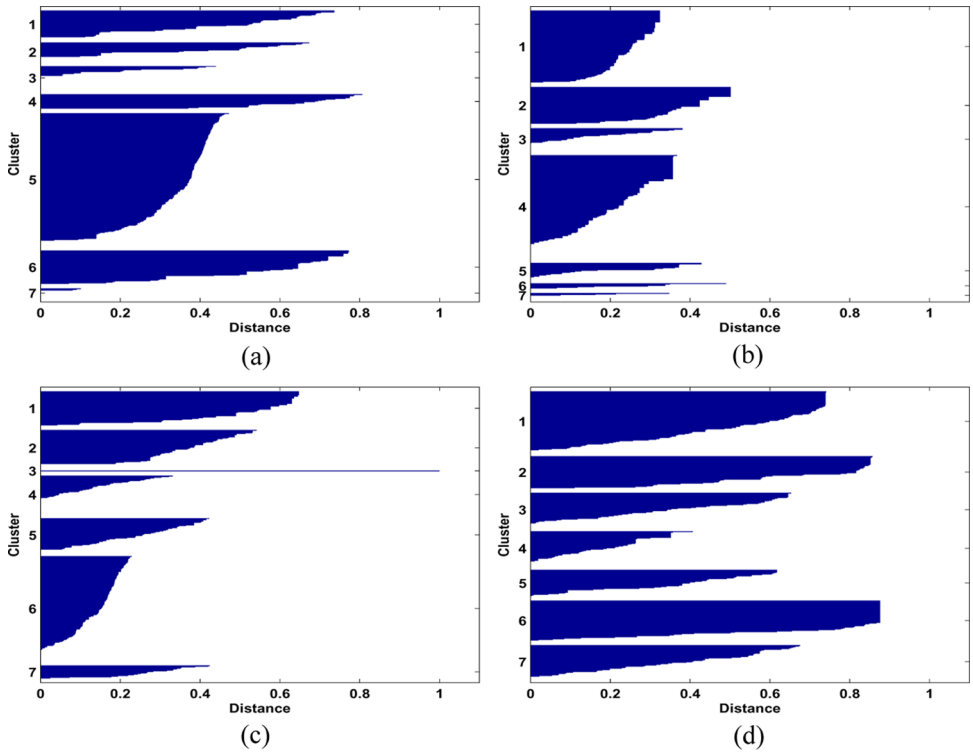
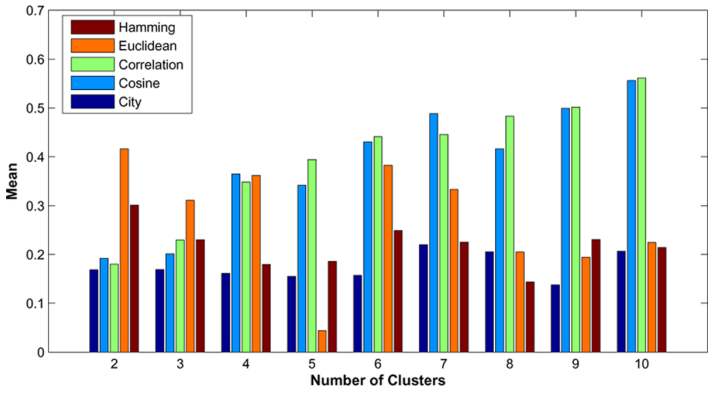


Figure 4 reveals the mean value of distances for different clusters. It is calculated by finding the average of distance values between points within clusters and their neighbouring clusters. The higher the mean value, the better clusters we get. It is worth noting that the information Figure 4 provides does not prove much on its own because it does not take into consideration points distribution within clusters.

The results of the distance plots in Figure 3, the distance mean values in Figure 4 as well as the entropy calculations all reveal that Cosine and Pearson

Correlation form better clusters than Euclidean, City Block and Hamming distance measures. Based on this information, we choose Cosine measures for all four data sets.

Figure 4 The mean value of distances for $2 \cdots 10$ clusters using different distance measures (see online version for colours)



5.2.2 *Determining the number of clusters*

Identifying the most appropriate number of clusters for all four data sets is a complex task because of lack of a one evaluation metric for the number of clusters. Different data sets with different number of categorised sessions leads to different results according to different number of clusters. Generally speaking, larger data sets with more sessions are best clustered using more clusters than smaller data sets (Gunduz and OZsu, 2003). Therefore, the number of clusters used for each data set was a result of applying k-means algorithm to each data set and, then applying ISODATA algorithm to the resulting clusters. For instance, we achieved best results for D1 when $k = 7$, for D2 when $k = 9$, for D3 when $k = 14$ and for D4 when $k = 10$. The test for best clusters results demonstrate that a larger number of web sessions is best clustered using a larger k. All clusters were attained using Cosine distance measure. Figure 5 depicts the different number of clusters for each data set. The figures plot the silhouette value represented by the cluster indices displaying a measure of how close each point in one cluster is to points in the neighbouring clusters. The silhouette measure ranges from $+1$, indicating points that are very distant from neighbouring clusters, to 0, indicating points that do not belong to a cluster.

5.2.3 *Determining* $z_{\alpha/2}$

Table 15 displays the results of the IMAC accuracy using different values for $z_{\alpha/2}$ using data set D1 data. It is clear that the accuracy increases at first with lower confidence threshold and therefore, larger minority class. However, after a certain point, accuracy starts to decrease when the majority class is reduced to the extent where it looses the advantage of the accuracy obtained by combining Markov Model and clustering. The optimal value for $z_{\alpha/2}$ is 1.15. Table 15 also reveals the number of states that are retained for association rule implementation.

Figure 5 Best clusters were achieved using different number of clusters for different data sets: (a) D1 with 7 clusters; (b) D2 with 9 clusters; (c) D3 with 14 clusters and (d) D4 with 10 clusters (see online version for colours)

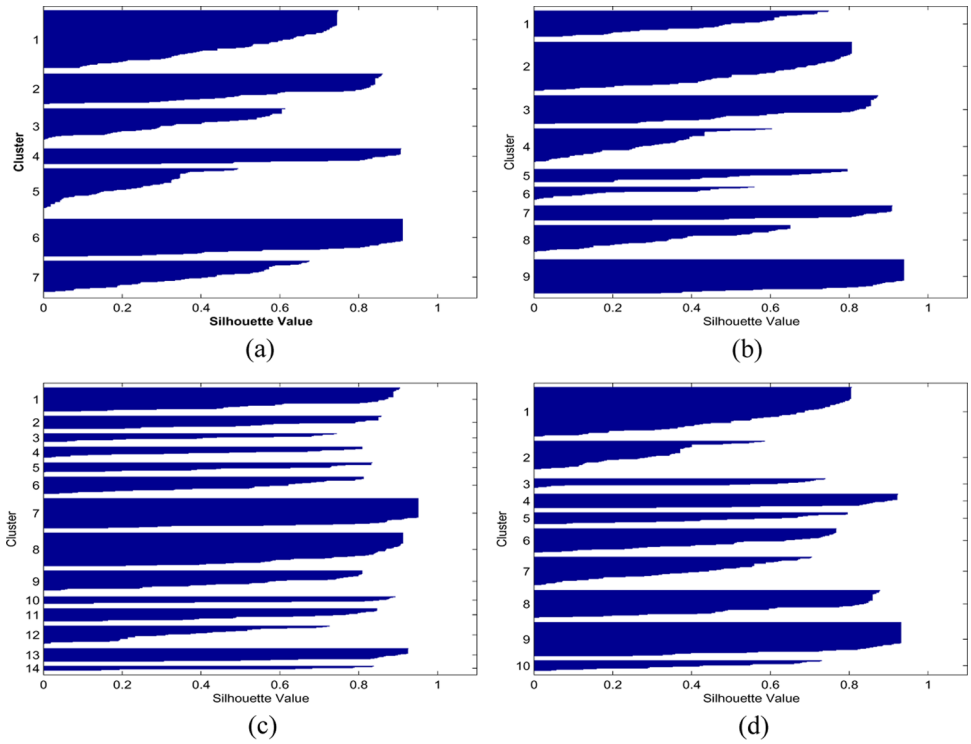


Table 15 prediction accuracy according to $z_{\alpha/2}$ value

$z_{\alpha/2}$	*Accuracy*	*No. of states*
0	31.29	9162
0.75	33.57	2061
0.84	35.45	1932
0.93	37.80	1744
1.03	40.60	1729
1.15	44.91	1706
1.28	43.81	1689
1.44	40.93	1614
1.64	38.85	1557
1.96	37.91	1479
2.57	36.81	1304

With $z_{\alpha/2} = 1.15$, the most probable pages range approximately between 80% and 40% with ϕ_c ranging between 47% and zero respectively given $n = 2$. This results in approximately 0.78 as the ratio of the majority class to the whole data set. This leaves space for 22% improvement using association rule mining not including instances that have zero matching states in the training data set.

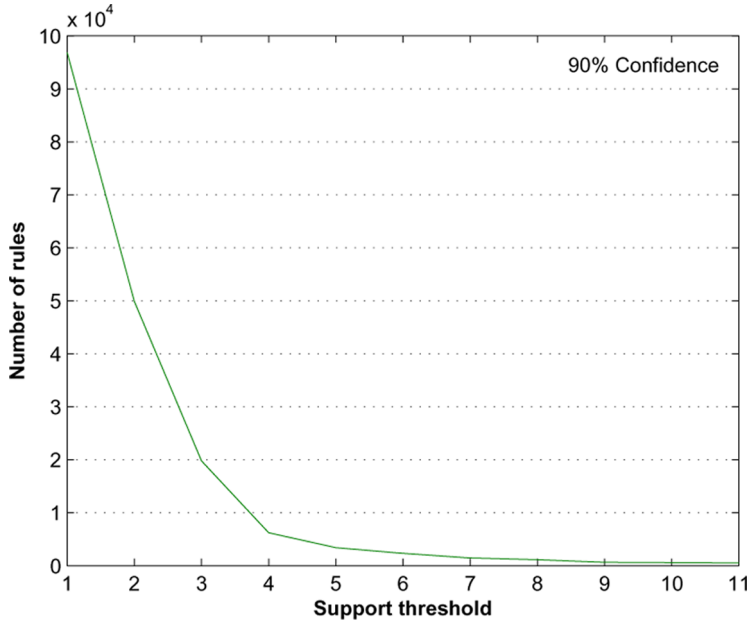
5.2.4 Determining the support factor for association rules

Since association rules techniques require the determination of a minimum support factor and a confidence factor, we used the experimental data to help determine such factors. We can only consider rules with certain support factor and above a certain confidence threshold. Using the D1, or EPA, data set, Figure 6 shows that the number of generated association rules dramatically decreases with the increase of the minimum support threshold with a fixed 90% confidence factor. Reducing the confidence factor results in an increase in the number of rules generated. This is apparent in Figure 7 where the number of generated rules decreases with the increase of the confidence factor while the support threshold is a fixed 4% value. It is also apparent from Figures 6 and 7 that the influence of the minimum support factor is much greater on the number of rules than the influence of the confidence factor. The association rules precision is calculated as a fraction of correct recommendations to total test cases used.

$$\text{Precision}(Te) = \frac{Te \cap Tr}{Te}. \tag{16}$$

Te represents the test cases whereas Tr represents training test cases or $(D - Te)$.

Figure 6 No. of rules generated according to different support threshold values and a fixed confidence factor: 90% (see online version for colours)



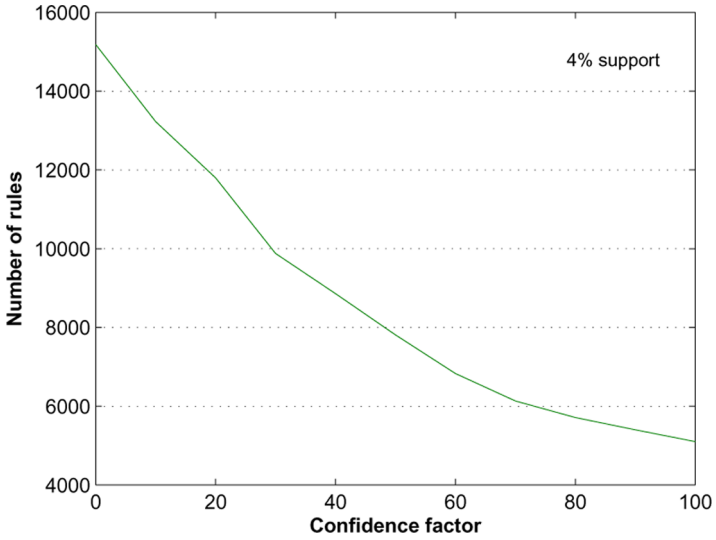
5.3 Experimental results

5.3.1 State complexity and accuracy of Markov Model

All clustering experiments were developed using MATLAB statistics toolbox. Since k-means computes different centroids each run and this yields different clustering

results each time, the best clustering solution with the least sum of distances is considered using ISODATA. Merging web pages by web services according to functionality reduces the number of unique pages and, accordingly, the number of sessions. Also, larger sessions are better clustered using larger number of clusters. Therefore, using Cosine distance measure with the number of clusters chosen ($k = 7$ for D1, $k = 9$ for D2, $k = 14$ for D3 and $k = 10$ for D4) leads to good clustering results.

Figure 7 No. of rules generated according to a fixed support threshold: 4% (see online version for colours)



Markov Model implementation was carried out for the original data in each cluster. Considering the Markov Model states S_j^l, the first order Markov Model contains S_j^1 which results in j number of states. The second order Markov Model contains $S_j^2 = j(j-1)/(1 \times 2) \approx j^2$ states. The third order Markov Model includes $S_j^3 = j(j-1)(j-2)/(3 \times 2 \times 1) \approx j^3$. The number of states increases at an exponential rate.

Using the four data sets, Tables 16 and 17 demonstrate the increase of the state space complexity as the order of all lth Markov Model increases. On the other hand, Figure 8 demonstrates the increase of accuracy as the order of all lth Markov Model increases.

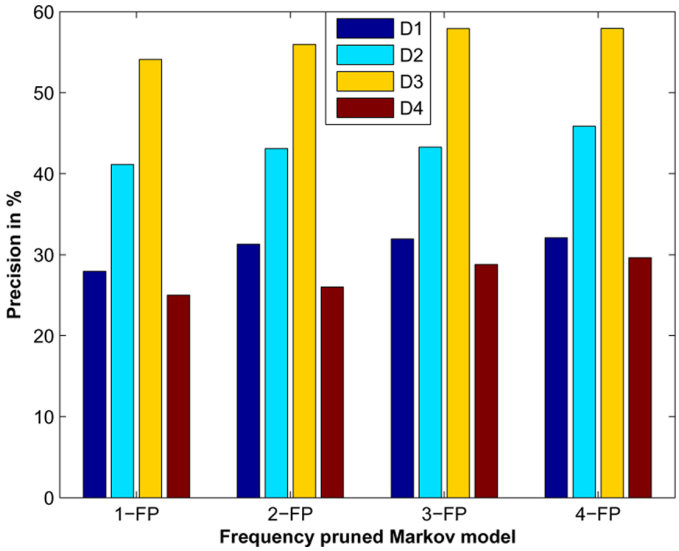
Table 16 Number of states of all 1- to 4-Markov Model orders

	1-MM	*2-MM*	*3-MM*	*4-MM*
D1	1945	39162	72524	101365
D2	1036	25060	89815	128516
D3	674	21392	50971	83867
D4	2054	34469	90123	131106

Table 17 Number of states of frequency pruned Markov Model orders

	1-PMM	*2-PMM*	*3-PMM*	*4-PMM*
D1	745	9162	14977	17034
D2	502	6032	18121	22954
D3	623	5290	11218	13697
D4	807	7961	19032	23541

Figure 8 Accuracy of all 1-, 2-, 3- and 4-frequency pruned Markov Model orders



Based on the accuracy increase represented in Figure 8, and based on the increase in the number of states represented in Tables 16 and 17, we use the all-2nd order Markov Model because it has better accuracy than that of the all-1st order Markov Model without the drawback of the state space complexity of the all-3rd and all-4th order Markov Model. For the purpose of this paper, we employ the frequency pruned Markov Model.

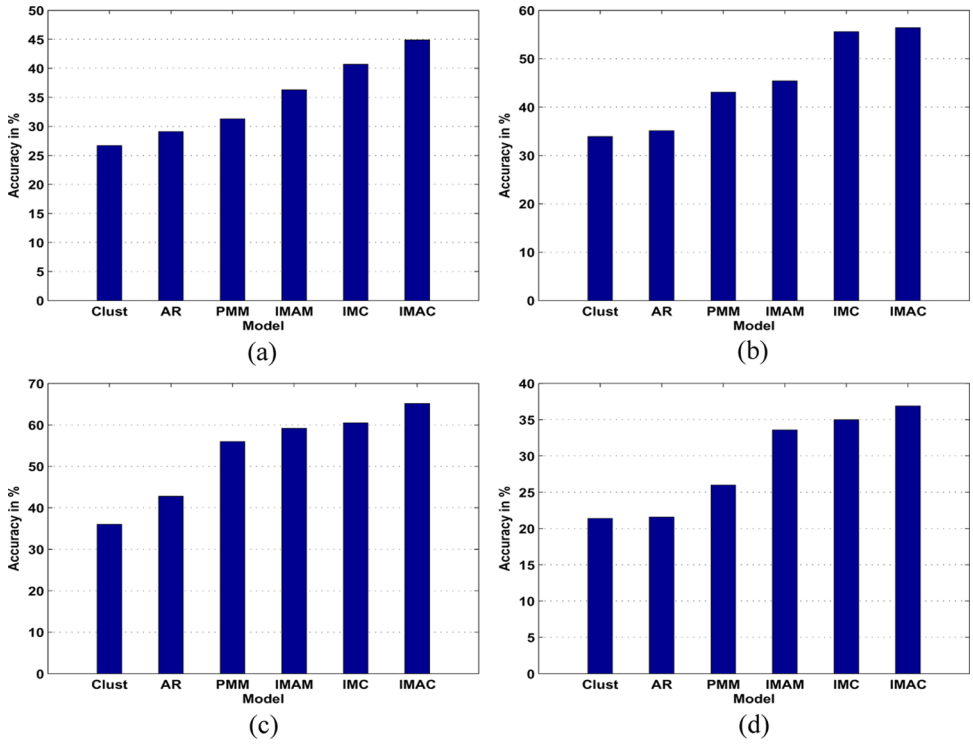
The clusters were divided into a training set and a test set each and 2-Markov Model accuracy was calculated accordingly. Then, using the test set, each session was considered as a new point and distance measures were calculated in order to define the cluster that the point belongs to. Next, l-Markov Model prediction accuracy was determined by using the Markov Model accuracy of that cluster. The Markov Model accuracy was calculated using a 10-fold cross validation. The data was partitioned into T for testing and $(D - T)$ for training where D represents the data set. This procedure was repeated 10 times, each time T is moved by T number of sessions. The mean cross validation was evaluated as the average over the 10 runs.

5.3.2 IMAC vs. individual and other integrated models

Figure 9 displays that IMAC results in better prediction accuracy than any of the other techniques individually using experiments based on all four data sets.

They also reveal that the increase in accuracy depends on the actual data set used. For instance, D1 and D4 reveal a more significant accuracy increase using IMAC over the individual models. On the other hand, D2 and D3 display a more consistent improvement in prediction accuracy. Prediction accuracy results were achieved using the maximum likelihood based on conditional probabilities.

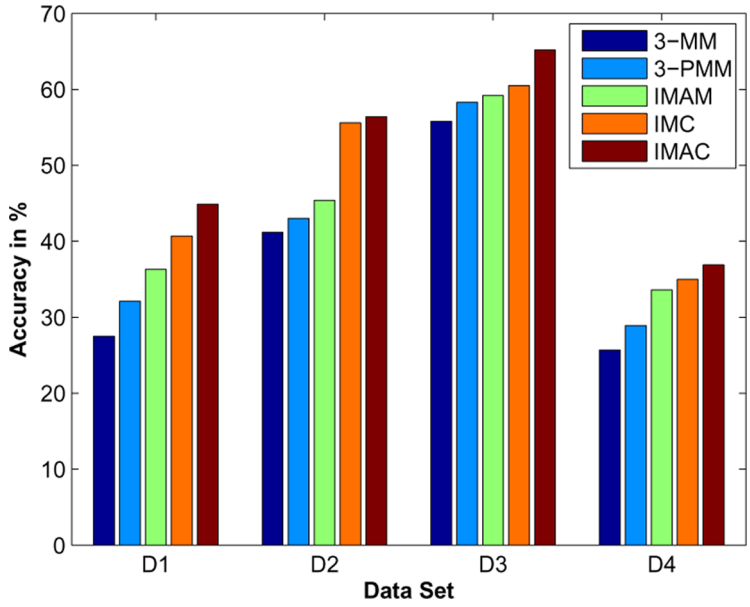
Figure 9 IMAC prediction accuracy compared to Clustering (Clust), Association Rules (AR), Pruned Markov Model (PMM), Integrated Markov Model and Association Rules (IMAM) and Integrated Markov Model and Clustering (IMC) for all four data sets: (a) Prediction accuracies for D1; (b) prediction accuracies for D2; (c) prediction accuracies for D3 and (d) prediction accuracies for D4 (see online version for colours)



To further emphasise the increase in prediction accuracy, it is essential to compare the IMAC model results to other models that rely on combining prediction models for web access prediction. For instance, IMAM (Khalil et al., 2006) combines Markov Model and association rules according to certain constraints. The results of IMAM model implementation proved an increase in web page access prediction accuracy over implementing association rule mining or Markov Model individually. The other model relies on combining clustering techniques with Markov Model. This model, Integration of Markov Model and Clustering (IMC) (Khalil et al., 2007), proved to achieve higher prediction accuracy than relying on Markov Model and clustering alone for prediction. All combination models, IMAM, IMC and IMAC are based on 2-order Markov Models. In comparing the IMAC model results to the other combination results of Markov Model and association rules

(IMAM) and Markov Model and clustering (IMC) as well as the individual models, we find that clustering techniques render the lowest web page prediction accuracy. This is evident in Figure 9. Although, association rule mining techniques prove to achieve better prediction accuracy results than clustering, the pruned all-2nd order Markov Model gives better results than association rules. As for the combination models, all models for all data sets showed a better increase in prediction accuracy using IMC than using IMAM and better prediction accuracy using IMAC than using IMC. It is important to note though that data set D2 displayed a more significant improvement of prediction accuracy using IMC. also, data set D3 revealed a more significant improvement of prediction accuracy using IMAC. Data set D1 demonstrated an overall consistent improvement of prediction accuracy using IMAM, then IMC, then IMAC respectively. On the other hand, the more significant improvement of prediction accuracy using IMAM over IMC and IMAC was apparent with data set D4. This is further manifested in Figure 10.

Figure 10 Accuracy of 3-MM and 3-PMM compared to that of IMAM, IMC and IMAC for all four data sets (see online version for colours)



5.4 *Comparing results to a higher order Markov Model*

Despite the efficient prediction accuracy results that were achieved using the three different integration models IMAM, IMC and IMAC, it was necessary to perform state space complexity analysis for the three models. The state space complexity analysis performed for IMAM model states included the summation of both Markov Model and association rules where applicable. Each association rule is considered as a state. Also, the IMC model states included both Markov Model and clustering states. Whereas, the IMAC model states were computed as the summation of the states of Markov Model, clustering and association rules where applicable. The results were compared to those of a higher order frequency pruned

Markov Model (3rd) using all four data sets. knowing that the frequency pruned Markov Model states are much less than those of Markov Model. The states results are shown in Table 18. Looking at Table 18, we notice that all three integration models involve fewer states than a higher order Markov Model. The number of states that are associated with the three integration models are less than those of the frequency pruned 3rd-order Markov Model using all data sets except for data set D3. The only apparent reason behind this result is that data set D3 has a large number of sessions with fewer number of pages. The increased number of web sessions results in higher clustering state space complexity for the clusters states are based on sessions and not pages. The increase in state space complexity for both IMC and IMAC models that implement clustering techniques asserts our findings. It is vindicated though that the number of states of the three integration models IMAM, IMC and IMAC are significantly less than those of the 3rd-order Markov Model.

Table 18 Number of states for 3-PMM, IMAM, IMC and IMAC and 3-MM using D1, D2, D3 and D4. States of integrated models include association rules. Each association rule is considered as a state

	D1	*D2*	*D3*	*D4*
3-PMM	14,977	18,121	11,218	19,032
IMAM	10,071	7,054	6,123	9,247
IMC	11,682	10,388	19,035	13,634
IMAC	13,388	11,511	20,020	15,116
3-MM	72,524	89,815	50,971	90,123

After verifying the increase of prediction accuracy using the IMAM, IMC and IMAC when compared to using Markov Model, association rule and clustering techniques individually, it was necessary to compare our prediction accuracy results to those of a higher order Markov models. We compared our results to those of 3rd-order Markov Model (3-MM) and frequency pruned 3rd-order Markov Model (3-PMM). Figure 10 depicts that our integration models deliver better prediction accuracy than a higher order Markov Model.

6 Conclusion

The method presented in this paper improves the web page access prediction accuracy by integrating all three prediction models: Markov Model, Clustering and association rules according to certain constraints. Our model, IMAC, integrates the three models using lower order Markov Model. Clustering is used to group homogeneous user sessions. Low order Markov models are built on clustered sessions. Association rules are used when Markov models could not make clear predictions. The integrated model has been demonstrated to be more accurate than all three models implemented individually, as well as, other integrated models. The integrated model has lass state space complexity and is more accurate than a higher order Markov Model.

References

Adami, G., Avesani, P. and Sona, D. (2003) 'Clustering documents in a web directory', *WIDM'03*, USA, pp.66–73.

Agrawal, R. and Srikant, R. (1994) 'Fast algorithms for mining association rules', *VLDB'94*, Chile, pp.487–499.

Agrawal, R. and Srikant, R. (1996) 'Mining sequential patterns', *International Conference on Data Engineering (ICDE)*, Taiwan, pp.3–11.

Albanese, M., Picariello, A., Sansone, C. and Sansone, L. (2004) 'web personalization based on static information and dynamic user behavior', *WIDM'04*, USA, pp.80–87.

Ball, G.H. and Hall, D.J. (1965) *Isodata, A Novel Method of Data Analysis and Classification*, Tech. Rep., Stanford University, Stanford, CA.

Basu, S., Bilenko, M. and Mooney, R.J. (2004) 'A probabilistic framework for semi-supervised clustering', *KDD'04*, USA, pp.59–68.

Bouras, C. and Konidaris, A. (2004) 'Predictive prefetching on the web and its potential impact in the wide area', *WWW: Internet and Web Information Systems*, Vol. 7, pp.143–179.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. (2003) 'Model-based clustering and visualization of navigation patterns on a web site', *Data Mining and Knowledge Discovery*, Vol. 7, No. 4, pp.399–424.

Casale, G. (2005) 'Combining queueing networks and web usage mining techniques for web performance analysis', *ACM Symposium on Applied Computing*, pp.1699–1703.

Chen, M., LaPaugh, A.S. and Singh, J.P. (2002) 'Predicting category accesses for a user in a structured information space', *SIGIR'02*, Finland, pp.65–72.

Deshpande, M. and Karypis, G. (2004) 'Selective Markov models for predicting web page accesses', *Transactions on Internet Technology*, Vol. 4, No. 2, pp.163–184.

Duda, R., Hart, P. and Stork, D. (2000) *Pattern Classification*, 2nd ed., John Wiley and Sons, USA, p.2.

Eick, C.F., Zeidat, N. and Zhao, Z. (2004) 'Supervised clustering – algorithms and benefits', *IEEE ICTAI'04*, Boca Raton, FL, USA, pp.774–776.

Eirinaki, M., Lampos, C., Paulakis, S. and Vazirgiannis, M. (2004) 'Web personalization integrating content semantics and navigational patterns', *WIDM'04*, Washington DC, USA, pp.2–9.

Eirinaki, M., Vazirgiannis, M. and Kapogiannis, D. (2005) 'Web path recommendations based on page ranking and Markov models', *WIDM'05*, Bremen, Germany, pp.2–9.

Finley, T. and Joachims, T. (2005) 'Supervised clustering with support vector machines', *22nd International Conference on Machine Learning*, USA, pp.217–224.

Gunduz, S. and OZsu, M.T. (2003) 'A web page prediction model based on click-stream tree representation of user behavior', *SIGKDD'03*, USA, pp.535–540.

Halkidi, M., Nguyen, B., Varlamis, I. and Vazirgiannis, M. (2003) 'Thesus: organizing web document collections based on link semantics', *The VLDB Journal*, Vol. 2003, No. 12, pp.320–332.

Jain, A.K., Murty, M.N. and Flynn, P.J. (1999) 'Data clustering: a review', *ACM Computing Surveys*, Vol. 31, No. 3, pp.264–323.

Jespersen, S., Pedersen, T.B. and Thorhauge, J. (2003) 'Evaluating the Markov assumption for web usage mining', *WIDM'03*, Louisiana, USA, pp.82–89.

Khalil, F., Li, J. and Wang, H. (2006) 'A framework of combining Markov model with association rules for predicting web page accesses', *Australasian Data Mining Conference (AusDM'06)*, Sydney, Australia, pp.177–184.

Khalil, F., Li, J. and Wang, H. (2007) 'Integrating Markov model with clustering for predicting web page accesses', *Australian World Wide Web (AusWeb'07)*, Coffs Harbour, Australia, pp.63–74.

Khalil, F., Li, J. and Wang, H. (2008) 'Integrating recommendation models for improved web page prediction accuracy', *Thirty-First Australasian Computer Science Conference (ACSC'08)*, Wollongong, Australia, pp.91–100.

Kim, D., Adam, N., Alturi, V., Bieber, M. and Yesha, Y. (2004) 'A clickstream-based collaborative filtering personalization model: towards a better performance', *WIDM'04*, Washington DC, USA, pp.88–95.

Lai, H. and Yang, T.C. (2000) 'A group-based inference approach to customized marketing on the web – integrating clustering and association rules techniques', *Hawaii International Conference on System Sciences*, Hawaii, USA, pp.37–46.

Liu, F., Lu, Z. and Lu, S. (2001) 'Mining association rules using clustering', *Intelligent Data Analysis*, Vol. 5, pp.309–326.

Lu, L., Dunham, M. and Meng, Y. (2005) 'Discovery of significant usage patterns from clusters of clickstream data', *WebKDD '05*, Illinois, USA, pp.139–142.

Mathur, V. and Apte, V. (2007) 'An overhead and resource contention aware analytical model for overloaded web servers', *WOSP'07*, Buenos Aires, Argentina, pp.26–37.

Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. (2000) 'Discovery of aggregate usage profiles for web personalization', *WebKDD'00*, Boston, USA, pp.61–82.

Papadakis, N.K. and Skoutas, D. (2005) 'STAVIES: a system for information extraction from unknown web data sources through automatic web warpper generation using clustering techniques', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 12, pp.1638–1652.

Pitkow, J. and Pirolli, P. (1999) 'Mining longest repeating subsequences to predict www surfing', *USENIX Annual Technical Conference*, California, USA, pp.139–150.

Pons, A.P. (2006) 'Object prefetching using semantic links', *The DATA BASE for Advances in Information Systems*, Vol. 37, No. 1, pp.97–109.

Rigou, M., Sirmakesses, S. and Tzimas, G. (2006) 'A method for personalized clustering in data intensive web applications', *APS'06*, Denmark, pp.35–40.

Sarukkai, R. (2000) 'Link prediction and path analysis using Markov chains', *9th International WWW Conference*, Amsterdam, pp.377–386.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. (2001) 'Itembased collaborative filtering recommendation algorithms', *10th International WWW Conference*, Hong Kong, pp.285–295.

Spiliopoulou, M., Faulstich, L.C. and Winkler, K. (1999) 'A data miner analysing the navigational behaviour of web users', *Workshop on Machine Learning in User Modelling of the ACAI'99*, Greece, pp.588–589.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. (2000) 'Web usage mining: discovery and applications of usage patterns from web data', *SIGDD Explorations*, Vol. 1, No. 2, pp.12–23.

Strehl, A., Ghosh, J. and Mooney, R.J. (2000) 'Impact of similarity measures on web-page clustering', *AI for Web Search*, AAAI Press, Menlo Park, California, USA, pp.58–64.

Wang, Q., Makaroff, D.J. and Edwards, H.K. (2004) 'Characterizing customer groups for an e-commerce website', *EC'04*, USA, pp.218–227.

Xiong, H., Wu, J. and Chen, J. (2006) '*K*-means clustering versus validation measures: a data distribution perspective', *KDD'06*, USA, pp.779–784.

Yang, Q., Li, T. and Wang, K. (2004) 'Building association-rule based sequential classifiers for web-document prediction', *Journal of Data Mining and Knowledge Discovery*, Vol. 8, No. 3, pp.253–273.

Zhao, Q., Bhomick, S.S. and Gruenwald, L. (2005) 'Wam miner: in the search of web access motifs from historical web log data', *CIKM'05*, Germany, pp.421–428.

Zhong, S. and Ghosh, J. (2003) 'A unified framework for model-based clustering', *Machine Learning Research*, Vol. 4, pp.1001–1037.

Zhu, J., Hong, J. and Hughes, J.G. (2002a) 'Using Markov chains for link prediction in adaptive websites', *Software 2002: Computing in an Imperfect World*, Belfast, Ireland, pp.60–73.

Zhu, J., Hong, J. and Hughes, J.G. (2002b) 'Using Markov models for website link prediction', *HT'02*, USA, pp.169–170.